Keysight Altera FPGA Dynamic Probe

Online Help

KEYSIGHT
TECHNOLOGIES

Notices

Trademarks

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

Revision History

August 2014

Available in electronic format only

Warranty

Technology Licenses

Restricted Rights Legend

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

Contents

Contents

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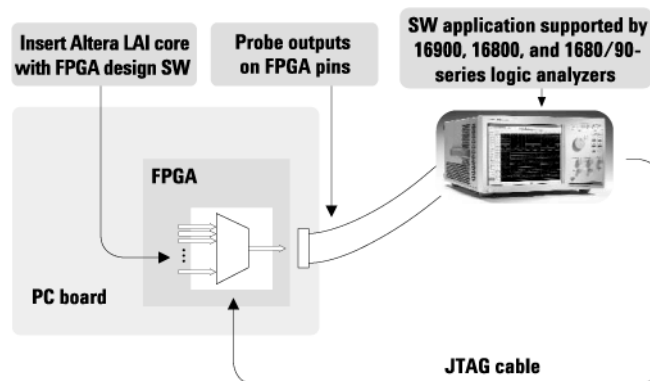
Create a time-saving FPGA measurement system. Insert an Altera LAI (Logic Analyzer Interface) core into your FPGA design. With the application running on your logic analyzer via JTAG, you control which group of internal signals to measure.

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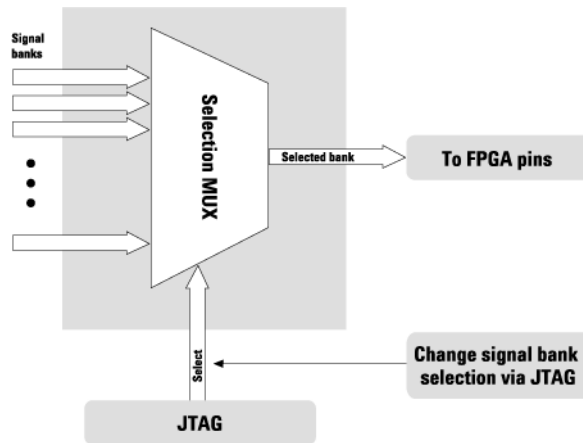
The Altera LAI core provides a multiplexer for selecting the signals that are output to FPGA debug pins. Core parameters determine the number of signals per bank and the number of banks.

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).

KEYSIGHT
TECHNOLOGIES

3 Installing and Licensing the FPGA Dynamic Probe

Before you can use the Keysight 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.keysifgt.com/find/la-sw-download"

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

KEYSIGHT
TECHNOLOGIES

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 46

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

Next

KEYSIGHT
TECHNOLOGIES

5 Preparation Steps

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

1 Install the JTAG cable programmer software (see)

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

Next   ·   Measurement Steps (see)

KEYSIGHT
TECHNOLOGIES

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 *Keysight 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 *Keysight 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 *Keysight 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 *Keysight 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)

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....
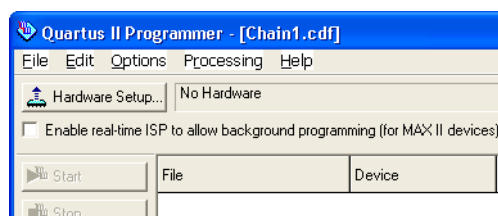
   c   In the Hardware Setup dialog, select the appropriate JTAG cable hardware.
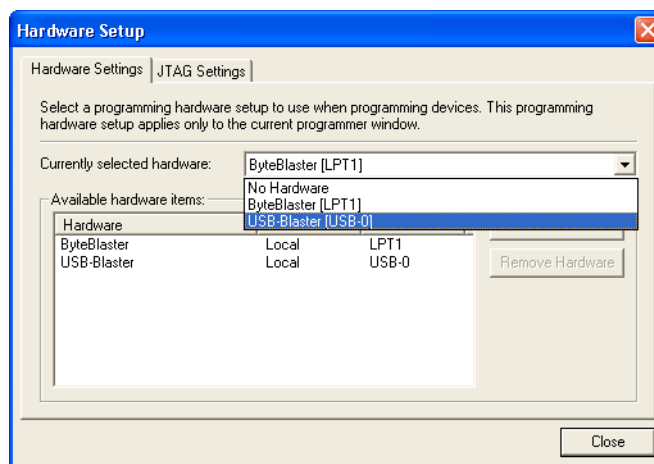
You may have to click Add Hardware... to add the appropriate JTAG cable hardware.

   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 *Keysight 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 26)

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 >.
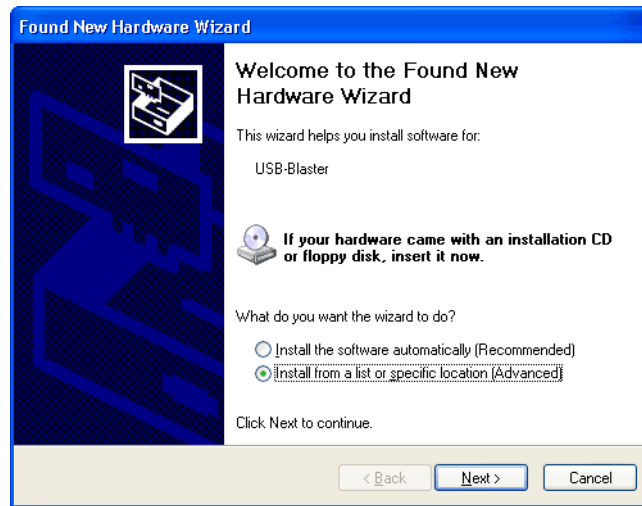
3   On the next page of the wizard, specify the drivers\usb-blaster\x32 (for a 32-bit operating system) or drivers\usb-blaster\x64 (for a 64-bit operating system) location underneath the Quartus II Programmer install directory; then, click Next >.
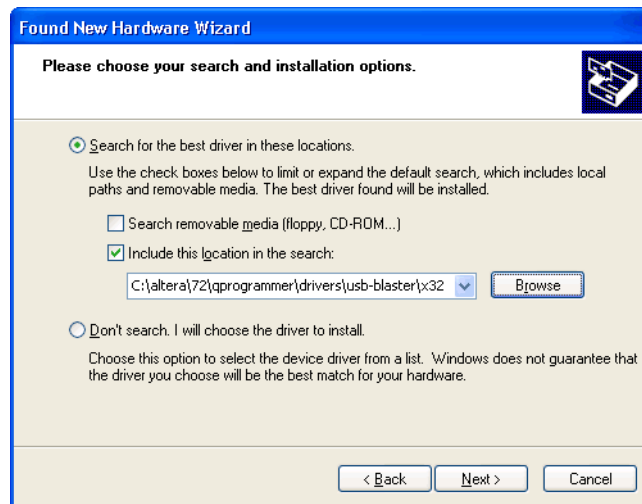
4   When the wizard has finished installing the driver, click Finish.
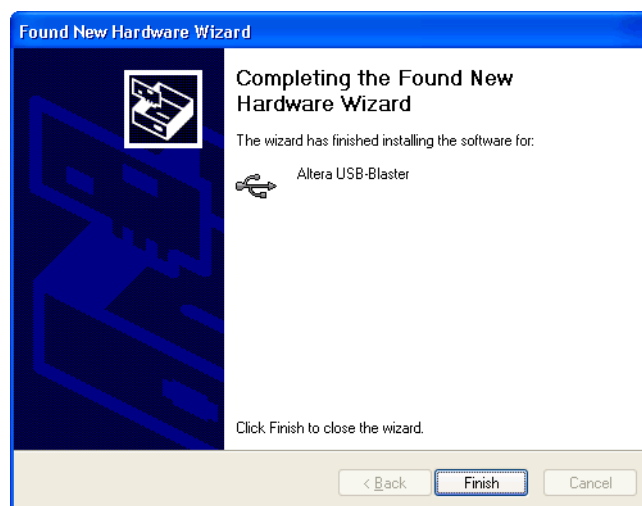
For more information on installing the USB Blaster driver, see
http://www.altera.com/support/software/drivers/usb-blaster/dri-usb-blaster-xp.html" (or
search for "USB Blaster driver" on the Altera web site).

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 *Keysight 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 *Keysight 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 *Keysight 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....
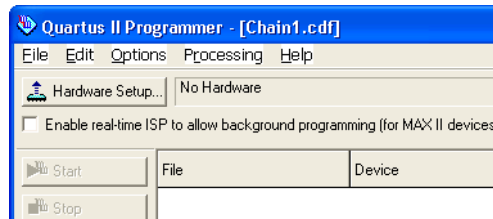
c   In the Hardware Setup dialog's JTAG Settings tab, click **Configure Local JTAG Server…**.
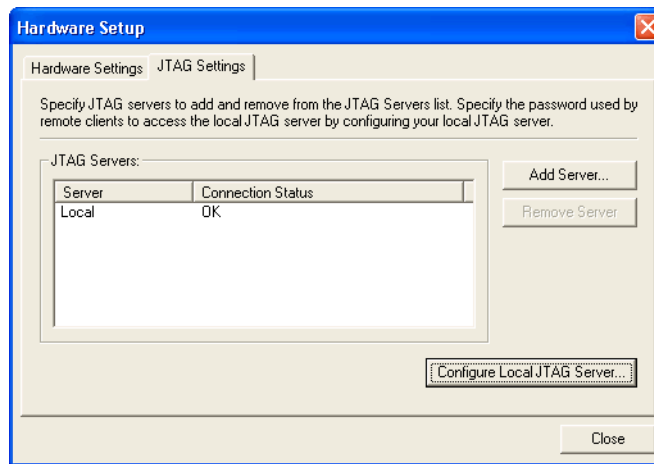
d   In the Configure local JTAG Server dialog, check **Enable remote clients to connect to the local JTAG server**, enter a password, and confirm the password.
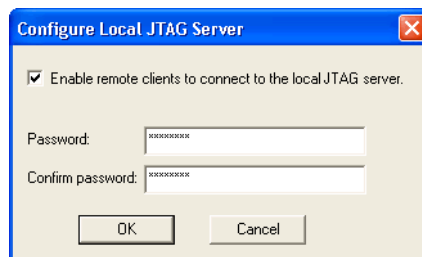
Record this password as you will need it later. Then, click **OK**.

