

## Overview

The LogicPort logic analyzer provides 34 sampled channels including two state mode clock inputs. Unused clock inputs are available as normal sampled channels in either timing or state mode. The LogicPort connects to a host PC's USB 1.1 or 2.0 port to form a complete high-speed logic analysis system.

## Hardware features

### Timing mode sample rate 1KHz to 500MHz

Real-time sample rates up to 500MHz capture today's high-speed digital signals.

### State mode sampling to 200MHz

Allows synchronous capture of signals based on an external clock provided by the hardware under test. Especially useful for DSP work when combined with the LogicPort's ability to represent a group of signals as an analog waveform.

### Qualified state mode sampling

Allows de-mux of buses, and capture of qualified data. Useful when working with microprocessors which multiplex data and address onto the same bus. Can also be used to capture data directed toward a particular device by using that device's chip select signal as a qualifier.

### Sample Compression (Transitional Sampling)

Data is compressed on its way into the sample buffer. This means much more data can be acquired. Compression is available both for timing and state mode.

### Logic threshold adjustable over a range of +6 to -6 volts

Combined with the LogicPort's high sensitivity and wide dynamic range, this allows compatibility with virtually all logic families including TTL, LVTTTL, CMOS, ECL, PECL, RS232 and others.

### Channel to channel skew <1ns

Gives an accurate representation of the timing relationships between different signals.

### High input impedance

Input resistance of 200k $\Omega$ , paralleled by <5pF provides minimal hardware loading.

### Multi-level trigger

Two trigger levels can be used individually, or logically combined in one of several modes. Each level can consist of multiple terms including:

**Edges** - Rising, Falling, Either (multiple channels, any combination)

**Pattern** - True, False, Entered, Exited

**Value** - Equal, Not Equal, Less Than, Greater Than, In Range, Not in Range

**Occurrence Count** - Range of 1 to 1,048,576 Consecutive or Cumulative

**Duration** - Equal, Less Than, Greater Than, In Range (samples or time)

### State mode Setup/Hold window adjustable

Sample time can be adjusted +/-2.5ns relative to clock edge allowing easy capture of signals with skewed clocks or narrow "data valid" windows.

### Hardware-based frequency counter

Allows real-time measurement of signals up to 300MHz regardless of the current sample mode and rate.

### USB Powered

Requires no external power supply.

## Software features

### User friendly (really)

A great deal of work has been put into making sure the LogicPort is simple and pleasant to use. Numerous context-sensitive menus are available with a right-click of the mouse. The user interface has been designed to maximize the display area available for sampled data. Plain-English trigger setup tames the beast which plagues many logic analyzers. You won't have to dedicate your life to learning how to use the LogicPort!

**Runs in Demo mode when no hardware is present**

Allows acquired data to be viewed off-line when the LogicPort is not connected. Send your acquired data to a colleague in the next room, or another country. They can use the LogicPort software in demo mode to view it exactly as it was acquired in your lab.

**Requires no registration**

The software may be installed on as many computers as you like. This allows the LogicPort to be conveniently moved from one workstation to another. Simply unplug from the USB port on one PC and plug it into another.

**Fast acquisition rate**

Typically 4 acquisitions per second on a 400MHz Pentium II desktop PC. Take that retired PC out of the closet and put it back to work as a high performance logic analyzer!

**Includes interpreters for I<sup>2</sup>C, SPI and RS232 protocols**

The software can interpret and display any combination of up to 17 I<sup>2</sup>C bus interfaces, 8 synchronous serial (SPI) interfaces, and 34 asynchronous serial (RS232) data channels.

**WYSIWYG Printing**

Print just the data you want to see.

**Simple file format**

Each project file contains settings, notes, and sampled data for the project. Sampled data can be exported to a separate file in a variety of formats. Data is exported as comma-separated values for easy import into other programs such as Microsoft Excel.

## Quickstart Tutorial

If you are familiar with the use of logic analyzers, this tutorial provides a brief explanation of LogicPort software features which will help you get up and running fast. The [Step by Step Example](#) in the next section provides detailed instructions for making your first measurement with the LogicPort.

### Terminology: About Wires, Signals, Groups and Interpreters

**Wires** are, as the name suggests, physical connections to the LogicPort. They have the fixed names D0 to D31, CLK1, and CLK2. These names are standardized to match the color-coded probe which is supplied with the LogicPort. Signal and ground wire colors are defined in the [Connector Pin Assignments](#) section of the appendix. It is important to understand that the ground wires have a [connection to the PC's chassis](#) through the USB cable.

Note that CLK1 and CLK2 have special functions *only* when sampling in state mode. When sampling in timing mode these inputs are normal sampled channels, and can be thought of as D32 and D33.

**Signals** can be thought of as aliases for wires. New projects start with a default set of signal names which you can edit as desired. Each wire has only one signal name associated with it.

**Groups** are collections of signals. A group may include any number of signals. Signals can be included in more than one group. You can skip wires when forming groups, but you cannot arrange them randomly. Here is an example of a legal group (wire names shown):

[D0, D1, D12, D13, D22]

You can also swap the MSB-LSB ordering of an entire group. The bit ordering can be changed by selecting the appropriate "Data Format", or by using the "Reverse Display Order" button in the group edit dialog. The following group is legal:

[D7, D6, D5, D1, D0]

The following group is NOT legal because the relative bit weights haven't been maintained:

[D0, D2, D1, D3, D4]

Note that the use of groups is entirely optional. All signals within a project can be displayed without using any groups.

**Interpreters** can be thought of as special groups. Interpreters are used to translate certain protocols and display the resulting high-level data. To learn more about interpreters see the [interpreter overview](#).

### About Project Files

Project files contain all settings and data for a given project. These files have the extension ".LPF". The file "Quickstart.LPF" is opened by default when the LogicPort software runs for the first time. This file can be edited, and the sampled data can be overwritten by a new acquisition (data will be simulated if in demo mode).

