Altera FPGA Dynamic Probe

Online Help

Agilent Technologies

Notices

Trademarks

Microsoft®, MS-DOS®, Windows®, Windows 2000®, and Windows XP® are U.S. registered trademarks of Microsoft Corporation.

Adobe®, Acrobat®, and the Acrobat Logo® are trademarks of Adobe Systems Incorporated.

Manual Part Number

Version 03.82.0000

Edition

April 10, 2009

Available in electronic format only

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

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.

Using the Altera FPGA Dynamic Probe

When the Altera LAI (Logic Analyzer Interface) core has been inserted into an FPGA, the FPGA dynamic probe lets a logic analyzer capture data on signals internal to the FPGA.

The FPGA dynamic probe lets you change probe points without recompiling or affecting the timing of the design, and it lets you import internal signal names from your FPGA design tool.

See Also

- Probe Control, COM Automation (see page 71)
- Probe Setup, XML Format (see page 73)

Contents

8 FPGA Dynamic Probe Concepts

9 FPGA Dynamic Probe Reference

10 Probe Control, COM Automation

11 Probe Setup, XML Format

1
Overview

The FPGA dynamic probe lets you:

- *View internal activity.*

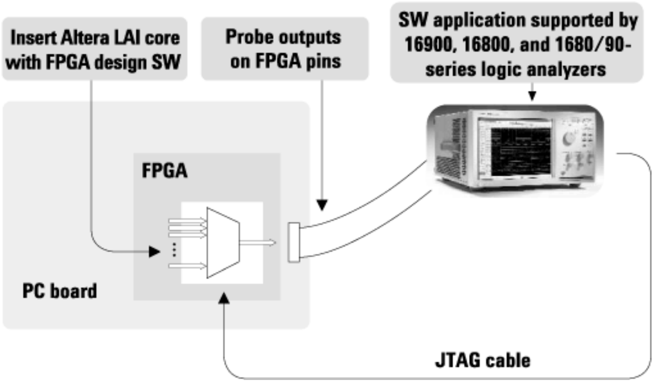
  With a logic analyzer, you are normally limited to measuring signals at the periphery of the FPGA. With the FPGA dynamic probe, you can now access signals internal to the FPGA. You can measure multiple internal signals for each external pin dedicated to debug, unlocking visibility into your design that you never had before.

- *Make multiple measurements in seconds.*

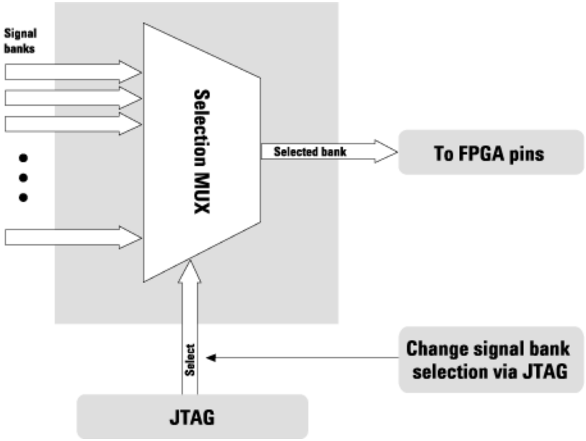
  Moving probe points internal to an FPGA used to be time consuming. Now, in less than a second you can easily measure a different set of internal signals — without design changes — and FPGA timing stays constant when you select new sets of internal signals for probing.

- *Leverage the work you did in your design environment.*

  The FPGA dynamic probe is the industry&#8217;s first tool that maps internal signal names from your FPGA design tool to your logic analyzer. Eliminate unintentional mistakes and save hours of time with this automatic setup of signal and bus names on your logic analyzer.


Insert Altera LAI core with FPGA design SW
Probe outputs on FPGA pins
SW application supported by 16900, 16800, and 1680/90-series logic analyzers
FPGA
PC board
JTAG cable

Create a time-saving FPGA measurement system. Insert an Altera LAI (Logic Analyzer Interface) core into your FPGA design. With the application running on your logic analyzer via JTAG, you control which group of internal signals to measure.



The Altera LAI core provides a multiplexer for selecting the signals that are output to FPGA debug pins. Core parameters determine the number of signals per bank and the number of banks.

2
Probing FPGA Debug Pins

The supported mechanisms for probing the FPGA debug pins with a logic analyzer are: soft touch (34-channel or 17 channel), Mictor, Samtec, and flying lead probes.

For more information on probing, see "Probing the Device Under Test" (in the online help).

3
Installing and Licensing the FPGA Dynamic Probe

Before you can use the Agilent B4656A FPGA dynamic probe for Altera FPGAs, you must install and license the software:

1 Install the FPGA dynamic probe software from the download web page at:

   "http://www.agilent.com/find/la-sw-download"

2 Follow the instructions on your entitlement certificate to redeem and install the FPGA dynamic probe license.

4
FPGA Design Steps

Altera LAI cores are inserted into FPGAs using Altera's Quartus II design software. For an example, see:

- "Example: Inserting an LAI Core" on page 48

For more information, see Altera's web site at " www.altera.com".

Next

- Preparation Steps (see page 17)

5
Preparation Steps

Before you can use the FPGA dynamic probe software with the *Agilent Logic Analyzer* application, you must take these preparation steps:

1 Install the JTAG cable programmer software (see page 18)

2 Set up the JTAG cable using the programmer software (see page 19)

Next

- Measurement Steps (see page 27)

Step 1. Install the JTAG cable programmer software

Before you can use the FPGA dynamic probe for Altera FPGAs, you must download and install the free Altera Quartus II Programmer software on the same logic analysis system or PC that the *Agilent Logic Analyzer* application runs on.

NOTE The PC that has the JTAG cable connection to the device under test does not have to be the logic analysis system or PC that the *Agilent Logic Analyzer* application runs on. In this case, the logic analysis system communicates via LAN with the system that has the JTAG server (and cable connection to the DUT).

1. Download the Altera Quartus II Programmer software from:

   "https://www.altera.com/support/software/download/programming/quartus2/dnl- quartus2_programmer.jsp"
2. Follow the installation instructions to install the software on the same logic analysis system or PC that the *Agilent Logic Analyzer* application runs on.
3. If the PC or logic analysis system that has the JTAG cable connection to the device under test is not the same computer that runs the *Agilent Logic Analyzer* application, and Quartus II design software is not already installed, install the Quartus II Programmer software there too.

Next

- Step 2. Set up the JTAG cable using the programmer software (see page 19)

Step 2. Set up the JTAG cable using the programmer software

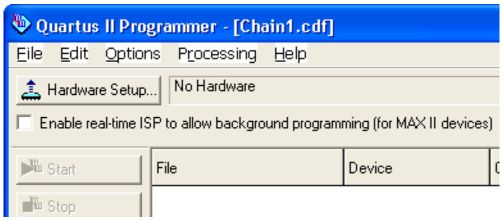
On the PC or logic analysis system that has the JTAG cable connection to the device under test:

1 If you are using the USB Blaster cable, driver files come with the Altera Quartus II Programmer software, but you must install them. See To install the USB Blaster device driver (see page 20).

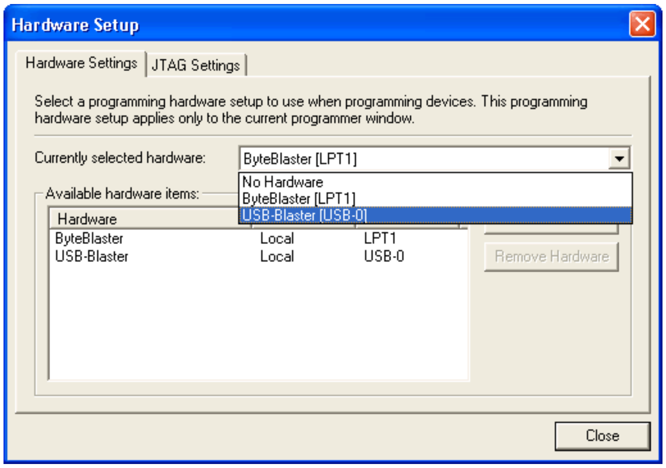
2 Open the Altera Quartus II Programmer (or design) software and set up the JTAG cable connection:

a From the Windows Start menu, choose **All Programs>Altera>Quartus II 7.2 Programmer and SignalTap II>Quartus II 7.2 Programmer**.

b In the Quartus II Programmer window, click **Hardware Setup...**.


Quartus II Programmer - [Chain1.cdf]
File Edit Options Processing Help
Hardware Setup...
No Hardware
Enable real-time ISP to allow background programming (for MAX II devices)
Start
Stop
File
Device

c In the Hardware Setup dialog, select the appropriate JTAG cable hardware.



Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: ByteBlaster [LPT1]

Hardware		
ByteBlaster	Local	LPT1
USB-Blaster	Local	USB-0

You may have to click **Add Hardware...** to add the appropriate JTAG cable hardware.

d After you have selected the appropriate JTAG cable hardware, click **Close** to close the Hardware Setup dialog.

e Choose **File>Exit** to close the Quartus II Programmer software.

If the PC or logic analysis system that has the JTAG cable connection to the device under test is not the same computer that will run the *Agilent Logic Analyzer* application, you must set up a JTAG server. See "To set up a JTAG Server" on page 21.

Next

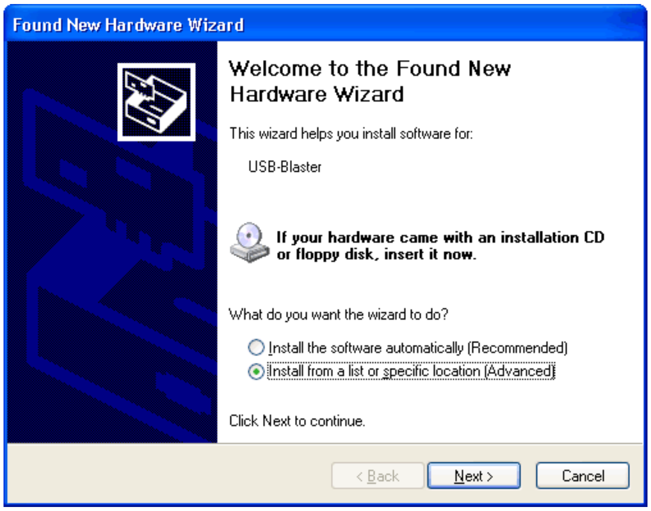
- Step 3. Establish connection between analyzer and JTAG cable (see page 28)

To install the USB Blaster device driver

On the PC or logic analysis system that has the JTAG cable connection to the device under test:

1 Connect the USB Blaster device to your computer.

2 In the Found New Hardware Wizard dialog, select **Install from a list or specific location**; then, click **Next >**.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

USB-Blaster

If your hardware came with an installation CD or floppy disk, insert it now.

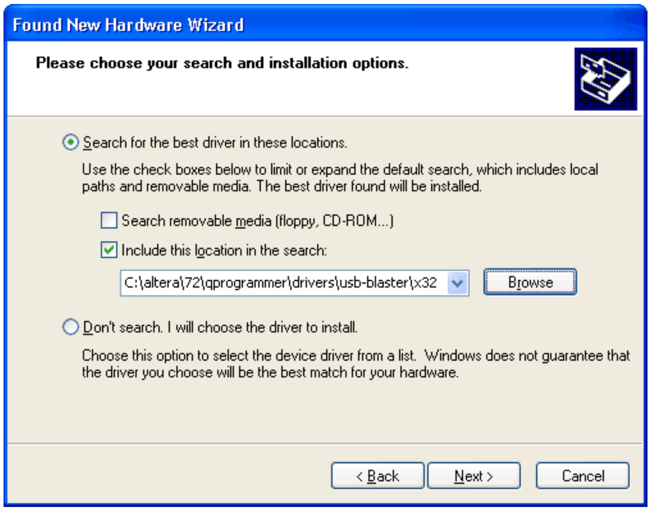
What do you want the wizard to do?

○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.

3 On the next page of the wizard, specify the **drivers\usb-blaster\x32** (for a 32-bit operating system) or **drivers\usb-blaster\x64** (for a 64-bit operating system) location underneath the Quartus II Programmer install directory; then, click **Next >**.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

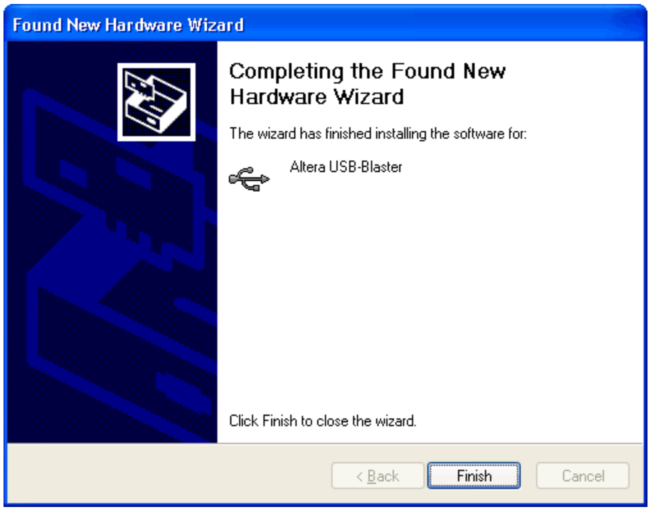
☑ Include this location in the search:

C:\altera\72\qprogrammer\drivers\usb-blaster\x32

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

4 When the wizard has finished installing the driver, click **Finish**.



Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Altera USB-Blaster

Click Finish to close the wizard.

For more information on installing the USB Blaster driver, see "http://www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-xp.html" (or search for "USB Blaster driver" on the Altera web site).

To set up a JTAG Server

The Altera FPGA Dynamic Probe can connect to JTAG cables over a TCP/IP connection. This is useful when you want to remotely connect to a logic analysis system that also has the JTAG cable connection to the device under test. In this case, the *Agilent Logic Analyzer* application (and the Altera FPGA Dynamic Probe software), must communicate remotely with the logic analysis system as well as the JTAG cable server.

If your JTAG cable is connected between the device under test and the logic analysis system or PC that is running the *Agilent Logic Analyzer* application, you do not need to set up a JTAG server.

To set up the Altera Programmer software to allow client/server connections to a remote JTAG cable:

1 Make sure the Altera Quartus II Programmer software is installed on both:

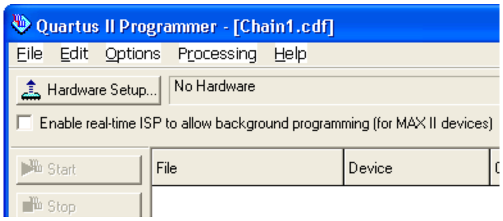
   - The *client computer* where you will be running the *Agilent Logic Analyzer* application (the JTAG Client).
   - The *remote computer* (logic analysis system or PC) that is physically connected to the JTAG cable (the JTAG Server).

   This should have already been done in "Step 1. Install the JTAG cable programmer software" on page 18.

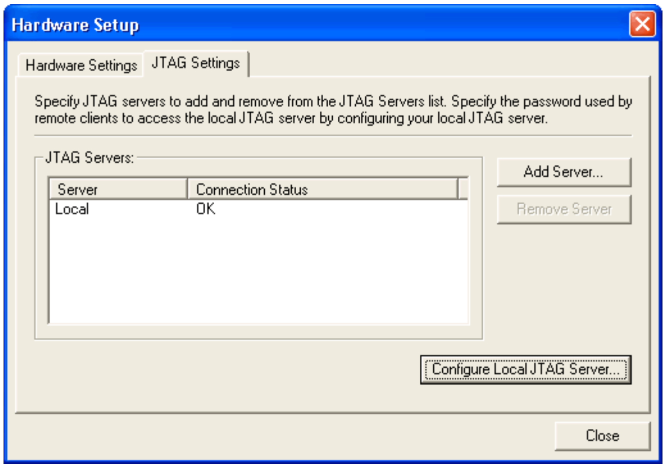
2 Set up the JTAG Server on the *remote computer*:

   a Start the Quartus II Programmer application from the Windows Start menu by choosing **All Programs>Altera>Quartus II 7.2 Programmer and SignalTap II>Quartus II 7.2 Programmer**.

   b In the Quartus II Programmer window, click **Hardware Setup...**.


Quartus II Programmer - [Chain1.cdf]
File Edit Options Processing Help
Hardware Setup...
No Hardware
Enable real-time ISP to allow background programming (for MAX II devices)
Start
Stop
File
Device

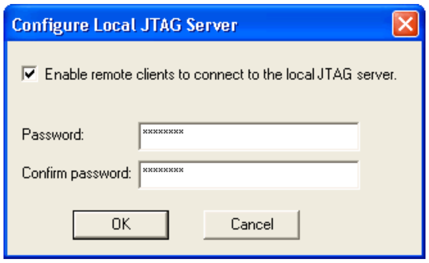
   c In the Hardware Setup dialog's JTAG Settings tab, click **Configure Local JTAG Server...**.



Specify JTAG servers to add and remove from the JTAG Servers list. Specify the password used by remote clients to access the local JTAG server by configuring your local JTAG server.

Server	Connection Status
Local	OK

d In the Configure local JTAG Server dialog, check **Enable remote clients to connect to the local JTAG server**, enter a password, and confirm the password.



☑ Enable remote clients to connect to the local JTAG server.

Password: ********

Confirm password: ********

Record this password as you will need it later. Then, click **OK**.

e Click **Close** to close the Hardware Setup dialog.

f Choose **File>Exit** to close the Quartus II Programmer software.

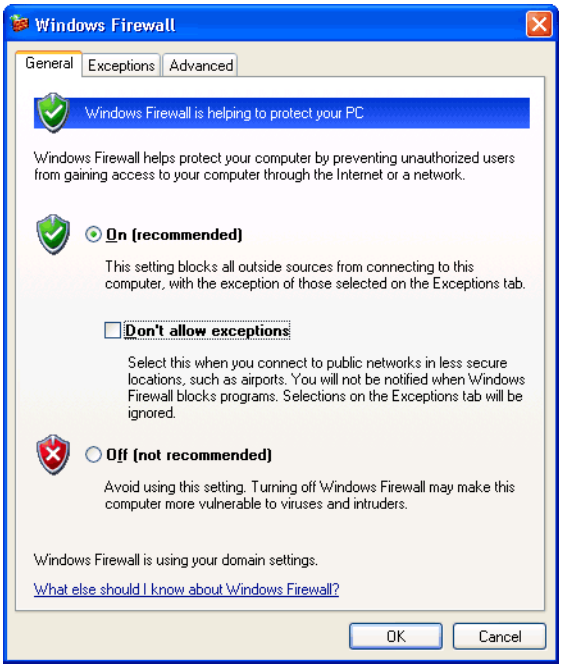
3 If the *remote computer* is running firewall software, allow access to the JTAG Server through the firewall. If the remote computer is not running firewall software, you can skip this step and proceed to step 4.

For example, if the *remote computer* is running Windows Firewall:

a Make sure you are logged in as an administrator.

b From the Windows Start menu, choose **Control Panel>Windows Firewall**.

c In the Windows Firewall dialog's General tab, uncheck **Don&#8217;t allow exceptions**.



Windows Firewall is helping to protect your PC

Windows Firewall helps protect your computer by preventing unauthorized users from gaining access to your computer through the Internet or a network.

● **On (recommended)**

This setting blocks all outside sources from connecting to this computer, with the exception of those selected on the Exceptions tab.

□ **Don't allow exceptions**

Select this when you connect to public networks in less secure locations, such as airports. You will not be notified when Windows Firewall blocks programs. Selections on the Exceptions tab will be ignored.

○ **Off (not recommended)**

Avoid using this setting. Turning off Windows Firewall may make this computer more vulnerable to viruses and intruders.

Windows Firewall is using your domain settings.

What else should I know about Windows Firewall?

