Quartus™

Programmable Logic Development Software

SignalTap™ User's Guide

Altera Corporation
101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Quartus SignalTap User's Guide
Version 1999.10 Revision 2
November 1999 P25-04733-01

Printed on Recycled Paper.

Contents

Introduction

The SignalTap™ Embedded Logic Analyzer megafunction, which is provided with the Quartus™ software and is used with APEX™ 20K programmable logic devices (PLDs), allows you to analyze and verify System-on-a-Programmable-Chip™ designs. The MasterBlaster™ and ByteBlasterMV™ communications cables provide analysis support, allowing you to transfer signal data to the Quartus software. Using the intuitive Quartus interface, you can select signals, set up triggering events, configure memory, and display waveforms. You can also use data retrieved by the Embedded Logic Analyzer to debug and verify your design.

This manual explains how to use the Embedded Logic Analyzer megafunction, and gives detailed, step-by-step procedures on how to set up and run the Embedded Logic Analyzer.

Quartus SignalTap Embedded Logic Analyzer Overview

The Quartus SignalTap Embedded Logic Analyzer is a powerful tool that lets you capture signals from internal PLD nodes while the device is running in system, which gives you non-intrusive access to signals from internal device nodes. You can optimize the SignalTap Embedded Logic Analyzer megafunction for various tasks. The Embedded Logic Analyzer megafunction works within the Quartus software for design development, debugging, and verification.

You can assign internal nodes for capture using the Quartus Node Finder. Once the Embedded Logic Analyzer megafunction is configured, you can compile and download it with the rest of the device design via the MasterBlaster or ByteBlasterMV communications cable. You can make triggering changes to the Embedded Logic Analyzer without recompiling your device design. The Quartus software displays data acquired by the Embedded Logic Analyzer as waveforms.

Configuring the Embedded Logic Analyzer Megafunction

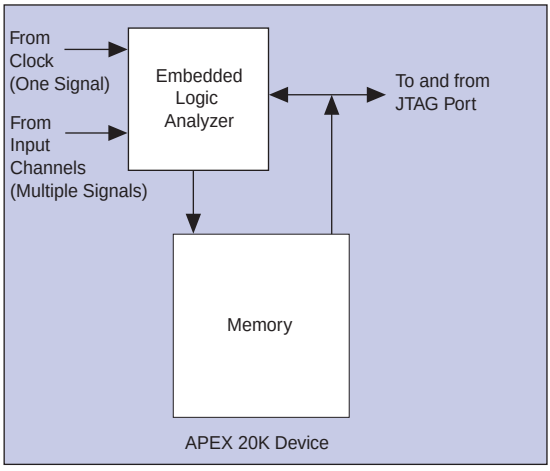
The Embedded Logic Analyzer megafunction is parameterized, which means you can customize it to fit a particular analysis task. Different methods for capturing signals allow you to configure device resources for maximum performance. The Embedded Logic Analyzer megafunction provides three methods for capturing signals, which you can use in any combination:

- Embedded Logic Analyzer
- Debugging port
- Trigger output

Embedded Logic Analyzer Megafunction

In the Embedded Logic Analyzer megafunction configuration, acquisition data is saved to internal device RAM, then streamed off-chip via the IEEE Std. 1149.1 Joint Test Action Group (JTAG) port. This configuration requires the greatest number of memory resources but the fewest number of I/O pins. Figure 1 shows the Embedded Logic Analyzer megafunction configuration.

Figure 1. Embedded Logic Analyzer Megafunction Configuration

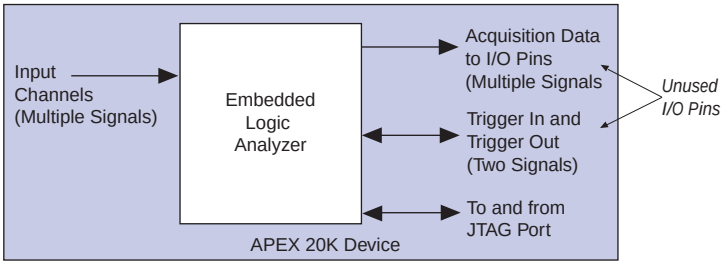


The Quartus software automatically assigns internal memory for acquisition storage when you specify that the Embedded Logic Analyzer acquisition memory should use one or more Embedded System Blocks (ESBs). See "Embedded System Block Usage" on page 5 and "Specifying the Sample Buffer Depth" on page 18 for more information.

Debug Port

When device RAM is limited, you can route internal signals to unused I/O pins for capture by an external analyzer. The debug port configuration conserves ESBs at the expense of I/O pins, and is useful for data-intensive applications in which the amount of saved data exceeds the available sample buffer depth. Figure 2 illustrates the I/O pin routing for the debug port mode.

Figure 2. Debug Port Configuration



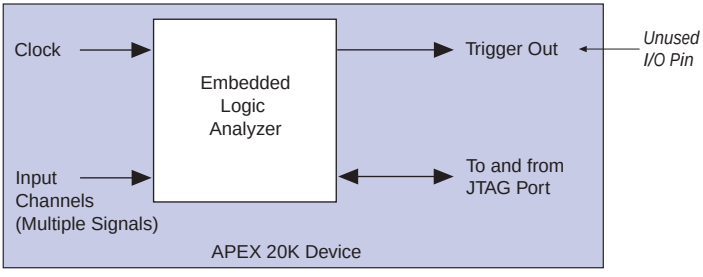
When you use the debug port configuration, the Quartus software automatically generates pins for signals that are selected for output via the debug port. To assign pins manually, use the Quartus Floorplan Editor.