The example files which are installed with the LogicPort are **read-only** files. Acquisition is disabled in these files, and changes to their settings will not be saved. This is to insure that the example files remain intact for future reference.

### Guided Tour

The best way to get to know the LogicPort is to take a quick tour of the software, clicking tabs, menus, and drop-down lists. Most controls have "Tool Tips" which give a brief explanation if you hover the mouse cursor over them for a short time. Note that when the help file is open you may need to click on the title bar of the LogicPort application before viewing tool tips.

### When in doubt, right-click

Many of the LogicPort's software functions are most easily accessed through context-sensitive "popup" menus. These menus open when you click in different areas of the window using the right mouse button.

Here is a suggested path for exploring the software:

#### Menus

You can create new project files from the "File" menu. When new files are created, they include a default set of signals and groups to make setup easier. User interface characteristics such as display colors and reference position are selected in the "Options" menu. Logic analyzer settings for a particular acquisition may be edited from the "Setup" menu.

## The Toolbar

Most commonly used items can be accessed by clicking a button in the toolbar. All buttons in the toolbar have equivalent menu items.

To the right of the toolbar buttons is the buffer position indicator. This shows the positions of the cursors and displayed sample data relative to the sample buffer extents.

## Status/Measurement panels

The [status bar](#) located at the bottom of the application window shows the current acquisition status, along with four selected measurements. Click the "Measurement Setup" button in the toolbar or click in one of the measurement panels to select which measurements are displayed. Measurements can be made between cursor, trigger and reference positions. Frequency and period can also be measured in real-time using the LogicPort's internal frequency counter.

## The Waveforms tab

The controls at the top of the tab are configurable, allowing you to select which parameters you want to adjust. Click the drop-down list above any of the scrollbars to see the available options.

The table area on the left is referred to as the waveform table. Here columns and signal, group and interpreter rows are added, removed, and rearranged. Patterns and edges used in trigger setup are edited in this area. Note that an edge or pattern can be displayed without being active. Pattern and edge terms are utilized only when they are enabled in the trigger setup dialog. Data and waveform formats are also selected in this area. Right-click in different areas of the waveform table to see the various context-sensitive menus. There are five different menus depending on where you click within the table. Some menus are accessed only from the top row. You can rearrange the columns by clicking and dragging the column by its heading. You can rearrange rows by clicking and dragging the row by its name. Double-clicking a group name, or clicking the "+" or "-" symbol at the left of a group name will expand/collapse the group to show/hide the signals it contains.

The area to the right of the waveform table is referred to as the waveform window. You can scroll the waveforms by clicking with the left mouse button and dragging them horizontally, or by using the scroll bar at the bottom of the waveform window. You can click and drag the cursors as well. Zoom in on a particular area of a waveform by left-clicking the area of interest while pressing the <Ctrl> key on the keyboard. Zoom out by right-clicking with the mouse while pressing the <Ctrl> key. You can also zoom in/out by pressing the +/- keys on the keyboard. See [Scrolling and Zooming](#) for more options.

If you hover the mouse cursor over a group waveform, the group's value at that sample's location appears in a tooltip. Right-click to see the context-sensitive menu for the waveform window.

## The State List tab

You can scroll the state list data and cursors by clicking and dragging them vertically. You can also scroll the data using the <Page Up> and <Page Down> keys on the keyboard or with the scrollbar at the right-hand edge of the window. You can rearrange the columns in the state list by dragging their name (note this also rearranges rows in the waveform table). There are four different context-sensitive menus available in the state list. Most of them are accessed from the top row.

## The Notes tab

This is just a scratchpad area where you can enter any notes you like, such as a description of how the sampled waveforms occurred. The notes are stored in the project file, along with sampled data and settings.

## Sample Mode Setup dialog

Open this dialog by clicking the "Sample Mode Setup" toolbar button (or menu option). From this dialog you can select timing or state mode, setup state mode parameters, and control sample compression.

## Trigger Setup dialog

Open this dialog by clicking the "Trigger Setup" toolbar button (or menu option). From this dialog you can setup trigger conditions. The drop-down list at the top determines which trigger levels are active, and if/how they will be logically combined. Levels are grayed-out when they are not active. Each trigger level has several terms which can be enabled. Within each level, terms are grayed-out when they are not enabled. Some terms are mutually exclusive. When a term is not available, its check-box will be grayed-out. Note that the patterns and edges referenced in this dialog are specified in the pattern and edge columns of the waveform table.

**This concludes the Quickstart Tutorial**

If you need more information about a particular feature, refer to the Detailed Operation section. It's a good idea to read through the FAQ section, and the Appendixes as well.

## Step-by-step: Exercise Objective

In this exercise we will create and setup a complete project from start to finish. The project will be based on an imaginary microprocessor circuit with the following characteristics:

8 Data lines

16 Address lines

RST, CLK, RD, WR and IRQ signals

25MHz CPU clock

CMOS Logic with a +5V power supply

We will setup to monitor all signals in timing mode, and to trigger on the rising edge of the IRQ signal only if the RST signal is low. For the purpose of this exercise we are most interested in the events which led up to the trigger event.

The LogicPort software should be running in Demo mode as you go through this exercise. Demo mode is only available when the LogicPort hardware is not connected. If you have the LogicPort connected, you can enter Demo mode by closing the LogicPort software, disconnecting the LogicPort, then restarting the software.

As you open different dialogs during this exercise, you may need to rearrange the help and dialog windows for more convenient access. You can use the left mouse button to drag any of the windows by their title bar.

Throughout this exercise, items shown as **Bold** are generally permanent items which can be found in the LogicPort GUI (menu items, buttons, etcetera). Items shown in **Green** represent potential user input.

Use the [>>](#) and [<<](#) links to navigate through the exercise.

## Step-by-step: Create a New Project



From the **File** menu select **New...** to create a new project file. A window will open asking for the name and location of your new project file. Specify the name **CPU** and click **Save**. You do not need to add an extension, the software will append the correct extension before saving the file.



