Agilent E9524A Trace Toolset for Xilinx MicroBlaze

Design Guide

Agilent Technologies

Notices

Manual Part Number

E9524-97001

This manual is available online at www.agilent.com. It is not an orderable part.

Revision History

E9524-97001, November 2006

Agilent Technologies, Inc.
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

Trademark Acknowledgements

Windows and MS Windows are U.S. registered trademarks of Microsoft Corporation.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

1
Introduction

In This Guide...

This *Design Guide* provides information to assist you in designing a board which will be compatible with the Agilent E9524A Inverse Assembler for Xilinx MicroBlaze. It tells you what signals are required by the decoder, and suggests how to route these signals to a header.

This guide assumes you are using a Xilinx MicroBlaze 5 soft processor core.

For information on using the decoder, see the online help which is installed with the decoder.

Product Overview

The inverse assembler, used with an Agilent Technologies logic analyzer, allows you to reconstruct program flow by capturing the instruction address of every executed instruction, then looking up the associated opcode in the object file, and then decoding the opcode into a MicroBlaze mnemonic. The inverse assembler output is scalable, according to which signals and buses you route to the logic analyzer.

The inverse assembler uses a small number of external FPGA signals, plus a clock signal. The number of signals depends on the size of the address space in which the program is executing.

Target System Requirements

The inverse assembler has been designed to work with target systems meeting the following requirements:

Supported processor

- Xilinx MicroBlaze 5 processor core.

Trace core

- Your FPGA design should include the MicroBlaze Trace Core, included with the Agilent FPGA Dynamic Probe for Xilinx FPGAs tool.

Object files

- You must have access to the object files for the code which is executing on your target system.
- You must know the memory address where the object code is loaded.

Headers

- You must provide a way to connect logic analyzer probes to the signals on your target system:
  - You must route the required signals from the MicroBlaze core to external pins on the FPGA.
  - You must provide a means to connect a logic analyzer probe to the signals.

Equipment Required

Logic analysis system

You need an Agilent Windows®-based logic analyzer (1680, 1690, 16800 or 16900 series).

Logic analyzer cards

The inverse assembler requires one logic analyzer card. The logic analyzer card you use must support the speed of the bus you are probing.

The logic analyzer card must provide enough channels to probe the headers on your target system.

Probes

You need an appropriate number of logic analyzer probes (“adapter cables”) to connect the logic analyzer cables to the header on your target system. The probe must match the type of connector you have placed on your board. Agilent recommends a MICTOR connector and an Agilent E5346A or E5380A probe.

Xilinx platform cable

You need to provide a Xilinx Platform Cable USB (recommended) or a Xilinx Parallel Cable. This cable allows the Agilent FPGA Dynamic Probe for Xilinx FPGAs tool on the logic analyzer to configure the signals in the MicroBlaze Trace Core.

Software Required

The Agilent E9524A Trace Toolset for Xilinx MicroBlaze requires the following additional software:

Agilent software

- Agilent B4655A FPGA Dynamic Probe for Xilinx. The Agilent MicroBlaze Trace Core (MTC) is included with this software.

Xilinx software

- Xilinx Platform Studio (XPS) 8.2i.
- Xilinx ChipScope Pro.

Compiler

- The gcc compiler which is shipped with Xilinx Platform Studio, or equivalent.

Suggested Design Process

1 Decide how many MicroBlaze trace signals to route to the pins of the FPGA. This decision will be a tradeoff between:
   - How many FPGA pins you can allocate to debug signals.
   - What information you need the inverse assembler to generate.

   See "Deciding Which Signals to Capture" on page 13 for details.

2 Choose what kind of connector to design into your board. This choice will depend on:
   - The amount of board space available for the connector.
   - Signal integrity requirements.
   - Which logic analyzer probes you already have or plan to buy.

   See "Designing the Headers" on page 23 for details.

3 Design your board. Be sure to route the appropriate number of signals from the FPGA pins to the logic analyzer connector.

4 When you are ready to begin making measurements, connect up the cables, then finish making the connections using software tools:

   a Use the Xilinx Platform Studio to include the chipscape_icon and agilent_mtc_v5 (Agilent MicroBlaze Trace Core) cores in your FPGA.

   b Use the Agilent FPGA Dynamic Probe to configure the MTC and route the necessary signals to the FPGA pins you have set aside for that purpose, and to automatically configure the logic analyzer to capture those signals.