For more information, see "Setting the Debug Port" on page 24.

Trigger Output Mode

The Embedded Logic Analyzer generates a trigger output signal when the trigger pattern has been recognized. You can set the pulse polarity to active high or active low.
Configuring the Embedded Logic Analyzer megafunction for trigger output mode requires no ESBs and only one I/O pin. See Figure 3.

Figure 3. Using the Embedded Logic Analyzer Megafunction as an Event Analyzer



For more information, see "Triggering the Embedded Logic Analyzer" on page 19.

Logic Element Usage

The number of input channels you assign to internal device nodes determines logic element (LE) usage. You can select any number of channels; however, the Embedded Logic Analyzer parameters are rounded up to the nearest power of two (1, 2, 4, 8, 16, 32, and so on). For example, if you specify 14 channels, the Embedded Logic Analyzer is parameterized for 16 channels.

Table 1 shows the LE usage for several channel configurations using an EP20K100 device.

Table 1. Logic Element Usage in an EP20K100 Device

Analyzer Channels	Logic Elements	Percent LE Usage
1	136	3.3
2	144	3.5
4	160	3.8
8	192	4.6
16	256	6.2
32	384	9.2
64	640	15.4

See "Assigning Signals to the Embedded Logic Analyzer File" on page 17 for instructions on assigning signals.

Embedded System Block Usage

In the Embedded Logic Analyzer configuration, acquisition data is saved to device ESBs, and then is transferred off-chip via the IEEE Std. 1149.1 JTAG port. The number of ESBs used for this function depends on the number of

input channels used and the depth of the sample buffer. Table 2 shows the number of ESBs used to store the values for different configurations for an APEX 20K device.

For instructions on setting the sample buffer depth, go to "Specifying the Sample Buffer Depth" on page 18.

Table 2. APEX 20K Embedded System Block Usage

Channels	Buffer Samples				
	128	256	512	1,024	2,048
1					1
2				1	2
4			1	2	4
8		1	2	4	8
16	1	2	4	8	16
32	2	4	8	16	32
64	4	8	16	32	64
128	8	16	32	64	128

Embedded Logic Analyzer Sample Rate

The Embedded Logic Analyzer samples synchronously to a specified global clock. The analyzer samples internal signals on the rising edge of the clock.

Go to "Selecting an Acquisition Clock Signal" on page 18 for information on selecting the Embedded Logic Analyzer clock.

Compilation Requirements

The Embedded Logic Analyzer megafunction is compiled with your APEX 20K project. Configuration changes, such as adding channels or changing the acquisition buffer depth, affect the Embedded Logic Analyzer

parameters and may require you to recompile your design. Table 3 lists possible changes to the Embedded Logic Analyzer and indicates whether the changes require recompilation.

Table 3. Compilation Requirements for Design Changes

Design Change	Recompilation Required
Changing the trigger pattern.	No
Running or stopping the logic analyzer.	No
Changing the number of input signals.	Yes
Changing the Embedded Logic Analyzer clock signal.	Yes
Changing the sample buffer depth.	Yes
Enabling the debug port.	Yes

MasterBlaster Communications Cable

The MasterBlaster communications cable is a hardware interface to a standard UNIX or PC serial port, or to a PC USB port (Windows 98 only), and is used for transferring configuration and programming data to Altera® devices. The MasterBlaster cable supports in-system debugging when you use it with the Embedded Logic Analyzer and APEX 20K devices. See Figure 4.

Figure 4. MasterBlaster Serial/USB Communications Cable

For more information on the MasterBlaster communications cable, see the *MasterBlaster Serial/USB Communications Cable Data Sheet*.

The MasterBlaster cable connects to the target circuit using a 10-pin female plug that connects to a 10-pin male header on the printed circuit board (PCB). The 10-pin male header has two rows of five pins, which are connected to the device's configuration pins. Table 4 shows the pin assignments for the MasterBlaster communications cable.

Table 4. Female Plug Pin Names & Download Modes (Part 1 of 2)

Pin	Passive Serial Mode		JTAG Mode	
	Signal	Description	Signal	Description
1	DCLK	Clock signal	TCK	Clock signal
2	GND	Signal ground	GND	Signal ground
3	CONF_DONE	Configuration control	TDO	Data from device
4	VCC	Power supply	VCC	Power supply
5	nCONFIG	Configuration control	TMS	JTAG state machine control

Table 4. Female Plug Pin Names & Download Modes (Part 2 of 2)

Pin	Passive Serial Mode		JTAG Mode	
	Signal	Description	Signal	Description
6	VIO	Reference voltage for output driver	VIO	Reference voltage for output driver
7	nSTATUS	Configuration status	–	No connect
8	–	No connect	–	No connect
9	DATA0	Data to device	TDI	Data to device
10	GND	Signal ground	GND	Signal ground

Figures 5 and 6 show the female and male connector dimensions.