## Step-by-step: Edit Signal Names



New project files start with a default set of signals. Usually you will want to edit the signal names to match your hardware. Though not necessary to complete the exercise, this might be a good time to learn about [Wires, Signals, Groups and Interpreters](#).

From the **Setup** menu select **Signal Names...** to open a dialog which enables you to edit the signal name for each wire. Wires **D0** through **D7** already have signal names **Data0** through **Data7** which are appropriate for our project, so we will leave the defaults for these wires. Click on **Data8** to edit the signal name for wire **D8**. Type **Address0** in the box at the top of the dialog. Continue editing the signal names for wires **D9** through **D23** in the same fashion using the signal names **Address1** through **Address15**. Note that you can use the Up and Down Arrow keys on the keyboard to scroll the signal list. Edit the signal names for wires **D24** through **D27** and **CLK1** as follows (**CLK1** is at the bottom of the list):

**D24 : RST**

**D25 : RD**

**D26 : WR**

**D27 : IRQ**

**CLK1 : CLK**

Click **OK** to dismiss the signal edit dialog.



## Step-by-step: Edit Groups



New project files start with a default set of groups which can be edited or deleted as desired. It's not necessary to delete unused groups.

From the **Setup** menu select **Groups...** to open a dialog which enables you to edit the groups contained in the project. The group named **Data[7..0]** is already setup appropriately for our project. Click to highlight the group named **Data[15..0]**. Now right-click to bring up the context-sensitive menu, and click **Edit Selected...** to edit the group. A dialog will open showing the group name **Data[15..0]** near the top. Change this name to **Address[15..0]**.

The default wires for this group are **D0** through **D15**. We need to edit this group to include only wires **D8** through **D23** (signals **Address0** through **Address15**). Start by selecting signals **Data0** through **Data7** in the **Signals In Group** list. You can use standard Windows editing shortcuts to select signals. For example click **Data0** then hold down the **Shift** key on the keyboard and click **Data7** to select the range of signals. Now click the **<** button to remove these signals from the group.

Next select signals **Address8** through **Address15** in the **Signals Available** list, and click the **>** button to add these signals to the group. The group should now include only signals **Address0** through **Address15**.

Click **OK** to dismiss the group edit dialog.

Select the **Data[31..0]** group, then right-click to bring up the context-sensitive menu. Click **Delete Selected** to delete this unused group. Notice that the group disappears from the waveform display as well.

Click **OK** to dismiss the dialog.



## Step-by-step: Setup Waveform Display



New projects start with a default set of signals and groups in the waveform display. For this exercise we will edit the waveform display to better suit our needs.

Right-click the **Clock2** cell in the waveform table to bring up a context-sensitive menu. Click **Remove Signal** to remove this signal from the display.

Right-click the **Signal** cell in the waveform table to bring up a context-sensitive menu. Click **Add Signal...** to open a dialog which enables you to select signals to be displayed in the waveform window. You can select multiple signals by holding down the **Control** key on the keyboard while selecting the signals. Scroll down the list and select all of the following signals:

**RST**

**RD**

**WR**

**IRQ**

Once all of these signals have been highlighted, release the **Control** key and click **OK** to dismiss the dialog and add them to the display.

Right-click the **Signal** cell in the waveform table to bring up a context-sensitive menu. Click **Add Group...** to open a dialog which enables you to select groups to be displayed in the waveform window. Hold down the **Control** key on the keyboard and select both groups **Address[15..0]** and **Data[7..0]**.

Release the **Control** key and click **OK** to dismiss the dialog and add these two groups to the display.

With the left mouse button, click and drag the **RST** cell up one row to rearrange the waveform table.

From the **Options** menu, select **Reference Position**, then click **Right/Bottom** to set the reference to the right-hand side of the display.



## Step-by-step: Setup Sample Conditions



From the **Setup** menu, select **Sample Mode...** to open the sample mode setup dialog.

Notice that timing mode is selected by default for new projects. Compression is also enabled with typical Pre/Post-Fill Time Limits selected. These settings are correct for this exercise, so dismiss the dialog by clicking the **OK** button.

The controls along the top of the Waveforms tab enable you to set other sample related parameters.

Since the CPU clock is 25MHz, we need a minimum sample rate of about 100MHz to acquire meaningful data. In this exercise, we are interested in seeing the timing relationships between signals so we will use a higher sample rate of 200MHz to get better resolution. Use the **Sample Rate** slider or drop-down list to select the **200MHz** sample rate.

The threshold for the CMOS logic in our imaginary circuit is 50% of the +5v power supply voltage. Use the **Logic Threshold** slider to set the threshold to **2.50v**.

For the purpose of this exercise, we are most interested in the events which led up to the trigger event, so use the **Pre-Trigger Buffer** control to reserve **99%** of the sample buffer for pre-trigger data (move the slider all the way to the right).



## Step-by-step: Setup Trigger Conditions

From the **Setup** menu, select **Trigger...** to open the trigger setup dialog.

For this exercise we only need one level of trigger conditions, so at the top of the dialog select **Trigger When level B is satisfied** from the drop-down list.

We will use edge B to detect the rising edge of the **IRQ** signal, and we will use pattern B to prevent trigger from occurring unless **RST** is low.

We don't need the pattern B term to be prequalified for this exercise, so make sure the **Prequalify Pattern/Value Terms** check-box is not selected (this is the default for new projects).

Confirm that the check-box next to **Edge B occurs** is "checked" to enable the edge term for trigger level B (this is the default for new projects). Since we are only interested in the first occurrence of edge B, it doesn't matter whether we select **Cumulative** or **Consecutive** for this exercise.

Click the check-box next to **Pattern B is** to enable the pattern term for trigger level B. Confirm that the drop down box is set to indicate **Pattern B is True** (this is the default for new projects).

Click **OK** to dismiss the trigger setup dialog.

We will set up trigger **Pattern B** and **Edge B** in the waveform table. The **Pattern A** and **Edge A** columns are included by default for new projects. Since these columns are not required for this exercise, we will set them to show **Pattern B** and **Edge B** instead.