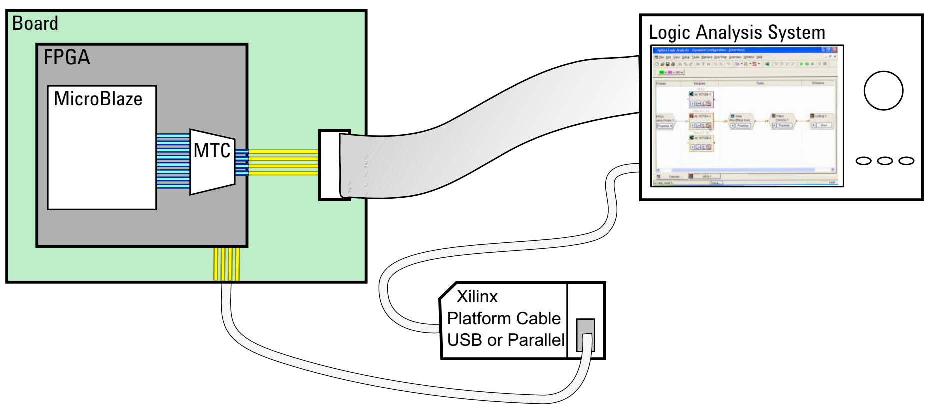
   See "Automatically Routing Signals to FPGA Pins With the MTC Core" on page 26 for an example.

2
Designing Your Board

This chapter describes the factors you need to consider when designing and preparing your target system for logic analysis.

Overview of the Connectors

You must provide a connector on your board for the logic analyzer.



The connector must match the logic analyzer probe you plan to use. Examples of such connectors include MICTOR, Samtec, and pads for an Agilent soft touch or pro-series soft touch connectorless probe.

MicroBlaze trace signals must be routed from the MicroBlaze Trace Core (MTC) to the pins of the FPGA, then to the logic analyzer connector.

Number of signals

The number of MicroBlaze trace signals that are routed to the logic analyzer is scalable. You can choose how many signals to route, depending on your pin budget and the debug information you need to see. By routing a larger number of signals, you increase the richness of the debug information generated by the inverse assembler. By routing a smaller number of signals, you can minimize the number of FPGA pins dedicated to MicroBlaze debug and the number of traces to the logic analyzer connector.

The MTC multiplexes the MicroBlaze trace signals 2:1. Thus the number of signals on the logic analyzer connector that are assigned to MicroBlaze trace is 1/2 the number of MicroBlaze trace signals that you want the logic analyzer to capture.

For information on choosing which signals to capture, see “Deciding Which Signals to Capture" on page 13. For instructions on how to configure the MTC to route the signals you have chosen, see “Example: Using XPS to Set Up an FPGA for the Inverse Assembler" on page 25.

Deciding Which Signals to Capture

The Agilent E9524A Inverse Assembler for Xilinx MicroBlaze is designed to work with trace signals from the MicroBlaze core. It will work with a full set of trace signals, or a subset of the trace signals. The inverse assembler will automatically scale its output in accordance with the buses and signals which are present.

Before you design your board, you must decide what subset of MicroBlaze trace signals you will provide to the inverse assembler and then determine the number of FPGA pins and logic analyzer connector pins that are required to support this subset.

The Agilent MicroBlaze Trace Core (MTC) is used to select a subset of MicroBlaze trace signals, and to mux these signals, 2:1, to a reduced set which must be routed to the pins of the FPGA and then to the logic analyzer connector. Thus, when you design your board, you need to provide FPGA pins and logic analyzer connector pins for 1/2 the maximum number of MicroBlaze trace signals that you plan to use for debug.

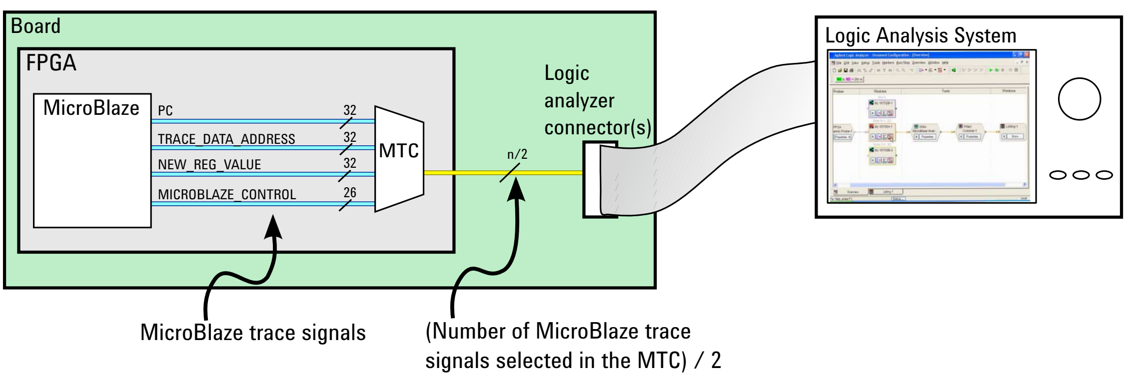
Overview of the MicroBlaze trace signals

The MicroBlaze trace signals are organized as four buses, shown in the following table.