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' t allow exceptions**.
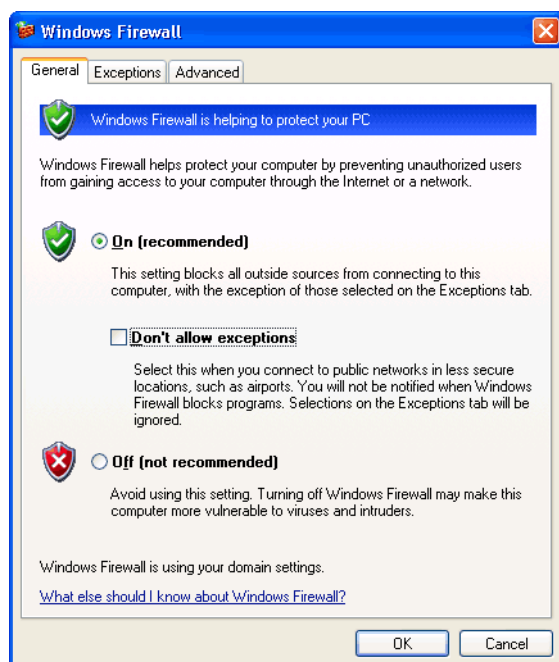
If the firewall is off, it may be that the computer is using different firewall software. In this case, you must enable access to the Quartus II Programmer port using the different firewall software.

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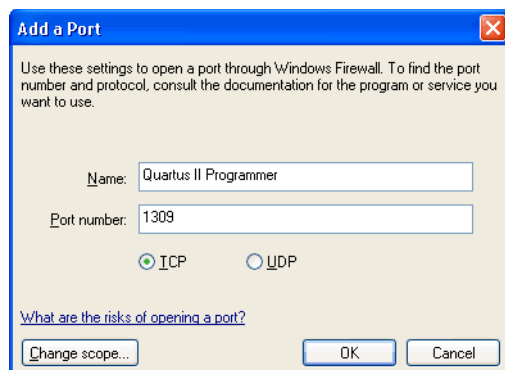
e Click **OK** to close the Windows Firewall dialog.

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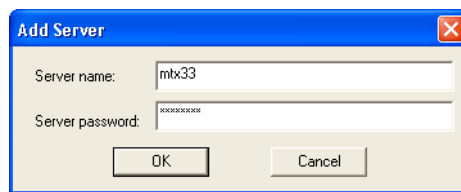
   d   Verify that the Hardware Setup dialog shows a Connection Status of "OK". If not, then there is a problem with the networking. Re-verify your firewall settings and server-name/server-password settings.
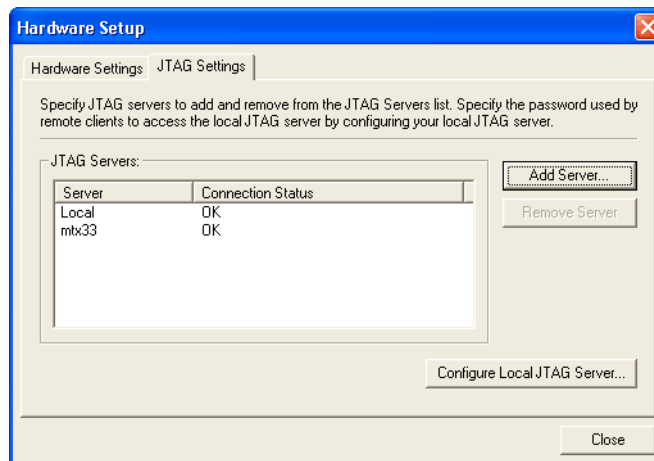
   e   Click Close to close the Hardware Setup dialog.

   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), 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 *Keysight Logic Analyzer* application:

1 Establish connection between analyzer and JTAG cable (see page 26)
2 Download configuration bits into FPGA (see page 28)
3 Import signal names (see page 29)
4 Map FPGA pins (see page 33)
5 Make the measurement (see page 37)

KEYSIGHT
TECHNOLOGIES

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.
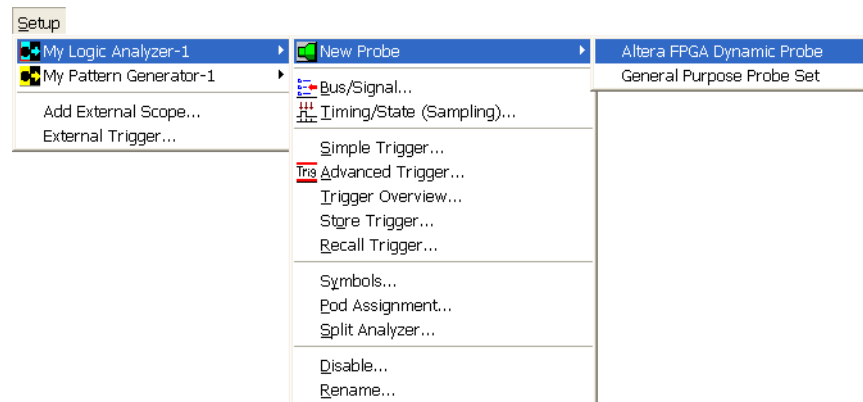
Or, in the Overview window, from a module's drop-down menu, choose New Probe>FPGA Dynamic Probe.
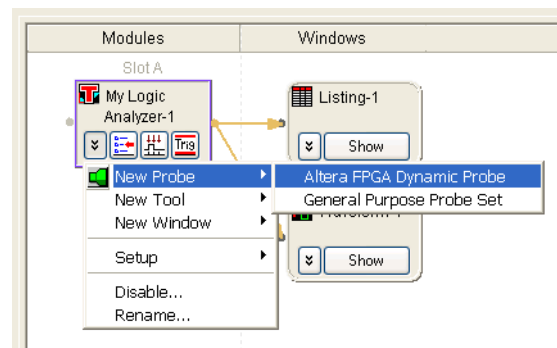
2   In the FPGA Dynamic Probe Setup dialog (see), click Cable Connection....
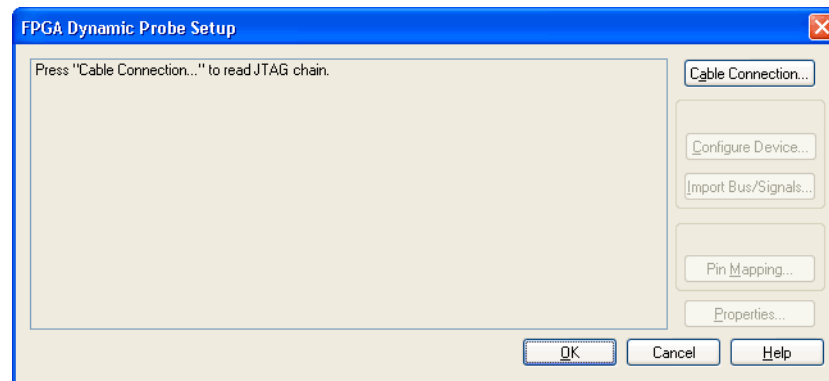
3   In the Cable Connection dialog (see), select the type of cable and, if necessary, specify any cable parameters; then, click OK.
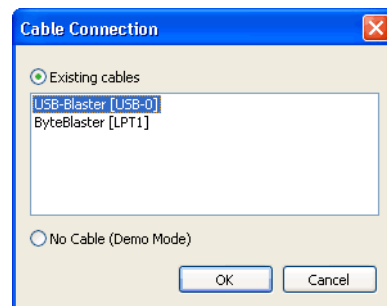
When the connection has been established, you will see the devices on the JTAG chain, and you can select the desired device.
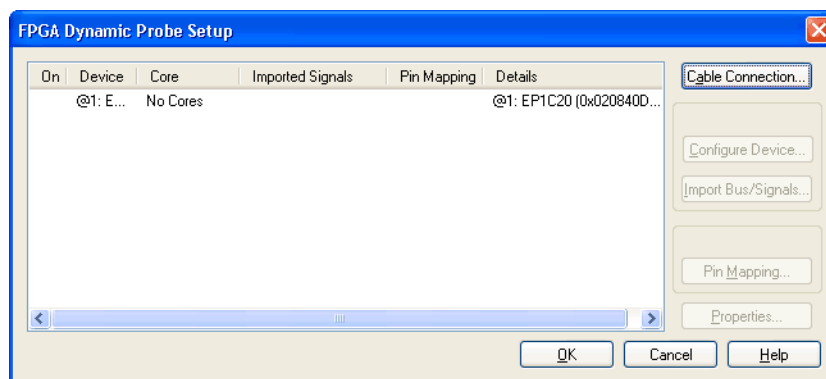
Next   ·   Step 4. Download configuration bits into FPGA (see)