You can do this by right-clicking the **Pattern A** cell at the top of the table to bring up a context-sensitive menu, then selecting **Set Column>Pattern>B**. Use the same method to set the **Edge A** column to show **Edge B**.

Click once in the **Pattern B** cell for the **RST** signal to setup pattern B.

Click once in the **Edge B** cell for the **IRQ** signal to setup edge B.

We have now set the trigger conditions such that trigger will occur on the rising edge of **IRQ**, but only when **RST** is low.

From the **File** menu, select **Save** to save all settings for the project file.

Note that in this exercise we have used trigger level B to demonstrate the fact that the two trigger levels have identical functionality, and to gain experience setting up columns in the waveform table. New projects are setup to use trigger level A by default. If we had setup this same trigger scheme using trigger level A, this entire setup process could have been accomplished with just two mouse clicks in the waveform table.

## Step-by-step: Connect to the Circuit



Because this exercise is based on an imaginary circuit, there is obviously no hardware to connect. Though we can't make an actual physical connection, we can go through the motions to learn how to connect real hardware.

First, expand the signal groups in the waveform table by right-clicking the **Signal** cell and then clicking **Expand All**. We do this to make the color codes for all wires visible in the **Wire ID** column.

Signal and ground wire colors are defined in the [Connector Pin Assignments](#) section of the appendix. The wires are organized as four groups of eight wires, plus two clock wires and four ground wires. Each group of eight wires shares a common color for the contact insulator. This color matches the background colors in the wire ID column. Each of the eight wires within these groups has a wire insulation color which matches the foreground (text) color in the wire ID column. You can use these color codes to determine which wires connect to specific signals in your hardware.

The ground wires are gray, and have a black contact insulator. For best signal fidelity, be sure to connect as many ground wires as possible to your hardware. It is important to understand that the ground wires have a [connection to the PC's chassis](#) through the USB cable.



## Step-by-step: Acquire Data



From the **Acquisition** menu, select **Single** to make a single data acquisition.

New data appears in the waveform window. Since the LogicPort software is running in Demo mode, this will be simulated data. Note that the content of the simulated data generated in Demo mode is not affected by any settings. It is there only to give an idea of how the LogicPort represents signals.

Now select **Recurring** from the **Acquisition** menu.

New (simulated) data is now acquired and displayed in the waveform window on a recurring basis.

Select **Halt** from the **Acquisition** menu to stop the recurring acquisition.

Note that with real hardware, acquisitions will not occur unless the trigger conditions are met.

The last method of initiating acquisition is the **Trigger Immediate** command. This command tells the LogicPort hardware to ignore the trigger settings and acquire new data *right now*. You may want to use this command if your trigger settings are preventing acquisition from occurring. With the **Trigger Immediate** command you can get a look at the hardware's activity without having to meet any trigger conditions. This is essentially the same as selecting **Trigger Immediately when acquisition starts** in the trigger setup dialog.



## Step-by-step: What's Next?

Here are a few other things you can do to get more acquainted with the LogicPort software:

Drag the waveforms back and forth using the left mouse button to see the interaction between the waveform position and the reference offset indicated at the reference position (when setting up this project we set the reference position to the right-hand edge of the graticule in the waveform window).

Make sure the LogicPort application has "System Focus" (click in the title bar to be sure), then move the mouse cursor to a position within the waveform window and press the **Control** key on the keyboard. Notice that the mouse cursor changes to a magnifying glass, indicating that you can zoom in/out by clicking within the waveform window. While pressing the **Control** key, left-click near **Cursor A** to zoom in and center that portion of the data within the display. Notice that the mouse cursor also moves to the center of the display so that it remains aligned with the area of interest. You can right-click to zoom out.

From the **View** menu, select **Zoom All** to zoom out and bring all of the acquired data into view.

From the **Options** menu, select **Show Cursors>None** to hide the cursors.

Notice that 99% of the data is to the left of the trigger cursor (indicated by the **T** with a line extending downward from it). Set the **Pre-Trigger Buffer** setting to **1%** and select **Single** from the **Acquisition** menu. Note that now only 1% of the data is to the left of the trigger cursor.

From the **Options** menu, select **Reference Position>Left/Top**. This moves the reference position to the left side of the display. Make sure the LogicPort application has "System Focus" (click in the title bar to be sure), then press the **T** key on the keyboard to scroll the waveforms so that the trigger cursor is at the reference position.

Right-click somewhere around the middle of the acquired data. From the context-sensitive menu select **Place Cursor>A**. This makes the cursors visible again and places cursor A at the position where you clicked to open the menu.

Drag the individual cursors around using the left mouse button and notice the result in the **Buffer Position** indicator at the top of the application. The buffer position indicator exists to help orient your current position within the sampled data, especially when the waveform scale is zoomed way in or out.

Move cursor A to a position within the sampled data. Make sure the LogicPort application has "System Focus" (click in the title bar to be sure), then press **A** on the keyboard to scroll the waveforms so that cursor A is at the reference position.

Click on the **State List** tab at the top of the window.

Notice that cursor A is centered at the reference position in the state list display as well.

Drag the data in the state list up or down using the left mouse button and notice the result in the **Buffer Position** indicator.

Click the **Notes** tab at the top of the window.

You may want to type some notes in this area to remind you of some of the things you have learned during this exercise. After you've entered your notes, select **Save** from the **File** menu to be sure your work is saved. Your notes will be saved in the project file.

You can use this project file as a template for another project by selecting **Save As...** from the **File** menu, and saving the file with a different name.

### This concludes the Step by Step example

Now that you have some familiarity with the LogicPort software and terminology, we recommend reviewing the [Quickstart Tutorial](#).

You can find more detailed information about specific topics in the Detailed Operation section which follows. It's a good idea to read through the various general discussion topics in the Appendix as well.

## Menu Item Descriptions

### File ->

#### **New**

Creates a new project with default settings and random data which can be used to setup signals and groups.

#### **Open...**

Opens an existing project file.

#### **Save**

Saves the current project.