If the firewall is off, it may be that the computer is using different firewall software. In this case, you must enable access to the Quartus II Programmer port using the different firewall software.

d In the Exceptions tab, click **Add Port...**.

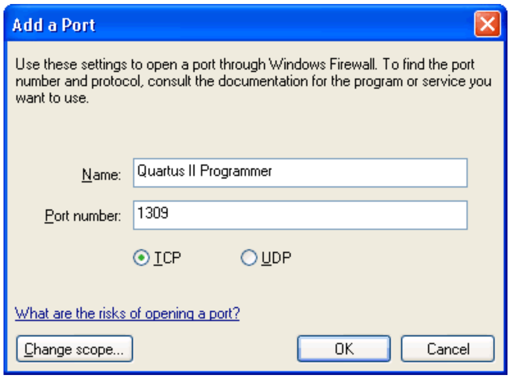
e In the Add a Port dialog:

i Enter the **Name**: Quartus II Programmer.

ii Enter the **Port number**: 1309.

iii Select **TCP**.

iv Click **OK** to close the Add a Port dialog.



Use these settings to open a port through Windows Firewall. To find the port number and protocol, consult the documentation for the program or service you want to use.

Name: Quartus II Programmer

Port number: 1309

● TCP ○ UDP

What are the risks of opening a port?

e Click **OK** to close the Windows Firewall dialog.

4 Get the hostname of the *remote computer* (JTAG Server):

a From the Windows Start menu, choose **Control Panel>System**.

b Click the **Computer Name** tab.

c Write down the "Full computer name". This is the hostname of your remote JTAG server which you will need shortly.

d Click **OK**.

5 Set up the JTAG Client on the *client computer*:

a Start the Quartus II Programmer application from the Windows Start menu by choosing **All Programs>Altera>Quartus II 7.2 Programmer and SignalTap II>Quartus II 7.2 Programmer**.

b In the Quartus II Programmer window, click **Hardware Setup...**.

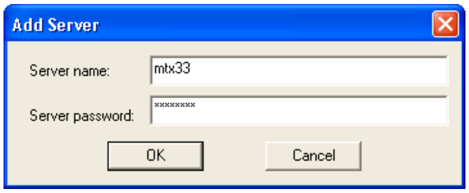
c In the Hardware Setup dialog's JTAG Settings tab, click **Add Server...**.

d In the Add Server dialog:

i Enter the full hostname of the remote JTAG Server you found in step 4.c as the **Server name**.

ii Enter the password you created in step 2.d as the **Server password**.

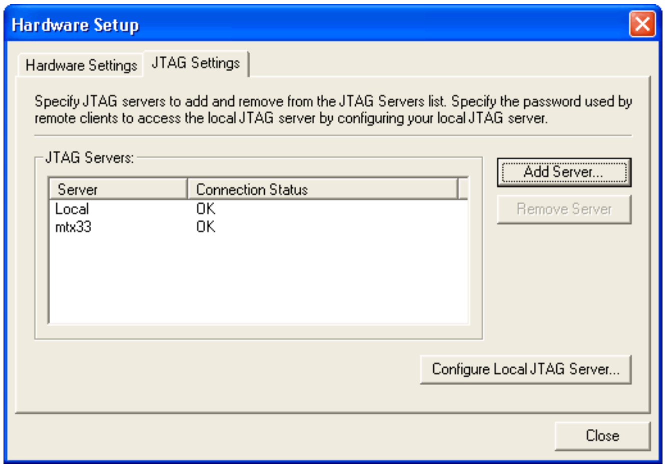
iii Click **OK**.



Server name: mtx33

Server password: ********

d Verify that the Hardware Setup dialog shows a Connection Status of "OK". If not, then there is a problem with the networking. Re-verify your firewall settings and server-name/server-password settings.



Specify JTAG servers to add and remove from the JTAG Servers list. Specify the password used by remote clients to access the local JTAG server by configuring your local JTAG server.

Server	Connection Status
Local	OK
mtx33	OK

- e Click **Close** to close the Hardware Setup dialog.
- f Choose **File>Exit** to close the Quartus II Programmer software.

Now, when you establish the connection between the analyzer and the JTAG cable (see "Step 3. Establish connection between analyzer and JTAG cable" on page 28), the remote JTAG cable will be listed in the Cable Connection dialog.

6
Measurement Steps

After you have completed the FPGA Design Steps (see page 15) and Preparation Steps (see page 17), you are ready to take these measurement steps in the *Agilent Logic Analyzer* application:

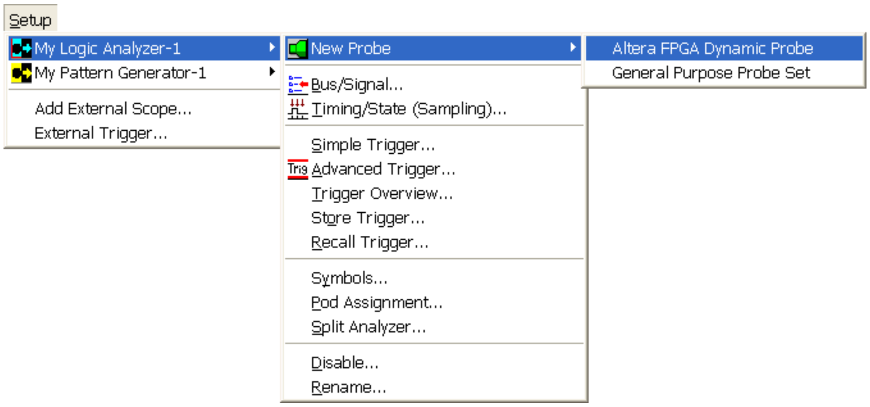
1. Establish connection between analyzer and JTAG cable (see page 28)
2. Download configuration bits into FPGA (see page 30)
3. Import signal names (see page 31)
4. Map FPGA pins (see page 35)
5. Make the measurement (see page 39)

Step 3. Establish connection between analyzer and JTAG cable

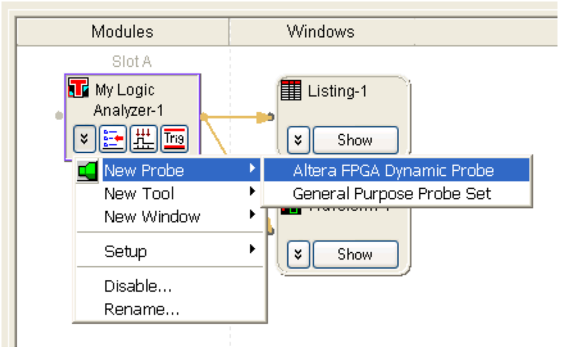
The FPGA dynamic probe application establishes a connection between the logic analyzer and a Xilinx cable. It also determines what devices are on the JTAG scan chain and lets you pick which one you wish to communicate.

To establish a connection between the logic analyzer and the Altera LAI core:

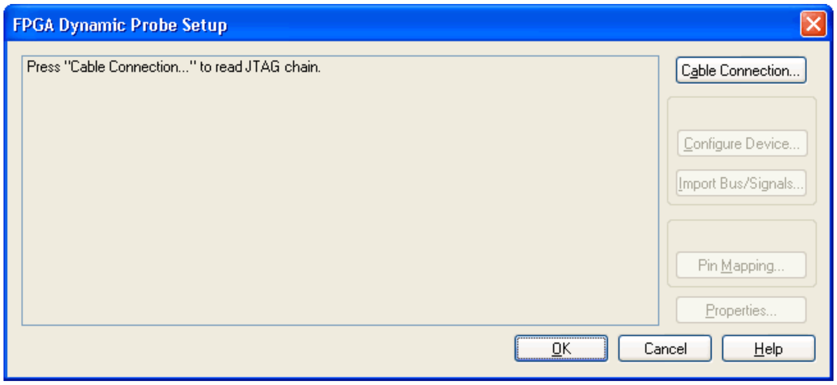
1 Add a new FPGA Dynamic Probe set by choosing **Setup>(Logic Analyzer Module)>New Probe>FPGA Dynamic Probe**.



Or, in the Overview window, from a module's drop-down menu, choose **New Probe>FPGA Dynamic Probe**.

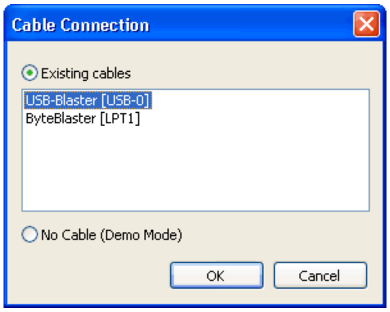


2 In the FPGA Dynamic Probe Setup dialog (see page 60), click **Cable Connection...**.



Press "Cable Connection..." to read JTAG chain.

3 In the Cable Connection dialog (see page 61), select the type of cable and, if necessary, specify any cable parameters; then, click **OK**.

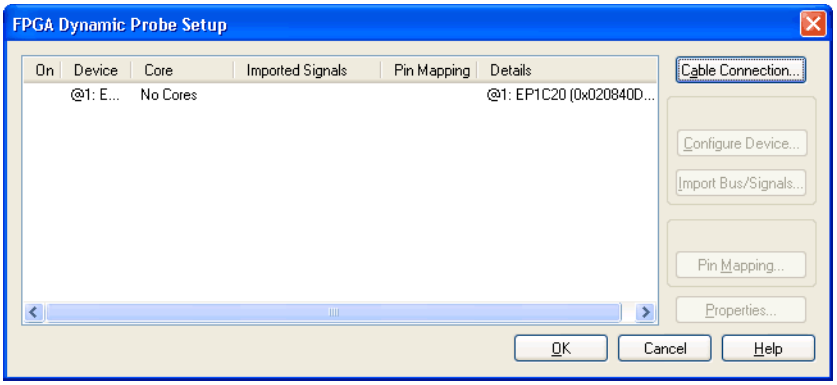


● Existing cables

USB-Blaster [USB-0]
ByteBlaster [LPT1]

○ No Cable (Demo Mode)

When the connection has been established, you will see the devices on the JTAG chain, and you can select the desired device.



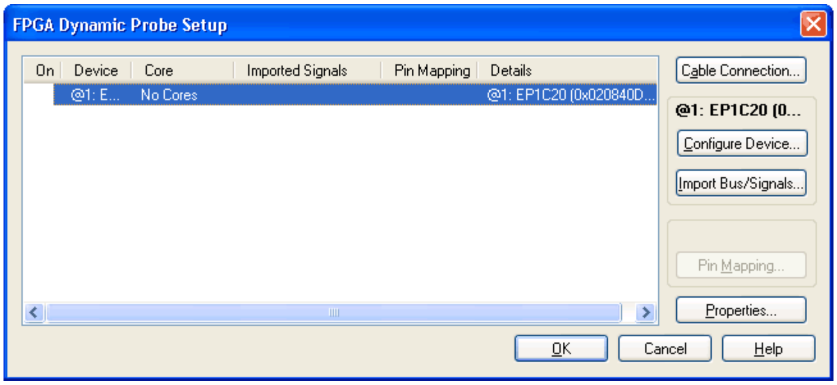
On	Device	Core	Imported Signals	Pin Mapping	Details
	@1: E...	No Cores			@1: EP1C20 (0x020840D...

Next • Step 4. Download configuration bits into FPGA (see page 30)

Step 4. Download configuration bits into FPGA

To download configuration bits into an FPGA:

1 In the FPGA Dynamic Probe Setup dialog (see page 60), select the FPGA device to which you wish to download configuration bits; then, click **Configure Device...**.



FPGA Dynamic Probe Setup

On	Device	Core	Imported Signals	Pin Mapping	Details
	@1: E...	No Cores			@1: EP1C20 (0x020840D...

@1: EP1C20 (0...

2 In the Select FPGA Configuration File dialog (see page 64), select the FPGA configuration file; then, click **Open**.

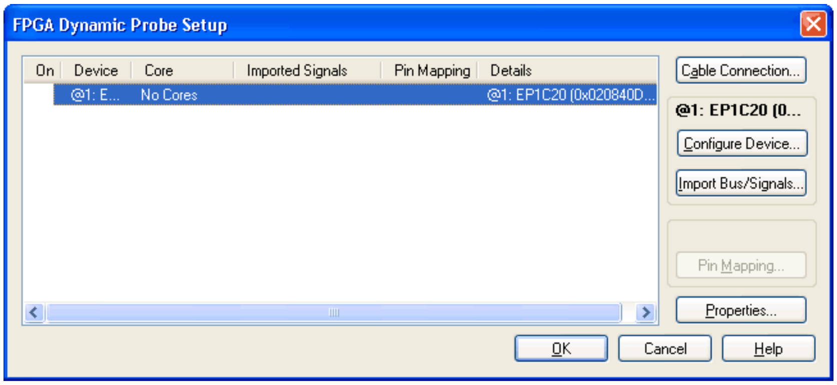
Next • Step 5. Import signal names (see page 31)

Step 5. Import signal names

The FPGA dynamic probe can automatically set up bus/signal names in the logic analyzer by reading a .lai file produced by Altera's Quartus II LAI (Logic Analyzer Interface) design software.

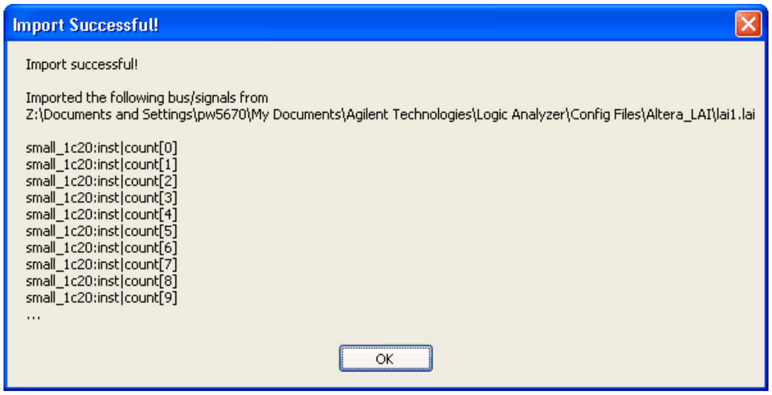
To import bus/signal names:

1 In the FPGA Dynamic Probe Setup dialog (see page 60), select the device whose bus/signal names you want to import; then, click **Import Bus/Signals...**.



2 In the Select Signal Import File dialog (see page 64), select the signal import file; then, click **Open**.

3 In the import results dialog, view the bus/signal import information; then, click **OK**.

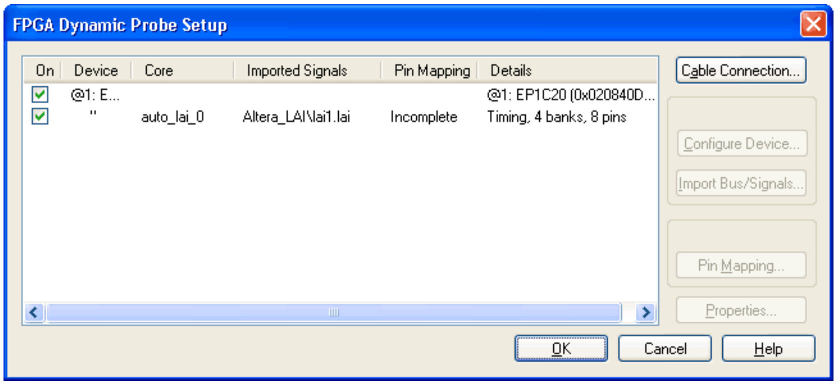


Import successful!

Imported the following bus/signals from
Z:\Documents and Settings\pw5670\My Documents\Agilent Technologies\Logic Analyzer\Config Files\Altera_LAI\lai1.lai

```
small_1c20:inst|count[0]
small_1c20:inst|count[1]
small_1c20:inst|count[2]
small_1c20:inst|count[3]
small_1c20:inst|count[4]
small_1c20:inst|count[5]
small_1c20:inst|count[6]
small_1c20:inst|count[7]
small_1c20:inst|count[8]
small_1c20:inst|count[9]
...
```

4 In the FPGA Dynamic Probe dialog, note that the defined cores now appear.



On	Device	Core	Imported Signals	Pin Mapping	Details
☑	@1: E...				@1: EP1C20 (0x020840D...
☑	"	auto_lai_0	Altera_LAI\lai1.lai	Incomplete	Timing, 4 banks, 8 pins

See Also

- To trim imported bus/signal names (see page 32)
- To rename imported bus/signal names (see page 32)
- To define additional FPGA bus/signal names (see page 33)

Next

- Step 6. Map FPGA pins (see page 35)

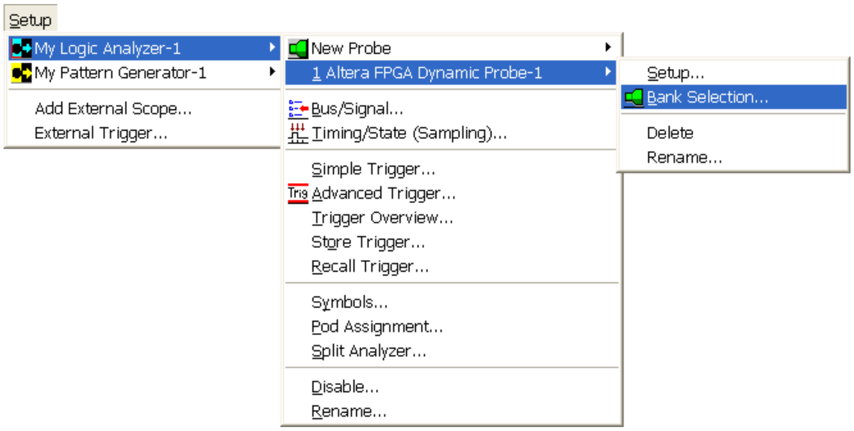
To trim imported bus/signal names

NOTE Trimming bus/signal names after initial bank selection may require manual insertion of bus/signals in Waveform and Listing display windows. We recommend that you trim bus/signal names before changing the bank selection.

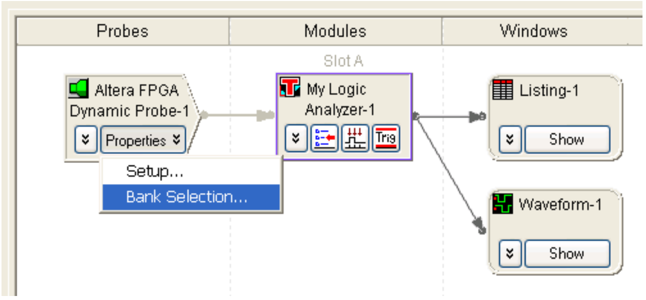
1. In the FPGA Dynamic Probe Bank Selection dialog (see page 67), click **Trim Bus/Signal Names...**.
2. In the Trim Bus/Signal Names dialog (see page 67), specify the bus/signal name characters to trim; then, click **OK**.

To rename imported bus/signal names

1. Open the Buses/Signals setup tab (see "Defining Buses and Signals" (in the online help)).
2. In the "FPGA Probe" bus/signal name folder, rename the bus/signal (see "To rename a bus or signal" (in the online help)).
3. Reopen the FPGA Dynamic Probe dialog by choosing **Setup>(Logic Analyzer Module)>(FPGA Dynamic Probe Name)>Bank Selection...**.



Or, in the Overview window, click the FGPA dynamic probe's **Properties** button; then, choose **Bank Selection...**.



Note also that you can triple-click signal names in the FPGA Dynamic Probe dialog to rename them (without having to do global trimming).