Table 1 MicroBlaze trace signals

MicroBlaze trace signals	Explanation
PC	Program counter (32 bits)
TRACE_DATA_ADDRESS	Data addresse bus (32 bits)
NEW_REG_VALUE	Data value bus (register writes and memory reads/writes, 32 bits)
MICROBLAZE_CONTROL	Assorted control and status signals (26 bits)

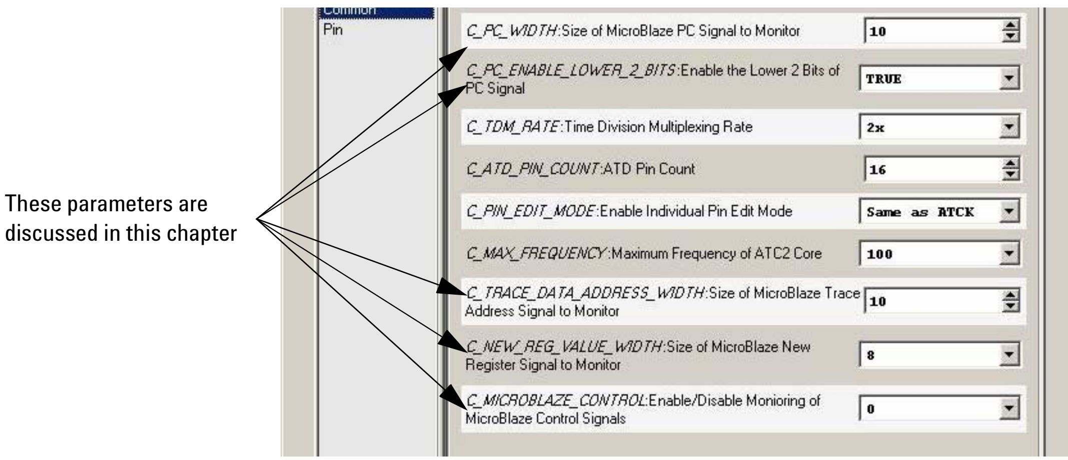
The relationship between the MicroBlaze core, the MicroBlaze trace signals and the Agilent MicroBlaze Trace Core (MTC) is shown in the following figure.



In addition to the signals shown here, there are four additional status signals and a clock signal.

Setting up the trace signals in XPS

The buses and signals available to the logic analyzer are determined by how you configure the FPGA in Xilinx Platform Studio (XPS). When you include the MTC in your FPGA design, you will specify which of the MicroBlaze trace signals you wish to route to the pins of the FPGA and to the logic connectors. Your selections will determine the number of signals that must be routed out of the FPGA and to the logic analyzer connector(s).



Note that by selecting a bus in XPS, several associated control signals will be automatically included in your design. See Table 3 on page 22 and “Descriptions of status signals automatically added by XPS" on page 19.

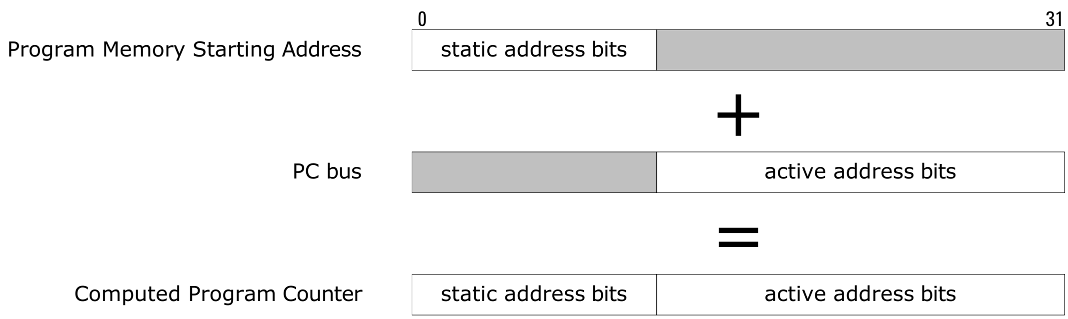
C_PC_WIDTH

C_PC_WIDTH defines the number of bits of PC to route to the logic analyzer. You do not need to route all 32 bits of the PC.

In any FPGA design, a certain number of upper Program Counter bits are static. Those bits do not need to be traced.

To reduce pin count, you can omit those upper, static bits of the program counter and specify the value of the static bits in the Properties dialog of the inverse assembler. The inverse assembler combines the value of the static bits with the address bits which are captured from the FPGA to generate the Address column of the listing. See the online help in the inverse assembler for more information.

The inverse assembler computes full 32-bit Program Counter (PC) addresses with less than 32 bits of PC by combining the most-significant bits of 'Program Start Address' with the least significant, active, n bits of 'PC' to form PC [0:31].



This bus is created by the MTC if you have selected a value other than 0 for C_PC_WIDTH in XPS.

Number of MicroBlaze trace signals If you configure C_PC_WIDTH = n, where n > 0, the number of trace signals will be:

n + 1

The extra 1 bit is for the status signal VALID_INSTR which is added automatically by XPS.

