LAX_x Logic Analyzer

Summary

This core reference describes how to place and use a Logic Analyzer instrument in an FPGA design.

Core Reference
CR0103 (v2.0) March 17, 2008

The LAX_x family of devices provide a range of 8- or 16-channel logic analyzer instruments for use in an FPGA design. The device family includes analyzers with pre-defined storage memories for captured data, as well as analyzers that provide a memory interface – allowing you the freedom to connect a block of RAM, the size of which is determined by an address bus of up to 20 bits.

Features

- 8- or 16-channel devices
- 6 instruments with predefined storage memories
  - 3, 8-channel Logic Analyzers with 1K, 2K or 4K storage memories (8-bit data)
  - 3, 16-channel Logic Analyzers with 1K, 2K or 4K storage memories (16-bit data)
- 2 instruments with synchronous/asynchronous access to off-core RAM
  - LAX_8 : 8-channel Logic Analyzer, 20-bit address bus, 8-bit data bus
  - LAX_16 : 16-channel Logic Analyzer, 20-bit address bus, 16-bit data bus
- Ability to use 16-channel devices in split mode
  - Split 8-channel analyzing with bit-level or magnitude triggering patterns
- External (Hardware) or internal (Software) triggering
- Ability to capture data at rate of system clock
- Analog and digital waveform generation
  - Support for continuous data capture from within waveform views
  - Cursor synchronization between waveform views
  - Pan and zoom synchronization between waveform views.

Available Devices

The following Logic Analyzer instruments are available:

LAX_8	-	8-channel Logic Analyzer with memory interface
LAX_16	-	16-channel Logic Analyzer with memory interface
LAX_1K8	-	8-channel Logic Analyzer with 1Kx8 predefined storage memory
LAX_2K8	-	8-channel Logic Analyzer with 2Kx8 predefined storage memory
LAX_4K8	-	8-channel Logic Analyzer with 4Kx8 predefined storage memory
LAX_1K16	-	16-channel Logic Analyzer with 1Kx16 predefined storage memory
LAX_2K16	-	16-channel Logic Analyzer with 2Kx16 predefined storage memory
LAX_4K16	-	16-channel Logic Analyzer with 4Kx16 predefined storage memory.

All of the devices in the LAX_x family can be found in the FPGA Instruments integrated library (`FPGA Instruments.IntLib`), located in the `\Library\Fpga` folder of the installation.

Functional Description

Symbol

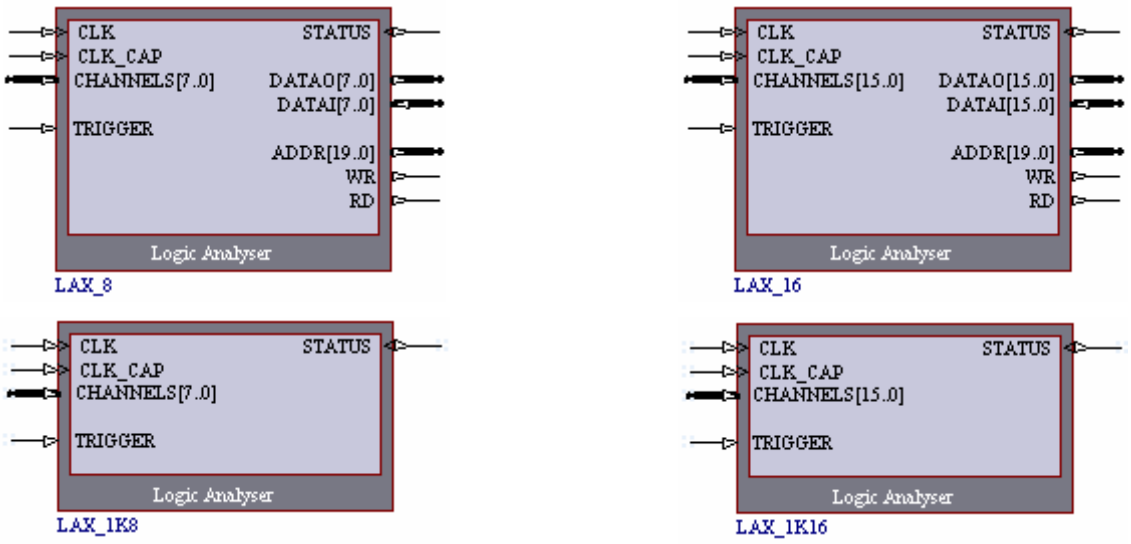


Figure 1. LAX_x Symbols

Note: The symbols for the LAX_2K8 and LAX_4K8 are identical to that of the LAX_1K8. The symbols for the LAX_2K16 and LAX_4K16 are identical to that of the LAX_1K16.

Pin Description

Table 1. LAX_x Pin description

Name	Type	Polarity/ Bus size	Description
Logic Analyzer Input Signals			
CLK	I	Rise	External system clock
CLK_CAP	I	Rise	Capture Clock input. The capture clock must be slower than the external system clock. If the **Capture Every Clock Edge** option is enabled in the *Logic Analyzer – Options* dialog (see *Capturing data*), this input is ignored and data will be captured at the same rate as the external system clock
CHANNELS	I	8/16	Input channels
Logic Analyzer Output Signals			
STATUS	O	Level	This signal conveys the current state of the Logic Analyzer – whether it is capturing data or not. 0 = Not capturing data 1 = Currently capturing data
External Trigger Signals			
TRIGGER	I	High	External trigger signal. Once the Logic Analyzer has been armed, a High level at this input will trigger the instrument and initiate capture of data. This signal allows for a hardware trigger to be defined, in addition to the software trigger that can be defined from the Logic Analyzer's Instrument panel.
Memory Interface Signals (LAX_8 and LAX_16 only)			
DATAO	O	8/16	Data bus output
DATAI	I	8/16	Data bus input

Name	Type	Polarity/ Bus size	Description
ADDR	O	20	Address bus output
WR	O	High	Memory write enable
RD	O	High	Memory read enable

Placing a LAX_x Device in a Design

The LAX_x family of devices provide a simple method of analyzing the logical levels of signals in a design. Figure 2 shows an example of how a Logic Analyzer device is wired into a design.