#### **Save As...**

Saves a copy of the current project using a specified name.

#### **Export Data...**

Opens the file export dialog.

#### **Print...**

Opens the print dialog enabling selection of the printer and page orientation. These settings are global options which are remembered between sessions. When "Auto Select" is specified for the Orientation, the software automatically determines which orientation will allow the image to fit best on the page.

#### **Print to Clipboard**

"Prints" an image of the current tab to the Windows Clipboard so that it can be pasted into documents.

#### **Recent Files**

Displays a list of recently used files for quick access

#### **Exit**

Exits the application.

### Options->

#### **Global Settings->**

Settings within this sub-menu are common to **all** projects. All other settings are unique to each project.

#### **Display->**

##### **Color Scheme**

Sets the color scheme used for graphic elements of the waveform window. The default waveform color can be changed for each individual row. Colors specified for individual rows are stored in the associated project file.

##### **Alternate Background->**

###### **Enable**

If selected, alternating background stripes are added to the waveform window.

###### **Adjust...**

Opens a dialog which enables adjustment of the alternating background color and intensity.

##### **Waveforms in Front**

When selected, the waveforms are painted last, after all cursors and other display elements. When not selected, the waveforms are painted first, with all other display elements in front.

##### **Large Waveforms**

Sets the displayed waveforms to a larger size to make distant viewing easier.

##### **Clear Before Acquisition**

Specifies that previously acquired data should be cleared from the display **and** project when a new sample sequence is started. Each time a sample sequence is initiated using the "Single", "Recurring" or "Trigger Immediate" option, the display will be cleared and **previously acquired data will be deleted from the project**.

##### **Sample Reference**

When "Left Edge" is selected, graticule and cursors reference the left edge of a given sample. This is the default setting. When "Center" is selected, graticule and cursors reference the center of a given sample. This is provided for personal preference, and has no effect on the data.

## Optimization ->

### Maximum Performance

Optimizes the LogicPort software for maximum acquisition rate and best GUI response. Usually the best choice for slower PCs or any PC running Windows98/WindowsME.

### Minimum CPU Loading

Optimizes the LogicPort software for minimum CPU loading. Use this option if you are running another program on your PC which needs more CPU time.

### Save On Exit

Sets the file-saving behavior when a project is exited. Files are exited when the application closes, and also when opening another project or creating a new one. The "Always" option is recommended. You may want to select another option if you are reviewing important sample data and don't want any chance of it getting overwritten when the file is exited.

## Reference Position

Selects the time/sample reference position in both the waveform window and state list. The graticule values are also relative to this position. Scroll and Zoom functions are relative to this position (except <Ctrl> + Left Click).

## Scale Factors->

### Relative to Reference

Specifies that the graticule indications should be relative to the reference position. Note that the reference position indication is *always* relative to the trigger event.

### Relative to Trigger

Specifies that graticule indications should be relative to the trigger event, rather than the reference position.

## State List Format->

### All Samples

Specifies that the state list should show all samples, even when they are identical.

### Unique States

Specifies that the state list should show a new state on each row (identical samples are skipped).

## Cursor Tracking->

### A..F Tracks

Specifies which cursor (if any) the associated cursor will track. You can temporarily disable cursor tracking by holding down the <Shift> key while dragging a cursor.

### Interlock All

Sets all cursors to track each other. When this feature is used, moving any cursor will move all cursors.

### Release All

Removes tracking assignments from all cursors. You can also *temporarily* disable cursor tracking by holding down the <Shift> key while dragging a cursor.

## Cursor Snap

When enabled, cursors will snap to the nearest displayed transition when the mouse button is released after dragging. If the mouse button is released on a particular waveform row, the cursor will snap to the nearest transition on *that* waveform. If the mouse button is released above or below all waveform rows, the cursor will snap to the nearest transition found in *any* waveform. You can temporarily disable cursor snap by holding down the <Alt> key while dragging a cursor.

## Show Graticule

If selected, vertical graticule lines are added to the waveform window.

## Show Trigger

If selected, a dotted vertical line indicates the trigger sample position in the waveform window.

## Show Cursors

Allows you to specify which (if any) of the six color-coded cursors are displayed in the waveform window and state list.

**Show Time**

Sets the format of interval indications for the waveform window, state list, and measurement panels. All indications will be in seconds calculated from the current sample rate (or the specified sample rate if in state mode).

**Show Samples**

Sets the format of interval indications for the waveform window, state list, and measurement panels. All indications will be in samples.

**Setup->**

**Signal Names...**

Opens the signal name editor.

**Groups...**

Opens the group edit dialog.

**Interpreters...**

Opens the interpreter edit dialog, where interpreters can be created, edited or deleted.

**Measurements...**

Opens the Measurement Setup dialog.

**Logic Sense...**

Opens a dialog which enables the logic sense to be individually specified for each input. Note that logic inversion occurs *before* capture.

**Sample Mode...**

Opens the Sample Mode Setup dialog.

**Trigger...**

Opens the Trigger Setup dialog.

**Acquisition->**

**Single**

Start a single acquisition sequence.

**Recurring**

Begins a recurring acquisition sequence. If trigger conditions are met, sample data will be acquired over and over until a Halt or Single acquisition command is received.

**Halt**

Halts the acquisition in progress and aborts recurring acquisition.

**Trigger Immediate**

Initiates an immediate acquisition without regard for acquisition mode or trigger conditions. If an acquisition is in progress, causes the current sample phase to be exited. See [Sample Mode Setup](#) for more information.

**Save On Acquisition->**

**Enable**

Enables the "Save On Acquisition" feature which allows acquired data to be exported, or a copy of the project to be saved when an acquisition completes.

**Settings...**

Opens a dialog which enables settings for the "Save On Acquisition" feature to be specified.

**View ->**

**Zoom In**

Decreases scale factor one step in a 1-2-5 sequence.

**Zoom Out**

Increases scale factor one step in a 1-2-5 sequence.

**Zoom All**

**Scroll to First Sample**

Scrolls waveforms and the state list to place the first sample at the display reference.