C_PC_ENABLE_LOWER_2_BITS

C_PC_ENABLE_LOWER_2_BITS determines whether the lower two bits of the PC will be routed to FPGA pins. To reduce pin count, you can choose to omit the lower two bits of the program counter, because instructions always begin on 4-byte boundaries.

Select TRUE if you want to output the lower two bits of PC. If you select TRUE, remember to include these two bits in C_PC_WIDTH.

Agilent recommends that you enable the lower two bits (select TRUE) unless the pin budget of the FPGA is severely restricted. Although the lower two bits are always zero, including them makes it much easier to trigger the logic analyzer. If you leave them out, every time you want to trigger on a PC value, you will need to shift the address by two bits.

C_TRACE_DATA_ADDRESS_WIDTH

C_TRACE_DATA_ADDRESS_WIDTH defines the number of bits of TRACE_DATA_ADDRESS to route to the logic analyzer. The TRACE_DATA_ADDRESS bus carries the address for data reads and writes. You do not need to route all 32 bits.

In any FPGA design, a certain number of upper data address bits are static. Those bits do not need to be traced.

To reduce pin count, you can omit those upper, static bits of the data address and specify the value of the static bits in the Properties dialog of the inverse assembler. The inverse assembler combines the value of the static bits with the address bits which are captured from the FPGA to generate the Address column of the listing. See the online help in the inverse assembler for more information.

This bus is 0 to 32 bits wide. This bus is created by the MTC if you have selected a value other than 0 for C_TRACE_DATA_ADDRESS_WIDTH in XPS.

Number of MicroBlaze trace signals If you configure C_TRACE_DATA_ADDRESS_WIDTH = n, where n > 0, the number of trace signals will be:

n + 3

The extra 3 bits are for the status signals VALID_INSTR , TRACE_DATA_ACCESS , and TRACE_DATA_READ which are added automatically by XPS.

C_NEW_REG_VALUE_WIDTH

C_NEW_REG_VALUE_WIDTH defines the number of bits of NEW_REG_VALUE to route to the logic analyzer.

The NEW_REG_VALUE bus carries the contents of the register which has been written to. Since all memory accesses go through a register, this bus provides the values of memory read and writes.

You can route 0, 8, 16, or 32 bits of NEW_REG_VALUE to the logic analyzer. Agilent recommends that you always choose 32 bits unless the pin budget of the FPGA is severely restricted. If you choose 0, 8, or 16 bits, you will not be able to see the upper bits of memory reads and writes in the logic analyzer listing.

This bus is created by the MTC if you have selected a value other than 0 for C_NEW_REG_VALUE_WIDTH in XPS.

Number of MicroBlaze trace signals If you configure C_NEW_REG_VALUE_WIDTH = n, where n > 0, the number of trace signals will be:

n + 4

The extra 4 bits are for the status signals VALID_INSTR , TRACE_DATA_ACCESS , TRACE_DATA_READ , and REG_WRITE which are added automatically by XPS.

C_MICROBLAZE_CONTROL

C_MICROBLAZE_CONTROL determines whether the 26 optional MicroBlaze control signals will be routed to FPGA pins. Xilinx Platform Studio allows you to bring many additional signals out of the FPGA. You can use those additional signals in triggers and filters, and you can display them in the listing, but they are not used by the inverse assembler.

In general, none of these additional signals are required, so you do not need to allocate any pins to these signals, and you can set the width of C_MICROBLAZE_CONTROL to 0. XPS does not allow these signals to be chosen individually; if you want to capture any of these signals, you must choose C_MICROBLAZE_CONTROL = 26.

Number of MicroBlaze trace signals If you configure C_MICROBLAZE_CONTROL = 26, the number of trace signals will be:

26

Descriptions of status signals automatically added by XPS

Depending on your selections in XPS, the following status signals may be automatically included:

CLOCK

The logic analyzer also uses the MicroBlaze CLOCK signal. A logic analyzer state is captured for each clock cycle; the clock is not displayed as a bus or signal.

REG_WRITE

Indicates that a state contains a register write. One bit wide.

This signal is automatically added by XPS if you have selected a value other than 0 for C_NEW_REG_VALUE_WIDTH in XPS.

TRACE_DATA_ACCESS

Indicates when a state contains a memory access. One bit wide.

This signal is automatically added by XPS if you have selected a value other than 0 for C_TRACE_DATA_ADDRESS_WIDTH or C_NEW_REG_VALUE_WIDTH in XPS.

TRACE_DATA_READ

When a state contains a data access, this signal indicates whether the data access is a read or a write. 1 indicates a read. One bit wide.

This signal is automatically added by XPS if you have selected a value other than 0 for C_TRACE_DATA_ADDRESS_WIDTH or C_NEW_REG_VALUE_WIDTH in XPS.

VALID_INSTR

Indicates that the logic analyzer state contains a complete instruction. One bit wide.

This signal is is always created by the Agilent MTC core.

Enabling multiplexing

To enable 2:1 multiplexing, you must set C_TDM_RATE to “2x” in XPS. Agilent recommends that you enable multiplexing.