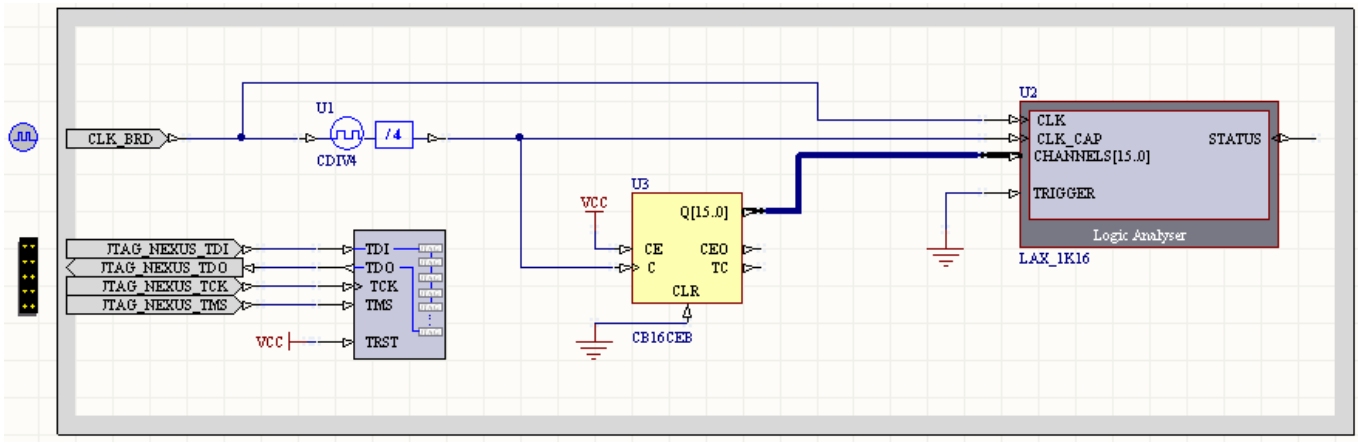


Figure 2. Using a LAX_x device to analyze signal levels in a design.

In the example circuit above, the Logic Analyzer device used is the LAX_1K16. This device has 1Kx16 storage memory built-in. All 16 input channels are used to monitor the corresponding outputs of a 16-bit cascadable binary counter.

The capture clock signal, CLK_CAP, is derived from the external system clock. Notice that a divide-by-4 clock divider has been used. This makes the CLK_CAP signal four times slower than the system clock. CLK_CAP must be slower than CLK, otherwise the Logic Analyzer will not capture data at all[1].

Note: In order to communicate with soft devices in a design (processors and/or virtual instruments) you must enable the soft devices JTAG chain within the design. This is done by placing a JTAG Port (`NEXUS_JTAG_CONNECTOR`) and corresponding Soft Nexus-Chain Connector (`NEXUS_JTAG_PORT`) on the top schematic sheet of the design, as shown in Figure 3.

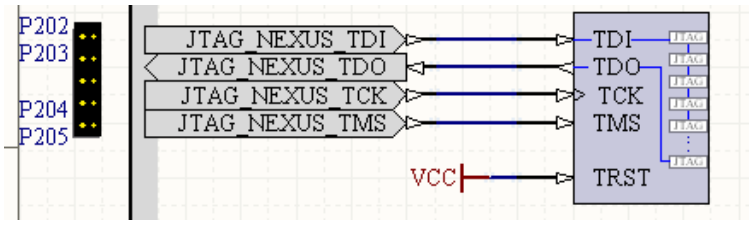


Figure 3. Implementing the soft devices chain within the design.

These devices can be found in the FPGA NB2DSK01 Port-Plugin (`FPGA NB2DSK01 Port-Plugin.IntLib`) and FPGA Generic (`FPGA Generic.IntLib`) integrated libraries respectively, both of which are located in the `\Library\Fpga` folder of the installation.

[1] To capture data at the same rate as CLK, enable the **Capture Every Clock Edge** option in the *Logic Analyzer – Options* dialog (see the section *Capturing Data*).

Connecting to External Memory

The LAX_8 and LAX_16 Logic Analyzer devices allow you to connect your own external memory. Both synchronous and asynchronous memories are supported.

The external RAM data width is equal to the number of input channels supported by the device. The size of the capture buffer organized in the external RAM block used is defined as:

Sample buffer size = $2^{Memsize}$

where Memsize equals the size, in bits, of the physically connected memory address bus.

The maximum supported memory address bus size is 20 bits, allowing the maximum sample buffer size of 1024K.

Whatever size block of RAM you use, the value for the memory address bus width must be entered into the **Memory Bus Width** field, in the *Logic Analyzer – Options* dialog (see *Capturing data*).

Accessing the Logic Analyzer Controls

The host computer is connected to the target Logic Analyzer instrument using the IEEE 1149.1 (JTAG) standard interface. This is the physical interface, providing connection to physical pins of the FPGA device in which the instrument has been embedded.

The Nexus 5001 standard is used as the protocol for communications between the host and all devices that are debug-enabled with respect to this protocol. This includes the Logic Analyzers, as well as other Nexus-compliant devices such as OCD-version microcontrollers, frequency generators, counters and digital I/O modules.

All such devices are connected in a chain – the Soft Devices chain – which is determined when the design has been implemented within the target FPGA device and presents in the **Devices** view (Figure 4). It is not a physical chain, in the sense that you can see no external wiring – the connections required between the Nexus-enabled devices are made internal to the FPGA itself.

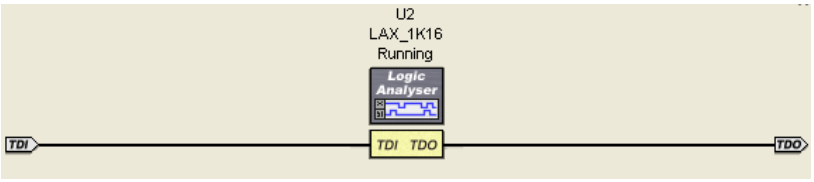


Figure 4. Nexus-enabled devices appearing in the Soft Devices chain.