Step 4. Download configuration bits into FPGA

To download configuration bits into an FPGA:

1   In the FPGA Dynamic Probe Setup dialog (see page 58), select the FPGA device to which you wish to download configuration bits; then, click Configure Device....
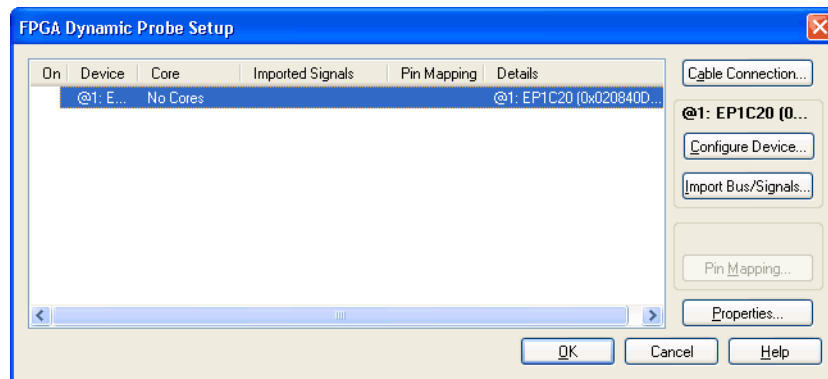
2   In the Select FPGA Configuration File dialog (see page 60), select the FPGA configuration file; then, click Open.

Next   ·   Step 5. Import signal names (see page 29)

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 58), select the device whose bus/signal names you want to import; then, click Import Bus/Signals....
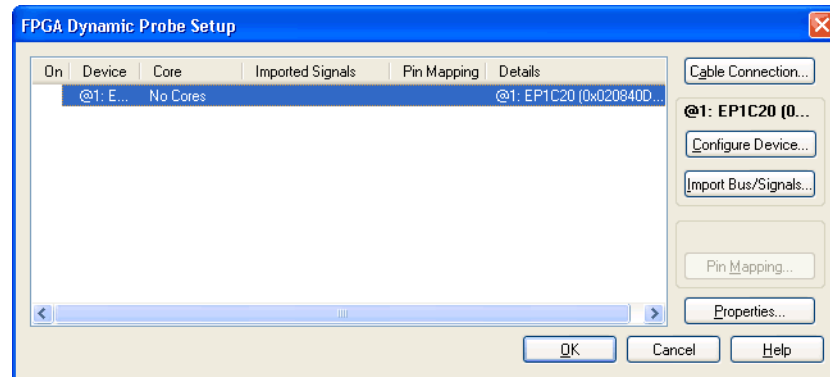
2   In the Select Signal Import File dialog (see page 61), select the signal import file; then, click Open.

3   In the import results dialog, view the bus/signal import information; then, click OK.
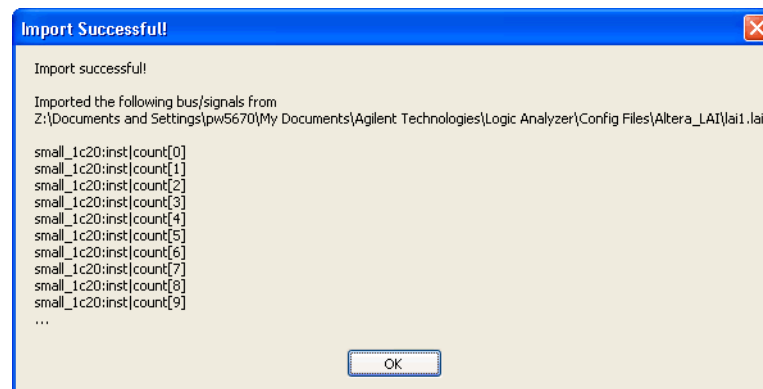
4   In the FPGA Dynamic Probe dialog, note that the defined cores now appear.
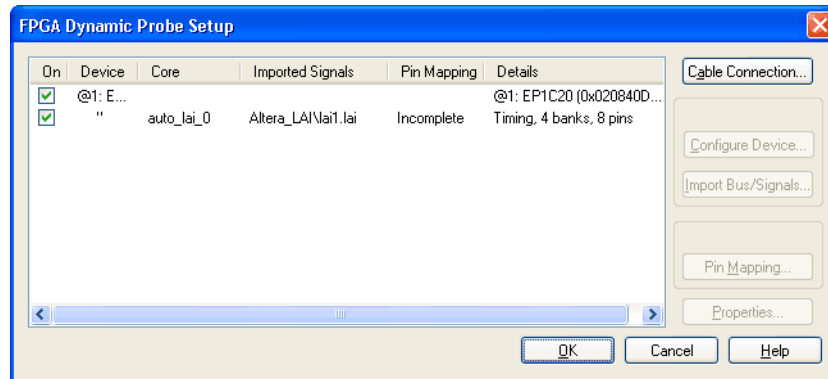
See Also · To trim imported bus/signal names (see page 30)

· To rename imported bus/signal names (see page 30)

· To define additional FPGA bus/signal names (see page 31)

Next · Step 6. Map FPGA pins (see page 33)

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 63), click Trim Bus/Signal Names....

2 In the Trim Bus/Signal Names dialog (see page 63), 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....
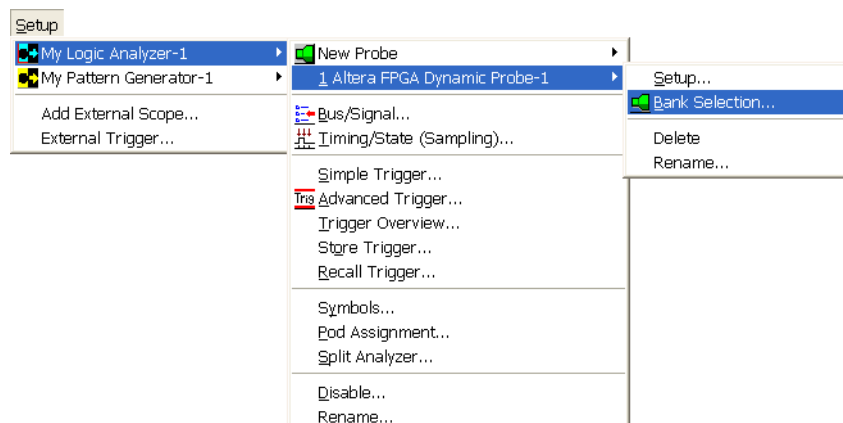
Or, in the Overview window, click the FGPA dynamic probe's Properties button; then, choose Bank Selection....
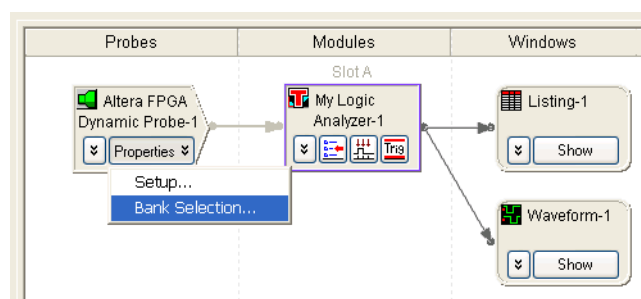
Note also that you can triple-click signal names in the FPGA Dynamic Probe dialog to rename them (without having to do global trimming).

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....
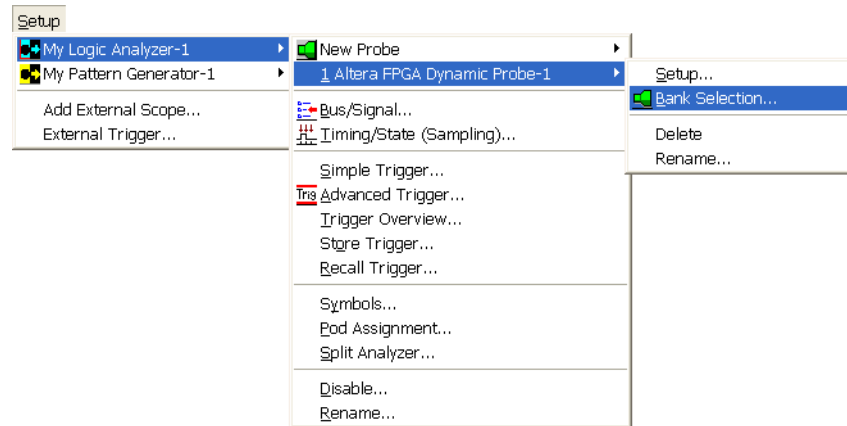
Or, in the Overview window, click the FGPA dynamic probe's Properties button; then, choose Bank Selection....
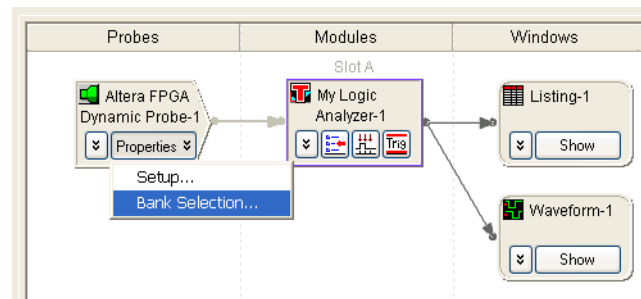
Whenever buses/signals are added to the "FPGA Probe" folder, they are associated with a specific bank. If you select another bank, the added buses/signals do not appear.