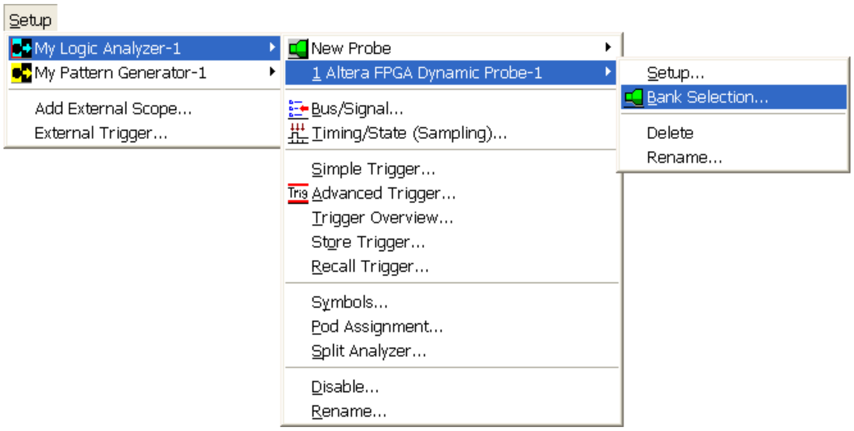
To define additional FPGA bus/signal names

1 Open the Buses/Signals setup tab (see "Defining Buses and Signals" (in the online help)).

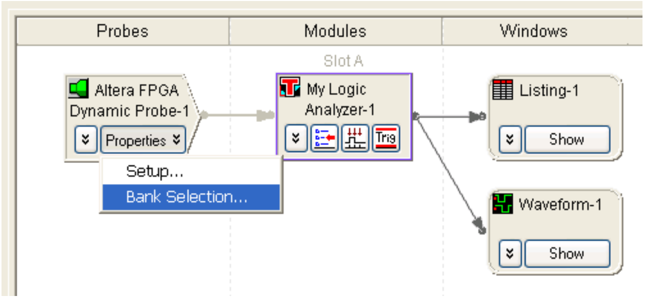
2 In the "FPGA Probe" bus/signal name folder, add a new bus/signal (see "To add a new bus or signal" (in the online help)).

3 Assign channels to the new bus/signal name (see "To assign channels in the default bit order" (in the online help) or "To assign channels, selecting the bit order" (in the online help)).

4 Reopen the FPGA Dynamic Probe dialog by choosing **Setup>(Logic Analyzer Module)>(FPGA Dynamic Probe Name)>Bank Selection...**.



Or, in the Overview window, click the FGPA dynamic probe's **Properties** button; then, choose **Bank Selection...**.



Whenever buses/signals are added to the "FPGA Probe" folder, they are associated with a specific bank. If you select another bank, the added buses/signals do not appear.

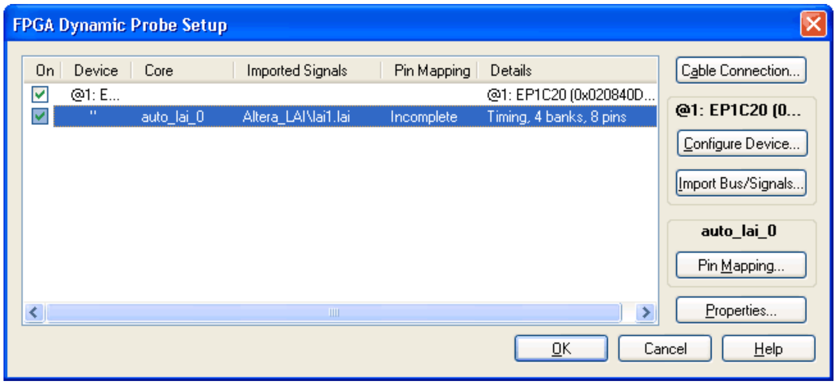
If you want to define buses/signals that apply to all banks, create them outside of the "FPGA Probe" folder. That way, the buses/signals are not associated with a bank.

Step 6. Map FPGA pins

Quickly specify how the FPGA pins (the signal outputs of Altera LAI) are connected to your logic analyzer. Select your probe type and rapidly provide the information needed for the logic analyzer to automatically track names of signals routed through the Altera LAI core.

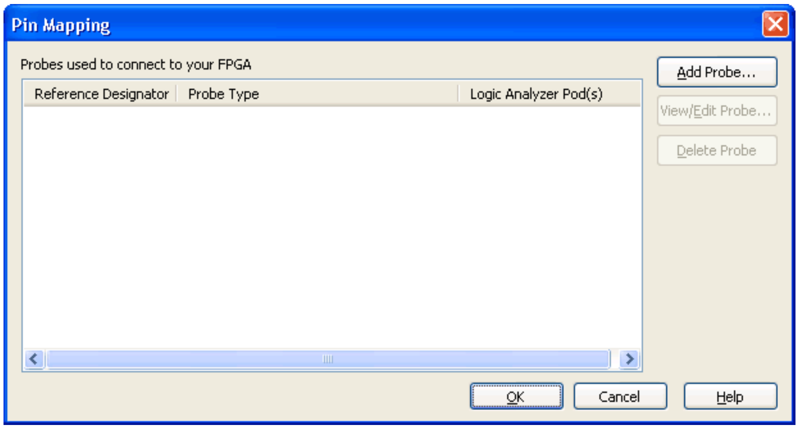
To map FPGA pins to logic analyzer probes:

1 In the FPGA Dynamic Probe Setup dialog (see page 60), select the Altera LAI core whose output pins you want to map; then, click **Pin Mapping...**.



On	Device	Core	Imported Signals	Pin Mapping	Details
☑	@1: E...				@1: EP1C20 (0x020840D...
☑	"	auto_lai_0	Altera_LAI\lai1.lai	Incomplete	Timing, 4 banks, 8 pins

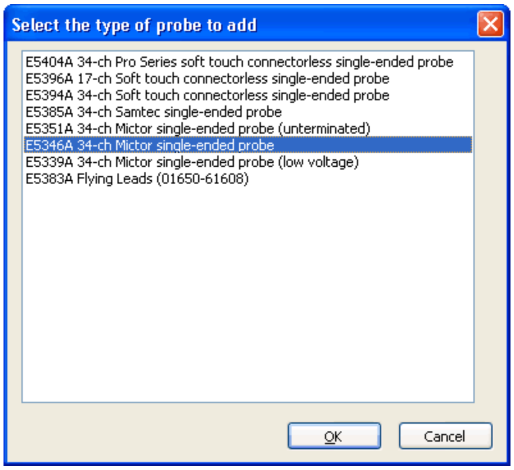
2 In the Pin Mapping dialog (see page 61), click **Add Probe...**.



Probes used to connect to your FPGA

Reference Designator	Probe Type	Logic Analyzer Pod(s)

3 In the "Select the type of probe to add" dialog, select the type of probe that is used to connect to your FPGA; then, click **OK**.

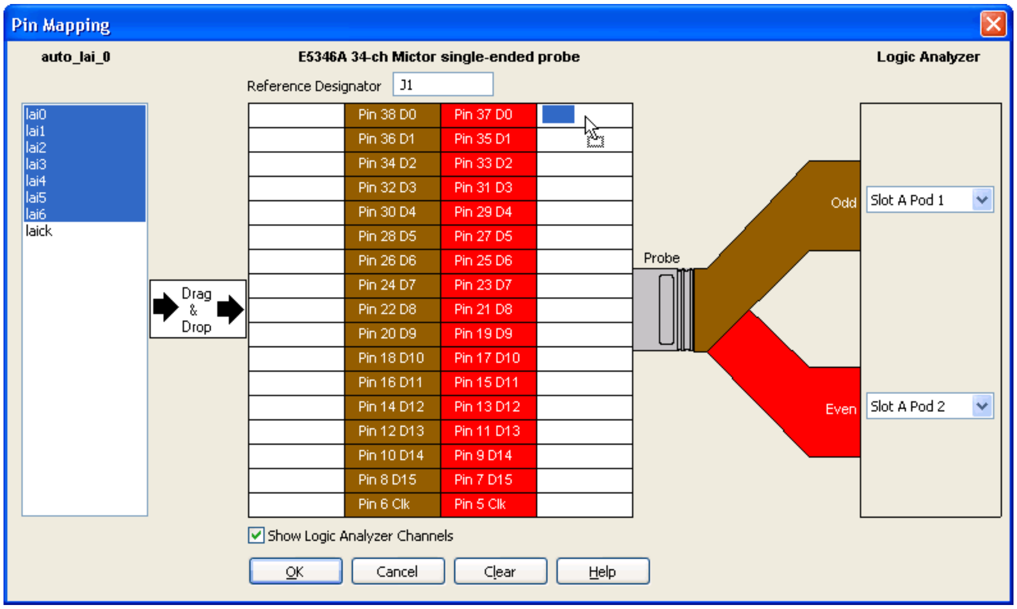


Select the type of probe to add

E5404A 34-ch Pro Series soft touch connectorless single-ended probe
E5396A 17-ch Soft touch connectorless single-ended probe
E5394A 34-ch Soft touch connectorless single-ended probe
E5385A 34-ch Samtec single-ended probe
E5351A 34-ch Mictor single-ended probe (unterminated)
E5346A 34-ch Mictor single-ended probe
E5339A 34-ch Mictor single-ended probe (low voltage)
E5383A Flying Leads (01650-61608)

If your probe doesn't appear in the list, you can "download the latest probe definitions from the web" (in the online help).

4 In the Pin Mapping Edit dialog (see page 62), select the FPGA pins (you can select multiple pins using Shift-click or Ctrl-click) and drag them on to the pin/pad map.



Pin Mapping

auto_lai_0 E5346A 34-ch Mictor single-ended probe Logic Analyzer

Reference Designator J1

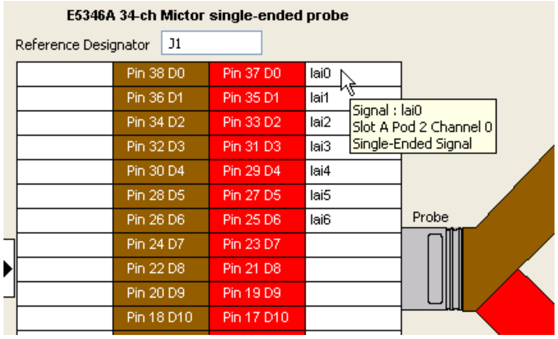
lai0
lai1
lai2
lai3
lai4
lai5
lai6
laick

	Pin 38 D0	Pin 37 D0	
	Pin 36 D1	Pin 35 D1	
	Pin 34 D2	Pin 33 D2	
	Pin 32 D3	Pin 31 D3	
	Pin 30 D4	Pin 29 D4	
	Pin 28 D5	Pin 27 D5	
	Pin 26 D6	Pin 25 D6	
	Pin 24 D7	Pin 23 D7	
	Pin 22 D8	Pin 21 D8	
	Pin 20 D9	Pin 19 D9	
	Pin 18 D10	Pin 17 D10	
	Pin 16 D11	Pin 15 D11	
	Pin 14 D12	Pin 13 D12	
	Pin 12 D13	Pin 11 D13	
	Pin 10 D14	Pin 9 D14	
	Pin 8 D15	Pin 7 D15	
	Pin 6 Clk	Pin 5 Clk	

Odd: Slot A Pod 1
Even: Slot A Pod 2

☑ Show Logic Analyzer Channels

After you've mapped FPGA pins to the probe, you can hover the mouse pointer over a pin description field to view a tool tip describing the FPGA debug pin name, the pod connection, the channel number, and the signal type (single-ended or differential).



E5346A 34-ch Mictor single-ended probe

Reference Designator J1

	Pin 38 D0	Pin 37 D0	lai0
	Pin 36 D1	Pin 35 D1	lai1
	Pin 34 D2	Pin 33 D2	lai2
	Pin 32 D3	Pin 31 D3	lai3
	Pin 30 D4	Pin 29 D4	lai4
	Pin 28 D5	Pin 27 D5	lai5
	Pin 26 D6	Pin 25 D6	lai6
	Pin 24 D7	Pin 23 D7	
	Pin 22 D8	Pin 21 D8	
	Pin 20 D9	Pin 19 D9	
	Pin 18 D10	Pin 17 D10	

Signal : lai0
Slot A Pod 2 Channel 0
Single-Ended Signal

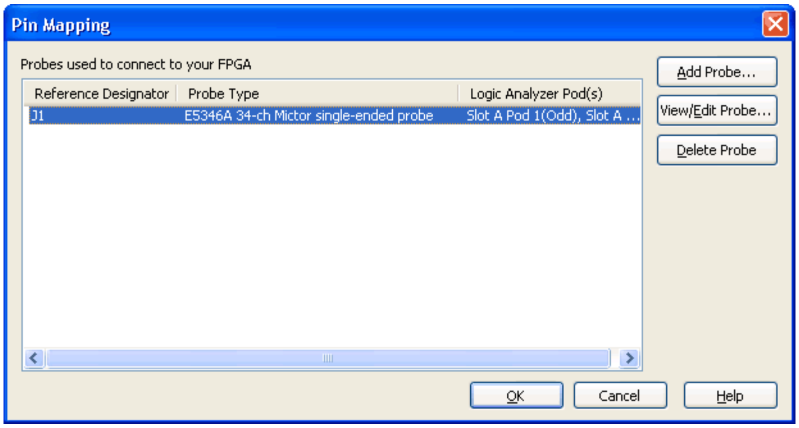
You can clear all FPGA pins that have been mapped to pins/pads by clicking **Clear**. You can clear individual pin mappings by dragging a pin from the pin/pad diagram back to the FPGA Pins list.

5 Select the logic analyzer pods that the probe is connected to.

When you wish to map multiple cores using different halves of the same probe, make sure you select **None** for the half that will be used by the other core.

NOTE For state (synchronous) cores, you need to make sure that the ATCK pin maps to one of the "Clk" pin/pad locations (which identify clock signal inputs) and that the associated logic analyzer pod is valid for clock inputs. (In the *General State Mode*, the clock lines on the first 4 pods of a logic analyzer can be used as clock inputs; in the *Turbo State Mode*, the clock line on the first pod can be used as a clock input.)

6 When you are done mapping FPGA pins, click **OK**. Note that your probe has been added to the list in the Pin Mapping dialog.



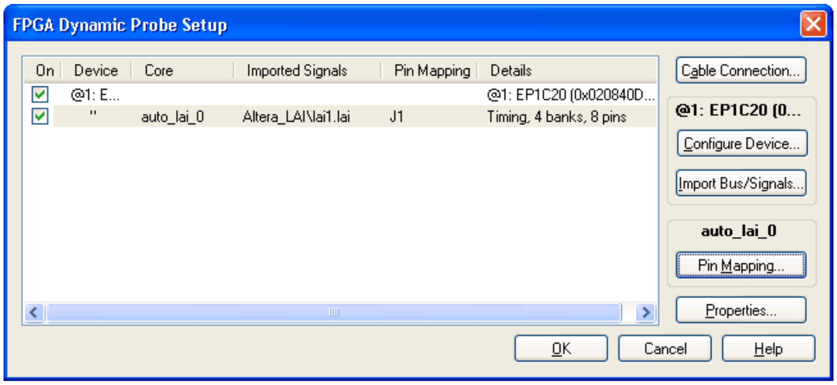
Pin Mapping

Probes used to connect to your FPGA

Reference Designator	Probe Type	Logic Analyzer Pod(s)
J1	E5346A 34-ch Mictor single-ended probe	Slot A Pod 1(Odd), Slot A ...

7 Click **OK** to close the Pin Mapping dialog.

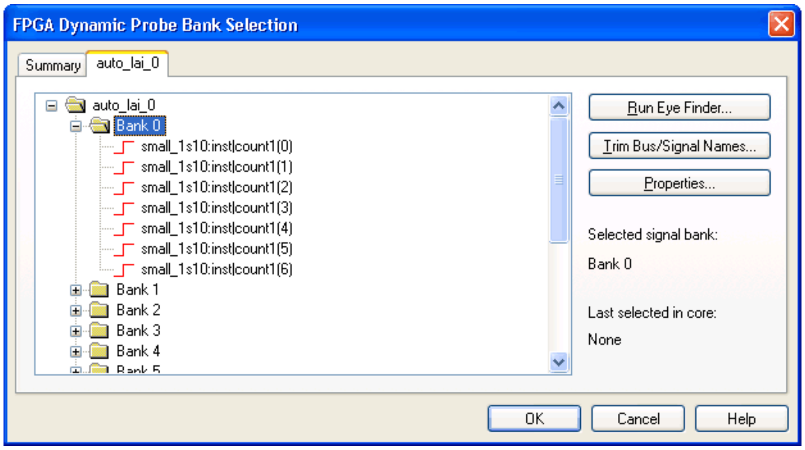
In the FPGA Dynamic Probe Setup dialog, notice pin mapping is no longer "Incomplete".



On	Device	Core	Imported Signals	Pin Mapping	Details
☑	@1: E...				@1: EP1C20 (0x020840D...
☑	"	auto_lai_0	Altera_LAI\lai1.lai	J1	Timing, 4 banks, 8 pins

8 Click **OK** to close the FPGA Dynamic Probe Setup dialog.

The FPGA Dynamic Probe dialog opens automatically. If you expand a bank, you see the imported bus/signal names.


FPGA Dynamic Probe Bank Selection
Summary
auto_lai_0
auto_lai_0
Bank 0
small_1s10:inst|count1(0)
small_1s10:inst|count1(1)
small_1s10:inst|count1(2)
small_1s10:inst|count1(3)
small_1s10:inst|count1(4)
small_1s10:inst|count1(5)
small_1s10:inst|count1(6)
Bank 1
Bank 2
Bank 3
Bank 4
Bank 5
Run Eye Finder...
Trim Bus/Signal Names...
Properties...
OK
Cancel
Help

Selected signal bank:

Bank 0

Last selected in core:

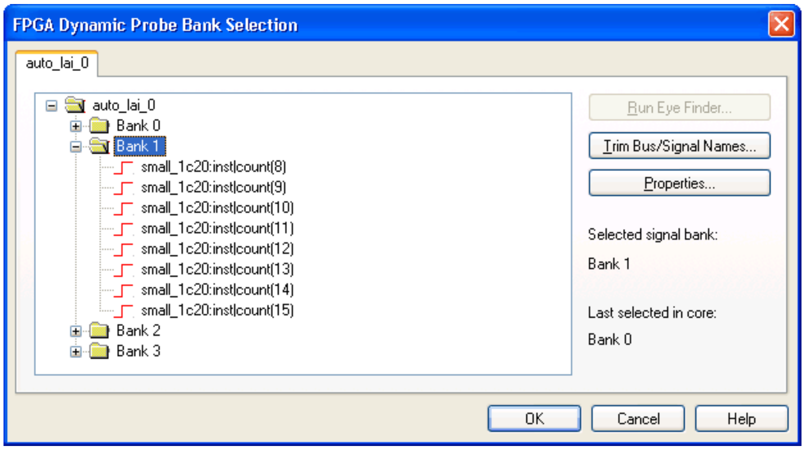
None

Next • Step 7. Make the measurement (see page 39)

Step 7. Make the measurement

At this point, you are ready to use the logic analyzer (as you would normally) to capture activity on internal FPGA signals.

You can tell the Altera LAI core to switch signal banks without affecting the timing of your design. When viewing the Probes toolbar (**View>Toolbars>Probes**), click to open the FPGA Dynamic Probe dialog. Then, select the signal bank to be routed to the logic analyzer and click **OK**. You can change signal banks as often as needed to make measurements throughout your FPGA.