For instruments such as the LAX_x, the Nexus protocol enables you to access the registers used for controlling the device. These registers are not exposed as such, rather input to them is provided through an instrument panel, which allows you to:

- Arm the Logic Analyzer ready for capture
- Reset the Logic Analyzer
- Access data capture options for the Logic Analyzer
- Access the LAX panel, from where you can define software triggering and view the content of the sample buffer
- Launch digital and/or analog waveform views for the captured data.

The controls for a Logic Analyzer used in a design can be accessed from the **Devices** view. Simply double click on the icon representing the Logic Analyzer whose controls you wish to access, in the **Soft Devices** region of the view. The Instrument Rack – Soft Devices panel will appear, with the chosen instrument added to the rack (Figure 5).

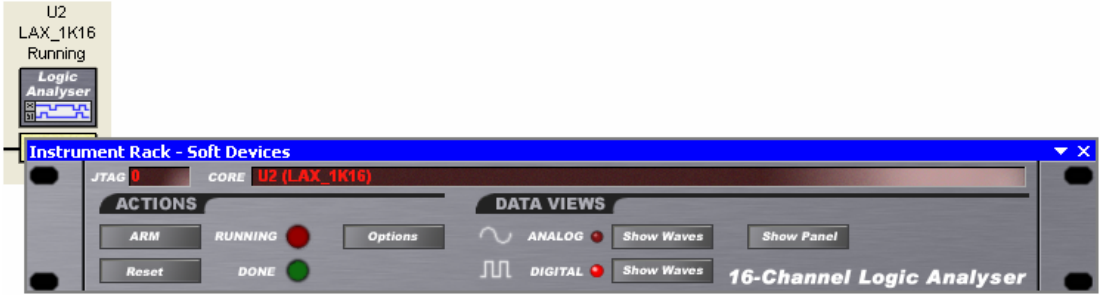


Figure 5. Accessing the Logic Analyzer Instrument panel.

Note: Each Logic Analyzer device that you have included in the design will appear, when double-clicked, as an Instrument in the rack (along with any other Nexus-enabled devices).

Operating the Logic Analyzer

The instrument panel for a Logic Analyzer device (Figure 6) contains various controls and options that allow you to effectively use the instrument in your design.

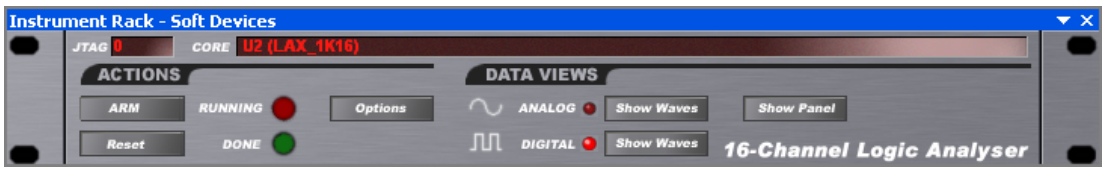


Figure 6. The Logic Analyzer Instrument Panel (16-channel instrument).

The Logic Analyzer can be triggered externally or internally, enabling both hardware and software triggering of the instrument. Selection between the two is determined by the status of the **Enable External Trigger** option, in the *Logic Analyzer – Options* dialog (see the section *Capturing data*).

Running the Logic Analyzer with Internal Triggering

Internal (or Software) triggering and data capture is carried out through the Analyzer's Instrument panel. To perform internal triggering of the instrument, ensure that the **Enable External Trigger** option, in the *Logic Analyzer – Options* dialog, is disabled.

Defining a Trigger Pattern

The software trigger pattern for the Logic Analyzer is defined in the **LAX** panel. This panel is accessed from the device's Instrument panel, by clicking on the **Show Panel** button, in the **Data Views** region.

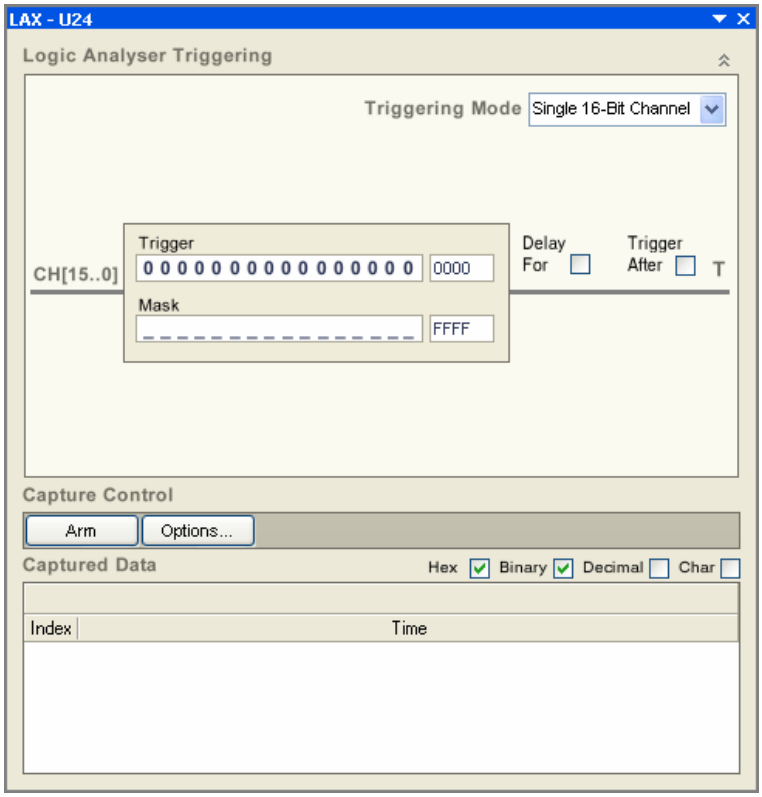


Figure 7. The LAX panel (showing trigger options for a 16-channel LAX device).