Calculating the number of pins required

The number of FPGA pins which must be routed to the logic analyzer connector(s) depends on the maximum values you plan to set in XPS. To calculate how many pins you need:

1 Calculate the "Number of MicroBlaze trace signals" for each selection you will make in XPS, using information from the previous sections.

   Make sure your calculation uses the maximum number of signals you will ever want to use at a given time.

   Do not count the same signal multiple times. For example, if you set both C_PC_WIDTH > 0 and C_TRACE_DATA_ADDRESS_WIDTH > 0, count VALID_INSTR only once.

2 Add all of those numbers.

3 Add 1 for the clock.

4 Divide the total by 2 (assuming you set C_TDM_RATE to 2x to enable multiplexing).

5 Round up to the nearest integer.

Example

Table 2 Example calculation of number of FPGA pins required

Value chosen in XPS	Number of MicroBlaze trace signals
C_PC_WIDTH = 10	10 + 1 = 11
C_TRACE_DATA_ADDRESS_WIDTH =8	8 + 2 = 10
C_NEW_REG_VALUE_WIDTH = 8	8 + 1= 9
C_MICROBLAZE_CONTROL = 0	0
	1 (CLOCK)
	Total = 31 / 2 = 15.5 → **16 pins**

Summary

The signals you choose to bring out will affect how much information the inverse assember can output.

Table 3 Effect of buses and signals on the listing

If you choose this bus in XPS	...these signals will be added automatically by XPS for use by the inverse assembler	...and the inverse assembler will output
PC	VALID_INSTR	Program trace (instruction addresses and mnemonics)
TRACE_DATA_ADDRESS	VALID_INSTR TRACE_DATA_ACCESS TRACE_DATA_READ	Data addresses for memory reads/writes. R/W indication in the MicroBlaze Inverse Assembly column.
NEW_REG_VALUE	VALID_INSTR TRACE_DATA_ACCESS TRACE_DATA_READ REG_WRITE	Data values for memory reads/writes. Data values and register numbers for register writes.
MICROBLAZE_CONTROL	MSR_REG (11 bits) PIPE_RUNNING (3 bits) EXCEPTION_TAKEN EXCEPTION_TYPE JUMP_TAKEN (4 bits) TRACE_DELAY_SLOT TRACE_DCACHE_REQ TRACE_DCACHE_HIT TRACE_ICACHE_REQ TRACE_ICACHE_HIT MB_HALTED	Not used by the inverse assembler. You may bring these signals out if you want to view them. In general, none of these signals are required.

If you remove one of the buses or signals in the logic analyzer, the inverse assembler will scale its output accordingly.

Designing the Headers

You must provide a sufficient number of logic analyzer connectors on your board to accomodate all of the MicroBlaze trace signals you wish to capture. See “Deciding Which Signals to Capture" on page 13 for help in deciding the number of signals.

Agilent recommends using a Mictor connector (for ruggedness and low cost), but other connectors may be used.

AMP MICTOR 38 connectors

Each MICTOR 38 connector carries 32 signals plus two clocks (CLK1 for two logic analyzer pods). Probes (part number E5346A, sometimes called “high-density termination cables”) are required to connect the logic analyzer cables to the MICTOR connector. These probes contain the required termination. One probe is required for every two logic analyzer pods.

To increase the structural support for the probes, you should use support shrouds on each connector.

For more information, including mechanical dimensions, see the *Agilent Technologies E5346A 38-Pin Probe and E5351A 38-Pin Adapter Cable Installation Note*, available from www.agilent.com.

Design Considerations

The connector must be close enough to the signal source so that the stub length created is less than $^1/_5$ the t_r (bus risetime). For PC board material, (er = 4.9) and Z_o in the range of 50 - 80Ω, use a propagation delay of 160 ps/inch of stub.

Each probed signal line must be able to supply a minimum of 600 mV to the probe tip and handle a minimum of 90 kΩ shunted by 10 pF. The maximum input voltage to the logic analyzer is ±40V peak

Other connectors

You may use other connector/probe combinations, including:

- Agilent soft touch connectorless probe
- Agilent pro series soft touch connectorless probe
- Samtec connector with Agilent Samtec probe

Design information for these connectors can be found on the web at www.agilent.com.

3
Example: Using XPS to Set Up an FPGA for the Inverse Assembler

This chapter gives an example of how to set up an FPGA and get the signals required by the inverse assembler to appear on the FPGA pins, using the Xilinx Platform Studio (XPS) 8.2i.

In this example, we will route the minimum required signals for inverse assembly to external FPGA pins. Those signals are PC (10 bits), VALID_INSTR (1 bit), and CLK (1 bit).

The example shows the two ways to route the signals to pins: manual routing and automatic routing using the MTC core.

NOTE This is only an example. The steps you need to follow for your system will be different.