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 58), select the Altera LAI core whose output pins you want to map; then, click Pin Mapping....
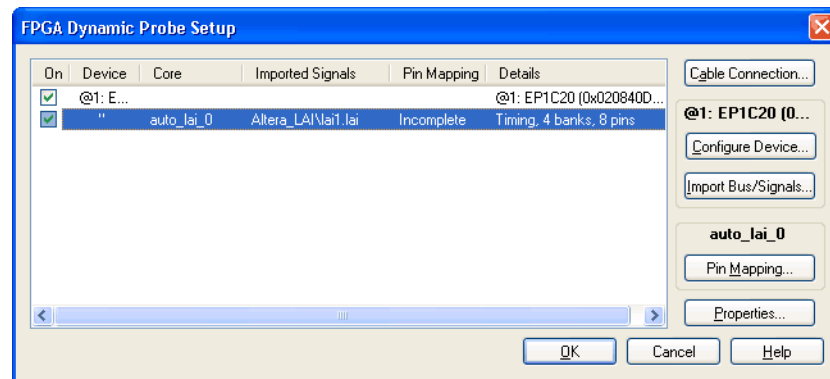
2   In the Pin Mapping dialog (see page 59), click Add Probe....
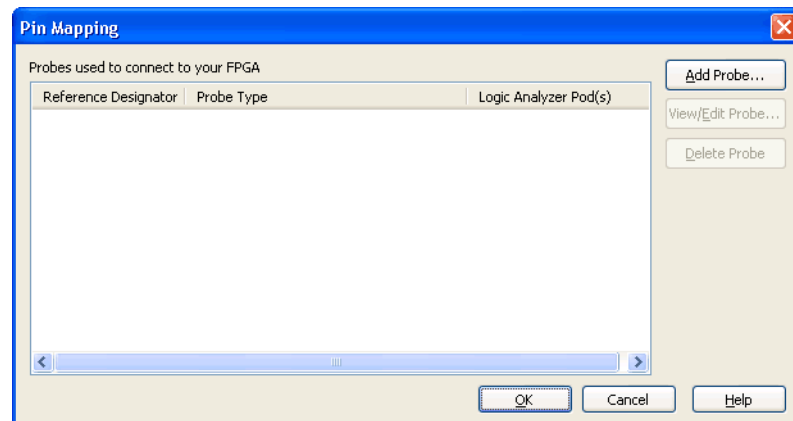
3   In the "Select the type of probe to add" dialog, select the type of probe that is used to connect to your FPGA; then, click **OK**.
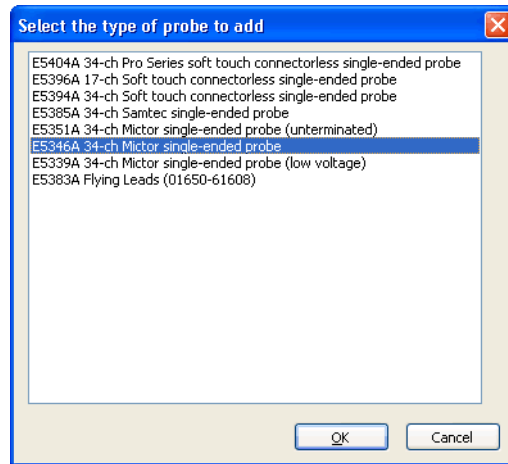
If your probe doesn't appear in the list, you can "download the latest probe definitions from the web" (in the online help).

4   In the Pin Mapping Edit dialog (see), select the FPGA pins (you can select multiple pins using Shift-click or Ctrl-click) and drag them on to the pin/pad map.
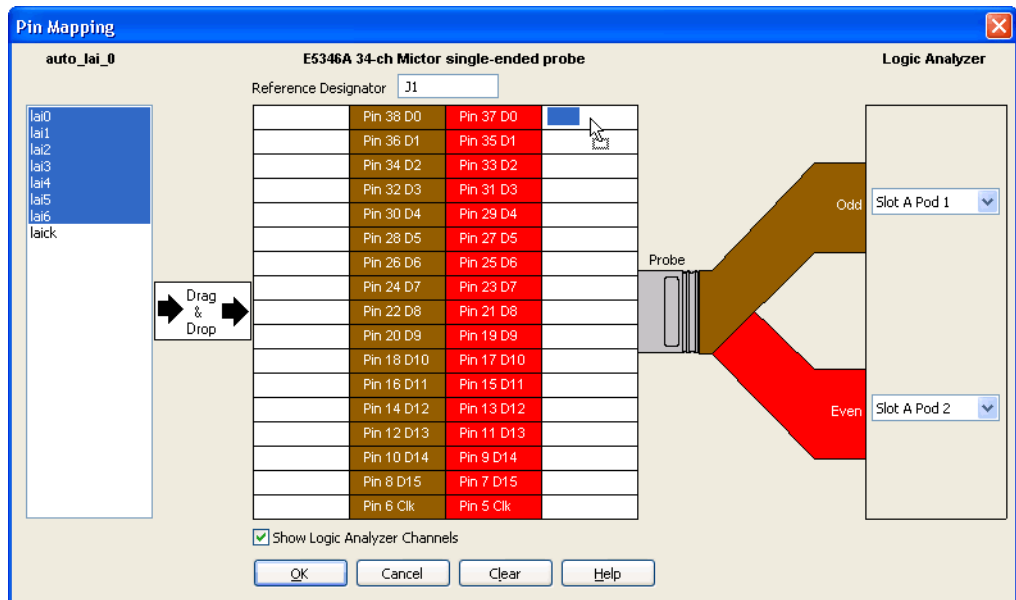
After you've mapped FPGA pins to the probe, you can hover the mouse pointer over a pin description field to view a tool tip describing the FPGA debug pin name, the pod connection, the channel number, and the signal type (single-ended or differential).
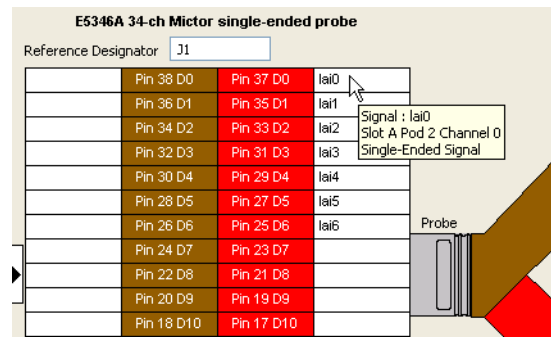
You can clear all FPGA pins that have been mapped to pins/pads by clicking Clear. You can clear individual pin mappings by dragging a pin from the pin/pad diagram back to the FPGA Pins list.

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.
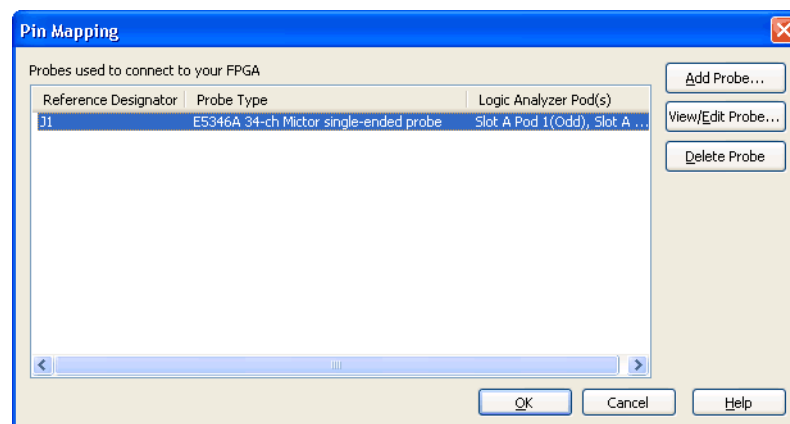
7   Click OK to close the Pin Mapping dialog.

In the FPGA Dynamic Probe Setup dialog, notice pin mapping is no longer "Incomplete".
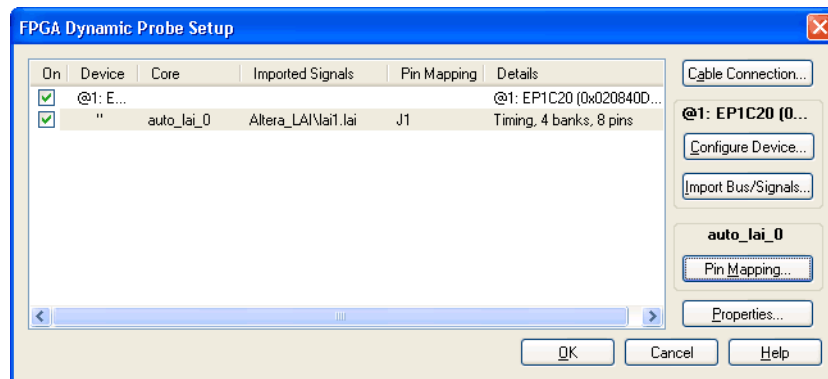
8   Click OK to close the FPGA Dynamic Probe Setup dialog.