Note: If there is more than one Logic Analyzer device in a design, each device will have its own **LAX** panel[2] associated with it (when launched via the **Show Panel** button on its corresponding Instrument panel).

For a software trigger, the levels of the signals appearing at the instruments' input channels are compared against a defined trigger and, when a match occurs, capture of data is initiated.

[2] The LAX panels are standard panels and, as such, can be readily accessed from the **View » Workspace Panels » Nexus** sub menu, or by clicking on the **Nexus** button at the bottom of the application window and choosing the required panel – multiple panels are distinguished by the designator assigned to the parent LAX device on the schematic sheet.

The actual pattern for the trigger is entered in the **Trigger** field. Depending on the particular LAX_x device being used – 8- or 16-channel – this field will contain 8 or 16 bits. Clicking on a bit entry will toggle its value between High (1) and Low (0). Define each bit, corresponding to each of the input signals monitored by the device, as required. The hexadecimal value for the pattern is displayed in the field to the right.

Figure 8. Defining the trigger pattern.

The **Mask** field contains a corresponding number of bits and is used to determine which of the Analyzer's input channels will be compared to the trigger pattern defined in the **Trigger** field. Each bit entry can either be enabled for specific comparison with the corresponding bit in the **Trigger** field (_) or can be left as a Don't Care (X). Again, the hexadecimal value for the pattern is displayed in the field to the right.

Figure 9. Defining the input channel mask.

When the Mask is defined, all input channels enabled for comparison by the mask must equal their corresponding entries in the **Trigger** field for data capture to start.

If all entries in the **Mask** field are left as Don't Cares (X), then data capture will start immediately.

Advanced Trigger Options

You may not want to capture data immediately upon a match arising between the masked input channels and the trigger pattern. By enabling the **Trigger After** option in the **LAX** panel, you are able to enter a value for the number of matches to occur before data capture begins.

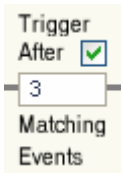


Figure 10. Delaying data capture.

If the value entered is 0 (default) or 1, capture will commence after the trigger pattern has been matched once. For values entered that are greater than 1, data capture will begin when the specified number of matches have been reached.

The **Delay For** option in the panel is used to obtain pre-samples – data samples captured before the sample that triggered acquisition of the data. The value entered in this field determines the index entry in the **Captured Data** region of the **LAX** panel at which the value that triggered the Analyzer will be stored.

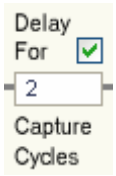


Figure 11. Requesting pre-sampling.

If this field is left blank, zero (default) is assumed and the first entry in the displayed data will be the sample that triggered the Analyzer. If a value greater than zero is entered, a corresponding number of pre-samples will be captured and displayed, before the one that triggered the Analyzer.

If the maximum value for the delay (required number of pre-samples) is reached or exceeded, only *Sample Buffer Size-1* pre-samples will be stored and the last sample in the data buffer will be the one that triggered the Analyzer.

For the Logic Analyzer devices with 1K, 2K and 4K built-in memory, the maximum values that can be entered in the **Delay For** option field are:

- 1K: 1023
- 2K: 2047
- 4K: 4095

For the LAX_8 and LAX_16 devices, where you can connect to an external memory space, the Sample Buffer Size is determined by:

$2^{Memsize}$

where Memsize is the size, in bits, of the physically connected memory address bus.

The maximum value that can be entered in the **Delay For** option field is therefore

$2^{Memsize} - 1$

Split-Trigger Mode

When using a 16-channel LAX device, you have the choice of using the Analyzer in one of two trigger modes:

- Single 16-Bit Channel (default, normal operational mode)
- Split 8-Bit Channels

The Split-trigger mode essentially turns a single 16-channel Logic Analyzer into two, 8-channel Analyzers, with advanced pre-trigger gating functionality that can be incorporated prior to actual data capture. Figure 12 shows the **Logic Analyzer Triggering** region of the **LAX** panel when the Triggering Mode is set to **Split 8-Bit Channels**.

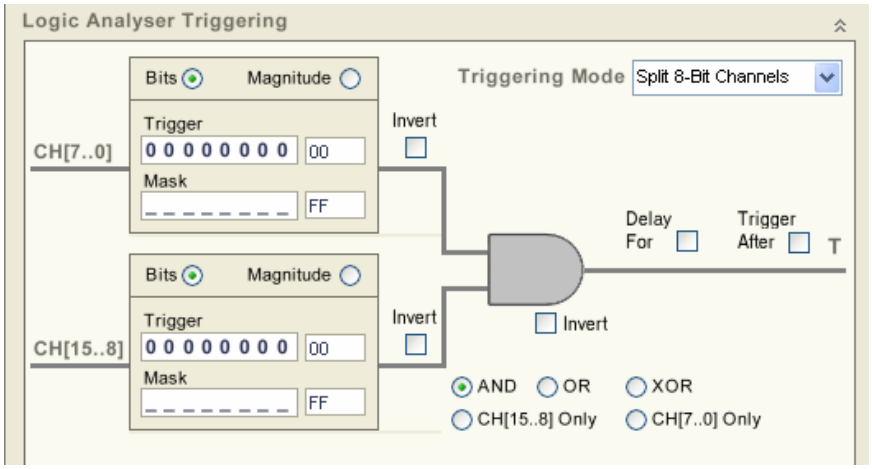


Figure 12. A 16-channel LAX device in Split-trigger mode.

Magnitude-Based Triggering

In Split-trigger mode, the software trigger for each 8-channel 'sub-Analyzer' can be defined in terms of either a distinct bit pattern, with input channel masking, or a magnitude-based trigger. By default, the trigger pattern is set to Bit mode. Change to Magnitude mode by enabling the corresponding option in the panel. The trigger definition region will change, as shown in Figure 13.

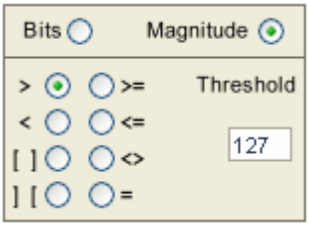


Figure 13. Using Magnitude-based triggering.