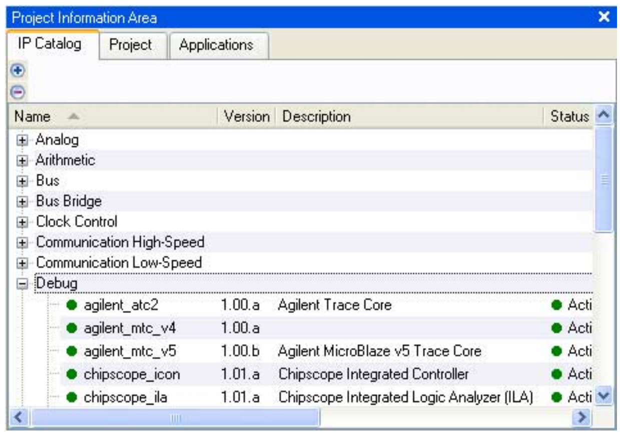
Automatically Routing Signals to FPGA Pins With the MTC Core

To route the MicroBlaze signals you want to capture to the external pins of the FPGA, use the Agilent MTC core.

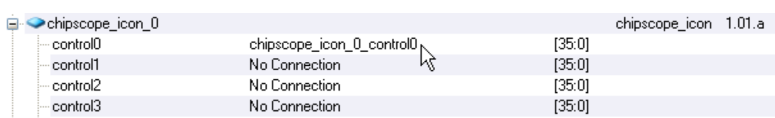
To do this, you need to include two cores in your design:

- chipscape_icon
- agilent_mtc_v5

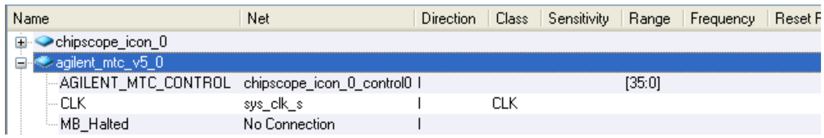
1 In the Program Information Area of XPS, select **IP Catalog**.
2 Expland DEBUG.



3 Drag chipscope_icon and agilent_mtc_v5 to the System Assembly window.
4 In the System Assembly window, expand chipscope_icon and set it up as shown:

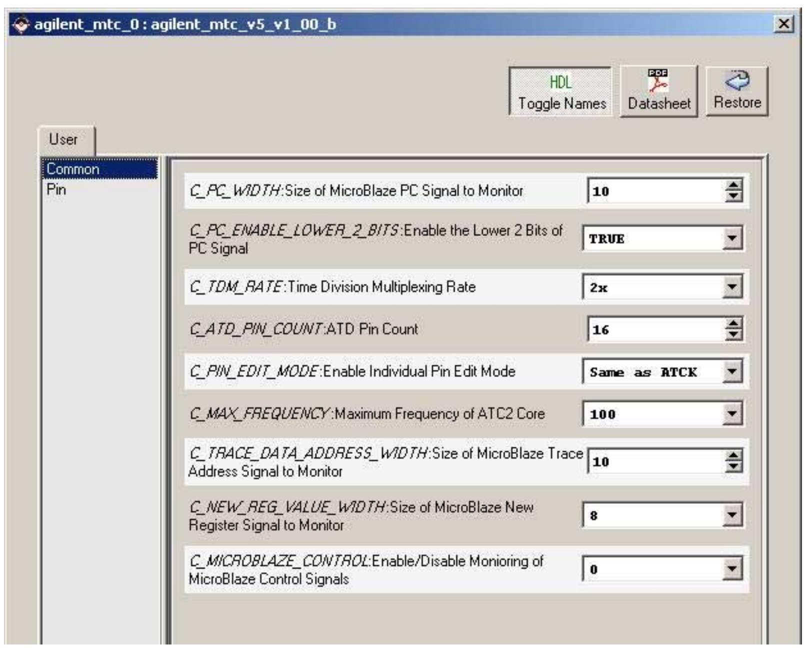


5 In the System Assembly window, expand agilent_mtc_v5 and set it up as shown:



6 Double-click agilent_mtc_v5 to display the setup dialog. Configure the buses and signals you want to bring out of the FPGA and measure with the logic analyzer.

In this example, we need 10 bits for PC, one clock bit, and one VALID_INSTR bit. These 12 bits can be brought out using 6 pins by using 2x multiplexing (C_TDM_RATE=2x). Set the pin location and I/O standard for each of the 6 pins.