FPGA Dynamic Probe Bank Selection

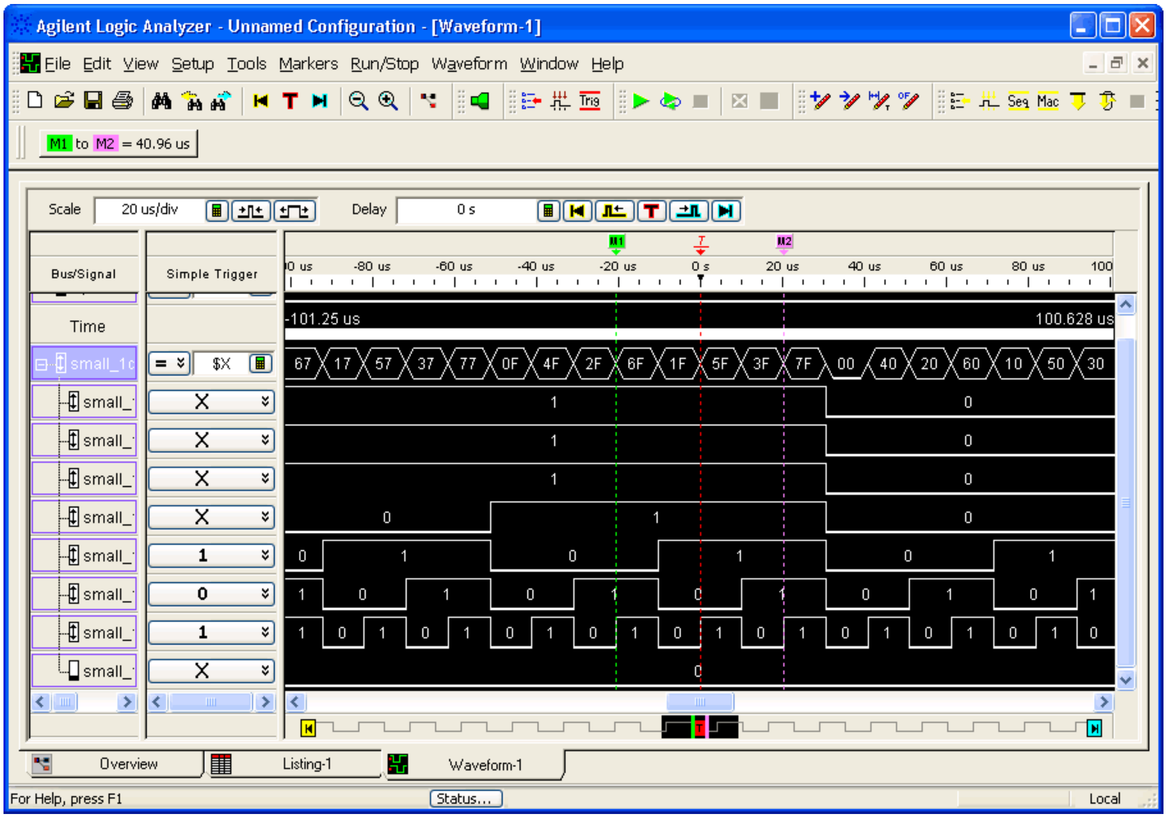
auto_lai_0

- auto_lai_0
  - Bank 0
  - Bank 1
    - small_1c20:inst|count(8)
    - small_1c20:inst|count(9)
    - small_1c20:inst|count(10)
    - small_1c20:inst|count(11)
    - small_1c20:inst|count(12)
    - small_1c20:inst|count(13)
    - small_1c20:inst|count(14)
    - small_1c20:inst|count(15)
  - Bank 2
  - Bank 3

Selected signal bank:
Bank 1

Last selected in core:
Bank 0

You can correlate internal FPGA activity with external measurements. With each new selection of a signal bank, the application updates new signal names from your design to the logic analyzer. View internal FPGA activity and time correlate internal FPGA measurements with external events in the surrounding system.


Agilent Logic Analyzer - Unnamed Configuration - [Waveform-1]
File Edit View Setup Tools Markers Run/Stop Waveform Window Help
Bus/Signal
Simple Trigger
Time
small_1c
small_
Overview
Listing-1
Waveform-1
For Help, press F1
Status...
Local

M1 to M2 = 40.96 us

Scale 20 us/div

Delay 0 s

-101.25 us

100.628 us

0 us
-80 us
-60 us
-40 us
-20 us
0 s
20 us
40 us
60 us
80 us
100
67 17 57 37 77 0F 4F 2F 6F 1F 5F 3F 7F 00 40 20 60 10 50 30

NOTE Timing zoom is automatically disabled when using the FPGA dynamic probe. You can re-enable timing zoom; however, because of the Altera LAI core, timing zoom does not provide an accurate representation of internal FPGA signals.

NOTE Captured data is invalidated whenever you:

- Select a different bank.
- Select a different core.
- Download configuration bits into an FPGA.
- Reopen a cable connection.
- Imported signal names.
- Trim imported bus/signal names.
- Change the FPGA pin mapping.

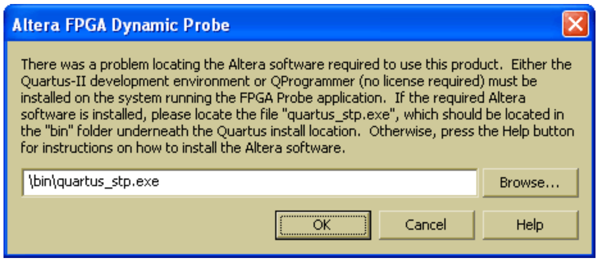
See Also

- "Capturing Data from the Device Under Test" (in the online help)
- "Analyzing the Captured Data" (in the online help)

7
FPGA Dynamic Probe Troubleshooting

- When I click the 'Cable Connection' button I get 'problem locating Altera software' dialog (see page 42)
- When I click the 'Cable Connection' button I get 'script error' dialog (see page 43)
- If you don't see activity in the logic analyzer (see page 44)
- If state mode measurements don't work (see page 45)

When I click the 'Cable Connection' button I get 'problem locating Altera software' dialog



Altera FPGA Dynamic Probe

There was a problem locating the Altera software required to use this product. Either the Quartus-II development environment or QProgrammer (no license required) must be installed on the system running the FPGA Probe application. If the required Altera software is installed, please locate the file "quartus_stp.exe", which should be located in the "bin" folder underneath the Quartus install location. Otherwise, press the Help button for instructions on how to install the Altera software.

\bin\quartus_stp.exe

Possible cause: Default install path was changed during Quartus II installation. Actions:

1 Follow the instructions displayed on the dialog. If the file 'quartus_stp.exe' does not exist anywhere on the machine, continue with step 2.

2 Close the error and 'Cable Connection' dialogs.

3 Use Windows Add/Remove Programs to "Modify" the Quartus II installation, checking both the QProgrammer and SignalTap boxes before confirming. Reboot.

When I click the 'Cable Connection' button I get 'script error' dialog

NOTE The blue light on the USB Blaster should blink when it is being identified.

Possible causes:

- There are no connected cables. Actions:
  - **a** Close the error and 'Cable Connection' dialogs.
  - **b** Re-seat the connection between the oscilloscope or PC and the Altera cable.
  - **c** Retry the 'Cable Connection' button.
- Cable driver installation is incomplete. Actions:
  - **a** Close the error and 'Cable Connection' dialogs.
  - **b** Verify: Windows Device Manager should show a USB controller named "Altera USB-Blaster". If this controller has a red X, it is disabled. Right-click and select 'Enable'. If it has a yellow exclamation mark, then the Altera driver has not been fully installed. There is probably a New Hardware Found dialog hidden beneath the display. Complete the instructions on that dialog. See To install the USB Blaster device driver (see page 20)
  - **c** Retry the 'Cable Connection' button.
- Selected cable not connected to DUT (device under test). Actions:
  - **a** Close the error and 'Cable Connection' dialogs.
  - **b** Re-seat connection between Altera cable and DUT (ensure plug on board is not reversed).
  - **c** Retry the 'Cable Connection' button.
- DUT is not powered up. Actions:
  - **a** Close the error dialog and 'Cable Connection' dialog.
  - **b** Power up DUT.
  - **c** Retry the 'Cable Connection' button.

If you don't see activity in the logic analyzer

If you get a dynamic status from the core that says everything is enabled and ready, but you see no activity on your logic analyzer, check your connections to and from your device under test to the logic analyzer pod cables.

If state mode measurements don't work

If you are unable to capture data in state mode, either look at the clock activity indicators and select the appropriate clock for state measurements, or make a timing measurement to determine which clock is the master clock.

8
FPGA Dynamic Probe Concepts

- "Example: Inserting an LAI Core" on page 48
- "Automated Logic Analyzer Set Up" on page 58

Example: Inserting an LAI Core

In this example, to create designs with the Logic Analyzer Interface (LAI) core, you need to download the Altera Quartus II Web Edition from:

"https://www.altera.com/support/software/download/altera_design/quartus_we/dnl-quartus_we.jsp"

The Quartus II software requires a free license that you can request from:

"https://www.altera.com/support/licensing/free_software/lic-q2web.jsp"

Here are the steps to insert a LAI core into your design:

1 "Creating a Logic Analyzer Interface" on page 48

2 "Configuring the Logic Analyzer Interface Core Parameters" on page 49

3 "Mapping the Clock Signal" on page 50

4 "Mapping the Logic Analyzer Interface File Pins to Available I/O Pins" on page 51

5 "Mapping Internal Signals to the Logic Analyzer Interface Banks" on page 53

6 "Saving the Logic Analyzer Interface" on page 55

7 "Enabling/Disabling the Logic Analyzer Interface" on page 55

8 "Compiling the Quartus II Project" on page 56

9 "FYI: Creating Multiple Logic Analyzer Interface Instances in One FPGA" on page 57

Creating a Logic Analyzer Interface

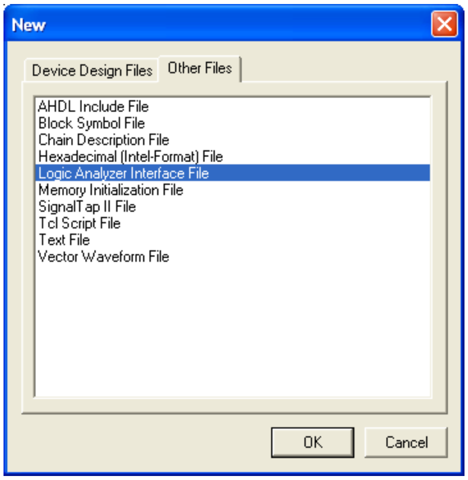
The Logic Analyzer Interface File (.lai) defines the interface between internal FPGA signals and the external logic analyzer. To define the Quartus II Logic Analyzer Interface, you can create a new LAI file or use an existing file. In this walkthrough, you open an existing design and add an LAI core.

To create a new Logic Analyzer Interface file, perform the following steps:

1 Double-click the Quartus II design file, small.qpf, located at C:\S800_altera

2 In the Quartus II **File** menu, click **New**.

3 In the New dialog, click the **Other Files** tab and select **Logic Analyzer Interface File**.



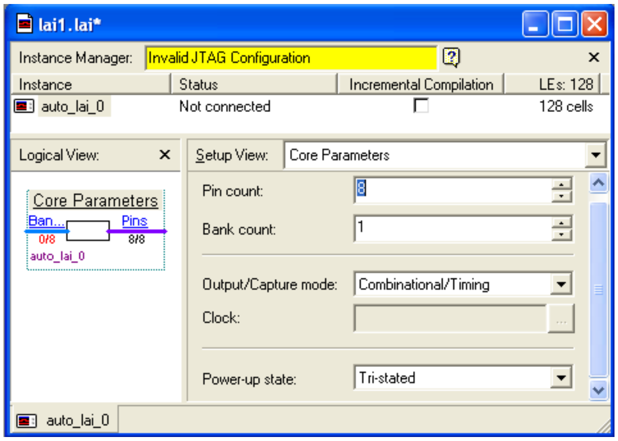
4 Click **OK**.

The Logic Analyzer Interface editor opens.

Next • "Configuring the Logic Analyzer Interface Core Parameters" on page 49

Configuring the Logic Analyzer Interface Core Parameters

After creating the LAI file, you must configure the Logic Analyzer Interface core parameters. To configure these parameters, from the **Setup View** list select Core Parameters.



Instance Manager: Invalid JTAG Configuration

Instance	Status	Incremental Compilation	LEs: 128
auto_lai_0	Not connected	□	128 cells

Setup View: Core Parameters

Pin count: 8

Bank count: 1

Output/Capture mode: Combinational/Timing

Clock:

Power-up state: Tri-stated

auto_lai_0

NOTE Screen resolution might cause your screen to look different; please resize or scroll within the window to view the parameter.

The parameters can be set depending on the user&#8217;s debug resources. In this example, the following parameters are used:

- **Pin count** = 13
- **Bank count** = 4
- **Capture mode** = State
- **Power- up state** = tristate

The Node Finder tool will be used in the next step to map the Clock signal.

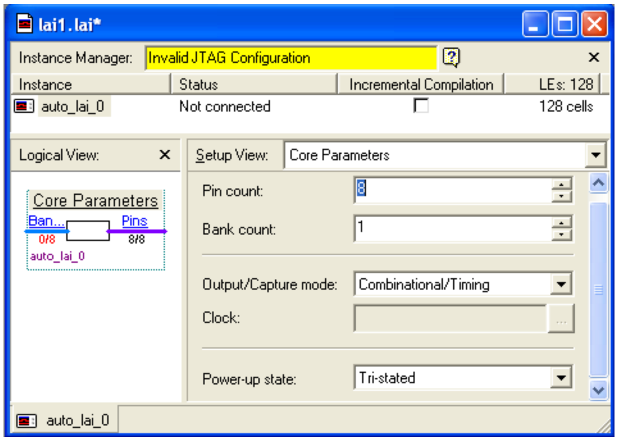
Parameter	Description
Pin Count	Signifies the number of pins dedicated to the Logic Analyzer Interface. The external pins connected to the users&#8217; debug header on the board. The Pin Count parameter can range from 1 to 256 pins.
Bank Count	Signifies the number of internal signals mapped to each pin. A Bank Count of 8 implies you will connect eight internal signals to each pin. The Bank Count parameter can range from 1 to 256 banks.
Output/Capture Mode	Signifies the type of acquisition the external logic analyzer will perform. There are two options to select from: • Combinational/Timing – This acquisition uses the logic analyzer&#8217;s internal clock to determine when to sample data. • Registered/State – This acquisition uses a signal from the system under test to determine when to sample. Registered/State acquisition samples data synchronously providing a functional view of the FGPA.
Clock	The clock parameter is available when the Output/Capture Mode is set to Registered State. The sample clock can be any signal in the design.
Power-Up State	The Power-Up State parameter specifies the power-up state of the Logic Analyzer Interface pins. The two options are tri-stated for all pins or selection a particular bank.

Next

- "Mapping the Clock Signal" on page 50

Mapping the Clock Signal

To assign the Clock, which will synchronize the state capture, click the ... button. This will launch the Node Finder tool.



Instance Manager: Invalid JTAG Configuration

Instance	Status	Incremental Compilation	LEs: 128
auto_lai_0	Not connected	□	128 cells

Setup View: Core Parameters

Pin count: 8

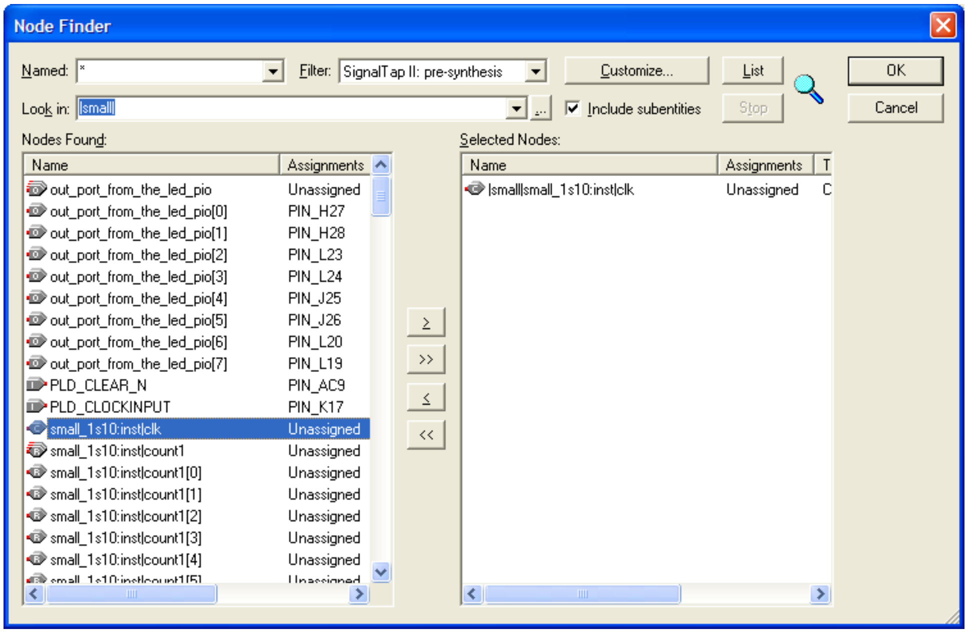
Bank count: 1

Output/Capture mode: Combinational/Timing

Clock:

Power-up state: Tri-stated

Click **List** to display the signals from the synthesized design. Search through signal names and locate the clock signal, small_1s10:inst|clk. When the signal is selected, click the **&#8805;** button to make the core assignment. Click **OK** to continue.



Named: *

Filter: SignalTap II: pre-synthesis

Look in: |small

☑ Include subentities

Nodes Found:

Name	Assignments
out_port_from_the_led_pio	Unassigned
out_port_from_the_led_pio[0]	PIN_H27
out_port_from_the_led_pio[1]	PIN_H28
out_port_from_the_led_pio[2]	PIN_L23
out_port_from_the_led_pio[3]	PIN_L24
out_port_from_the_led_pio[4]	PIN_J25
out_port_from_the_led_pio[5]	PIN_J26
out_port_from_the_led_pio[6]	PIN_L20
out_port_from_the_led_pio[7]	PIN_L19
PLD_CLEAR_N	PIN_AC9
PLD_CLOCKINPUT	PIN_K17
small_1s10:inst|clk	Unassigned
small_1s10:inst|count1	Unassigned
small_1s10:inst|count1[0]	Unassigned
small_1s10:inst|count1[1]	Unassigned
small_1s10:inst|count1[2]	Unassigned
small_1s10:inst|count1[3]	Unassigned
small_1s10:inst|count1[4]	Unassigned
small_1s10:inst|count1[5]	Unassigned

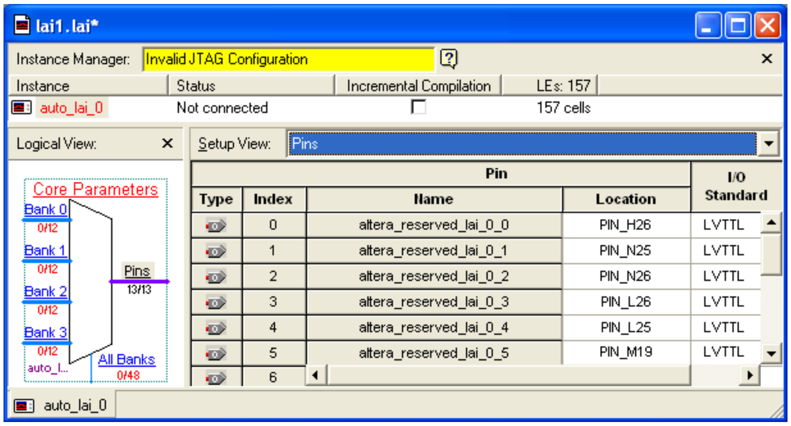
Selected Nodes:

Name	Assignments	T
|small|small_1s10:inst|clk	Unassigned	C

Next • "Mapping the Logic Analyzer Interface File Pins to Available I/O Pins" on page 51

Mapping the Logic Analyzer Interface File Pins to Available I/O Pins

To configure the LAI I/O pins, from the **Setup View** list select **Pins**.



lai1.lai*

Instance Manager: Invalid JTAG Configuration

Instance	Status	Incremental Compilation	LEs: 157
auto_lai_0	Not connected	□	157 cells

Setup View: Pins

Pin				I/O Standard
Type	Index	Name	Location	
	0	altera_reserved_lai_0_0	PIN_H26	LVTTL
	1	altera_reserved_lai_0_1	PIN_N25	LVTTL
	2	altera_reserved_lai_0_2	PIN_N26	LVTTL
	3	altera_reserved_lai_0_3	PIN_L26	LVTTL
	4	altera_reserved_lai_0_4	PIN_L25	LVTTL
	5	altera_reserved_lai_0_5	PIN_M19	LVTTL
	6			

To assign pin locations double-click inside the **Location** column next to the reserved pins in the **Name** column. This action opens the Pin Planner tool.