Figure 5. 10-Pin Female Plug Header Dimensions (Inches)

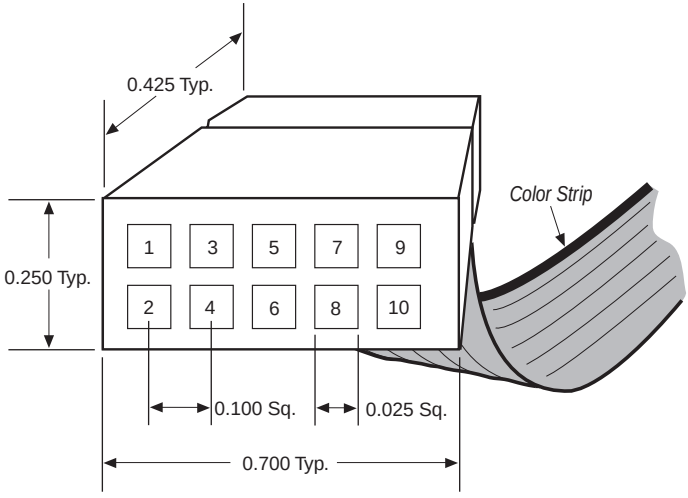
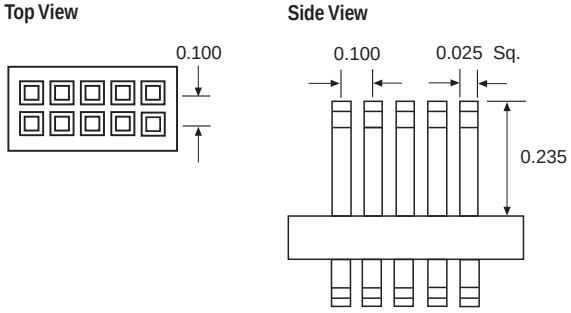


Figure 6. 10-Pin Male Header Dimensions (Inches)



ByteBlasterMV Parallel Port Download Cable

The ByteBlasterMV parallel port download cable is a hardware interface to a standard PC parallel port and is used for transferring configuration and programming data to Altera devices. The ByteBlasterMV cable also supports in-system debugging when you use it with the Embedded Logic Analyzer and APEX 20K devices.

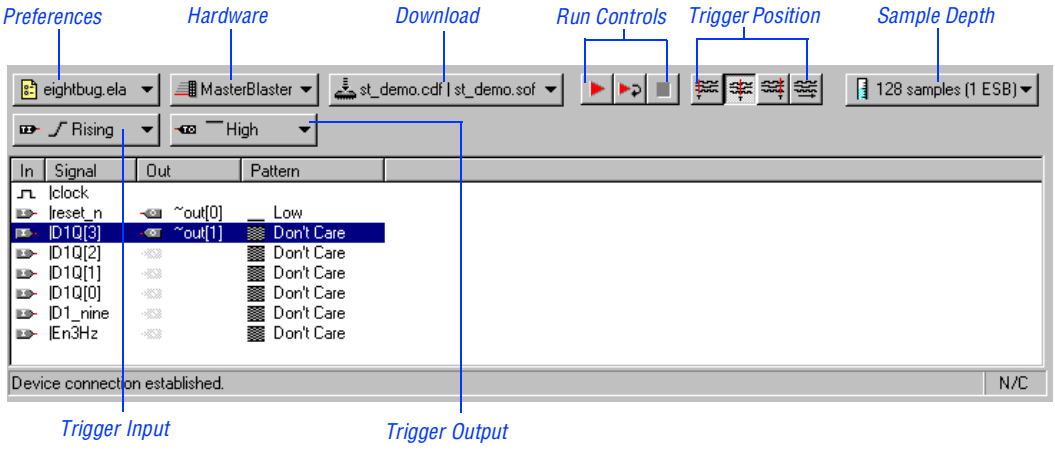
The ByteBlasterMV cable has a 25-pin male header that connects to the PC parallel port, and a 10-pin female plug that connects to the circuit board.

For more information on the ByteBlasterMV parallel port download cable, see the *ByteBlasterMV Parallel Port Download Cable Data Sheet*.

Using the Embedded Logic Analyzer

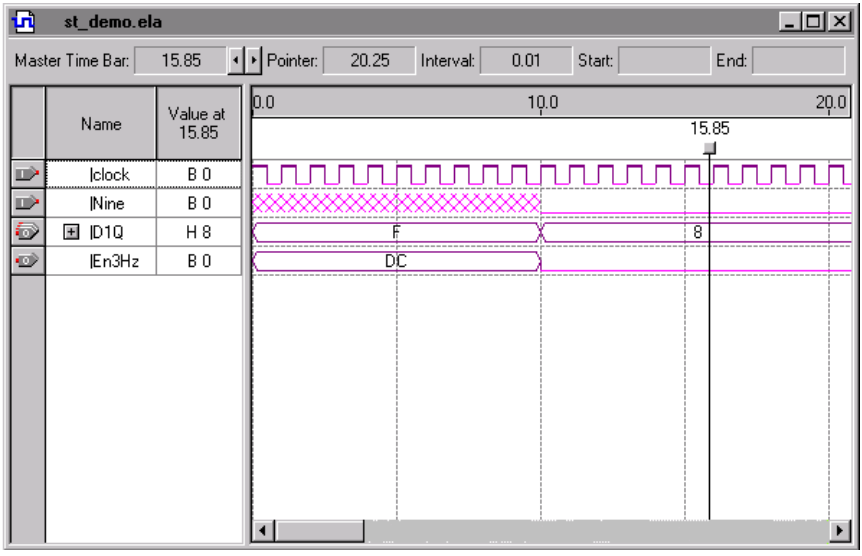
The Quartus SignalTap Embedded Logic Analyzer user interface and the Embedded Logic Analyzer controls are integrated into the Quartus software and provide access to trigger setup, sample depth selection, and run controls. Figure 7 shows the Embedded Logic Analyzer window in the Quartus software.