**Scroll to Trigger**

Scrolls waveforms and the state list to place the trigger sample at the display reference.

**Scroll to Last Sample**

Scrolls waveforms and the state list to place the last sample at the display reference.

**Scroll to Cursor**

Scrolls waveforms and the state list to place the selected cursor at the display reference.

**Help->**

**Contents...**

Opens the LogicPort help file.

**Help Language**

Selects the language for the LogicPort help file.

**Visit Website...**

Opens the system's default internet browser and navigates to the Intronix home page.

**Software Updates->**

**Check Now**

Accesses the Intronix website to determine whether a newer software build is available (requires internet access).

**Check at Startup**

When enabled, the Intronix website is accessed each time the software is started to determine whether there is a newer software build available (requires internet access). This is the easiest way to make sure that your software stays up to date with all of the latest features.

**About...**

Displays current information about the LogicPort software.

**[Waveforms/State List Tab menus](#)**

## Scrolling and Zooming

### Scrolling Waveforms and Cursors

There are many options for positioning waveforms, cursors, and state list data.

Waveforms can be scrolled by clicking within the waveform window using the left mouse button, then dragging them horizontally (rows can be dragged vertically in the state list). You can click and drag cursors using the same method.

You can also use the scrollbar at the bottom of the waveform window or right side of the state list. Dragging the scrollbar allows you to scroll across the entire range of the acquired data. Clicking the arrows at the ends of the scrollbar scrolls the display one pixel in the waveform window, or one row in the state list.

You can scroll to the next or previous transition by typing N or P on the keyboard. You can scroll to the next or previous transition occurring on a *specific* waveform by first right-clicking on that waveform, then choosing the appropriate option from the popup menu. See the [waveform tab menu descriptions](#) for more details.

You can use the Right/Left arrow keys to scroll within the waveform window. If you hold down one of these keys the scrolling accelerates.

You can use the Up/Down arrow keys or Page Up/Down keys to scroll vertically within the state list.

Cursors can be positioned by right-clicking in the desired cursor location and then choosing "Place Cursor" from the popup menu.

You can quickly scroll to a particular cursor or the trigger position by typing the appropriate letter A, B, C, D or T on the keyboard.

Scrolling of waveforms and cursors affects both the Waveforms tab and the State List tab simultaneously. Additional options for scrolling and positioning cursors are included in popup menus and the toolbar (right-click within the waveform window or state list table).

### Zooming

You can zoom in on a particular area of a waveform by left-clicking the area of interest while pressing the <Ctrl> key on the keyboard (this also adjusts the reference offset to center the area of interest within the display). Zoom out by right-clicking while pressing the <Ctrl> key. You can also zoom in/out by pressing the +/- keys on the keyboard, by clicking menu and toolbar items, or by using the mouse scroll wheel while pressing the <Shift> key. You can "Zoom All" from the View menu or toolbar. The Zoom All feature automatically sets the scale factor as required to fit all of the sample buffer within the waveform window, then adjusts the reference offset to center the data within the window. Zoom All will also move the cursors into view if they are beyond the buffer's extents.

### Cursor Snap

Cursor snap makes it easy to align cursors with transitions within the display. With cursor snap enabled, you can simply drag the cursor to a position near the transition of interest, then release the mouse button. When the mouse button is released, the cursor automatically moves to the nearest transition.

If the mouse button is released in a location above or below all of the displayed waveforms, it will snap to the nearest transition found on *any* waveform. If the mouse button is released on a particular waveform row, the cursor will snap to the nearest transition found on *that* waveform. Cursors will only snap to transitions visible within the display.

You can temporarily disable cursor snap by holding down the <Alt> key while dragging a cursor. Cursor snap does not apply when using the "Place Cursor" feature.

### Cursor Tracking

Cursor tracking allows you to move two or more cursors at once, while maintaining the relative position between them. Cursors can be assigned to track each other in any combination. For example, you could assign cursors B, C and D to track cursor A. When cursor A is moved, cursors B, C and D will follow. In this example if cursor B, C or D is moved, no other cursors will follow.

You could also assign all cursors to track each other. For example if you assign A to track B, B to track C, C to track D, D to track E, E to track F, and F to track A, all cursors are interlocked. When any cursor is moved, all other cursors will follow. You can quickly make this particular set of tracking assignments by selecting **Options>Cursor Tracking>Interlock All**.

You can temporarily disable cursor tracking for a particular cursor by holding down the <Shift> key while dragging the cursor. This will prevent other cursors from tracking the selected cursor. This allows you to adjust a cursor's position relative to others without having to remove their tracking assignments.



## About Wires, Signals, Groups and Interpreters

**Wires** are, as the name suggests, physical connections to the LogicPort. They have the fixed names D0 to D31, CLK1, and CLK2. These names are standardized to match the color-coded probe which is supplied with the LogicPort. Signal and ground wire colors are defined in the [Connector Pin Assignments](#) section of the appendix. It is important to understand that the ground wires have a [connection to the PC's chassis](#) through the USB cable.

Note that CLK1 and CLK2 have special functions *only* when sampling in state mode. When sampling in timing mode these inputs are normal sampled channels, and can be thought of as D32 and D33.

**Signals** can be thought of as aliases for wires. New projects start with a default set of signal names which you can edit as desired. Each wire has only one signal name associated with it.

**Groups** are collections of signals. A group may include any number of signals. Signals can be included in more than one group. You can skip wires when forming groups, but you cannot arrange them randomly. Here is an example of a legal group (wire names shown):

[D0, D1, D12, D13, D22]

You can also swap the MSB-LSB ordering of an entire group. The bit ordering can be changed by selecting the appropriate "Data Format", or by using the "Reverse Display Order" button in the group edit dialog. The following group is legal:

[D7, D6, D5, D1, D0]

The following group is NOT legal because the relative bit weights haven't been maintained:

[D0, D2, D1, D3, D4]

Note that the use of groups is entirely optional. All signals within a project can be displayed without using any groups.