NOTE

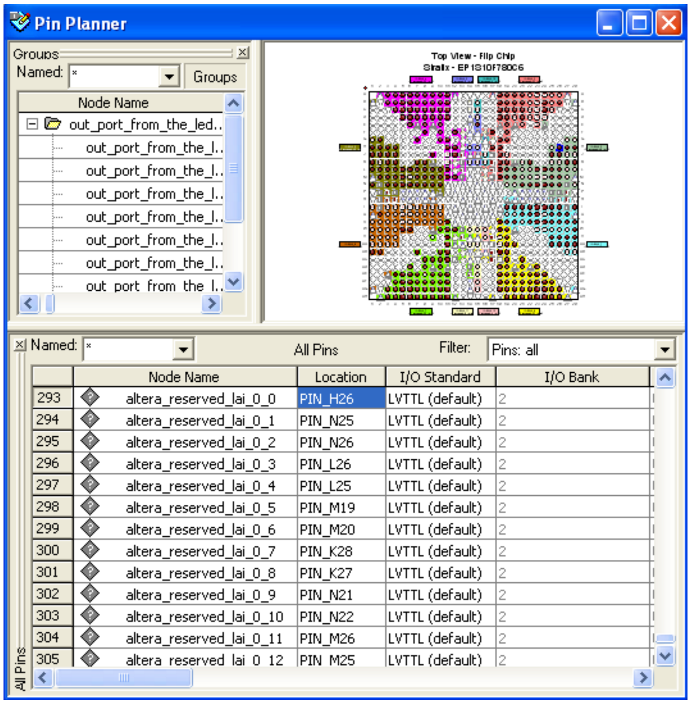
Please resize the window as necessary.

You can assign pins to the LAI by double-clicking in the **Location** column within Pin Planner. The pin assignment is typed in as PIN_XYZ or selected from the drop-down list.

The pins are assigned according to the following table.

Node Name	Location	I/O Standard
altera_reserved_lai_0_0	PIN_H26	LVTTL
altera_reserved_lai_0_1	PIN_N25	LVTTL
altera_reserved_lai_0_2	PIN_N26	LVTTL
altera_reserved_lai_0_3	PIN_L26	LVTTL
altera_reserved_lai_0_4	PIN_L25	LVTTL
altera_reserved_lai_0_5	PIN_M19	LVTTL
altera_reserved_lai_0_6	PIN_M20	LVTTL
altera_reserved_lai_0_7	PIN_K28	LVTTL
altera_reserved_lai_0_8	PIN_K27	LVTTL
altera_reserved_lai_0_9	PIN_N21	LVTTL
altera_reserved_lai_0_10	PIN_N22	LVTTL

Node Name	Location	I/O Standard
altera_reserved_lai_0_11	PIN_M26	LVTTL
altera_reserved_lai_0_12	PIN_M25	LVTTL



		Node Name	Location	I/O Standard	I/O Bank
293	◈	altera_reserved_lai_0_0	PIN_H26	LVTTL (default)	2
294	◈	altera_reserved_lai_0_1	PIN_N25	LVTTL (default)	2
295	◈	altera_reserved_lai_0_2	PIN_N26	LVTTL (default)	2
296	◈	altera_reserved_lai_0_3	PIN_L26	LVTTL (default)	2
297	◈	altera_reserved_lai_0_4	PIN_L25	LVTTL (default)	2
298	◈	altera_reserved_lai_0_5	PIN_M19	LVTTL (default)	2
299	◈	altera_reserved_lai_0_6	PIN_M20	LVTTL (default)	2
300	◈	altera_reserved_lai_0_7	PIN_K28	LVTTL (default)	2
301	◈	altera_reserved_lai_0_8	PIN_K27	LVTTL (default)	2
302	◈	altera_reserved_lai_0_9	PIN_N21	LVTTL (default)	2
303	◈	altera_reserved_lai_0_10	PIN_N22	LVTTL (default)	2
304	◈	altera_reserved_lai_0_11	PIN_M26	LVTTL (default)	2
305	◈	altera_reserved_lai_0_12	PIN_M25	LVTTL (default)	2

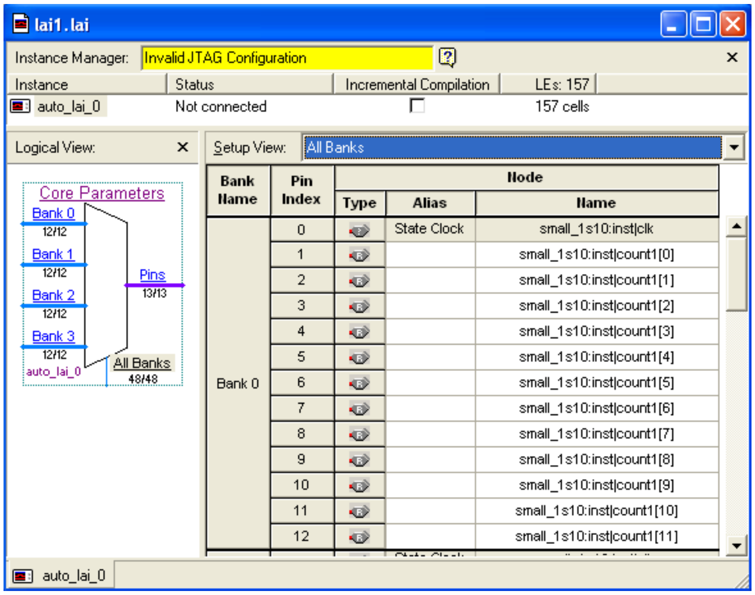
Once all LAI nodes have been assigned pin locations, close the Pin Planner window. The pin assignments will be reflected in the Setup View.

Next

- "Mapping Internal Signals to the Logic Analyzer Interface Banks" on page 53

Mapping Internal Signals to the Logic Analyzer Interface Banks

Having specified the number of banks to use in the Core Parameters, internal signals must be assigned for each bank in the Logic Analyzer Interface. Click the **Setup View** arrow and select **Bank n** to assign signals a bank at a time or select **All Banks**.



lai1.lai

Instance Manager: Invalid JTAG Configuration

Instance	Status	Incremental Compilation	LEs: 157
auto_lai_0	Not connected	□	157 cells

Logical View:

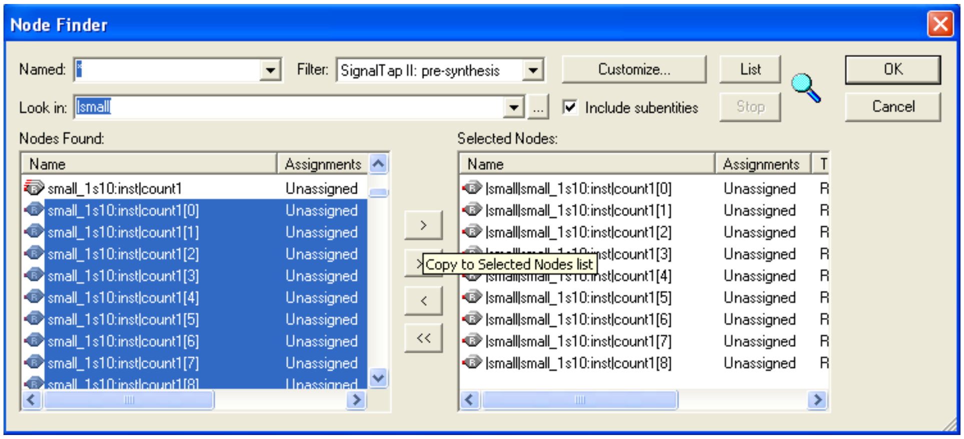
Setup View: All Banks

Bank Name	Pin Index	Node		
		Type	Alias	Name
Bank 0	0		State Clock	small_1s10:inst|clk
	1			small_1s10:inst|count1[0]
	2			small_1s10:inst|count1[1]
	3			small_1s10:inst|count1[2]
	4			small_1s10:inst|count1[3]
	5			small_1s10:inst|count1[4]
	6			small_1s10:inst|count1[5]
	7			small_1s10:inst|count1[6]
	8			small_1s10:inst|count1[7]
	9			small_1s10:inst|count1[8]
	10			small_1s10:inst|count1[9]
	11			small_1s10:inst|count1[10]
	12			small_1s10:inst|count1[11]

auto_lai_0

NOTE If using a state core, bit 0 of each bank will be reserved for the clock.

To begin assigning signals, double-click in the **Name** column to launch Node Finder. Within Node Finder click **List** to display all the design signals. Find the signals of interest, and select the signals from the Node Finder dialog box. As signals are assigned in Node Finder, the LAI schematic in the Logical View begins to reflect the assignments.



Node Finder

Named: *

Filter: SignalTap II: pre-synthesis

Look in: |small

☑ Include subentities

Nodes Found:

Name	Assignments
small_1s10:inst|count1	Unassigned
small_1s10:inst|count1[0]	Unassigned
small_1s10:inst|count1[1]	Unassigned
small_1s10:inst|count1[2]	Unassigned
small_1s10:inst|count1[3]	Unassigned
small_1s10:inst|count1[4]	Unassigned
small_1s10:inst|count1[5]	Unassigned
small_1s10:inst|count1[6]	Unassigned
small_1s10:inst|count1[7]	Unassigned
small_1s10:inst|count1[8]	Unassigned

Copy to Selected Nodes list

Selected Nodes:

Name	Assignments	T
|small|small_1s10:inst|count1[0]	Unassigned	R
|small|small_1s10:inst|count1[1]	Unassigned	R
|small|small_1s10:inst|count1[2]	Unassigned	R
|small|small_1s10:inst|count1[3]	Unassigned	R
|small|small_1s10:inst|count1[4]	Unassigned	R
|small|small_1s10:inst|count1[5]	Unassigned	R
|small|small_1s10:inst|count1[6]	Unassigned	R
|small|small_1s10:inst|count1[7]	Unassigned	R
|small|small_1s10:inst|count1[8]	Unassigned	R

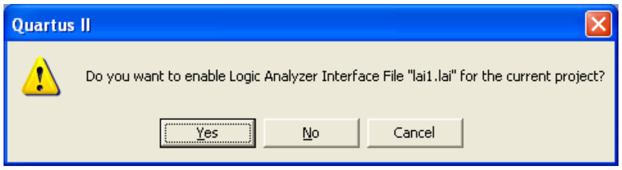
The design contains four counters; count1 and count3 are up counters and count2 and count4 are down counters. Assign a counter output per bank, for example, count1 to Bank0, count2 to Bank1, count3 to Bank2, and count4 to Bank3.

Next • "Saving the Logic Analyzer Interface" on page 55

Saving the Logic Analyzer Interface

To save the LAI file perform the following steps:

1. In the **File** menu, click **Save As**, the save dialog box opens.
2. In the **File name** box, enter your desired file name. Click **Save**.
3. When prompted, click **Yes** to enable the Logic Analyzer Interface file for the current project.



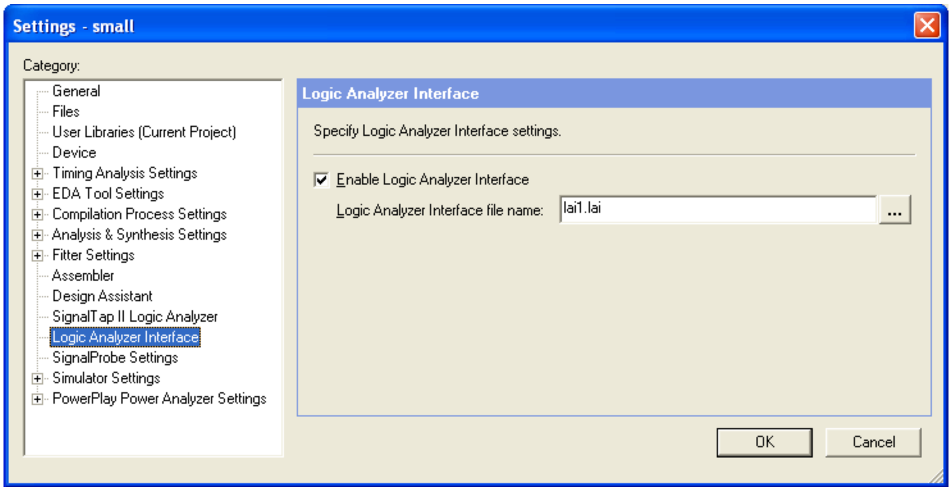
Do you want to enable Logic Analyzer Interface File "lai1.lai" for the current project?

Next • "Enabling/Disabling the Logic Analyzer Interface" on page 55

Enabling/Disabling the Logic Analyzer Interface

The Logic Analyzer Interface can be enabled and disabled to include and remove the core from the design. This step can be preformed as follows:

1. On the **Assignments** menu click **Settings**. The Settings dialog box opens.
2. Under **Category**, click **Logic Analyzer Interface**. The Logic Analyzer Interface displays.
3. Make sure **Enable Logic Analyzer Interface** is checked to include the core.
4. The **Logic Analyzer Interface file name** displays the full path name of the LAI file.



Logic Analyzer Interface

Specify Logic Analyzer Interface settings.

☑ Enable Logic Analyzer Interface

Logic Analyzer Interface file name: lai1.lai

Click **OK** to continue.

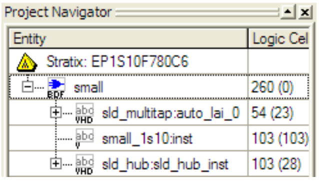
Next • "Compiling the Quartus II Project" on page 56

Compiling the Quartus II Project

The next step is to compile the project. On the **Processing** menu click **Start Compilation**.

NOTE Warnings during compilation are okay.

To ensure the Logic Analyzer Interface is properly compiled with the project, expand the entity hierarchy in the Project Navigator. If the Logic Analyzer Interface is compiled with the design, the *sld_hub* and *sld_multitap* entities will be shown in the project navigator.



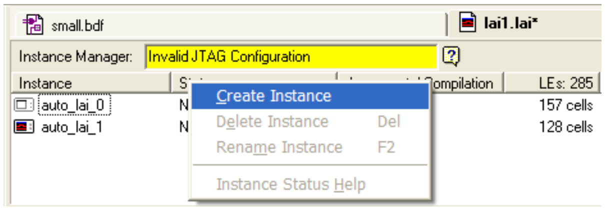
Entity	Logic Cel
Stratix: EP1S10F780C6	
small	260 (0)
sld_multitap:auto_lai_0	54 (23)
small_1s10:inst	103 (103)
sld_hub:sld_hub_inst	103 (28)

Next • "FYI: Creating Multiple Logic Analyzer Interface Instances in One FPGA" on page 57

FYI: Creating Multiple Logic Analyzer Interface Instances in One FPGA

The Logic Analyzer Interface supports multiple interfaces in a single FPGA. This feature is particularly useful when you want to build LAI configurations that contain different settings. For example, build one LAI instance to perform Registered/State analysis and build another instance that performs Combinational/Timing analysis on the same set of signals. Another example would be to perform Registered/State analysis on portions of the design that are in different clock domains.

To create multiple Logic Analyzer Interfaces, right-click in the **Instance Manger** window and select **Create Instance**.


Instance Manager: Invalid JTAG Configuration

Instance	S	LEs: 285
auto_lai_0	N	157 cells
auto_lai_1	N	128 cells

Automated Logic Analyzer Set Up

The FPGA dynamic probe automatically sets up the logic analyzer for the type of Altera LAI core it connects to.

For timing (asynchronous) cores, the FPGA dynamic probe automatically sets up:

- Timing mode.

For state (synchronous) cores, the FPGA dynamic probe automatically sets up:

- State mode.
- Master clock mode.
- Clock signal and edge.

For all types of cores, the FPGA dynamic probe automatically sets up:

- Bus/signal names for the selected bank.

In the logic analyzer module's "Analyzer Setup dialog" (in the online help), you can rename buses/signals (see page 32) and define additional bus/signal names (see page 33), but changing any of the other settings made by the FPGA dynamic probe will interfere with its operation.

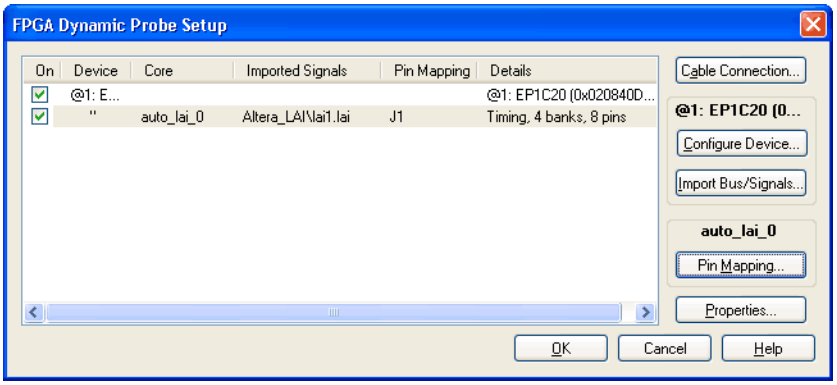
You are free to change settings that are untouched by the FPGA dynamic probe (like memory depth, trigger position, or sampling positions); they will not affect the FPGA dynamic probe.

9
FPGA Dynamic Probe Reference

FPGA Dynamic Probe Setup Dialog

The FPGA Dynamic Probe (see page 9) dialog lets you:

- Establish a connection between the logic analyzer and an FPGA with one or more Altera LAI cores.
- Configure the FPGA with a new design file.
- Map FPGA pins to probe pins/pads.
- Import signal names from the FPGA design tool.



On	Device	Core	Imported Signals	Pin Mapping	Details
☑	@1: E...				@1: EP1C20 (0x020840D...
☑	"	auto_lai_0	Altera_LAI\lai1.lai	J1	Timing, 4 banks, 8 pins

On (in FPGA device list)	The check boxes in this column let you enable or disable a core for use. If you do not want to use a particular core (typically in the multiple core case), you can uncheck its box, and the outputs of the core will be disabled. If a core's "always on" mode is enabled (see the core Details (see page 65) within its Properties dialog (see page 65)), the core is always enabled and can be probed at powerup (bank 0 will be the selected bank). In this case, the core cannot be disabled, and the check box cannot be unchecked.
Cable Connection...	Opens the Cable Connection dialog (see page 61) for establishing a connection between the logic analyzer and the Altera LAI core.
Configure Device...	Opens the Select FPGA Configuration File dialog (see page 64) for downloading a design into the selected FPGA device.
Import Bus/Signals...	Opens the Select Signal Import File dialog (see page 64) for importing internal FPGA bus/signal names.

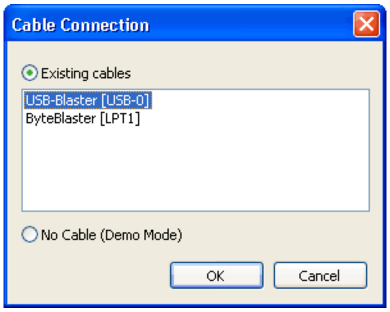
Pin Mapping...	Opens the Pin Mapping dialog (see page 61) for defining the logic analyzer probes that are used to connect to the FPGA and setting up the pin mapping (see Step 6. Map FPGA pins (see page 35)).
Properties...	Opens the Properties dialog (see page 65) which lets you rename devices and cores as well as display information about the selected Altera LAI core.