Figure 7. Embedded Logic Analyzer User Interface in the Quartus Software



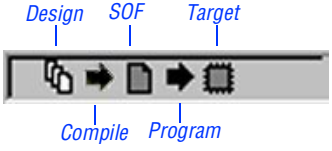
The Embedded Logic Analyzer captures and stores data in an Embedded Logic Analyzer File (**.ela**). This data is displayed in the Embedded Logic Analyzer Waveform window. See Figure 8.

Figure 8. Data Displayed as Waveforms by the Quartus Software



The Quartus status bar displays at the bottom of the Quartus window if you have turned on **Display status bar** in the **General** tab of the **Options** dialog box (Tools menu). The Embedded Logic Analyzer icons on the status bar indicate your stage in the Embedded Logic Analyzer debug process. When you point to an icon on the status bar, a brief description of that item's function appears next to it. See Figure 9.

Figure 9. Embedded Logic Analyzer Icons on the Quartus Status Bar



The following Embedded Logic Analyzer icons are available from the Quartus status bar:

Icon:	Name:	Description:
	**Design**	When you open a project and it contains a design file, the **Design** icon changes from dimmed to normal to indicate that there is an open design file.
	**Compile**	After you make design changes or Embedded Logic Analyzer configuration changes that require recompilation, the **Compile** arrow on the status bar is highlighted in yellow to indicate that you must compile.
	**SOF**	After you compile the design, the **SOF** icon changes from dimmed to normal to indicate that you have generated a programming file.
	**Program**	After you compile the design, the **Program** arrow is highlighted in yellow to indicate that you must download the design.
	**Target**	After you have downloaded the design, the **Target** icon changes from dimmed to normal to indicate that the Embedded Logic Analyzer is ready to run.

Embedded Logic Analyzer Setup

To set up the Embedded Logic Analyzer, follow these steps:

1. Create or open an Embedded Logic Analyzer File (.**ela**).
2. If necessary, save the ELA File.
3. If necessary, set the Embedded Logic Analyzer preferences.
4. Assign signals to the Embedded Logic Analyzer.
5. Select an Embedded Logic Analyzer clock signal.
6. Configure the sample buffer depth.
7. Set the trigger pattern.
8. Set the trigger position.
9. Configure the communications cable.
10. If necessary, compile the SignalTap Embedded Logic Analyzer megafunction into your file.
11. Download the Embedded Logic Analyzer.

12. Run the Embedded Logic Analyzer.

Creating, Opening & Closing an Embedded Logic Analyzer File

In order to set up and modify the Embedded Logic Analyzer, you must first create, name, and save an Embedded Logic Analyzer File (.**ela**) or open an existing ELA File.

Before opening or creating an ELA File, you must create or open a project. See "Creating a New Project" in Quartus Help for more information.

To create an ELA File, follow these steps:

1. Choose **New** (File menu).

   or

   Click **New** on the Standard Compile Mode toolbar.

2. In the **New** dialog box, click the **Other Files** tab.

3. Select **Embedded Logic Analyzer File**.

4. Click **OK**.

5. In the **File name** box, type a file name.

6. Click **Save**.

To open an existing ELA File, follow these steps:

1. Choose **Open** (File menu).

   or

   Click **Open** on the Standard Compile Mode toolbar.

2. In the **Open** dialog box, select **Waveform/Vector Files** from the **Files of Type** list.

3. In the **Files** box, select the desired file.

   or

   In the **File name** box, type the desired file name.

4. Click **Open**.

To close an open ELA File, follow these steps:

1. Select the desired ELA File.
2. Choose **Close** (File menu).

Saving, Copying & Renaming an Embedded Logic Analyzer File

Once you create and edit an ELA File, you can save any changes to the existing ELA File or you can copy and rename the file.

To save changes to an existing file:

✓ Choose **Save** (File menu).

or

✓ Click the **Save** button on the Standard Compile Mode toolbar.

To save an untitled ELA File or to copy and rename a previously saved file under a new name, follow these steps:

1. Choose **Save As** (File menu).
2. In the **Save As** dialog box, select the desired target directory from the **Save in** list, if necessary.
3. In the **Files** list, specify a file name, or in the **File name** box, type a file name.
4. In the **Save as type** list, select **Embedded Logic Analyzer File (.ela)**, if necessary.

5. Turn on **Add file to current project**.

6. Click **Save**.

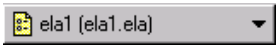
Setting the Embedded Logic Analyzer Preferences

You can set the preferences for the Embedded Logic Analyzer by using the **Preferences** list. Enabling these preferences allows the Embedded Logic Analyzer to prompt you to compile the Embedded Logic Analyzer, and/or download SRAM Object Files (**.sof**) or Chain Description Files (**.cdf**) when it is necessary. If you do not enable these preferences, the Embedded Logic Analyzer performs the next step without prompting you first. In the default settings, all of the options in the **Preferences** list are enabled.