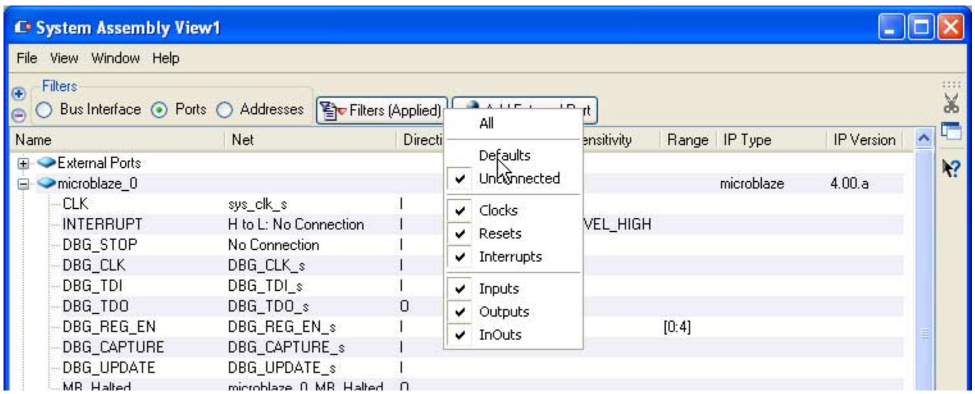


Manually Routing Signals to FPGA Pins

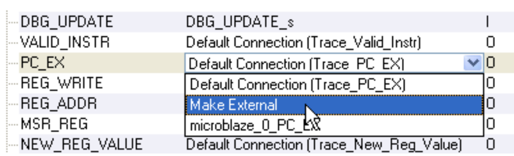
If you do not use the MTC core, you must route the signals to external pins by adding external ports and modifying the UCF file:

1 In the XPS System Assembly window, select the **Ports** tab.

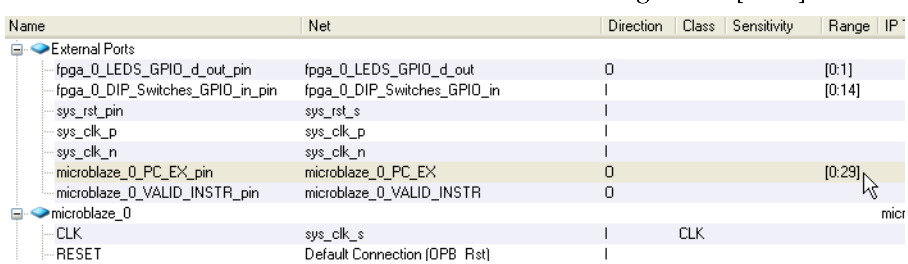
2 Click **Filters** and select **Defaults**.



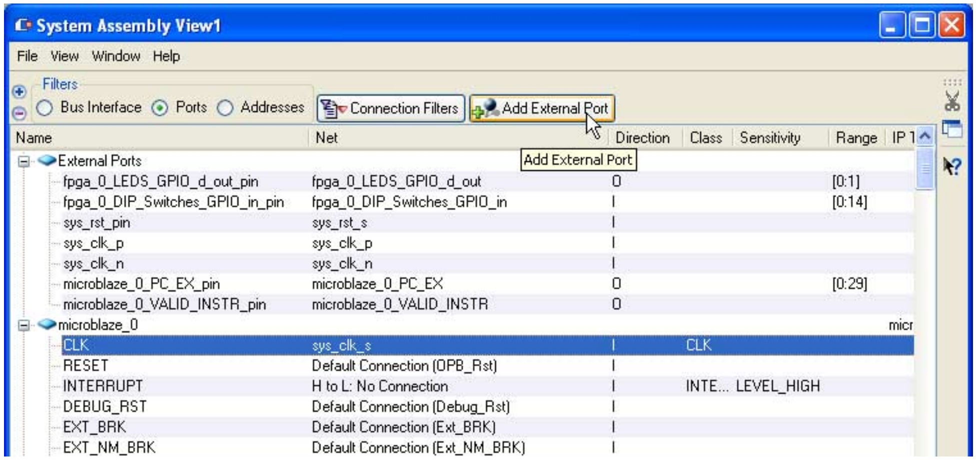
3 Right-click “VALID_INSTR” and select **Make External**.



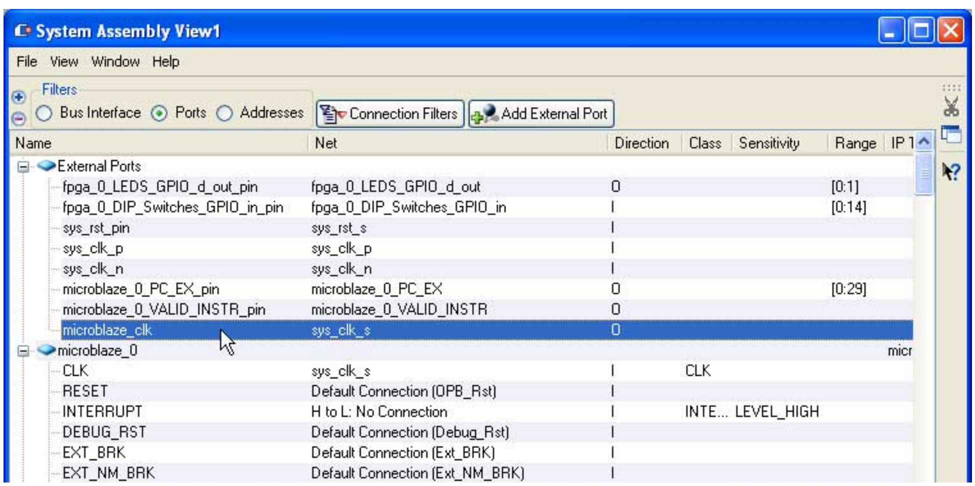
4 Do the same for PC. For PC set the range to be [0:29].