When using this mode for triggering, the decimal value obtained from the 8-bit input is compared against a specified Threshold value. Triggering depends on the particular comparison test enabled and whether the result of that comparison is True. Table 2 summarizes the various comparison tests that can be used in this mode.

>	Greater Than	-	Compares the decimal input value to the specified Threshold value and triggers if it is greater than the threshold.
<	Less Than	-	Compares the decimal input value to the specified Threshold value and triggers if it is less than the threshold
[ ]	Inside Specified Range	-	Compares the decimal input value against a range specified by Upper and Lower Threshold values and triggers if it is inside the range
] [	Outside Specified Range	-	Compares the decimal input value against a range specified by Upper and

			Lower Threshold values and triggers if it is outside the range
>=	Greater Than Or Equal To	-	Compares the decimal input value to the specified Threshold value and triggers if it is greater than or equal to the threshold
<=	Less Than Or Equal To	-	Compares the decimal input value to the specified Threshold value and triggers if it is less than or equal to the threshold
<>	Not Equal	-	Compares the decimal input value to the specified Threshold value and triggers if it is not equal to the threshold
=	Equal	-	Compares the decimal input value to the specified Threshold value and triggers if it is equal to the threshold

The value entered for the Threshold can be anywhere in the range 0-255, corresponding to the decimal value that can be obtained from the 8-bit input to the Analyzer. This range for the Threshold value is cyclic. If you enter a value for the Threshold that is greater than 255, you will effectively be mapped to the corresponding value in the 0-255 range. For example, if you enter 256, the Threshold will in fact be 0. Likewise, entering 300 will equate to 44.

Logical Gating

Prior to the pre-sampling (**Delay For**) and capture delay (**Trigger After**) options, the Split-trigger mode also offers gating functionality, to enable a more sophisticated level of triggering to be defined. Of course, you may wish to only trigger using 8 channels of the device, without the need for a logical combination of the two 8-channel sub-Analyzers. For this purpose, options are provided in the **LAX** panel to trigger using either the upper or lower byte of the 16 input channels – **CH[15..8] Only** and **CH[7..0] Only** respectively.

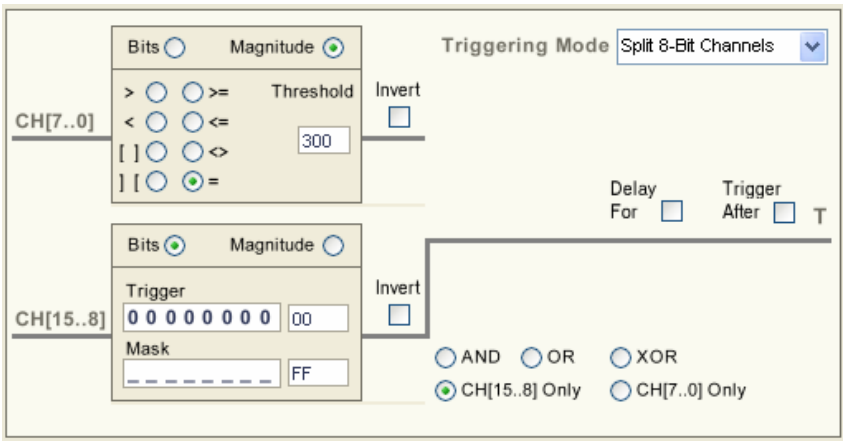


Figure 14. Single 8-bit channel triggering in Split-trigger mode.

This may resemble triggering using a standard 8-channel Logic Analyzer, but there is the added bonus of being able to use Magnitude-based triggering and the fact that you can enable either or both sub-Analyzers at the click of an option, rather than having to upgrade the Logic Analyzer instrument from 8-channel to 16-channel within the design.

Combinatorial logic gating of the outputs of the two trigger sections is provided through the **AND**, **OR** and **XOR** options. An **Invert** option next to the image for the chosen gate enables three additional logic functions to be obtained – **NAND**, **NOR** and **XNOR**.

Invert options are also provided at the output of each trigger matching condition section. When the condition for a trigger is met – either the specified bit pattern is matched or the result of a Magnitude comparison is True – the output of the section will be High (‘1’). The added Invert option in each case allows you further control when logically gating the outputs to provide the overall trigger.

Figure 15 shows an example of using the logical AND of the two trigger sections to provide a more advanced level of triggering.

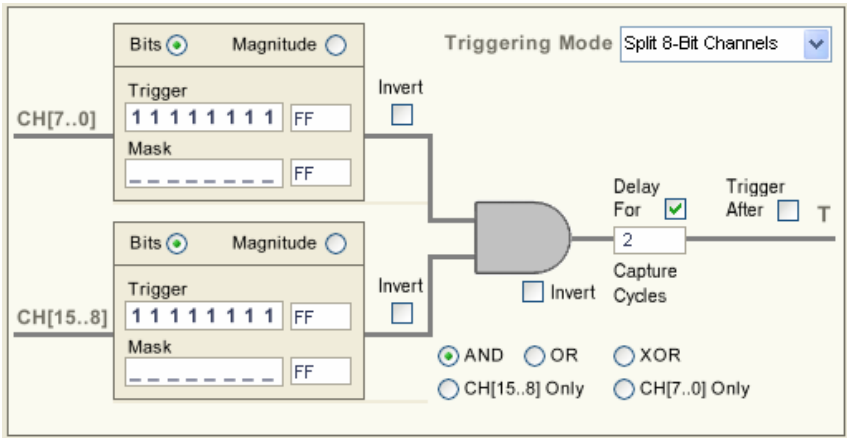
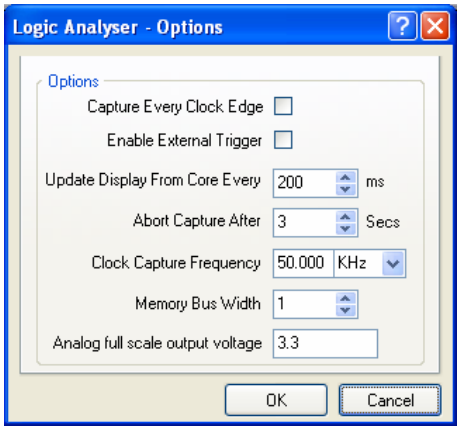


Figure 15. Advanced triggering using logical gating of the 8-channel splits.