**Interpreters** can be thought of as special groups. Interpreters are used to translate certain protocols and display the resulting high-level data. To learn more about interpreters see the [interpreter overview](#).

## Configurable Controls

The controls at the top of the Waveforms tab are configurable, allowing you to select which parameters you want to adjust at any given time. The following parameters may be selected from the drop-down list for each of the controls:

### Sample Rate/Period

Controls the timing mode sample rate (has no effect in state mode). Sample rate can be displayed as frequency or period as appropriate.

### Logic Threshold

Controls the input threshold level. This is used to set a threshold compatible with the logic family being tested. It can also be used to improve sampling of noisy or distorted signals. See also [Logic Sense Setup](#).

### Pre-Trigger Buffer

Specifies what portion of the sample buffer should be viewable before the trigger event. This setting has no effect on previously acquired data. Changes will take effect with the next acquisition. Note that when compression is enabled, buffer usage is calculated *after* sample data is compressed. This means the pre-trigger buffer may appear larger or smaller if compression is much more successful before or after the trigger event.

### Measurement Source

Specifies which channel the selected measurement will process. This setting has no effect if the selected measurement doesn't require a channel assignment (the Interval between cursors for example).

Once a parameter has been selected, its value can be changed using the associated slider control, or by selecting a specific value from the drop down list (drop down list is not available for some parameters).

## About Rows and Columns

The first row in the waveform table contains column headings. Right-clicking in this area opens a menu which enables columns to be added, removed, or changed. The first column on the left is always the "Signal" column. This column is reserved for the names of signal and group rows in the table. The following types of columns can be added to the table:

### Pattern

Displays a trigger pattern and allows it to be edited. Click in a cell to cycle through the possible values. Note that patterns A and B apply to trigger levels A and B respectively.

### Edge

Displays selected trigger edges and allows them to be edited. Click in a cell to cycle through the available edges. Note that edges A and B apply to trigger levels A and B respectively.

### Cursor

Shows data values at the cursor location.

### Wire ID

Shows wire name and color-code for each signal.

### Wire Status

Shows the current wire status for each signal. The inputs are constantly sampled at 250MHz regardless of acquisition mode, sample mode or rate. The wire status is updated approximately 10 times per second. A "T" indicates that the wire has been both high and low since the last update (it has Transitioned). If you are sampling at a rate of less than 250MHz, the wire status may indicate "T" when no transitions are apparent in the sampled data. This means there are pulses in the data which are too narrow to be reliably captured at the current sample rate. You may need to increase the sample rate if you are concerned with capturing these glitches.

Below the first row are signal, group and interpreter rows which have been added to the waveform table. You can add or edit rows by right-clicking in this area.

## Resizing Analog Waveforms

When a group row's waveform style is set to "Analog" you can resize the row by clicking and dragging the lower boundary of the cell containing the group's name.

## Expanding/Collapsing Groups and Interpreters

Each group and interpreter in the waveform table has a "+" or "-" symbol next to it. Clicking in the area of this symbol will expand/collapse the group or interpreter. You can also double-click a group or interpreter's name to toggle its expanded/collapsed state. When a group or interpreter is expanded, all of its signals are displayed below it.

## Rearranging the Waveform Table

You can rearrange columns by clicking and dragging them by their heading in the first row. You can rearrange rows by clicking and dragging them by their name in the first column (note this also rearranges columns in the state list).

## Waveforms Tab Menus

There are five context-sensitive menus which can be accessed by right-clicking in different areas of the waveform table. Column related menus are accessed from the top row only. The cell in which you right-click determines what row or column the menu items apply to. The following items can be found in the various waveform table menus (listed here in alphabetical order):

### **Add Column**

Adds a column of the selected type as the last column in the waveform table.

### **Add Group...**

Opens the group selection dialog. Adds the selected group(s) at the bottom of the waveform table. New groups can also be created from the group selection dialog.

### **Add Interpreter...**

Opens the interpreter selection dialog. Interpreters selected from this dialog will be added at the bottom of the waveform table. Interpreters can also be created, edited or removed from within this dialog.

### **Add Signal...**

Opens the signal selection dialog. Adds the selected signal(s) at the bottom of the waveform table.

### **Clear Edge**

Clears all previously selected edges for the associated trigger level.

### **Clear Pattern**

Sets all pattern bits for the associated trigger level to "X".

### **Collapse All**

Simultaneously collapses all groups and interpreters in the waveform table.

### **Data Format**

Specifies the radix and bit ordering for the selected group. The specified radix will be used when displaying group values in the waveform window, when displaying cursor values, and when specifying values in the trigger setup dialog. The "Signed" option causes the group's data to be interpreted as 2's complement signed data. This option only applies to decimal radix numbers and to analog waveforms. Binary and hex radix values are always shown without modification. The wire ordering options specify whether higher numbered wires appear in the left-most or right-most bits of group data.

### **Display Order**

Specifies whether higher numbered wires for the selected group will be displayed at the top or the bottom of the group when it's expanded. This has no effect on the group's data format.

### **Edit Group...**

Opens an edit dialog for the selected group. The group edit dialog allows renaming the group and editing the signal list and display order for the group. You can also edit signals from within the group edit dialog (right-click within either selection list).

### **Edit Interpreter...**

Opens a dialog which enables settings for a particular interpreter to be specified.

### **Edit Signal...**

Opens an edit dialog for the selected signal. The edit dialog allows renaming of other signals as well.

### **Expand All**

Simultaneously expands all groups and interpreters in the waveform table.

### **Insert Column**

Inserts a column of the selected type in the location where you right-clicked to open the menu. Existing columns are shifted to the right.

### **Insert Group...**

Opens the group selection dialog. The selected groups will be inserted wherever you right-clicked to open the menu. Existing signals and groups will be shifted down. Groups can also be Created, Edited, and Deleted from within the selection dialog (right-click within the selection list).

### **Insert Interpreter...**

Opens the interpreter selection dialog. The selected interpreter(s) will be inserted at the position where you right-clicked to open the menu. Interpreters can also be created, edited or removed from within this dialog.