The FPGA Dynamic Probe dialog opens automatically. If you expand a bank, you see the imported bus/signal names.
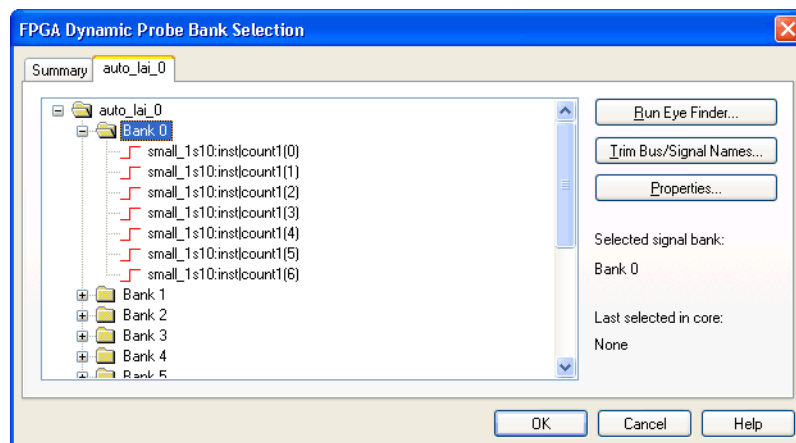
Next   · Step 7. Make the measurement (see page 37)

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 [icon] 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.
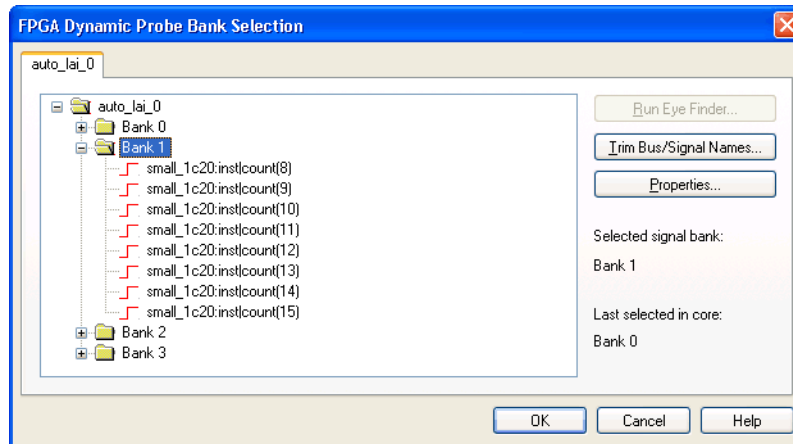
You can correlate internal FPGA activity with external measurements. With each new selection of a signal bank, the application updates new signal names from your design to the logic analyzer. View internal FPGA activity and time correlate internal FPGA measurements with external events in the surrounding system.
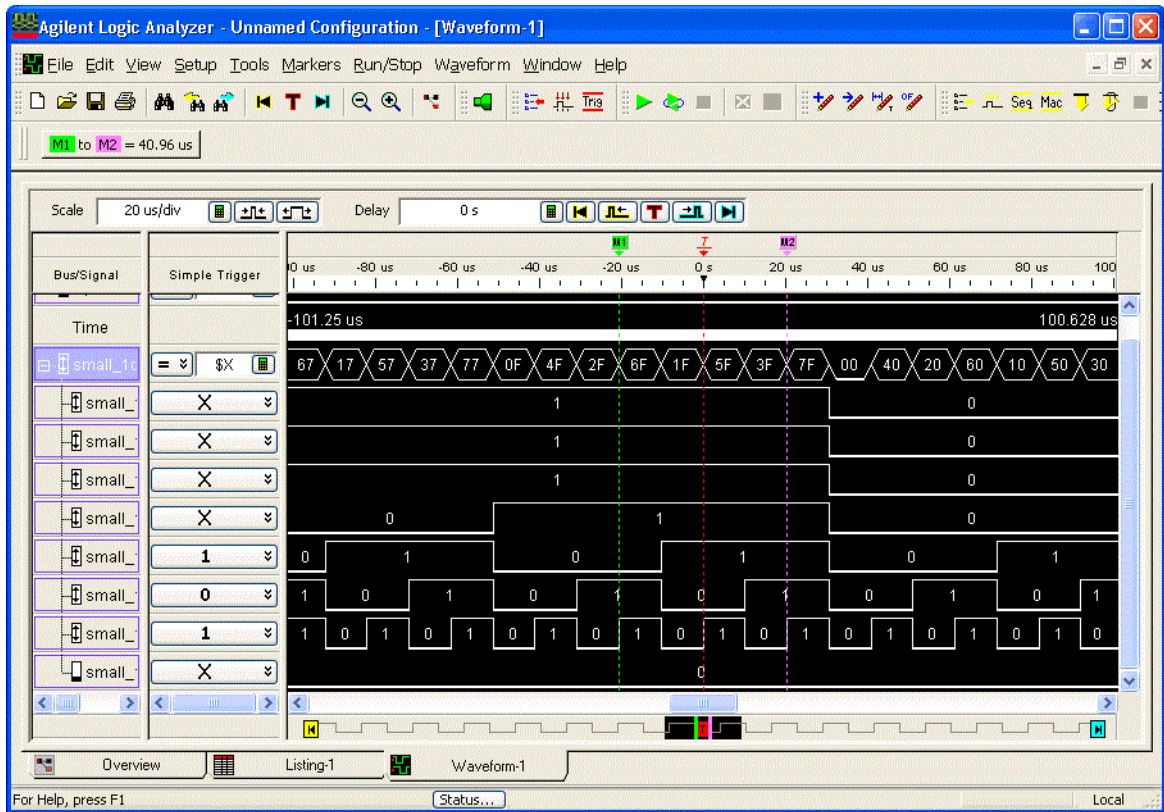
NOTE   Timing zoom is automatically disabled when using the FPGA dynamic probe. You can re-enable timing zoom; however, because of the Altera LAI core, timing zoom does not provide an accurate representation of internal FPGA signals.

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

KEYSIGHT
TECHNOLOGIES

When I click the 'Cable Connection' button I get 'problem locating Altera software' dialog
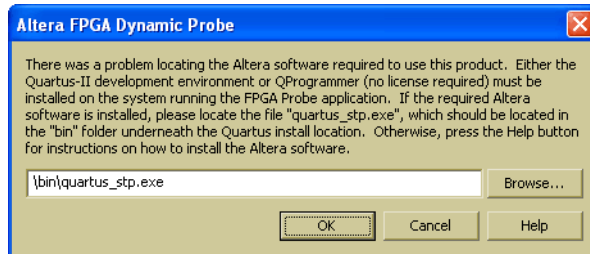
Possible cause: Default install path was changed during Quartus II installation. Actions:

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)
  - 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

KEYSIGHT
TECHNOLOGIES

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:

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**.
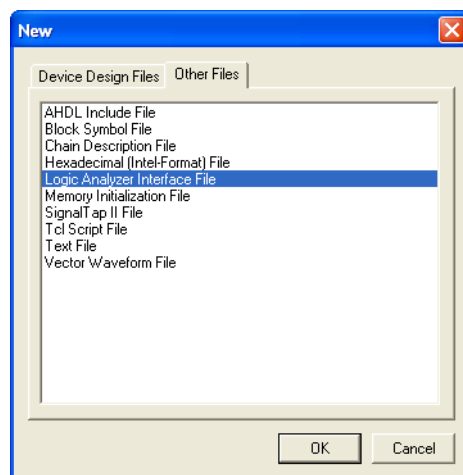
4   Click **OK**.

The Logic Analyzer Interface editor opens.

Next   ·

Configuring the Logic Analyzer Interface Core Parameters

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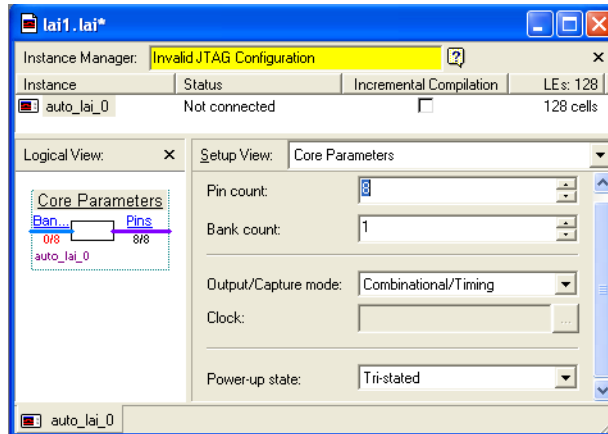
Screen resolution might cause your screen to look different; please resize or scroll within the window to view the parameter.

The parameters can be set depending on the user' 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' 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.