To set the Embedded Logic Analyzer preferences, follow these steps:

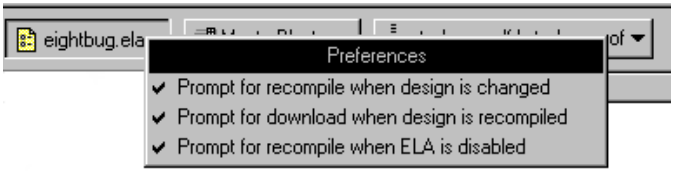
1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click the **Preferences** button on the Embedded Logic Analyzer toolbar.



3. In the **Preferences** list, select the preferences that you want to enable. See Figure 10.

Figure 10. Embedded Logic Analyzer Preferences



Assigning Signals to the Embedded Logic Analyzer File

The Embedded Logic Analyzer captures signals from any internal device node (including I/O pins). However, before capturing signals, you must first assign internal nodes to the input channels. The number of signals you assign determines the amount of device resources used.

Go to "Logic Element Usage" on page 5 and "Embedded System Block Usage" on page 5 for more information on resource usage.

To assign an internal node signal to the ELA File, follow these steps:

1. Open or create an ELA File.
2. Choose **Node or Bus** (Insert menu).
3. In the **Insert Node** dialog box, click **Node Finder**.
4. In the **Nodes Found** list, select one or more nodes or groups of nodes after performing a sorted search.
5. Click the ≥ button to copy the selected nodes or groups to the **Selected Nodes** list.
6. Click **OK**.
7. Click **OK** in the **Insert Node or Bus** dialog box.

Go to "Copying Node Names to an Embedded Logic Analyzer File with the Node Finder" in Quartus Help for more information on using the Node Finder.

Deleting Signals from the Embedded Logic Analyzer File

To delete signals from the ELA File, follow these steps:

1. Open an ELA File.

2. Select the node or group you wish to remove.

3. Choose **Delete** (Edit menu).

   or

   Choose **Delete** (right button pop-up menu).

Selecting an Acquisition Clock Signal

You must select and designate one signal as the Embedded Logic Analyzer acquisition clock. All input channels are sampled on the rising edge of the acquisition clock signal. Altera recommends using a global clock signal as the acquisition signal.

To select the acquisition clock signal, follow these steps:

1. Open an ELA File.

2. To add the signal you want to use as the acquisition clock signal to the ELA File, choose **Insert Node or Bus** (Insert menu).

3. In the Embedded Logic Analyzer window, select the appropriate acquisition clock signal.

4. Press the right mouse button on the signal name and choose **Clock** (right button pop-up menu).

Go to "Copying Node Names to an Embedded Logic Analyzer File with the Node Finder" in Quartus Help for more information on using the Node Finder.

Specifying the Sample Buffer Depth

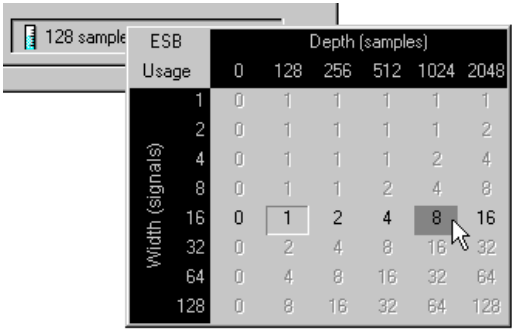
You must specify the number of ESBs that the Embedded Logic Analyzer uses for saving acquisition data. The number of ESBs used depends on the depth of the sample buffer and the number of internal nodes assigned to the Embedded Logic Analyzer. To configure the sample buffer depth, follow these steps:

1. Open an ELA File.

2. Click the **Sample Depth** button on the Embedded Logic Analyzer toolbar.

3. Select the desired sample depth. See Figure 11.

Figure 11. Configuring the Sample Buffer Depth

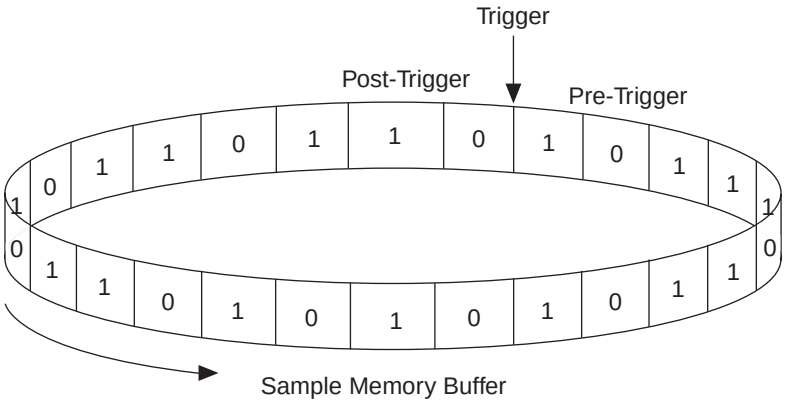


Using ESBs for acquisition memory decreases the number of ESBs available for other applications.

Triggering the Embedded Logic Analyzer

When you run the Embedded Logic Analyzer, the SignalTap Embedded Logic Analyzer megafunction continually samples input signals and places data in a circular buffer with new samples replacing old samples. See Figure 12. When the trigger pattern is recognized, the Embedded Logic Analyzer stops immediately, or continues sampling to refill the entire buffer, or continues sampling indefinitely, depending on the trigger position setting.