5 Add the MicroBlaze system clock (CLK) to an output. For this case the clock signal is called “sys_clk_s”. Begin by clicking **Add External Port**.



Now configure the new port to look like this:



6 Modify your UCF file. Add pin locations for the MicroBlaze trace outputs as shown in the example below. Note that the actual pin locations are user-defined. (This example shows PC_EX for MicroBlaze4; for MicroBlaze5, this bus bus is called PC).

```
# IA FPGA pins
Net microblaze_clk LOC=L16;
Net microblaze_0_VALID_INSTR LOC=L15;
Net microblaze_0_PC_EX<29> LOC=C12;
Net microblaze_0_PC_EX<28> LOC=B13;
Net microblaze_0_PC_EX<27> LOC=D16;
Net microblaze_0_PC_EX<26> LOC=D15;
Net microblaze_0_PC_EX<25> LOC=D14;
Net microblaze_0_PC_EX<24> LOC=E16;
Net microblaze_0_PC_EX<23> LOC=E15;
Net microblaze_0_PC_EX<22> LOC=E14;
Net microblaze_0_PC_EX<21> LOC=F16;
Net microblaze_0_PC_EX<20> LOC=F15;
Net microblaze_0_PC_EX<19> LOC=G16;
Net microblaze_0_PC_EX<18> LOC=G15;
Net microblaze_0_PC_EX<17> LOC=K16;
Net microblaze_0_PC_EX<16> LOC=K15;
Net microblaze_0_PC_EX<15> LOC=B4;
Net microblaze_0_PC_EX<14> LOC=A5;
Net microblaze_0_PC_EX<13> LOC=B5;
Net microblaze_0_PC_EX<12> LOC=C5;
Net microblaze_0_PC_EX<11> LOC=A6;
Net microblaze_0_PC_EX<10> LOC=B6;
Net microblaze_0_PC_EX<9> LOC=D6;
Net microblaze_0_PC_EX<8> LOC=A7;
Net microblaze_0_PC_EX<7> LOC=B7;
Net microblaze_0_PC_EX<6> LOC=C7;
Net microblaze_0_PC_EX<5> LOC=A10;
Net microblaze_0_PC_EX<4> LOC=B10;
Net microblaze_0_PC_EX<3> LOC=D10;
Net microblaze_0_PC_EX<2> LOC=A11;
Net microblaze_0_PC_EX<1> LOC=B11;
Net microblaze_0_PC_EX<0> LOC=C11;
```

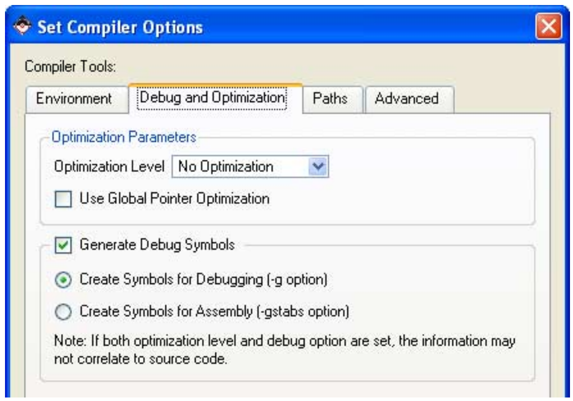
Building the Software

Once you have routed the signals to external pins on the FPGA, build your software and download the FPGA bit file.

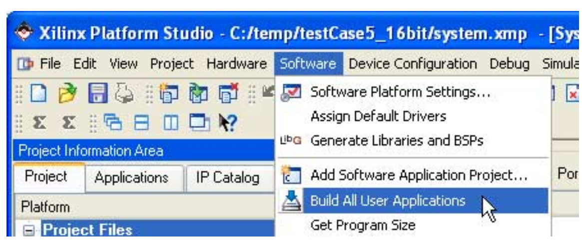
1 In XPS, select **Hardware>Generate Bitstream**.

2 Select **Software>Generate Libraries and Bsps**.

3 Set the compiler options.



4 When you are done creating the application and setting the options, select **Software>Build All User Applications** in XPS to build all of the applications.



5 After building the applications, XPS allows you to select the application that must be initialized in the generated bitstream. To initialize the hello_world_app executable in the bitstream, right-click on the project name hello_world_app in the tree view and select **Mark to Initialize BRAMs**. From the XPS main window, select **Device Configuration>Update Bitstream** to initialize the BRAMs with the application's executable information.

6 Download FPGA bit file “implementation/download.bit”

Index