Parameter	Description
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' 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

To assign the Clock, which will synchronize the state capture, click the … button. This will launch the Node Finder tool.
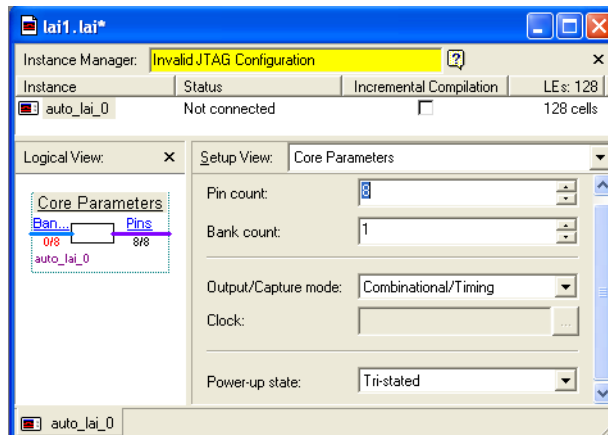
Click **List** to display the signals from the synthesized design. Search through signal names and locate the clock signal, small_1s10:inst|clk. When the signal is selected, click the ≥ button to make the core assignment. Click **OK** to continue.
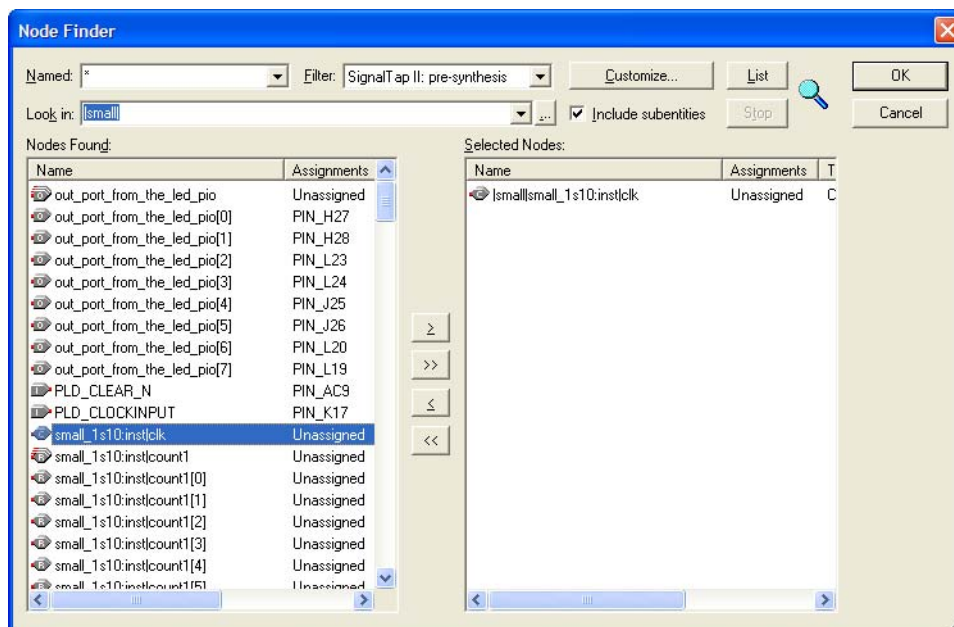
Mapping the Logic Analyzer Interface File Pins to Available I/O Pins

To configure the LAI I/O pins, from the **Setup View** list select **Pins**.
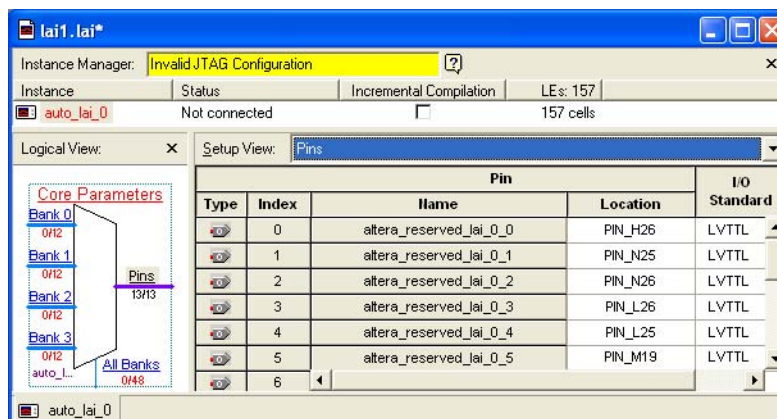
To assign pin locations double-click inside the **Location** column next to the reserved pins in the **Name** column. This action opens the Pin Planner tool.

| 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
altera_reserved_lai_0_11	PIN_M26	LVTTL
altera_reserved_lai_0_12	PIN_M25	LVTTL
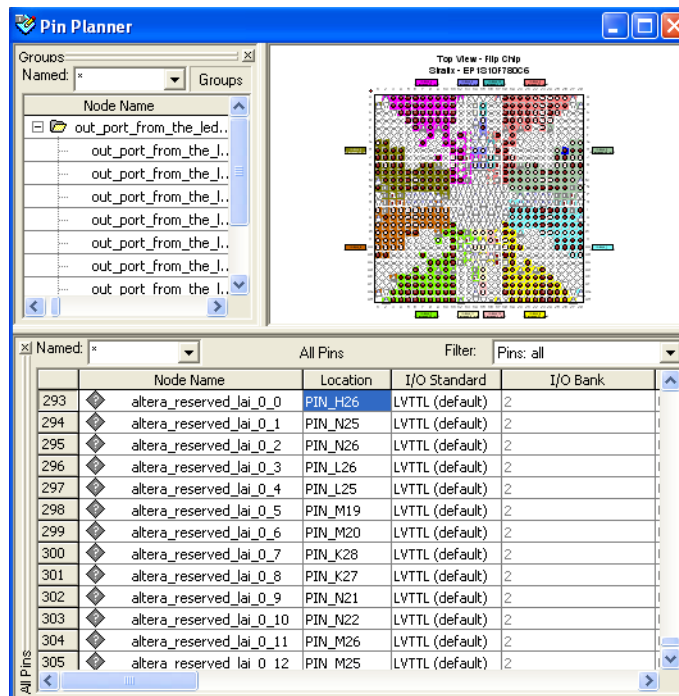
Once all LAI nodes have been assigned pin locations, close the Pin Planner window. The pin assignments will be reflected in the Setup View.

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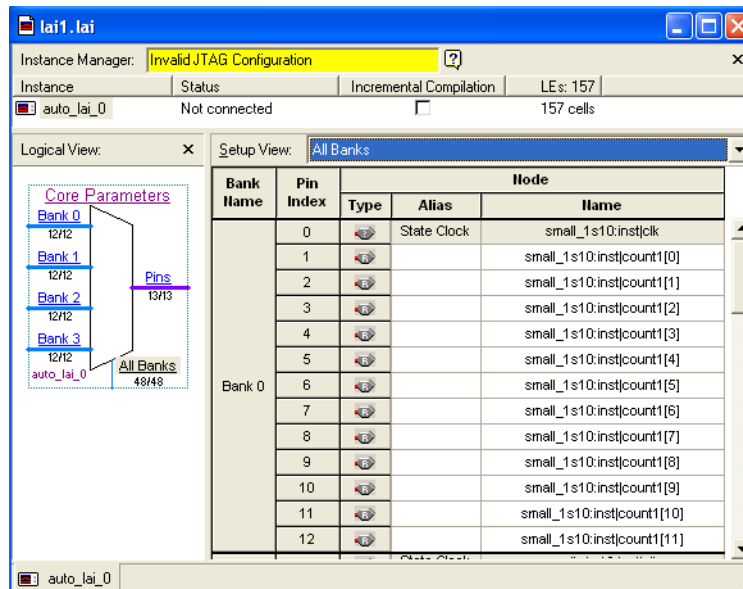
NOTE	If using a state core, bit 0 of each bank will be reserved for the clock.

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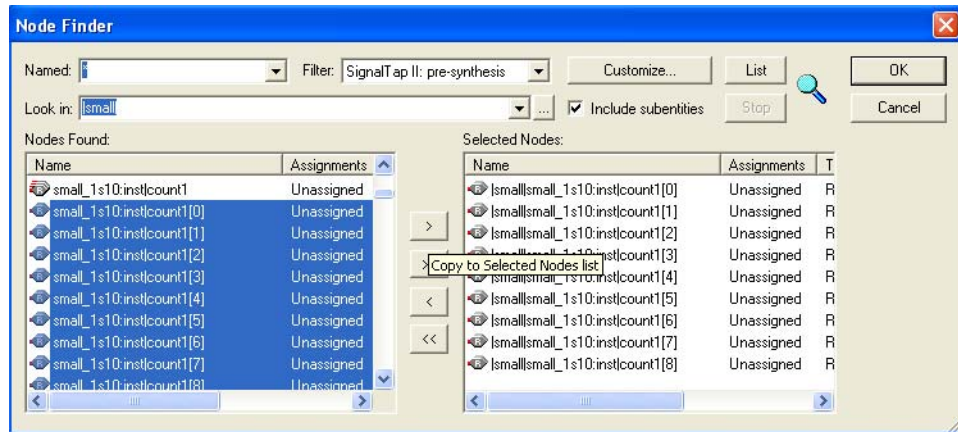
The design contains four counters; count1 and count3 are up counters and count2 and count4 are down counters. Assign a counter output per bank, for example, count1 to Bank0, count2 to Bank1, count3 to Bank2, and count4 to Bank3.