Capturing Data

Once the Trigger pattern has been defined, the Logic Analyzer can be set ready for data capture. This is achieved using the controls in either the **Actions** region of the Instrument panel or the **Capture Control** region of the **LAX** panel.

Pressing the **Options** button will launch the *Logic Analyzer – Options* dialog.



From here, you can define the following capture-related options:

- **Capture Every Clock Edge** – use this option to capture data at the same rate as the external system clock. When this option is enabled, the CLK_CAP input is ignored.
- **Enable External Trigger** – use this option to select whether the device will operate using external (Hardware) triggering or internal (Software) triggering. If enabled, the Analyzer will be triggered externally, using the TRIGGER input. In this case, any defined software trigger pattern – on the **LAX** panel – will be ignored. If this option is disabled, external trigger events will be ignored and the trigger defined on the **LAX** panel will be used.
- **Update Display From Core Every** – use this field to define the frequency with which the fields on the Instrument panel will be refreshed – i.e. read back from the corresponding registers to which they were written and stored, including the read back of data in the sample buffer. Note that read back will only occur when the Analyzer is not armed.
- **Abort Capture After** – use this field to determine how long the Logic Analyzer continues to capture data after capture has been initiated.
- **Clock Capture Frequency** – use this field to define the data capture rate. This value does not affect data capture, rather it is used for synchronization purposes when displaying the data as analog and/or digital waveforms. The value should be set to that of the incoming clock signal on the CLK_CAP pin.
- **Memory Bus Width** – this field becomes available only when using the LAX_8 or LAX_16 devices. Use it to enter the value, in bits, corresponding to the size of the physically connected memory address bus.
- **Analog full scale output voltage** – use this field to define the full-scale voltage swing for all analog waveforms compiled from the captured data.

Once the options have been defined as required, use the **Reset** button on the Analyzer's Instrument panel to effectively flush the sample buffer. Now press the **Arm** button. The Logic Analyzer will now be set ready to capture data when the defined trigger pattern is matched and in accordance with other options that may have been set as described earlier.

As the Logic Analyzer sits ready in the armed state, the RED 'Running' LED will appear lit and the STATUS output will go High. As soon as data is captured and the sample buffer is filled, the GREEN 'Done' LED will light and the STATUS output will go Low.

Running the Logic Analyzer with External Triggering

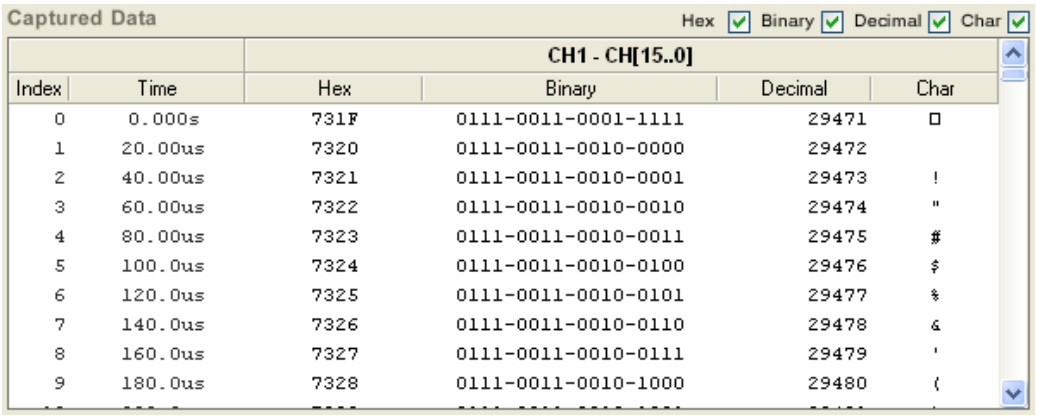
External (or Hardware) triggering and data capture is carried out using the Analyzer's TRIGGER input. To perform external triggering of the instrument, ensure that the **Enable External Trigger** option, in the *Logic Analyzer – Options* dialog, is enabled.

Irrespective of the trigger mode used (internal or external), the device is armed in exactly the same way – by clicking the **Arm** button on either the Instrument panel or **LAX** panel.

In this armed state, triggering of the instrument now depends solely on the TRIGGER input. Triggering is level-driven. When a High level is detected at the TRIGGER input, data capture is initiated. Data will be captured until the entire sample buffer for the instrument is filled.

Viewing Captured Data

Captured data, stored in the sample buffer, will appear in the **Captured Data** region of the **LAX** panel.


Captured Data Hex Binary Decimal Char

		CH1 - CH[15..0]			
Index	Time	Hex	Binary	Decimal	Char
0	0.000s	731F	0111-0011-0001-1111	29471	□
1	20.00us	7320	0111-0011-0010-0000	29472	
2	40.00us	7321	0111-0011-0010-0001	29473	!
3	60.00us	7322	0111-0011-0010-0010	29474	"
4	80.00us	7323	0111-0011-0010-0011	29475	#
5	100.0us	7324	0111-0011-0010-0100	29476	$
6	120.0us	7325	0111-0011-0010-0101	29477	%
7	140.0us	7326	0111-0011-0010-0110	29478	&
8	160.0us	7327	0111-0011-0010-0111	29479	'
9	180.0us	7328	0111-0011-0010-1000	29480	(

Data is listed in terms of its Index - its position within the sample buffer - and can appear in the following formats, the display of which is enabled using the corresponding check boxes in the panel:

- Hexadecimal
- Binary
- Decimal
- ASCII Character.

The time at which each sample is captured is also shown.

The number of samples displayed will be equal to the size of memory that is connected to the Logic Analyzer (LAX_8 and LAX_16) or comes predefined as part of the Logic Analyzer device (1K, 2K and 4K devices).