Figure 12. Trigger Sampling



Setting Trigger Positions

The Trigger Position controls offer four choices:

Icon:	Name:	Description:
	**Pre-trigger**	Captures signals immediately before triggering (92% pre-trigger, 8% post-trigger).
	**Center**	Captures signals before and after triggering (half pre-trigger and half post-trigger).
	**Post-trigger**	Captures signals that occur immediately after triggering (12% pre-trigger, 88% post-trigger).
	**Continuous trigger**	Captures signals indefinitely (which is useful when using Trigger Out).

To set the trigger position, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).
2. Click the desired **Trigger position** button on the Embedded Logic Analyzer toolbar.

Trigger Pattern

The Embedded Logic Analyzer uses a single pattern recognizer for triggering, which is referred to as the "trigger pattern." You can set a logic condition (High, Low, Rising Edge, Falling Edge, or Don't Care) for each input signal to specify the trigger pattern. The Embedded Logic Analyzer is triggered when the input signals match the trigger pattern. The trigger pattern appears as a column in the Embedded Logic Analyzer window.

Defining the Trigger Pattern

To define the triggering pattern, you must assign a logic condition for each input channel. By default, all bits of a trigger pattern are set to the "Don't Care" setting, masking them from trigger recognition. You can also set the bits to High, Low, Rising Edge, Falling Edge, or Either Edge. Table 5 describes these conditions.

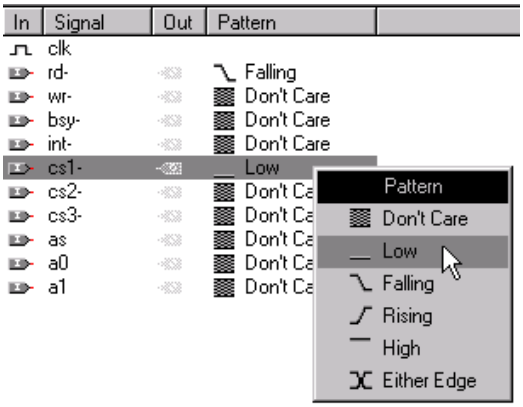
Table 5. Triggering Pattern Logic Conditions

Condition	Description
Don't Care	Signal condition doesn't matter.
High	Signal must be high for the trigger to occur.
Low	Signal must be low for the trigger to occur.
Falling Edge	Signal must be falling for the trigger to occur.
Rising Edge	Signal must be rising for the trigger to occur.
Either Edge	Signal must be either rising or falling for the trigger to occur.

To define the trigger pattern, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).
2. In the **Pattern** column, select the desired signal.
3. Choose a trigger condition from the right button pop-up menu. See Figure 13.

Figure 13. Defining the Trigger Pattern



Setting the Trigger Input & Output Signals

You can use trigger input and trigger output signals to synchronize the Embedded Logic Analyzer with external equipment, such as an oscilloscope or system-level logic analyzer, allowing you to synchronize the capture of internal and external events.

Setting the Trigger Input

You can use an I/O pin to trigger the Embedded Logic Analyzer. You can set Trigger In to recognize a High, Low, Rising Edge, Falling Edge, Either Edge, or Don't Care condition on that I/O pin.

When you assign a signal condition to the trigger input signal, the global pin `~trig_in` is generated in the design. You must assign this signal to a device pin and compile. You can change the signal condition (High, Low, and so on) after you assign the node without requiring recompilation.

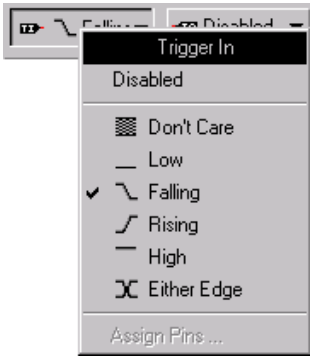
To enable the trigger input signal, perform the following steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click the **Trigger Input** button on the Embedded Logic Analyzer toolbar.

3. In the **Trigger In** list, select the desired **Trigger In** condition (such as Low, Falling, Rising, and so on). See Figure 14.

Figure 14. Enabling Trigger Input



Setting the Trigger Output

You can use a spare I/O pin as a trigger output signal to indicate that a trigger pattern has occurred. You can also specify the polarity of the output pulse as high or low.

When you enable the trigger output signal, the global node `~trig_out` is generated in the design. You must assign this signal to a device pin and compile. You can then change the output pulse settings without recompiling.

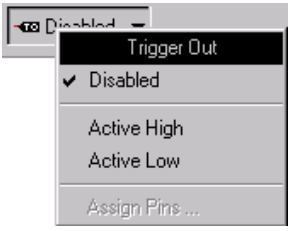
To enable the trigger output signal, perform the following steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click the **Trigger Output** button on the Embedded Logic Analyzer toolbar.

3. In the **Trigger Out** list, select the appropriate polarity (**Active High** or **Active Low**). See Figure 15.

Figure 15. Selecting Trigger Output



Setting the Debug Port

Using the debug port setting, you can route any Embedded Logic Analyzer input signal to a spare I/O pin for capture by external equipment. When you assign an Embedded Logic Analyzer input signal to the debug port, the Embedded Logic Analyzer automatically generates a pin in the device design. The debug port pin is named "`~out[`*n*`]`," where *n* is a number representing the order in which the debug port pin occurs in the signal list. By default, no input signals are routed to the debug port.

You must assign debug port nodes to device pins, and compile the device after adding or deleting debug port signals.