Next   ·   "Saving the Logic Analyzer Interface" on page 52

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.
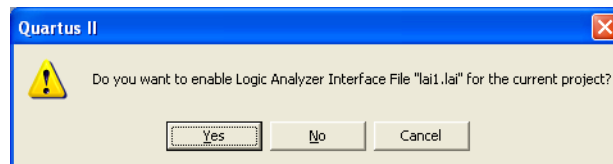
Next   ·   "Enabling/Disabling the Logic Analyzer Interface" on page 52

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.
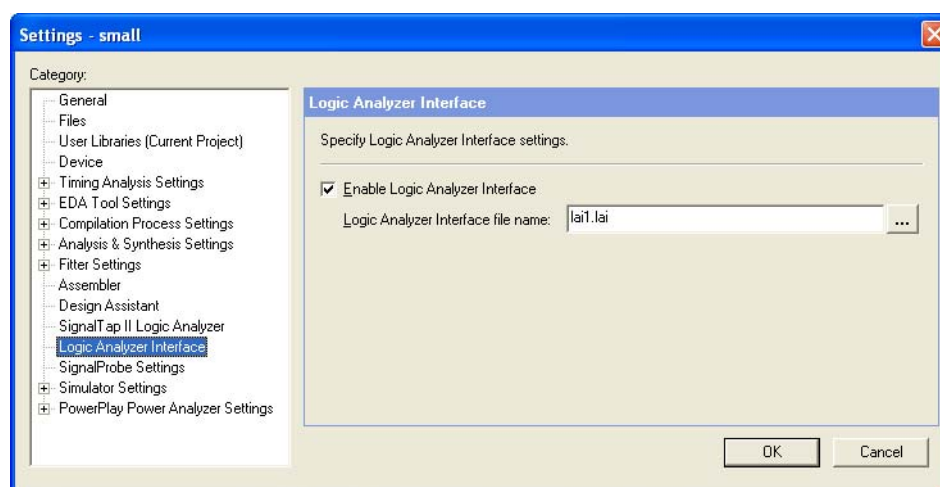
Click **OK** to continue.

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.

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**.
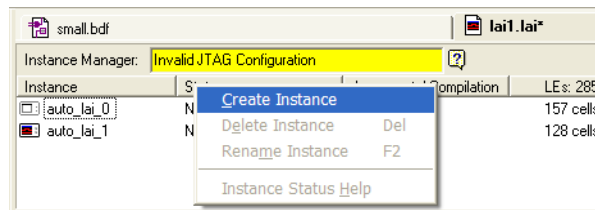
Automated Logic Analyzer Set Up

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 30) and define additional bus/signal names (see page 31), 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

KEYSIGHT
TECHNOLOGIES

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 (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 62) within its Properties dialog (see page 61)), 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 58) for establishing a connection between the logic analyzer and the Altera LAI core.
Configure Device...	Opens the Select FPGA Configuration File dialog (see page 60) for downloading a design into the selected FPGA device.
Import Bus/Signals...	Opens the Select Signal Import File dialog (see page 61) for importing internal FPGA bus/signal names.
Pin Mapping...	Opens the Pin Mapping dialog (see page 59) 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 33)).
Properties...	Opens the Properties dialog (see page 61) which lets you rename devices and cores as well as display information about the selected Altera LAI core.

See Also   · Measurement Steps (see page 25)

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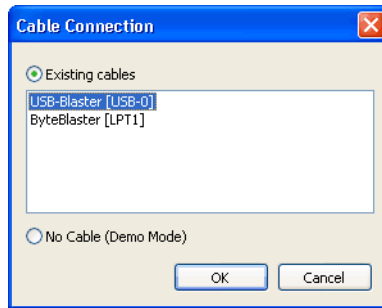
Existing cables	Lets you select a cable that has been set up using the Altera Quartus II Programmer (or design) software.
No Cable (Demo Mode)	Simulates a cable connection for demonstration purposes.

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 26)

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.

Add Probe...	Opens the "Select the type of probe to add" dialog; when you click **OK**, the Pin Mapping Edit dialog (see page 60) 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 60) 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 33)

Pin Mapping Edit Dialog

The Pin Mapping Edit dialog lets you map the FPGA output pins to the logic analyzer probe pins/pads.
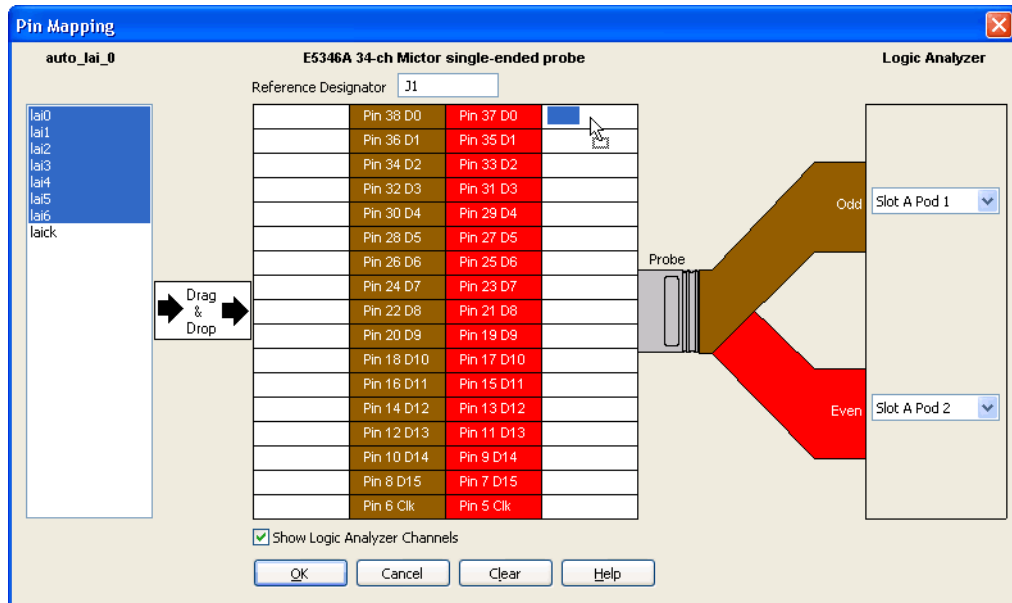
Reference Designator	Identifies the reference designator (in the device under test) of the probe connector, the connectorless probe retention module, or pins probed by flying leads.
FPGA Pins	Lists the FPGA pins used for the Altera LAI core outputs. When dragging these pins onto the pin/pad map, you 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)

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.
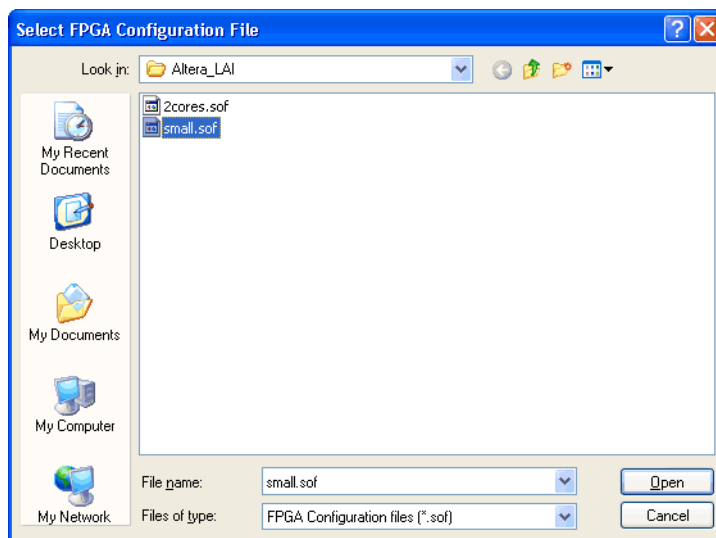
Select Signal Import File Dialog

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.
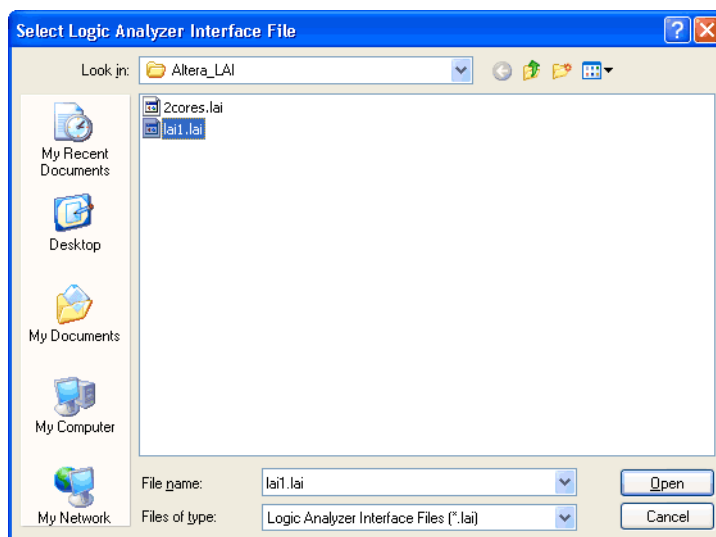
See Also   ·   Step 5. Import signal names (see page 29)

Properties Dialog

The Properties dialog lets you change device, core, and bank names as well as view detailed information about the core.
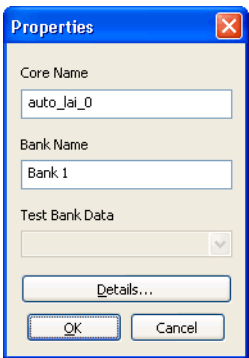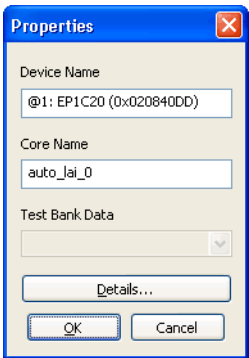
Device Name	Lets you rename the selected device.
Core Name	Lets you rename the selected core.
Bank Name	Lets you rename the selected bank.
Test Bank Data	Not available with Altera LAI cores.
Details...	Opens the Core Details dialog (see page 62) which displays information about the selected Altera LAI core.