Remember that the Index of the event that actually triggered the instrument is determined by the **Delay For** option, in the **Logic Analyzer Triggering** region of the **LAX** panel. For example, if this option is disabled, the trigger-event sample will have Index = 0. If the **Delay For** option is enabled with a value of 2, the trigger-event sample will have Index = 2, and so on. Figure 16 demonstrates this.

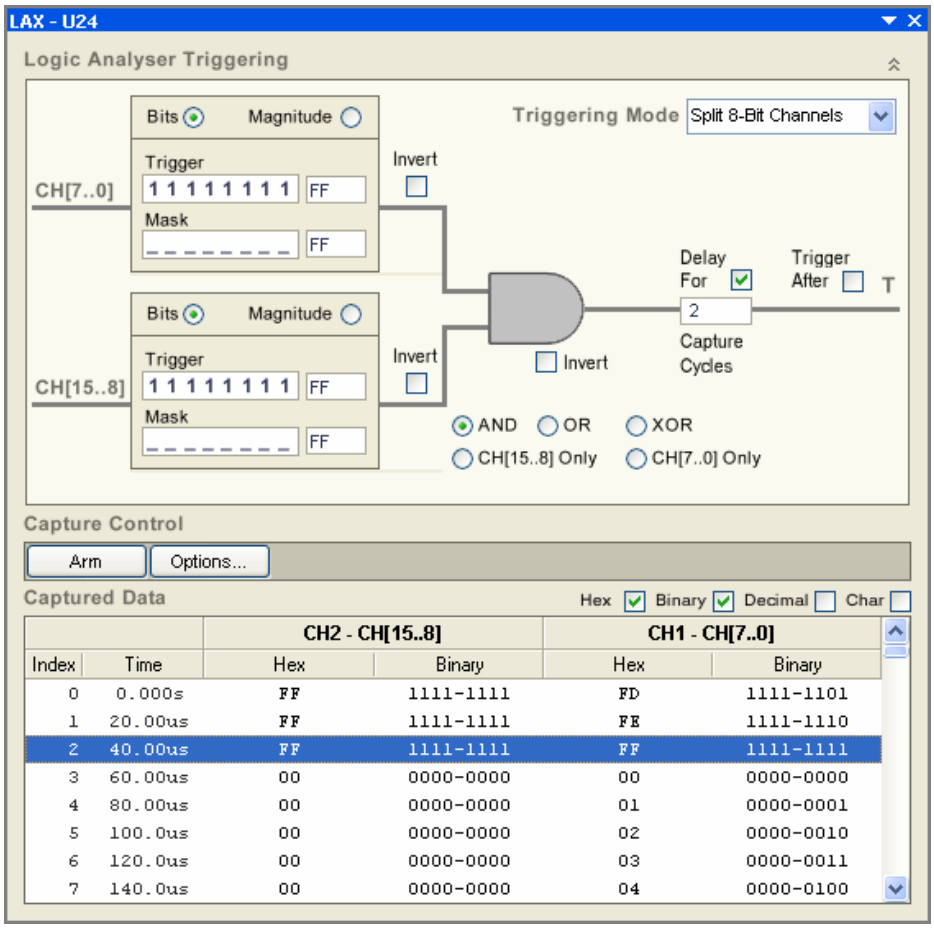


Figure 16. Locating the trigger event sample within the captured data.

Graphical Display of the Captured Data - Waveformsy

Captured data can be displayed in both analog and digital waveform format. Access to the respective waveform views is made through the Analyzer's Instrument panel.

Digital Waveforms

Pressing the **Show Waves** button associated to Digital output, in the **Data Views** region of the Logic Analyzer's Instrument panel, will load the captured data into a digital wave file (`*.LaxDig`) and open the file as the active document view. The data captured for all input channels will be graphically displayed in the Waveform Analysis window.

Note: If a 16-channel device has been used in split-trigger mode, two groups of signals – consisting of the upper and lower bytes – will be generated and displayed.

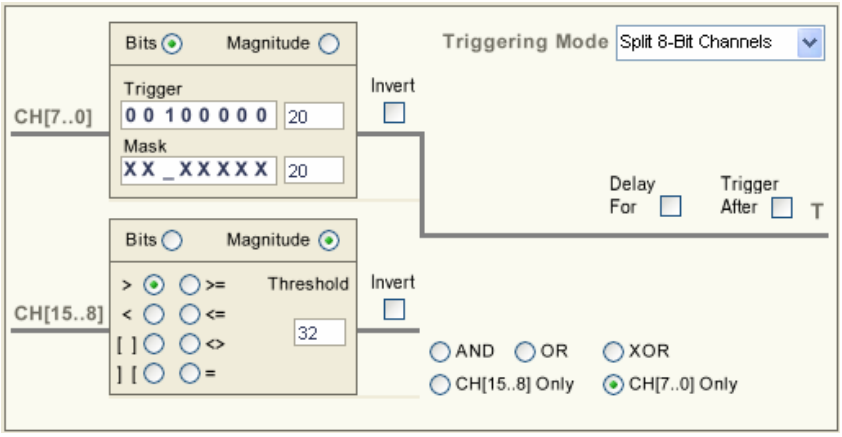
The document can be toggled between the open and closed state, simply by clicking anywhere within the DIGITAL region[3]. If modifications have been made to the document, a confirmation dialog will appear allowing you to save the changes.

Figure 17 shows example digital waveforms for captured data obtained from the AudioCodec FlashMemory – TSK165 example project (which can be found in the `\Examples\NB1 Examples\Processor Examples` folder of the installation).

[3] By default the digital wave view is set in the open state so that upon data capture, the digital wave file will be loaded and opened straight away.

Figure 17. Digital waveforms for the sampled input channels.

In this case, the Logic Analyzer was configured in split-trigger mode, with only channels 7..0 used for triggering and specifically set to trigger on the SPI_DIN line (D5).



As can be seen in Figure 17, the captured data is represented in two groups, reflecting the 16-channel Analyzer having been configured into Split 8-bit channels.