See Also

- Measurement Steps (see page 27)

Cable Connection Dialog

The Cable Connection dialog lets you specify the the type of cable used to connect the logic analyzer to the device under test JTAG port.



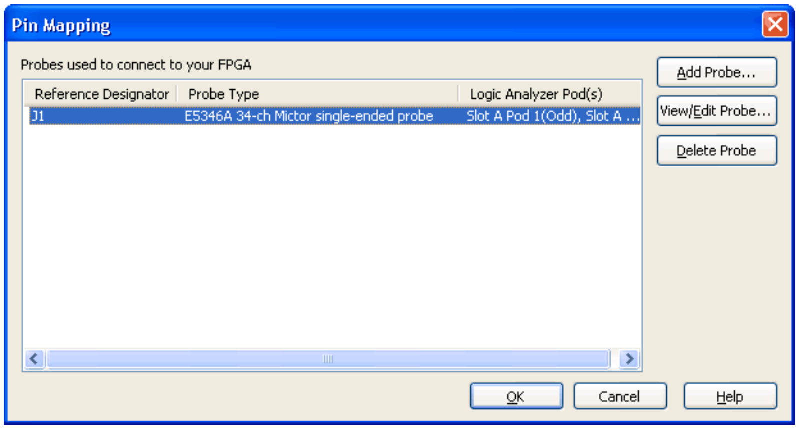
Existing cables	Lets you select a cable that has been set up using the Altera Quartus II Programmer (or design) software.
No Cable (Demo Mode)	Simulates a cable connection for demonstration purposes.

See Also

- Step 2. Set up the JTAG cable using the programmer software (see page 19)
- Step 3. Establish connection between analyzer and JTAG cable (see page 28)

Pin Mapping Dialog

The Pin Mapping dialog lets you define the logic analyzer probes that are used to connect to the FPGA, and it lets you set up the FPGA pin to probe pin/pad mapping.



Pin Mapping

Probes used to connect to your FPGA

Reference Designator	Probe Type	Logic Analyzer Pod(s)
J1	E5346A 34-ch Mictor single-ended probe	Slot A Pod 1(Odd), Slot A ...

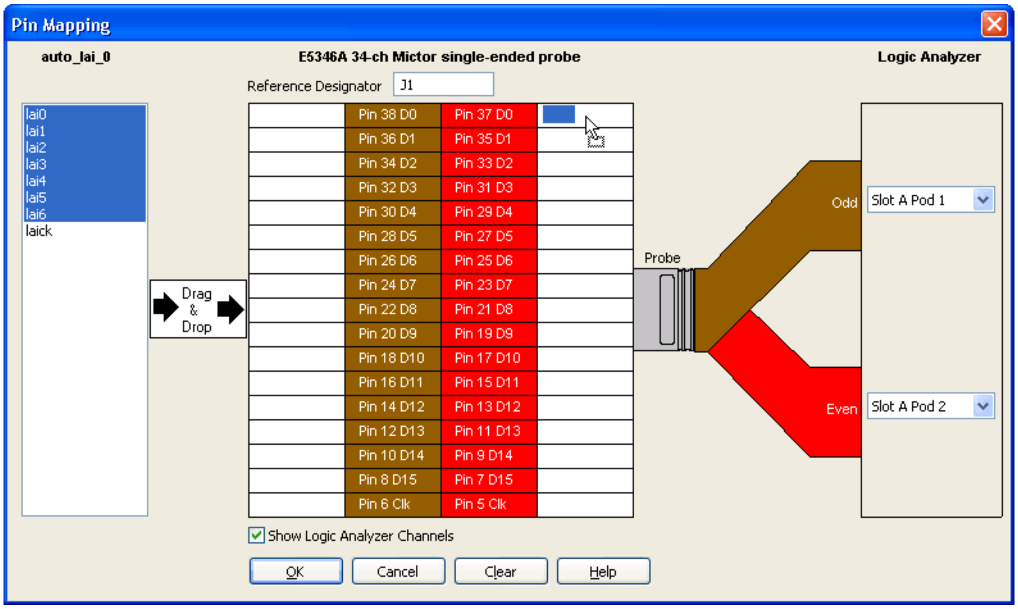
Add Probe...	Opens the "Select the type of probe to add" dialog; when you click **OK**, the Pin Mapping Edit dialog (see page 62) is opened for mapping the FPGA output pins to the probe pins/pads. If your probe doesn't appear in the list of probe types, you can "download the latest probe definitions from the web" (in the online help).
Edit Probe...	For the selected probe, opens the Pin Mapping Edit dialog (see page 62) for editing the FPGA output pins to probe pin/pad mapping.
Delete Probe	Deletes the selected probe.

See Also

- Step 6. Map FPGA pins (see page 35)

Pin Mapping Edit Dialog

The Pin Mapping Edit dialog lets you map the FPGA output pins to the logic analyzer probe pins/pads.



Pin Mapping

auto_lai_0

E5346A 34-ch Mictor single-ended probe

Logic Analyzer

Reference Designator J1

lai0
lai1
lai2
lai3
lai4
lai5
lai6
laick

	Pin 38 D0	Pin 37 D0	
	Pin 36 D1	Pin 35 D1	
	Pin 34 D2	Pin 33 D2	
	Pin 32 D3	Pin 31 D3	
	Pin 30 D4	Pin 29 D4	
	Pin 28 D5	Pin 27 D5	
	Pin 26 D6	Pin 25 D6	
	Pin 24 D7	Pin 23 D7	
	Pin 22 D8	Pin 21 D8	
	Pin 20 D9	Pin 19 D9	
	Pin 18 D10	Pin 17 D10	
	Pin 16 D11	Pin 15 D11	
	Pin 14 D12	Pin 13 D12	
	Pin 12 D13	Pin 11 D13	
	Pin 10 D14	Pin 9 D14	
	Pin 8 D15	Pin 7 D15	
	Pin 6 Clk	Pin 5 Clk	

Odd: Slot A Pod 1

Even: Slot A Pod 2

☑ Show Logic Analyzer Channels

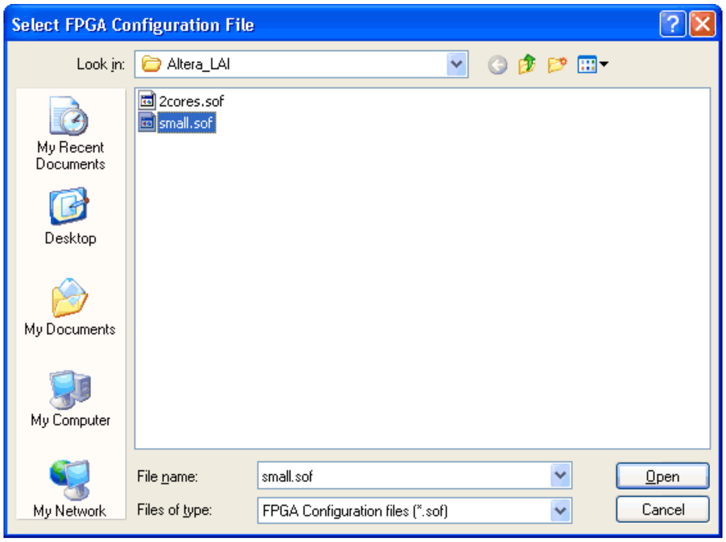
Reference Designator	Identifies the reference designator (in the device under test) of the probe connector, the connectorless probe retention module, or pins probed by flying leads.
FPGA Pins	Lists the FPGA pins used for the Altera LAI core outputs. When dragging these pins onto the pin/pad map, you can select multiple pins using Shift-click or Ctrl-click.
Probe Pin/Pad Diagram	Diagrams probe pins/pads, flying-lead channels, or termination adapter pins, and provides fields for dropping FPGA pin numbers.
Logic Analyzer Slot, Pod	Lets you select the logic analyzer module slots/pods to which the probe, flying leads, or termination adapter is connected.
Show Logic Analyzer Channels	When checked, the logic analyzer pod channel numbers are displayed in the probe pin/pad diagram next to the pin/pad numbers.
Clear	Clears all FPGA pins that have been mapped to pins/pads. You can clear individual pin mappings by dragging a pin from the pin/pad diagram back to the FPGA Pins list.

See Also

- Step 7. Map FPGA pins (see page 39)

Select FPGA Configuration File Dialog

The Select FPGA Configuration File dialog lets you select a design file for downloading into an FPGA device on the JTAG chain.



Select FPGA Configuration File

Look in: Altera_LAI

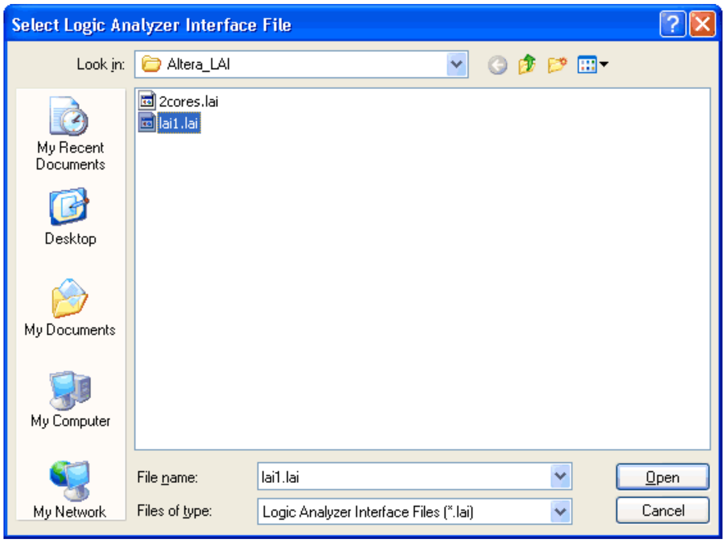
2cores.sof
small.sof

File name: small.sof

Files of type: FPGA Configuration files (*.sof)

Select Signal Import File Dialog

The Select Signal Import File dialog lets you select the file (from the FPGA design tool) that contains the names of the internal buses/signals that appear on Altera LAI core inputs.



Select Logic Analyzer Interface File

Look in: Altera_LAI

2cores.lai
lai1.lai

File name: lai1.lai

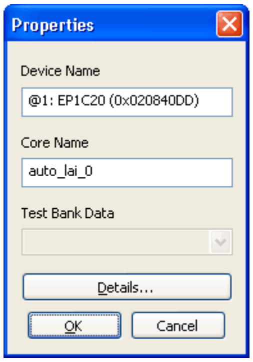
Files of type: Logic Analyzer Interface Files (*.lai)

See Also

- Step 5. Import signal names (see page 31)

Properties Dialog

The Properties dialog lets you change device, core, and bank names as well as view detailed information about the core.


Properties

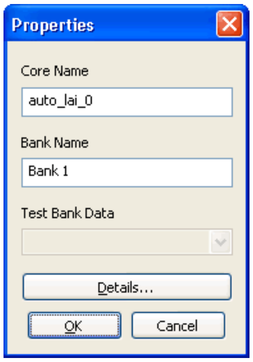
Device Name

@1: EP1C20 (0x020840DD)

Core Name

auto_lai_0

Test Bank Data


Properties

Core Name

auto_lai_0

Bank Name

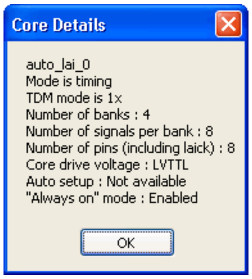
Bank 1

Test Bank Data

Device Name	Lets you rename the selected device.
Core Name	Lets you rename the selected core.
Bank Name	Lets you rename the selected bank.
Test Bank Data	Not available with Altera LAI cores.
Details...	Opens the Core Details dialog (see page 65) which displays information about the selected Altera LAI core.

Core Details Dialog

The Core Details dialog displays information about the selected Altera LAI core.

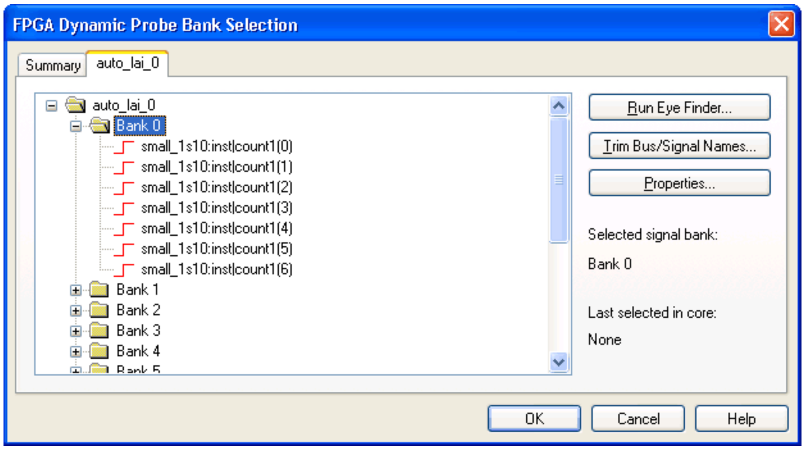


auto_lai_0
Mode is timing
TDM mode is 1x
Number of banks : 4
Number of signals per bank : 8
Number of pins (including laick) : 8
Core drive voltage : LVTTL
Auto setup : Not available
"Always on" mode : Enabled

FPGA Dynamic Probe Bank Selection Dialog

The FPGA Dynamic Probe (see page 9) Bank Selection dialog lets you:

- Select a different bank of internal signals to probe.
- Rename individual signals (without having to do global trimming) by triple-clicking the signal name.


FPGA Dynamic Probe Bank Selection
Summary
auto_lai_0
auto_lai_0
Bank 0
small_1s10:inst|count1(0)
small_1s10:inst|count1(1)
small_1s10:inst|count1(2)
small_1s10:inst|count1(3)
small_1s10:inst|count1(4)
small_1s10:inst|count1(5)
small_1s10:inst|count1(6)
Bank 1
Bank 2
Bank 3
Bank 4
Bank 5
Run Eye Finder...
Trim Bus/Signal Names...
Properties...
OK
Cancel
Help

Selected signal bank:
Bank 0

Last selected in core:
None

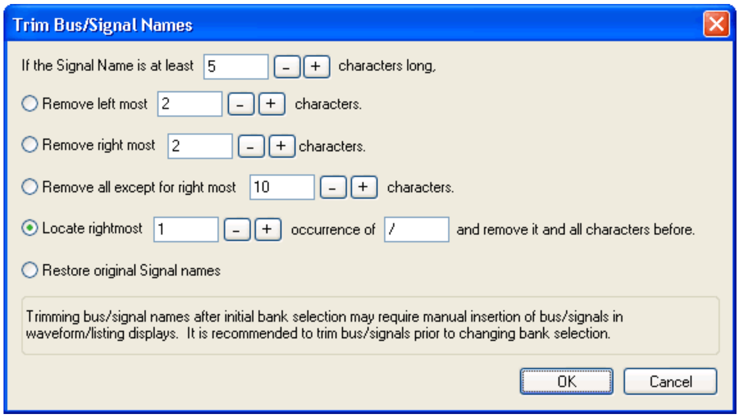
Run Eye Finder...	Opens the "Thresholds and Sample Positions dialog" (in the online help) for running *eye finder* to automatically adjust the state mode sampling positions. This button is available when probing only state (synchronous) cores. If there are any timing (asynchronous) cores, the logic analyzer is set up in timing mode, and *eye finder* is not available.
Trim Bus/Signal Names...	Opens the Trim Bus/Signal Names dialog (see page 67) for shortening imported FPGA internal bus/signal names.
Properties...	Opens the Properties dialog (see page 65) which lets you rename cores and banks as well as display information about the selected Altera LAI core.

See Also

- Measurement Steps (see page 27)

Trim Bus/Signal Names Dialog

The Trim Bus/Signal Names dialog lets you specify how imported bus/signal names should be shortened.



If the Signal Name is at least 5 characters long.

○ Remove left most 2 characters.

○ Remove right most 2 characters.

○ Remove all except for right most 10 characters.

● Locate rightmost 1 occurrence of / and remove it and all characters before.

○ Restore original Signal names

Trimming bus/signal names after initial bank selection may require manual insertion of bus/signals in waveform/listing displays. It is recommended to trim bus/signals prior to changing bank selection.

If the Signal Name is at least	Only bus/signal names longer than this value will be trimmed.
Remove left most	The number of characters to remove from the beginning of the names.
Remove right most	The number of characters to remove from the end of the names.
Remove all except right most	The number of characters to leave at the end of the names.
Locate right most Nth occurrence of the string	The string before which all characters from the names are stripped.
Restore original Signal names	Undoes the bus/signal name trimming.

See Also

- Step 5. Import signal names (see page 31)

Specifications and Characteristics

The FPGA dynamic probe for Altera FPGAs has these specifications and characteristics:

- Supported Logic Analyzers (see page 69)
- FPGA Dynamic Probe Software Application (see page 69)

Supported Logic Analyzers

Standalone logic analyzers:	1680 Series, 1690 Series, 16800 Series
Modular logic analysis systems:	16900A, 16901A, 16902A, 16902B, 16903A with one or more of the following cards: • 16740A, 16741A, 16742A. • 16750A, 16751A, 16752A, 16753A, 16754A, 16755A, 16756A. • 16760A. • 16910A, 16911A, 16950A, 16950B, 16951B. A single node-locked FPGA dynamic probe license will enable all modules within a 16900 Series system.
Triggering capabilities:	Determined by logic analyzer.
Supported probing mechanisms:	Soft touch (34-channel and 17-channel), Mictor, Samtec, Flying lead

FPGA Dynamic Probe Software Application

Maximum number of devices supported on a JTAG scan chain:	256
Maximum number of Altera LAI cores supported per FPGA device:	15

10
Probe Control, COM Automation

The *Agilent Logic Analyzer* application includes the COM Automation Server. This software lets you write programs that control the *Agilent Logic Analyzer* application from remote computers on the Local Area Network (LAN).

In a COM automation program, you can configure a probe by:

- Loading a configuration file (which configures the complete logic analyzer setup).
- Using the "Probe" (in the online help) object's "DoCommands" (in the online help) method with an XML-format string parameter (see Probe Setup, XML Format (see page 73)).

You can get information about a probe's configuration using the Probe object's "QueryCommand" (in the online help) method. Queries supported by the FPGA dynamic probe are listed below.

For more information about logic analyzer COM automation and probe objects in general, see "COM Automation" (in the online help).

XML-Based Queries Supported

The FPGA dynamic probe supports the following XML-based queries (made with the "Probe" (in the online help) object's "QueryCommand" (in the online help) method).

Query	Description
GetAllSetup	Returns the current setup, using the full tag set, used for writing generic configuration files (see the XML format <Properties> element (see page 98)).
GetProperties	Returns the current setup, using the full tag set, equivalent to "GetAllSetup" (see the XML format <Properties> element (see page 98)).

See Also

- "COM Automation" (in the online help)
- Probe Setup, XML Format (see page 73)

11
Probe Setup, XML Format

When you save logic analyzer configurations to XML format files, setup information for the FPGA dynamic probe is included.

This XML format setup information is also used when writing COM automation programs to control the logic analyzer from a remote computer.