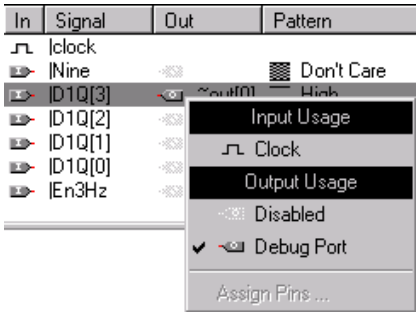
To add or remove a signal from the debug port, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Select a signal name.

3. Choose **Debug Port** (right button pop-up menu). See Figure 16.

Figure 16. Setting the Debug Port



Compiling the Embedded Logic Analyzer Megafunction

You must compile the project or design entity before running the Embedded Logic Analyzer, if you have not already done so, or if you made changes that require recompilation. Go to "Compilation Requirements" on page 6 for information on whether your changes require recompilation.

To compile the Embedded Logic Analyzer megafunction into your design, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click **Run** on the Embedded Logic Analyzer toolbar.

3. Click **Yes** in the dialog box that prompts, **The design has changed. Do you wish to recompile before running the logic analyzer?**

Downloading a Design

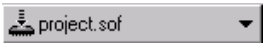
You can download a design to an APEX device in a JTAG chain consisting of a single device or multiple Altera devices.

To download a design, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Select a communications cable to be used for download. See "Selecting a Communications Cable" on page 31.

3. Click the **Download** button on the Embedded Logic Analyzer toolbar.



4. If the JTAG chain contains a single Altera device, in the **Download** list, select **Program with current CDF/SOF.**

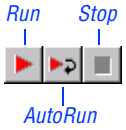
   or

   If the JTAG chain contains multiple Altera devices, in the **Download** list, select the appropriate Chain Description File (**.cdf**), and then select the desired SRAM Object File (**.sof**).

Running the Embedded Logic Analyzer

The Embedded Logic Analyzer run controls consist of **Run**, **Auto Run**, and **Stop**. These controls allow you to start and stop the analysis. Figure 17 shows the run controls.

Figure 17. SignalTap Embedded Logic Analyzer Run Controls



During compilation, the Embedded Logic Analyzer run controls are dimmed to indicate that they are not available.

Run

When you press the **Run** button, the Embedded Logic Analyzer samples the specified input signals on the rising edge of the selected clock. See "Selecting an Acquisition Clock Signal" on page 18 for more details.

If you set the trigger position to "Continuous," the Embedded Logic Analyzer samples until you press the **Stop** button. See "Setting Trigger Positions" on page 20 for more details.

If you use ESBs to save acquisition data, the Embedded Logic Analyzer captures and stores samples of the input signals until triggering occurs and the trigger position is satisfied. The Embedded Logic Analyzer then transfers acquisition data from device ESBs to the host computer via the communications cable. The acquisition data is stored in the ELA File and is displayed by the Waveform window. The Embedded Logic Analyzer then returns to an idle state.

If you choose not to use device ESBs to save acquisition data, the Embedded Logic Analyzer stops sampling and returns to idle when triggered.

Running the Embedded Logic Analyzer does not affect the rest of the device logic.

To run the Embedded Logic Analyzer, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click **Run** on the Embedded Logic Analyzer toolbar.

Auto Run

You can use the **Auto Run** control for continuous display updates from the Embedded Logic Analyzer. **Auto Run** works like **Run** except that, instead of remaining idle after acquisition, **Auto Run** automatically restarts.

To disable **Auto Run**, you must stop the Embedded Logic Analyzer manually. See "Stop," next.

To enable **Auto Run**, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click **Auto Run** on the Embedded Logic Analyzer toolbar.

The Embedded Logic Analyzer is halted while uploading acquisition data to the Quartus software. Signal activity that occurs during this time is not captured.

Stop

You can stop the Embedded Logic Analyzer manually by using the **Stop** button. Any data saved to internal memory is transferred immediately to the host computer and is displayed in the ELA File. The Embedded Logic Analyzer returns to an idle state.

To stop the Embedded Logic Analyzer, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click **Stop** on the Embedded Logic Analyzer toolbar.

Viewing Waveforms

Acquisition data is displayed in the Embedded Logic Analyzer Waveform window. See the "Waveform Editor" section in Quartus Help for more information.

Enabling and Disabling the Embedded Logic Analyzer Megafunction

When enabled, the Embedded Logic Analyzer megafunction is compiled into the design and provides access to signals from internal nodes. When disabled, it is removed from the design during the next compilation.

By default, the Embedded Logic Analyzer megafunction is enabled in the design. You can disable the Embedded Logic Analyzer megafunction at any time by closing the Embedded Logic Analyzer window.

To disable the Embedded Logic Analyzer megafunction, follow these steps:

1. Close the Embedded Logic Analyzer window.

2. When you are asked, **The Embedded Logic Analyzer has been disabled. Do you wish to recompile to remove it from the design?**, click **Yes.**

Communications Hardware

The Quartus software communicates with APEX devices via the MasterBlaster and ByteBlasterMV communications cables. You can download designs, control the Embedded Logic Analyzer, and retrieve acquisition data. The **Hardware** button on the Embedded Logic Analyzer toolbar allows you to specify settings required for communication.

Running the Embedded Logic Analyzer in Demo Mode

Running the Embedded Logic Analyzer in Demo mode causes random data to appear in your ELA File. This random data allows you to view or demonstrate how the Embedded Logic Analyzer works without using a target board or communications cable.