Analog Waveforms

Pressing the **Show Waves** button associated to Analog output, in the **Data Views** region of the Logic Analyzer's Instrument panel, will load the captured data into an analog wave file (`*.LaxAn`) and open the file as the active document view. The data captured for all input channels will be graphically displayed in the Waveform Analysis window.

Note: If a 16-channel device has been used in split-trigger mode, two analog waveforms – generated from the upper and lower bytes – will be displayed.

The document can be toggled between the open and closed state, simply by clicking anywhere within the ANALOG region. If modifications have been made to the document, a confirmation dialog will appear allowing you to save the changes.

Figure 18 shows example analog waveforms for captured data, once again obtained from the AudioCodec example project.

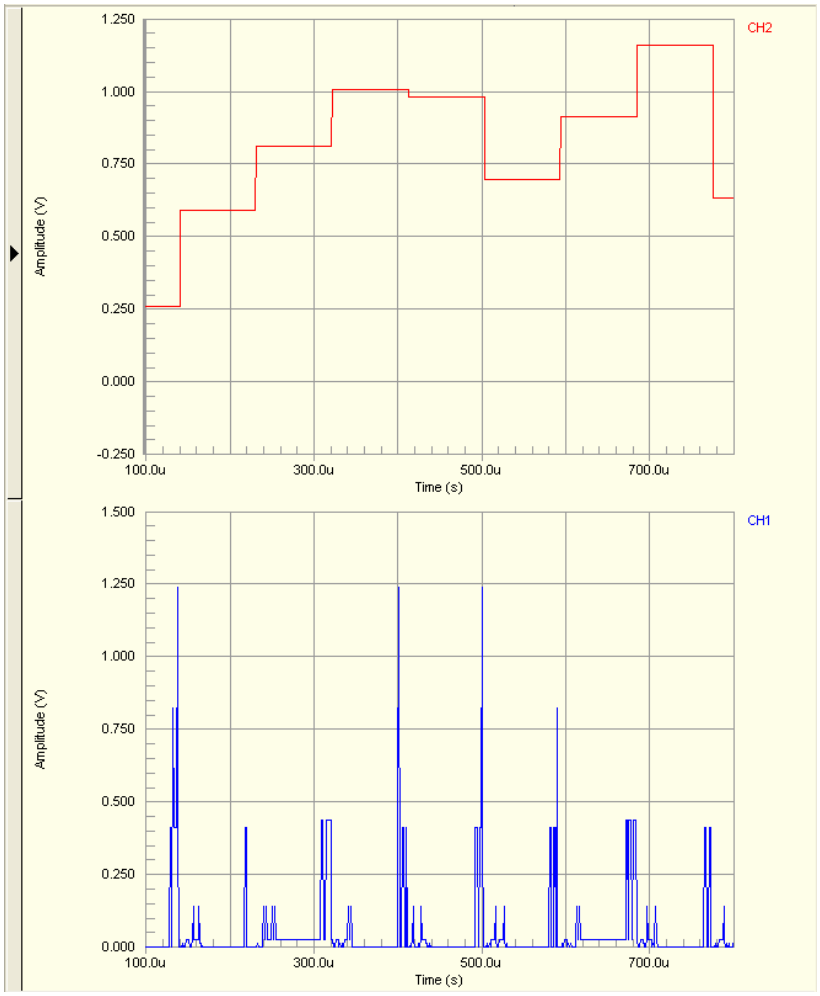


Figure 18. Generated analog waveforms for the sampled inputs.

In this case, the Logic Analyzer was configured in split-trigger mode, with only channels 15..8 used for triggering and specifically set to trigger with a magnitude greater than 32.

An analog waveform is created from its constituent digital signals (for the input channels it represents) by first normalizing to 1 and then scaling to the range specified by the value entered in the **Analog full scale output voltage** field, defined in the *Logic Analyzer – Options* dialog.

The two previous examples show triggering of the device on analog or digital data. You could quite easily set up a sophisticated trigger that is based on a combination of the two.

Capture Control

With either the digital or analog wave file open as the active document, the Logic Analyzer can be controlled using the following commands available from the **Logic Analyzer** menu:

Single Capture - Run the Logic Analyzer in Single-Shot mode. The Waveform Analysis window and the Captured Data region on the associated LAX panel will be refreshed with the new data accordingly.

Continuous Capture - Run the Logic Analyzer in Continuous Capture mode. The Waveform Analysis window and the Captured Data region on the associated LAX panel will be refreshed with the new data accordingly after each capture. Capture will continue until either the Stop button is pressed or the wave file is closed.

Stop Capture - Stop Logic Analyzer when running in Continuous Capture mode.

Logic Analyzer Options - Setup Logic Analyzer. Clicking this button opens the *Logic Analyzer – Options* dialog, from where you can define various options with respect to the capture and viewing of data (see *Capturing data*).

If both Analog and Digital waveform views are open simultaneously (i.e. split windows), these controls will affect both views at the same time. In this way you can see continuously captured digital and analog data side-by-side.

Analyzing Waveforms

When analyzing the captured data, a number of features are available within the waveform viewers that enable combined analysis of both the analog and digital waveforms:

- If the `*.LaxDig` and `*.LaxAn` documents are open side-by-side, panning or zooming within one document will affect the other document at the same time
- If you enable cursors in the two documents, the active cursor in the digital waveform document will be synchronized to Cursor A in the analog waveform document

The active cursor in the digital waveform view and Cursor A in the analog waveform view are also synchronized to the Captured Data region of the associated **LAX** panel. Moving the cursor in either document will move the cursor in the other document and move up or down through the sample buffer display in the panel accordingly, and vice versa. Similarly, selecting a captured sample in the **LAX** panel will cause the cursor in either or both waveform documents (depending on whether they are open side-by-side) to jump to the corresponding point along the waveform for that particular sample.

Revision History

Date	Version No.	Revision
20-Jan-2004	1.0	New product release
21-Jun-2005	1.1	Updated for Altium Designer SP4
12-Dec-2005	1.2	Path references updated for Altium Designer 6
17-Mar-2008	2.0	Updated for Altium Designer Summer 08