XML elements for the FPGA dynamic probe have the following hierarchy:

```
<Properties> (see page 98)
   <ATC_II> (see page 75)
   <JTAG_Chain> (see page 88)
   <Devices> (see page 87)
      <Device> (see page 85)
         <Cores> (see page 83)
            <Core> (see page 81)
               <Banks> (see page 80)
                  <Bank> (see page 76)
                     <Signals> (see page 102)
                        <Signal> (see page 100)
                           <Labels> (see page 91)
                              <Label> (see page 89)
                     <NonATCLabels> (see page 92)
                        <Label> (see page 90)
                           <Assignment> (see page 74)
                     <WindowInfo> (see page 107)
                     <SymbolInfo> (see page 105)
                     <TriggerInfo> (see page 106)
               <PinMapping> (see page 93)
                  <DefinedProbes> (see page 84)
                     <Probe> (see page 97)
                        <Pods> (see page 96)
                           <Pod> (see page 95)
                        <Signals> (see page 104)
                           <Signal> (see page 101)
```

See Also

- "XML Format" (in the online help)
- Probe Control, COM Automation (see page 71)

<Assignment> Element

The <Assignment> element describes the logic analyzer pod and channel assignments for a Label element (under NonATCLabels).

Attributes

Name	Description
Channel	'number'
Pod	'number'

Parents This element can have the following parents: <Label> (see page 90).

Example

```
<NonATCLabels>
   <Label Name='My Bus 2' Comment='' Handle='131'>
      <Assignment Pod='1' Channel='0' />
      <Assignment Pod='1' Channel='1' />
      <Assignment Pod='1' Channel='2' />
      <Assignment Pod='1' Channel='3' />
      <Assignment Pod='1' Channel='4' />
      <Assignment Pod='1' Channel='5' />
      <Assignment Pod='1' Channel='6' />
      <Assignment Pod='1' Channel='7' />
   </Label>
</NonATCLabels>
```

<ATC_II> Element

The <ATC_II> element describes the selected device, the selected core, and the cable type.

Attributes

Name	Description
CableType	'number'
ParallelCablePort	'number'
ParallelCableSpeed	'number'
ParallelCableType	'number'
SelectedCore	'number'
SelectedDevice	'number'
USBCableSpeed	'number'

Parents This element can have the following parents: <Properties> (see page 98).

Example

```
<ATC_II SelectedDevice='1' SelectedCore='0' CableType='0'
      ParallelCableType='0' ParallelCableSpeed='2' ParallelCablePort='0'
      USBCableSpeed='0'/>
```

<Bank> Element

The <Bank> element describes a bank within an Altera LAI core.

Attributes

Name	Description
CalibrationBank	'F' (false) or 'T' (true)
Name	'string'
NumDataPins	'number'
State2X	'F' (false) or 'T' (true)

Children This element can have the following children: <Signals> (see page 102), <NonATCLabels> (see page 92), <WindowInfo> (see page 107), <SymbolInfo> (see page 105), <TriggerInfo> (see page 106).

Parents This element can have the following parents: <Banks> (see page 80).

Example

```
<Bank Name='Bank-0' CalibrationBank='F' NumDataPins='16' State2X='F'>
   <Signals>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='0' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='1' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='2' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='3' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='4' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='5' />
         </Labels>
      </Signal>
      <Signal>
```

```
         <Labels>
            <Label Name='/s2mon/tid' Bit='6' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/tid' Bit='7' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='0' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='1' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='2' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='3' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='4' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='5' />
         </Labels>
      </Signal>
      <Signal>
         <Labels>
            <Label Name='/s2mon/lastackid' Bit='6' />
         </Labels>
      </Signal>
      <Signal>
         <Labels />
      </Signal>
   </Signals>
   <NonATCLabels />
   <WindowInfo WindowHandle='1' WindowSettings='
   &lt;Setup&gt;
      &lt;Sampling PerDivision=&apos;5 ns&apos;
            Delay=&apos;0 s&apos;/&gt;
      &lt;BusSignals&gt;
         &lt;Clear/&gt;
         &lt;BusSignal Module=&apos;My 1682D-1&apos;
               Name=&apos;My Bus 1&apos; DefaultBase=&apos;Hex&apos;
               Color=&apos;hFFFFFF&apos; Height=&apos;30&apos;/&gt;
         &lt;BusSignal Module=&apos;My 1682D-1&apos;
               Name=&apos;/s2mon/tid&apos; DefaultBase=&apos;Hex&apos;
               Color=&apos;hFFFFFF&apos; Height=&apos;30&apos;/&gt;
         &lt;BusSignal Module=&apos;My 1682D-1&apos;
               Name=&apos;/s2mon/s2mstate&apos;
               DefaultBase=&apos;Hex&apos; Color=&apos;hFFFFFF&apos;
               Height=&apos;30&apos;/&gt;
         &lt;BusSignal Name=&apos;Time&apos; Color=&apos;hFFFFFF&apos;
               Height=&apos;30&apos;/&gt;
      &lt;/BusSignals&gt;
   &lt;/Setup&gt;
   '/>
   <WindowInfo WindowHandle='2' WindowSettings='
   &lt;BusSignals&gt;
      &lt;Clear/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;Sample Number&apos; Color=&apos;hFFFFFF&apos;
            Alignment=&apos;Right&apos; Width=&apos;112&apos;/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;My Bus 1&apos; DefaultBase=&apos;Hex&apos;
            Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
            Width=&apos;113&apos;/&gt;
      &lt;BusSignal Name=&apos;Time&apos;
            DefaultBase=&apos;Absolute&apos; Color=&apos;hFFFFFF&apos;
            Alignment=&apos;Right&apos; Width=&apos;152&apos;/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;/s2mon/tid&apos; DefaultBase=&apos;Hex&apos;
            Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
            Width=&apos;113&apos;/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;/s2mon/s2mstate&apos;
            DefaultBase=&apos;Hex&apos; Color=&apos;hFFFFFF&apos;
            Alignment=&apos;Right&apos; Width=&apos;120&apos;/&gt;
   &lt;/BusSignals&gt;
   '/>
   <SymbolInfo ModuleHandle='1' SymbolSettings='
   &lt;Module&gt;
      &lt;BusSignalSetup&gt;
         &lt;BusSignals&gt;
            &lt;BusSignal Name=&apos;My Bus 1&apos;/&gt;
            &lt;Folder Name=&apos;Core 0 FPGA Probe&apos;
                 Comment=&apos;Created by FPGA Dynamic Probe-1&apos;&gt;
               &lt;BusSignal Name=&apos;/s2mon/tid&apos;/&gt;
               &lt;BusSignal Name=&apos;/s2mon/s2mstate&apos;/&gt;
            &lt;/Folder&gt;
         &lt;/BusSignals&gt;
         &lt;NetlistImport/&gt;
      &lt;/BusSignalSetup&gt;
      &lt;Config TimeOfTrigger=&apos;1.110240661 Gs&apos;
            CorrelatedTriggerTime=&apos;0 s&apos;
            UserSkewTime=&apos;0 s&apos;
            SystemTrigger=&apos;T&apos;/&gt;
   &lt;/Module&gt;
   '/>
   <TriggerInfo ModuleHandle='1' TriggerSettings='
   &lt;Module&gt;
```

```
      &lt;Trigger Mode=&apos;State&apos; Type=&apos;Normal&apos;&gt;
         &lt;StoreQual Mode=&apos;Custom&apos;&gt;
            &lt;Event ParensNeeded=&apos;F&apos;&gt;
               &lt;Anything/&gt;
            &lt;/Event&gt;
         &lt;/StoreQual&gt;
         &lt;Step Number=&apos;1&apos;&gt;
            &lt;If&gt;
               &lt;Event ParensNeeded=&apos;F&apos;&gt;
                  &lt;BusSignal Name=&apos;/s2mon/tid&apos;
                        Bit=&apos;All&apos; Operator=&apos;Equals&apos;
                        Value=&apos;h3F&apos;/&gt;
               &lt;/Event&gt;
               &lt;Occurrence Value=&apos;1&apos;
                     Mode=&apos;Eventual&apos;/&gt;
               &lt;Action&gt;
                  &lt;TriggerAction Operator=&apos;Fill Memory&apos;&gt;
                     &lt;StoreQual Mode=&apos;Custom&apos;&gt;
                        &lt;Event ParensNeeded=&apos;F&apos;&gt;
                           &lt;DefaultStore/&gt;
                        &lt;/Event&gt;
                     &lt;/StoreQual&gt;
                  &lt;/TriggerAction&gt;
               &lt;/Action&gt;
            &lt;/If&gt;
         &lt;/Step&gt;
      &lt;/Trigger&gt;
      &lt;Config TimeOfTrigger=&apos;1.110240661 Gs&apos;
            CorrelatedTriggerTime=&apos;0 s&apos;
            UserSkewTime=&apos;0 s&apos;
            SystemTrigger=&apos;T&apos;/&gt;
   &lt;/Module&gt;
   '/>
</Bank>
```

<Banks> Element

The <Banks> element contains descriptions of banks within an Altera LAI core.

Children This element can have the following children: <Bank> (see page 76).

Parents This element can have the following parents: <Core> (see page 81).

Example

```
<Banks>
   <Bank Name='Bank 0' CalibrationBank='F' NumDataPins='16' State2X='F'>
      ...
   </Bank>
   <Bank Name='Bank 1' CalibrationBank='F' NumDataPins='16' State2X='F'>
      ...
   </Bank>
   <Bank Name='Bank 2' CalibrationBank='F' NumDataPins='16' State2X='F'>
      ...
   </Bank>
   <Bank Name='Bank 3' CalibrationBank='F' NumDataPins='16' State2X='F'>
      ...
   </Bank>
   <Bank Name='Test Bank' CalibrationBank='T' NumDataPins='16'
         State2X='F'>
      ...
   </Bank>
</Banks>
```

<Core> Element

The <Core> element describes a core within a device on the JTAG chain.

Attributes

Name	Description
ClockPodIndex	'number'
CoreCannotBeMaster	'F' (false) or 'T' (true)
CoreIsMaster	'F' (false) or 'T' (true)
Latency	'number'
MinimumPeriod	'number'
Name	'string'
NumBanks	'number'
NumPins	'number'
NumSignals	'number'
SelectedBank	'number'
SelectedForUse	'F' (false) or 'T' (true)
SelectedSignal	'number'
StateMode	'F' (false) or 'T' (true)
TDM_1X	'F' (false) or 'T' (true)
TestBankAvailable	'F' (false) or 'T' (true)
TestBankMode	'number'
Type	'number'
ThresholdCode	'number'

Children This element can have the following children: <Banks> (see page 80), <PinMapping> (see page 93).

Parents This element can have the following parents: <Cores> (see page 83).

Example

```
<Core Name='Core-1' Type='0' SelectedBank='0' SelectedSignal='-1'
      NumBanks='5' NumSignals='16' NumPins='17' StateMode='T' TDM_1X='T'
      ThresholdCode='92' TestBankAvailable='T' ClockPodIndex='0'
      SelectedForUse='T' CoreIsMaster='T' CoreCannotBeMaster='F'
      MinimumPeriod='-1' TestBankMode='2' Latency='4'>
   <Banks>
      <Bank Name='Bank 0' CalibrationBank='F' NumDataPins='16'
            State2X='F'>
         ...
      </Bank>
      <Bank Name='Bank 1' CalibrationBank='F' NumDataPins='16'
```

```
            State2X='F'>
         ...
      </Bank>
      <Bank Name='Bank 2' CalibrationBank='F' NumDataPins='16'
            State2X='F'>
         ...
      </Bank>
      <Bank Name='Bank 3' CalibrationBank='F' NumDataPins='16'
            State2X='F'>
         ...
      </Bank>
      <Bank Name='Test Bank' CalibrationBank='T' NumDataPins='16'
            State2X='F'>
         ...
      </Bank>
   </Banks>
   <PinMapping Attached='T' ModuleHandle='1'>
      <DefinedProbes>
         <Probe Name='J1' Type='E5346A 34-ch Mictor single-ended probe'>
            <Pods>
               <Pod Index='0' />
               <Pod Index='1' />
            </Pods>
            <Signals>
               ...
            </Signals>
         </Probe>
      </DefinedProbes>
   </PinMapping>
</Core>
```

<Cores> Element

The <Cores> element contains descriptions of cores within a device on the JTAG chain.

Children This element can have the following children: <Core> (see page 81).

Parents This element can have the following parents: <Device> (see page 85).

Example

```
<Cores>
   <Core Name='Core 0' Type='0' SelectedBank='0' SelectedSignal='-1'
         NumBanks='5' NumSignals='16' NumPins='9' StateMode='T'
         TDM_1X='F' ThresholdCode='92' TestBankAvailable='T'
         ClockPodIndex='0' SelectedForUse='T' CoreIsMaster='F'
         CoreCannotBeMaster='F' MinimumPeriod='-1' TestBankMode='2'
         Latency='4'>
      <Banks>
         ...
      </Banks>
      <PinMapping Attached='T' ModuleHandle='1'>
         ...
      </PinMapping>
   </Core>
   <Core Name='Core 1' Type='0' SelectedBank='0' SelectedSignal='-1'
         NumBanks='2' NumSignals='9' NumPins='9' StateMode='F'
         TDM_1X='T' ThresholdCode='92' TestBankAvailable='F'
         ClockPodIndex='2' SelectedForUse='T' CoreIsMaster='T'
         CoreCannotBeMaster='F' MinimumPeriod='-1' TestBankMode='0'>
      <Banks>
         ...
      </Banks>
      <PinMapping Attached='T' ModuleHandle='1'>
         ...
      </PinMapping>
   </Core>
</Cores>
```

<DefinedProbes> Element

The <DefinedProbes> element contains defined probes.

Children This element can have the following children: <Probe> (see page 97).

Parents This element can have the following parents: <PinMapping> (see page 93).

Example

```
<DefinedProbes>
   <Probe Name='J1' Type='E5346A 34-ch Mictor single-ended probe'>
      <Pods>
         <Pod Index='0'/>
         <Pod Index='1'/>
      </Pods>
      <Signals>
         <Signal Name='ATD0' Pin='38' PinMapIndex='0'
               ClockChannel='F' />
         <Signal Name='ATD1' Pin='36' PinMapIndex='1'
               ClockChannel='F' />
         <Signal Name='ATD2' Pin='34' PinMapIndex='2'
               ClockChannel='F' />
         <Signal Name='ATD3' Pin='32' PinMapIndex='3'
               ClockChannel='F' />
         <Signal Name='ATD4' Pin='30' PinMapIndex='4'
               ClockChannel='F' />
         <Signal Name='ATD5' Pin='28' PinMapIndex='5'
               ClockChannel='F' />
         <Signal Name='ATD6' Pin='26' PinMapIndex='6'
               ClockChannel='F' />
         <Signal Name='ATD7' Pin='24' PinMapIndex='7'
               ClockChannel='F' />
         <Signal Name='ATD8' Pin='22' PinMapIndex='8'
               ClockChannel='F' />
         <Signal Name='ATD9' Pin='20' PinMapIndex='9'
               ClockChannel='F' />
         <Signal Name='ATD10' Pin='18' PinMapIndex='10'
               ClockChannel='F' />
         <Signal Name='ATD11' Pin='16' PinMapIndex='11'
               ClockChannel='F' />
         <Signal Name='ATD12' Pin='14' PinMapIndex='12'
               ClockChannel='F' />
         <Signal Name='ATD13' Pin='12' PinMapIndex='13'
               ClockChannel='F' />
         <Signal Name='ATD14' Pin='10' PinMapIndex='14'
               ClockChannel='F' />
         <Signal Name='ATCK' Pin='6' PinMapIndex='16'
               ClockChannel='T' />
      </Signals>
   </Probe>
</DefinedProbes>
```

<Device> Element

The <Device> element describes a device on the JTAG chain.

Attributes

Name	Description
CDCFilename	'string'
Configurable	'F' (false) or 'T' (true)
FPGAFilename	'string'
IRLength	'number'
Name	'string'
NumCores	'number'
SelectedForUse	'F' (false) or 'T' (true)
Type	'number'
UserRegNum	'number'

Children This element can have the following children: <Cores> (see page 83).

Parents This element can have the following parents: <Devices> (see page 87).

Example

```
<Device Name='Device 1' Type='0' NumCores='2' UserRegNum='1'
      Configurable='T' IRLength='6' CDCFilename='C:\Documents and
      Settings\user\My Documents\Agilent Technologies\Logic
      Analyzer\Config Files\demo\Altera_demo_V8.cdc'
      FPGAFilename='C:\Documents and Settings\user\My Documents\Agilent
      Technologies\Logic Analyzer\Config Files\demo\Altera_demo_V8.bit'
      SelectedForUse='F'>
   <Cores>
      <Core Name='Core 0' Type='0' SelectedBank='0' SelectedSignal='-1'
            NumBanks='5' NumSignals='16' NumPins='9' StateMode='T'
            TDM_1X='F' ThresholdCode='92' TestBankAvailable='T'
            ClockPodIndex='0' SelectedForUse='T' CoreIsMaster='F'
            CoreCannotBeMaster='F' MinimumPeriod='-1' TestBankMode='2'
            Latency='4'>
         <Banks>
            ...
         </Banks>
         <PinMapping Attached='T' ModuleHandle='1'>
            ...
         </PinMapping>
      </Core>
      <Core Name='Core 1' Type='0' SelectedBank='0' SelectedSignal='-1'
            NumBanks='2' NumSignals='9' NumPins='9' StateMode='F'
            TDM_1X='T' ThresholdCode='92' TestBankAvailable='F'
            ClockPodIndex='2' SelectedForUse='T' CoreIsMaster='T'
            CoreCannotBeMaster='F' MinimumPeriod='-1' TestBankMode='0'>
         <Banks>
            ...
         </Banks>
         <PinMapping Attached='T' ModuleHandle='1'>
            ...
         </PinMapping>
      </Core>
   </Cores>
</Device>
```

<Devices> Element

The <Devices> element contains descriptions of the devices on the JTAG chain.

Children This element can have the following children: <Device> (see page 85).

Parents This element can have the following parents: <Properties> (see page 98).

Example

```
<Devices>
   <Device Name='Device 0' Type='0' NumCores='0' UserRegNum='0'
         Configurable='F' IRLength='8' CDCFilename='' FPGAFilename=''
         SelectedForUse='F'>
      <Cores/>
   </Device>
   <Device Name='Device 1' Type='0' NumCores='2' UserRegNum='1'
         Configurable='T' IRLength='6' CDCFilename='C:\Documents and
         Settings\user\My Documents\Agilent Technologies\Logic
         Analyzer\Config Files\demo\Altera_demo_V8.cdc'
         FPGAFilename='C:\Documents and Settings\user\My
         Documents\Agilent Technologies\Logic Analyzer\Config
         Files\demo\Altera_demo_V8.bit' SelectedForUse='F'>
      <Cores>
         <Core Name='Core 0' Type='0' SelectedBank='0'
               SelectedSignal='-1' NumBanks='5' NumSignals='16'
               NumPins='9' StateMode='T' TDM_1X='F' ThresholdCode='92'
               TestBankAvailable='T' ClockPodIndex='0'
               SelectedForUse='T' CoreIsMaster='F'
               CoreCannotBeMaster='F' MinimumPeriod='-1'
               TestBankMode='2' Latency='4'>
            <Banks>
               ...
            </Banks>
            <PinMapping Attached='T' ModuleHandle='1'>
               ...
            </PinMapping>
         </Core>
         <Core Name='Core 1' Type='0' SelectedBank='0'
               SelectedSignal='-1' NumBanks='2' NumSignals='9'
               NumPins='9' StateMode='F' TDM_1X='T' ThresholdCode='92'
               TestBankAvailable='F' ClockPodIndex='2'
               SelectedForUse='T' CoreIsMaster='T'
               CoreCannotBeMaster='F' MinimumPeriod='-1'
               TestBankMode='0'>
            <Banks>
               ...
            </Banks>
            <PinMapping Attached='T' ModuleHandle='1'>
               ...
            </PinMapping>
         </Core>
      </Cores>
   </Device>
</Devices>
```

<JTAG_Chain> Element

The <JTAG_Chain> element describes the number of devices and the number of Altera LAI cores on the JTAG chain.