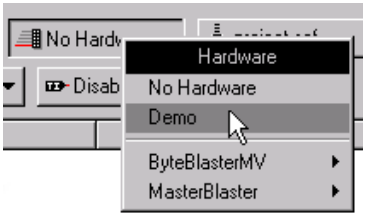
To run the Embedded Logic Analyzer in Demo mode, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Click the **Hardware** button on the Embedded Logic Analyzer toolbar.

3. In the **Hardware** list, select **Demo**. See Figure 18.

Figure 18. Running Demo Mode



4. Click the **Run** button on the Embedded Logic Analyzer toolbar.

Selecting a Communications Cable

You can download data using the MasterBlaster or ByteBlasterMV communications cable. To select a communications cable, follow these steps:

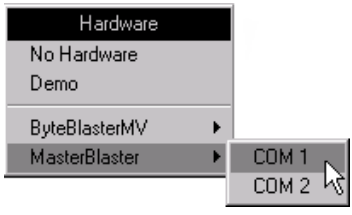
1. Select the communications cable in the Programmer window. Refer to "Changing the Hardware Setup" in Quartus Help for information on selecting a communications cable in the Programmer.

2. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

3. Click the **Hardware** button on the Embedded Logic Analyzer toolbar.

4. In the **Hardware** list, select the desired communications cable and then select the appropriate communications port from the submenu. If you are using Windows 98, **USB** is also listed as an available communications port. See Figure 19.

Figure 19. Selecting a Communications Cable & Communications Port



Selecting the MasterBlaster Baud Rate

By default, the MasterBlaster communications cable connects at 115k bps. You can set the baud rate manually if you need to lower the speed.

To set the baud rate manually, follow these steps:

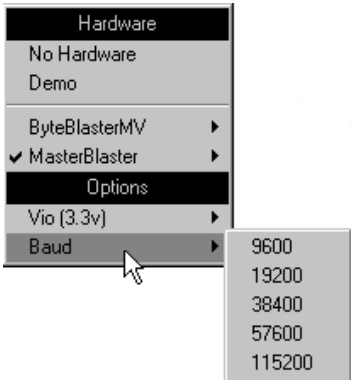
1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).

2. Select the MasterBlaster cable as the communications cable. See "Selecting a Communications Cable" on page 31.

3. Click the **Hardware** button on the Embedded Logic Analyzer toolbar.

4. In the **Hardware** list, select **Baud**, and then select the desired baud rate. See Figure 20.

Figure 20. Selecting a Baud Rate



Selecting the MasterBlaster I/O Voltage

The MasterBlaster communications cable supports a wide range of device I/O voltages. The I/O voltage can be set automatically to correspond to the reference voltage on the VIO pin of the JTAG cable (see Table 4 on page 8) or you can set it manually to a specific voltage.

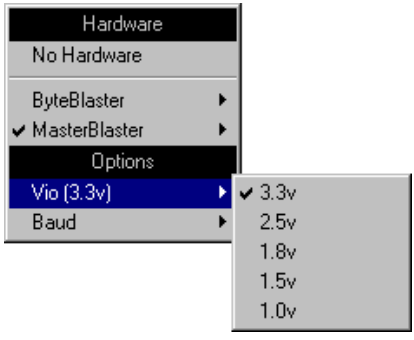
The voltage on the VIO pin overrides any manual setting. If you want to set the I/O voltage manually, leave the VIO pin unconnected.

To set the device I/O voltage, follow these steps:

1. To open the Embedded Logic Analyzer window, choose **Auxiliary Windows > Embedded Logic Analyzer** (View menu).
2. Select the MasterBlaster cable as the communications cable. See "Selecting a Communications Cable" on page 31 for more information.
3. Click the **Hardware** button on the Embedded Logic Analyzer toolbar.

4. In the **Hardware** list, select **Vio (External)**, and then select the appropriate I/O voltage or select **Vio**. See Figure 21.

Figure 21. Setting the I/O Voltage Level Manually



Contacting Altera

If you have any additional questions about Altera products, contact Altera for technical support and product information.

Technical Support

If you need technical support, call or fax the Altera Applications Department Monday through Friday.

Tel:	(800) 800-EPLD	(6:00 a.m. to 6:00 p.m. Pacific Time)
	(408) 544-7000	(7:30 a.m. to 5:30 p.m. Pacific Time)
Fax:	(408) 544-6401	

You can also e-mail the Altera Applications Department at **support@altera.com.**

In addition, you can visit the **Atlas** online solutions database, which is available from the Altera web site at **http://www.altera.com**.

For additional technical support for the Quartus software, refer to the Quartus support web site by choosing **Altera on the Web > Quartus Home Page** (Help menu), or by pointing your browser to **https://websupport.altera.com**. This web site provides information on how to register your software and obtain license information, and it provides other support information.

Product Information

If you need the latest Altera product information or literature, use the Altera web site, which is available 24 hours a day, seven days a week.

Go to "Contacting Altera" in Quartus Help for complete information on Altera technical support services.

Revision History

The information contained in the Quartus SignalTap User's Guide version 1999.10 revision 2 supersedes information published in previous versions.

On pages 23 and 24, all references to assigning pins in the Trigger Input, Trigger Output, and Debug Port lists have been deleted.

Index

T

W