Core Details Dialog

The Core Details dialog displays information about the selected Altera LAI core.
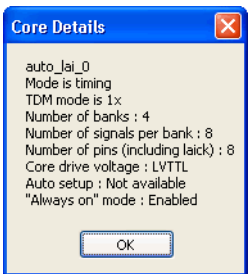
FPGA Dynamic Probe Bank Selection Dialog

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.

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 63) for shortening imported FPGA internal bus/signal names.
Properties...	Opens the Properties dialog (see page 61) which lets you rename cores and banks as well as display information about the selected Altera LAI core.

See Also   ·   Measurement Steps (see page 25)

Trim Bus/Signal Names Dialog

The Trim Bus/Signal Names dialog lets you specify how imported bus/signal names should be shortened.
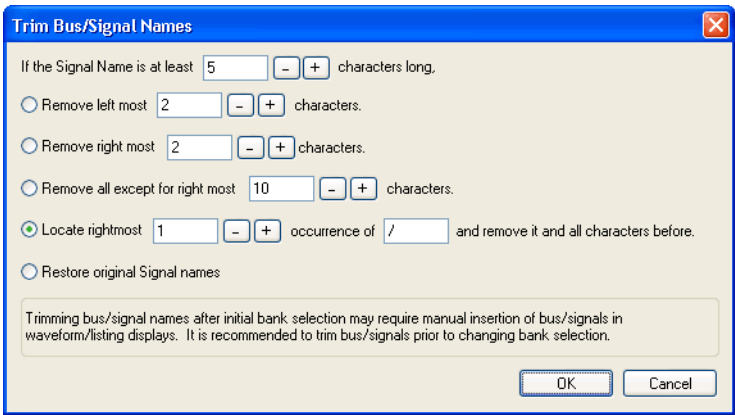
If the Signal Name is at least	Only bus/signal names longer than this value will be trimmed.
Remove left most	The number of characters to remove from the beginning of the names.
Remove right most	The number of characters to remove from the end of the names.
Remove all except right most	The number of 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)

Specifications and Characteristics

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

- Supported Logic Analyzers (see page 65)
- FPGA Dynamic Probe Software Application (see page 65)

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 *Keysight Logic Analyzer* application includes the COM Automation Server. This software lets you write programs that control the *Keysight 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 69)).

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 94)).
GetProperties	Returns the current setup, using the full tag set, equivalent to "GetAllSetup" (see the XML format ‹Properties› element (see page 94)).

See Also

· "COM Automation" (in the online help)
· Probe Setup, XML Format (see page 69)

KEYSIGHT
TECHNOLOGIES

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 94)
   <ATC_II> (see  page 71)
   <JTAG_Chain> (see  page 84)
   <Devices> (see  page 83)
      <Device> (see  page 81)
         <Cores> (see  page 79)
            <Core> (see  page 77)
               <Banks> (see  page 76)
                  <Bank> (see  page 72)
                     <Signals> (see  page 98)
                        <Signal> (see  page 96)
                           <Labels> (see  page 87)
                              <Label> (see  page 85)
                     <NonATCLabels> (see  page 88)
                        <Label> (see  page 86)
                           <Assignment> (see  page 70)
                     <WindowInfo> (see  page 103)
                     <SymbolInfo> (see  page 101)
                     <TriggerInfo> (see  page 102)
               <PinMapping> (see  page 89)
                  <DefinedProbes> (see  page 80)
                     <Probe> (see  page 93)
                        <Pods> (see  page 92)
                           <Pod> (see  page 91)
                        <Signals> (see  page 100)
                           <Signal> (see  page 97)
```

See Also
- "XML Format" (in the online help)
- Probe Control, COM Automation (see page 67)

KEYSIGHT
TECHNOLOGIES

\<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 86).

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 94).

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 98), `<NonATCLabels>` (see page 88), `<WindowInfo>` (see page 103), `<SymbolInfo>` (see page 101), `<TriggerInfo>` (see page 102).

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

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 72).

Parents   This element can have the following parents: `<Core>` (see page 77).

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 76), `<PinMapping>` (see page 89).

Parents   This element can have the following parents: `<Cores>` (see page 79).

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 77).

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

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).

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

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 79).

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

Example
```
<Device Name='Device 1' Type='0' NumCores='2' UserRegNum='1'
      Configurable='T' IRLength='6' CDCFilename='C:\Documents and
      Settings\user\My Documents\Keysight Technologies\Logic
      Analyzer\Config Files\demo\Altera_demo_V8.cdc'
      FPGAFilename='C:\Documents and Settings\user\My Documents\Keysight
      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 81).

Parents This element can have the following parents: `<Properties>` (see page 94).

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\Keysight Technologies\Logic
          Analyzer\Config Files\demo\Altera_demo_V8.cdc'
          FPGAFilename='C:\Documents and Settings\user\My
          Documents\Keysight 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).

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).

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 70).

Parents This element can have the following parents: `<NonATCLabels>` (see page 88).

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 85).

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

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 86).

Parents   This element can have the following parents: `<Bank>` (see page 72).

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 80).

Parents   This element can have the following parents: \<Core\> (see page 77).

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 92).

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 91).

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

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 92),`<Signals>` (see page 100).

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

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 71), `<JTAG_Chain>` (see page 84), `<Devices>` (see page 83).

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\Keysight Technologies\Logic
            Analyzer\Config Files\demo\Altera_demo_V8.cdc'
            FPGAFilename='C:\Documents and Settings\user\My
            Documents\Keysight 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 87).

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

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 100).

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 96).

Parents This element can have the following parents: \<Bank> (see page 72).

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).

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

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 72).

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 72).

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 72).

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

Symbols

.bit FPGA configuration file, 60
.lai FPGA signal import file, 61

A

activity not on FPGA dynamic probe, 42
Altera FPGA dynamic probe, 3
Altera JTAG cable programmer software, 18
Altera LAI core, selecting, 26
Altera LAI core, selecting signal banks, 37
Altera LAI cores, enabling/disabling, 58
Altera software, problem locating, 40
Assignment, XML element, 70
ATC_II, XML element, 71
automated logic analyzer set up (FPGA dynamic probe), 55

B

B4656A FPGA dynamic probe for Altera FPGAs, 13
Bank, XML element, 72
banks (Altera LAI core), selecting, 37
Banks, XML element, 76
bus/signal names, adding FPGA, 31
bus/signal names, renaming FPGA, 30
bus/signal names, trimming FPGA, 30

C

Cable Connection dialog, 58
cable connection, selecting type of, 26
captured data, invalidation of, 37
client, JTAG, 21
Clk pin/pad locations, 33
clock activity indicators, 43
COM automation, FPGA dynamic probe, 67
concepts, FPGA dynamic probe for Altera FPGAs, 45
configuration file, FPGA, 60
configure FPGA device, 28
connection to Altera LAI core, establishing, 26
Core Details dialog, 62
core name, 61

Core, XML element, 77
cores (Altera LAI), enabling/disabling, 58
cores, multiple, mapping to two halves of the same probe, 33
Cores, XML element, 79

D

data, invalidation of, 37
debug pins (FPGA), mapping to probes and pods, 33
debug pins (FPGA), probing, 11
debug pins, FPGA, 60
DefinedProbes, XML element, 80
demo mode, FPGA dynamic probe, 58
design (FPGA) tools, 65
design steps for FPGA dynamic probe, 15
device name, 61
Device, XML element, 81
Devices, XML element, 83
disabling/enabling Altera LAI cores, 58

E

enabling/disabling Altera LAI cores, 58
error, script, 41

F

firewall, Quartus II Programmer port number, 23
FPGA bus/signal names, adding, 31
FPGA configuration file, 60
FPGA debug pins, 60
FPGA debug pins, mapping to probes and pods, 33
FPGA debug pins, probing, 11
FPGA Dynamic Probe Bank Selection dialog, 63
FPGA dynamic probe characteristics, 65
FPGA Dynamic Probe Setup dialog, 58
FPGA dynamic probe specifications, 65
FPGA dynamic probe, automated logic analyzer set up, 55
FPGA dynamic probe, design steps, 15
FPGA dynamic probe, installing and licensing, 13
FPGA dynamic probe, measurement steps, 25

FPGA dynamic probe, no activity, 42
FPGA dynamic probe, state mode measurements, 43
FPGA imported bus/signal names, renaming, 30
FPGA imported bus/signal names, trimming, 30
FPGA Probe, bus/signal name folder, 31
FPGA signal names, importing from LAI file, 29
FPGA, downloading configuration bits, 28

H

hostname of JTAG server, 24

I

imported FPGA bus/signal names, renaming, 30
imported FPGA bus/signal names, trimming, 30
importing FPGA signal names from LAI file, 29
installing the FPGA dynamic probe, 13
invalidation of captured data, 37

J

JTAG cable programmer software, Altera, 18
JTAG cable, establishing a connection, 26
JTAG cable, setting up, 19
JTAG scan chain, devices on, 26
JTAG scan chain, maximum number of devices, 65
JTAG server, 21
JTAG_Chain, XML element, 84

L

Label, XML element, 85
Label, XML element (under NonATCLabels), 86
Labels, XML element, 87
LAI core, compiling the project (example), 53
LAI core, configuring parameters (example), 47