Attributes

Name	Description
NumATC_II	'number'
NumDevices	'number'

Parents This element can have the following parents: <Properties> (see page 98).

Example

```
<JTAG_Chain NumDevices='3' NumATC_II='2' />
```

<Label> Element

The <Label> element describes a label within a signal.

Attributes

Name	Description
Bit	'number'
Name	'string'

Parents This element can have the following parents: <Labels> (see page 91).

Example

```
<Label Name='/s2mon/tid' Bit='0' />
```

<Label> Element (under NonATCLabels)

The <Label> element describes a bus/signal name within the NonATCLabels element.

Attributes

Name	Description
Comment	'string'
Handle	'number'
Name	'string'

Children This element can have the following children: <Assignment> (see page 74).

Parents This element can have the following parents: <NonATCLabels> (see page 92).

Example

```
<NonATCLabels>
   <Label Name='My Bus 2' Comment='' Handle='131'>
      <Assignment Pod='1' Channel='0' />
      <Assignment Pod='1' Channel='1' />
      <Assignment Pod='1' Channel='2' />
      <Assignment Pod='1' Channel='3' />
      <Assignment Pod='1' Channel='4' />
      <Assignment Pod='1' Channel='5' />
      <Assignment Pod='1' Channel='6' />
      <Assignment Pod='1' Channel='7' />
   </Label>
</NonATCLabels>
```

<Labels> Element

The <Labels> element contains descriptions of labels within a signal.

Children This element can have the following children: <Label> (see page 89).

Parents This element can have the following parents: <Signal> (see page 100).

Example

```
<Labels>
   <Label Name='/s2mon/tid' Bit='0' />
</Labels>
```

<NonATCLabels> Element

The <NonATCLabels> element contains a bank's bus/signal names that were not imported from a .cdc file or were renamed (for example, the Calibration Bus or any other renamed or additionally defined bus/signal names).

Children This element can have the following children: <Label> (see page 90).

Parents This element can have the following parents: <Bank> (see page 76).

Example

```
<NonATCLabels>
   <Label Name='My Bus 2' Comment='' Handle='131'>
      <Assignment Pod='1' Channel='0' />
      <Assignment Pod='1' Channel='1' />
      <Assignment Pod='1' Channel='2' />
      <Assignment Pod='1' Channel='3' />
      <Assignment Pod='1' Channel='4' />
      <Assignment Pod='1' Channel='5' />
      <Assignment Pod='1' Channel='6' />
      <Assignment Pod='1' Channel='7' />
   </Label>
</NonATCLabels>
```

<PinMapping> Element

The <PinMapping> element contains descriptions of pin mapping within an Altera LAI core.

Attributes

Name	Description
Attached	'F' (false) or 'T' (true)
ModuleHandle	'number'

Children This element can have the following children: <DefinedProbes> (see page 84).

Parents This element can have the following parents: <Core> (see page 81).

Example

```
<PinMapping Attached='T' ModuleHandle='1'>
   <DefinedProbes>
      <Probe Name='J1' Type='E5346A 34-ch Mictor single-ended probe'>
         <Pods>
            <Pod Index='0'/>
            <Pod Index='1'/>
         </Pods>
         <Signals>
            <Signal Name='ATD0' Pin='38' PinMapIndex='0'
                  ClockChannel='F' />
            <Signal Name='ATD1' Pin='36' PinMapIndex='1'
                  ClockChannel='F' />
            <Signal Name='ATD2' Pin='34' PinMapIndex='2'
                  ClockChannel='F' />
            <Signal Name='ATD3' Pin='32' PinMapIndex='3'
                  ClockChannel='F' />
            <Signal Name='ATD4' Pin='30' PinMapIndex='4'
                  ClockChannel='F' />
            <Signal Name='ATD5' Pin='28' PinMapIndex='5'
                  ClockChannel='F' />
            <Signal Name='ATD6' Pin='26' PinMapIndex='6'
                  ClockChannel='F' />
            <Signal Name='ATD7' Pin='24' PinMapIndex='7'
                  ClockChannel='F' />
            <Signal Name='ATD8' Pin='22' PinMapIndex='8'
                  ClockChannel='F' />
            <Signal Name='ATD9' Pin='20' PinMapIndex='9'
                  ClockChannel='F' />
            <Signal Name='ATD10' Pin='18' PinMapIndex='10'
                  ClockChannel='F' />
            <Signal Name='ATD11' Pin='16' PinMapIndex='11'
                  ClockChannel='F' />
            <Signal Name='ATD12' Pin='14' PinMapIndex='12'
                  ClockChannel='F' />
            <Signal Name='ATD13' Pin='12' PinMapIndex='13'
                  ClockChannel='F' />
            <Signal Name='ATD14' Pin='10' PinMapIndex='14'
```

```
                  ClockChannel='F' />
            <Signal Name='ATCK' Pin='6' PinMapIndex='16'
                  ClockChannel='T' />
         </Signals>
      </Probe>
   </DefinedProbes>
</PinMapping>
```

<Pod> Element

The <Pod> element describes the pod index used within a defined probe.

Attributes

Name	Description
Index	'number'

Parents This element can have the following parents: <Pods> (see page 96).

Example

```
<Pod Index='0'/>
```

<Pods> Element

The <Pods> element contains the pods used by a defined probe.

Children This element can have the following children: <Pod> (see page 95).

Parents This element can have the following parents: <Probe> (see page 97).

Example

```
<Pods>
   <Pod Index='0'/>
   <Pod Index='1'/>
</Pods>
```

<Probe> Element

The <Probe> element describes a defined probe.

Attributes

Name	Description
Name	'string' (name of connector in device under test)
Type	'string' (name of probe)

Children

This element can have the following children: <Pods> (see page 96), <Signals> (see page 104).

Parents

This element can have the following parents: <DefinedProbes> (see page 84).

Example

```
<Probe Name='J1' Type='E5346A 34-ch Mictor single-ended probe'>
   <Pods>
      <Pod Index='0'/>
      <Pod Index='1'/>
   </Pods>
   <Signals>
      <Signal Name='ATD0' Pin='38' PinMapIndex='0' ClockChannel='F' />
      <Signal Name='ATD1' Pin='36' PinMapIndex='1' ClockChannel='F' />
      <Signal Name='ATD2' Pin='34' PinMapIndex='2' ClockChannel='F' />
      <Signal Name='ATD3' Pin='32' PinMapIndex='3' ClockChannel='F' />
      <Signal Name='ATD4' Pin='30' PinMapIndex='4' ClockChannel='F' />
      <Signal Name='ATD5' Pin='28' PinMapIndex='5' ClockChannel='F' />
      <Signal Name='ATD6' Pin='26' PinMapIndex='6' ClockChannel='F' />
      <Signal Name='ATD7' Pin='24' PinMapIndex='7' ClockChannel='F' />
      <Signal Name='ATD8' Pin='22' PinMapIndex='8' ClockChannel='F' />
      <Signal Name='ATD9' Pin='20' PinMapIndex='9' ClockChannel='F' />
      <Signal Name='ATD10' Pin='18' PinMapIndex='10' ClockChannel='F' />
      <Signal Name='ATD11' Pin='16' PinMapIndex='11' ClockChannel='F' />
      <Signal Name='ATD12' Pin='14' PinMapIndex='12' ClockChannel='F' />
      <Signal Name='ATD13' Pin='12' PinMapIndex='13' ClockChannel='F' />
      <Signal Name='ATD14' Pin='10' PinMapIndex='14' ClockChannel='F' />
      <Signal Name='ATCK' Pin='6' PinMapIndex='16' ClockChannel='T' />
   </Signals>
</Probe>
```

<Properties> Element

The <Properties> element contains setup information for the FPGA dynamic probe.

Children This element can have the following children: <ATC_II> (see page 75), <JTAG_Chain> (see page 88), <Devices> (see page 87).

Parents This element can have the following parents: "<Probe>" (in the online help).

When used in COM automation, this element is returned by the "QueryCommand method" (in the online help)'s GetAllSetup and GetProperties queries. You can also use this element string as an XMLCommand with the "DoCommands method" (in the online help) to configure the FPGA dynamic probe.

Example

```
<Properties>
   <ATC_II SelectedDevice='1' SelectedCore='1' CableType='0'
         ParallelCableType='0' ParallelCableSpeed='2'
         ParallelCablePort='0' USBCableSpeed='0'/>
   <JTAG_Chain NumDevices='2' NumATC_II='2'/>
   <Devices>
      <Device Name='Device 0' Type='0' NumCores='0' UserRegNum='0'
            Configurable='F' IRLength='8' CDCFilename=''
            FPGAFilename='' SelectedForUse='F'>
         <Cores/>
      </Device>
      <Device Name='Device 1' Type='0' NumCores='2' UserRegNum='1'
            Configurable='T' IRLength='6' CDCFilename='C:\Documents
            and Settings\user\My Documents\Agilent Technologies\Logic
            Analyzer\Config Files\demo\Altera_demo_V8.cdc'
            FPGAFilename='C:\Documents and Settings\user\My
            Documents\Agilent Technologies\Logic Analyzer\Config
            Files\demo\Altera_demo_V8.bit' SelectedForUse='F'>
         <Cores>
            <Core Name='Core 0' Type='0' SelectedBank='0'
                  SelectedSignal='-1' NumBanks='5' NumSignals='16'
                  NumPins='9' StateMode='T' TDM_1X='F'
                  ThresholdCode='92' TestBankAvailable='T'
                  ClockPodIndex='0' SelectedForUse='T'
                  CoreIsMaster='F' CoreCannotBeMaster='F'
                  MinimumPeriod='-1' TestBankMode='2' Latency='4'>
               <Banks>
                  ...
               </Banks>
               <PinMapping Attached='T' ModuleHandle='1'>
                  ...
               </PinMapping>
            </Core>
            <Core Name='Core 1' Type='0' SelectedBank='0'
                  SelectedSignal='-1' NumBanks='2' NumSignals='9'
                  NumPins='9' StateMode='F' TDM_1X='T'
                  ThresholdCode='92' TestBankAvailable='F'
                  ClockPodIndex='2' SelectedForUse='T' CoreIsMaster='T'
                  CoreCannotBeMaster='F' MinimumPeriod='-1'
                  TestBankMode='0'>
               <Banks>
                  ...
               </Banks>
               <PinMapping Attached='T' ModuleHandle='1'>
                  ...
               </PinMapping>
            </Core>
         </Cores>
      </Device>
   </Devices>
</Properties>
```

<Signal> Element (under Bank)

The <Signal> element describes a signal within a bank.

Children This element can have the following children: <Labels> (see page 91).

Parents This element can have the following parents: <Signals> (see page 102).

Example

```
<Signal>
   <Labels>
      <Label Name='/s2mon/tid' Bit='0' />
   </Labels>
</Signal>
```

<Signal> Element (under Probe)

The <Signal> element describes a signal within a defined probe.

Attributes

Name	Description
ClockChannel	'F' (false) or 'T' (true)
Name	'string'
PinMapIndex	'number'

Parents This element can have the following parents: <Signals> (see page 104).

Example

```
<Signal Name='ATD0' Pin='38' PinMapIndex='0' ClockChannel='F' />
```

<Signals> Element (under Bank)

The <Signals> element contains descriptions of signals within a bank.

Children This element can have the following children: <Signal> (see page 100).

Parents This element can have the following parents: <Bank> (see page 76).

Example

```
<Signals>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='0' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='1' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='2' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='3' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='4' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='5' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='6' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/tid' Bit='7' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='0' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='1' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='2' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='3' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='4' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='5' />
      </Labels>
   </Signal>
   <Signal>
      <Labels>
         <Label Name='/s2mon/lastackid' Bit='6' />
      </Labels>
   </Signal>
   <Signal>
      <Labels />
   </Signal>
</Signals>
```

<Signals> Element (under Probe)

The <Signals> element contains the signals used by a defined probe.

Children This element can have the following children: <Signal> (see page 101).

Parents This element can have the following parents: <Probe> (see page 97).

Example

```
<Signals>
   <Signal Name='ATD0' Pin='38' PinMapIndex='0' ClockChannel='F' />
   <Signal Name='ATD1' Pin='36' PinMapIndex='1' ClockChannel='F' />
   <Signal Name='ATD2' Pin='34' PinMapIndex='2' ClockChannel='F' />
   <Signal Name='ATD3' Pin='32' PinMapIndex='3' ClockChannel='F' />
   <Signal Name='ATD4' Pin='30' PinMapIndex='4' ClockChannel='F' />
   <Signal Name='ATD5' Pin='28' PinMapIndex='5' ClockChannel='F' />
   <Signal Name='ATD6' Pin='26' PinMapIndex='6' ClockChannel='F' />
   <Signal Name='ATD7' Pin='24' PinMapIndex='7' ClockChannel='F' />
   <Signal Name='ATD8' Pin='22' PinMapIndex='8' ClockChannel='F' />
   <Signal Name='ATD9' Pin='20' PinMapIndex='9' ClockChannel='F' />
   <Signal Name='ATD10' Pin='18' PinMapIndex='10' ClockChannel='F' />
   <Signal Name='ATD11' Pin='16' PinMapIndex='11' ClockChannel='F' />
   <Signal Name='ATD12' Pin='14' PinMapIndex='12' ClockChannel='F' />
   <Signal Name='ATD13' Pin='12' PinMapIndex='13' ClockChannel='F' />
   <Signal Name='ATD14' Pin='10' PinMapIndex='14' ClockChannel='F' />
   <Signal Name='ATCK' Pin='6' PinMapIndex='16' ClockChannel='T' />
</Signals>
```

<SymbolInfo> Element

The <SymbolInfo> elements contain the module bus/signal symbol settings associated with a particular bank.

Attributes

Name	Description
ModuleHandle	'number'
SymbolSettings	'string'

Parents This element can have the following parents: <Bank> (see page 76).

Example

```
<SymbolInfo ModuleHandle='1' SymbolSettings='
&lt;Module&gt;
   &lt;BusSignalSetup&gt;
      &lt;BusSignals&gt;
         &lt;BusSignal Name=&apos;My Bus 1&apos;/&gt;
         &lt;Folder Name=&apos;Core 0 FPGA Probe&apos;
               Comment=&apos;Created by FPGA Dynamic Probe-1&apos;&gt;
            &lt;BusSignal Name=&apos;/s2mon/tid&apos;/&gt;
            &lt;BusSignal Name=&apos;/s2mon/s2mstate&apos;/&gt;
         &lt;/Folder&gt;
      &lt;/BusSignals&gt;
      &lt;NetlistImport/&gt;
   &lt;/BusSignalSetup&gt;
   &lt;Config TimeOfTrigger=&apos;1.110240661 Gs&apos;
         CorrelatedTriggerTime=&apos;0 s&apos;
         UserSkewTime=&apos;0 s&apos; SystemTrigger=&apos;T&apos;/&gt;
&lt;/Module&gt;
'/>
```

See Also

- "<Module> Element (under Configuration Setup)" (in the online help)

<TriggerInfo> Element

The <TriggerInfo> elements contain the module trigger settings associated with a particular bank.

Attributes

Name	Description
ModuleHandle	'number'
TriggerSettings	'string'

Parents This element can have the following parents: <Bank> (see page 76).

Example

```
<TriggerInfo ModuleHandle='1' TriggerSettings='
&lt;Module&gt;
   &lt;Trigger Mode=&apos;State&apos; Type=&apos;Normal&apos;&gt;
      &lt;StoreQual Mode=&apos;Custom&apos;&gt;
         &lt;Event ParensNeeded=&apos;F&apos;&gt;
            &lt;Anything/&gt;
         &lt;/Event&gt;
      &lt;/StoreQual&gt;
      &lt;Step Number=&apos;1&apos;&gt;
         &lt;If&gt;
            &lt;Event ParensNeeded=&apos;F&apos;&gt;
               &lt;BusSignal Name=&apos;/s2mon/tid&apos;
                     Bit=&apos;All&apos; Operator=&apos;Equals&apos;
                     Value=&apos;h3F&apos;/&gt;
            &lt;/Event&gt;
            &lt;Occurrence Value=&apos;1&apos;
                  Mode=&apos;Eventual&apos;/&gt;
            &lt;Action&gt;
               &lt;TriggerAction Operator=&apos;Fill Memory&apos;&gt;
                  &lt;StoreQual Mode=&apos;Custom&apos;&gt;
                     &lt;Event ParensNeeded=&apos;F&apos;&gt;
                        &lt;DefaultStore/&gt;
                     &lt;/Event&gt;
                  &lt;/StoreQual&gt;
               &lt;/TriggerAction&gt;
            &lt;/Action&gt;
         &lt;/If&gt;
      &lt;/Step&gt;
   &lt;/Trigger&gt;
   &lt;Config TimeOfTrigger=&apos;1.110240661 Gs&apos;
         CorrelatedTriggerTime=&apos;0 s&apos;
         UserSkewTime=&apos;0 s&apos; SystemTrigger=&apos;T&apos;/&gt;
&lt;/Module&gt;
'/>
```

See Also

- "<Module> Element (under Configuration Setup)" (in the online help)

<WindowInfo> Element

The <WindowInfo> elements contain the display window settings associated with a particular bank.

Attributes

Name	Description
WindowHandle	'number'
WindowSettings	'string'

Parents This element can have the following parents: <Bank> (see page 76).

Example

```
<WindowInfo WindowHandle='1' WindowSettings='
&lt;Setup&gt;
   &lt;Sampling PerDivision=&apos;5 ns&apos; Delay=&apos;0 s&apos;/&gt;
   &lt;BusSignals&gt;
      &lt;Clear/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;My Bus 1&apos; DefaultBase=&apos;Hex&apos;
            Color=&apos;hFFFFFF&apos; Height=&apos;30&apos;/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;/s2mon/tid&apos; DefaultBase=&apos;Hex&apos;
            Color=&apos;hFFFFFF&apos; Height=&apos;30&apos;/&gt;
      &lt;BusSignal Module=&apos;My 1682D-1&apos;
            Name=&apos;/s2mon/s2mstate&apos;
            DefaultBase=&apos;Hex&apos; Color=&apos;hFFFFFF&apos;
            Height=&apos;30&apos;/&gt;
      &lt;BusSignal Name=&apos;Time&apos; Color=&apos;hFFFFFF&apos;
            Height=&apos;30&apos;/&gt;
   &lt;/BusSignals&gt;
&lt;/Setup&gt;
'/>
<WindowInfo WindowHandle='2' WindowSettings='
&lt;BusSignals&gt;
   &lt;Clear/&gt;
   &lt;BusSignal Module=&apos;My 1682D-1&apos;
         Name=&apos;Sample Number&apos; Color=&apos;hFFFFFF&apos;
         Alignment=&apos;Right&apos; Width=&apos;112&apos;/&gt;
   &lt;BusSignal Module=&apos;My 1682D-1&apos;
         Name=&apos;My Bus 1&apos; DefaultBase=&apos;Hex&apos;
         Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
         Width=&apos;113&apos;/&gt;
   &lt;BusSignal Name=&apos;Time&apos; DefaultBase=&apos;Absolute&apos;
         Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
         Width=&apos;152&apos;/&gt;
   &lt;BusSignal Module=&apos;My 1682D-1&apos;
         Name=&apos;/s2mon/tid&apos; DefaultBase=&apos;Hex&apos;
         Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
         Width=&apos;113&apos;/&gt;
   &lt;BusSignal Module=&apos;My 1682D-1&apos;
         Name=&apos;/s2mon/s2mstate&apos; DefaultBase=&apos;Hex&apos;
         Color=&apos;hFFFFFF&apos; Alignment=&apos;Right&apos;
         Width=&apos;120&apos;/&gt;
```

```
&lt;/BusSignals&gt;
'/>
```

See Also

- "<Window> Element (under Configuration Setup)" (in the online help)

Index