#### **Insert Signal...**

Opens the signal selection dialog. The selected signal(s) will be inserted wherever you right-clicked to open the menu. Existing signals and groups will be shifted down.

#### **Remove All**

Removes all signals and groups from the waveform and state list windows.

#### **Remove Column**

Removes the selected column from the waveform table.

#### **Remove Group**

Removes the selected group from the display (but does not delete the group from the project).

#### **Remove Interpreter**

Removes the selected interpreter from the display (but does not remove it from the project).

#### **Remove Signal**

Removes the selected signal from the display.

#### **Set Column**

Changes the column type for the column you right-clicked to open the menu.

#### **Show Signal Column Only**

Temporarily hides all columns except the signal column to maximize the amount of data which can be viewed in the waveform window.

#### **Waveform Style**

Specifies whether the selected group's waveform will be a traditional digital "envelope" type display, or an analog representation of the data. Analog waveforms can be sized vertically by clicking and dragging the divider at the bottom of the group row. Analog waveforms can also be translated from 2's complement signed data (see "Data Format" above). If you hover the mouse cursor over a group waveform, the group's value at that location appears in a tooltip.

#### **Waveform Color**

Selects the waveform color for a particular row in the waveform window. Signals displayed within a group inherit the color of the group waveform..

**The waveform window has one context-sensitive menu containing the following items:**

#### **Place Cursor**

Places the selected cursor at the position where you right-click to open the menu.

#### **Scroll to Next Transition**

Scrolls to the next transition found in the acquired data. If you clicked within a particular row to open the menu, the reference position will be scrolled to the next transition found on *that* particular waveform, starting from the reference position. If you clicked above or below all displayed rows to open the menu, the reference position will be scrolled to the next transition found on *any* displayed waveform, starting from the reference position.

#### **Scroll to Previous Transition**

Scrolls to the most recent previous transition found in the acquired data. If you clicked within a particular row to open the menu, the reference position will be scrolled to the previous transition found on *that* particular waveform, starting from the reference position. If you clicked above or below all displayed rows to open the menu, the reference position will be scrolled to the previous transition found on *any* displayed waveform, starting from the reference position.

#### **Scroll to First Sample**

Scrolls waveforms and the state list to place the first sample at the display reference.

#### **Scroll to Trigger**

Scrolls waveforms and the state list to place the trigger sample at the display reference.

#### **Scroll to Last Sample**

Scrolls waveforms and the state list to place the last sample at the display reference.

#### **Scroll to Cursor**

Scrolls waveforms and the state list to place the selected cursor at the display reference.

## State List Tab

The state list tab can be used to display data acquired in either state *or* timing mode. The state list shows data in a tabular format. Displayed data can be condensed to show only unique states, or can be expanded to show all samples. In either case, the data is always accompanied by a time stamp which can be set to indicate time or samples.

When set to indicate time, the time stamp's scale is based upon the acquisition rate. If the data was acquired in timing mode, the sample rate at the time of acquisition will be used. If the data was acquired in state mode, the clock rate currently specified in the "Sample Mode Setup" dialog is used.

The time stamp can be set to indicate relative to the trigger event or relative to the reference position as desired.

When the "State List Format" is set to "Unique States", duplicate samples are not displayed. This allows much more data to be displayed, and in many cases makes the data easier to interpret. This feature can be especially handy when used with interpreters because it allows the state list to summarize bus transactions spanning a very large time interval.

When the "State List Format" is set to "All Samples" the state list shows data one sample per row, even if this means filling the list with identical samples.

You can scroll the state list data and cursors by clicking and dragging them vertically. You can also scroll the data using the <Page Up> and <Page Down> keys on the keyboard or with the scrollbar at the right-hand edge of the window. You can rearrange the columns in the state list by dragging their name (note this also rearranges rows in the waveform table).

There are four context-sensitive menus available in the state list. Most of them are accessed from the top row. Menu items for the State List tab are a subset of the items available in the [Waveforms Tab menus](#).

## Notes Tab

This is just a scratch-pad area where you can enter any notes you like, such as a description of how the sampled waveforms occurred. The notes are stored in the project file, along with sampled data and settings.

## Logic Sense Setup

By default, the LogicPort uses Positive logic when evaluating its inputs. That is, when the input voltage is higher than the selected threshold voltage, it is interpreted as a logical 1 or TRUE. If the input voltage is below the threshold voltage, it is interpreted as a logical 0 or FALSE.

The "Logic Sense Setup" dialog enables you to invert one or more of the LogicPort's input channels. This option is useful when working with systems which use Negative logic, and in cases where you are only able to connect to an inverted version of a signal. This option allows the trigger magnitude comparators to be used with Negative logic. This would not otherwise be possible since the magnitude of a group is not mathematically comparable to its bitwise inverted value.

Inverting a channel has the effect of placing an infinitely fast NOT gate between your hardware and the LogicPort's input. This includes state mode clock inputs and qualifiers.

When a channel is inverted, the signal will be interpreted as a logical 0 or FALSE when the input voltage is *higher* than the threshold voltage. This inversion occurs in hardware, *before* the data is sampled. **Once the data has been acquired, its logic sense cannot be changed.** Changes to logic sense settings have no effect until the next acquisition. Data is always displayed with the logic sense that was in effect at the time of acquisition.

Note that *all* settings and indications apply *after* inversion. For example, if you invert a channel and then specify that the LogicPort should trigger on the rising edge of that signal, it should be no surprise that the trigger will actually occur on the *falling* edge of the applied signal. Because the signal has been inverted before sampling, the displayed data will show a rising edge as the trigger event.

## Sample Mode Setup

This dialog enables you to choose between timing mode and state mode. You can specify other sample related parameters as well. Click OK or click the "Sample Mode Setup" button in the toolbar to close the dialog. The Apply button applies changes without closing the dialog. You must click Apply or close the dialog before new settings take effect.

### Timing Mode (Internal Sample Clock)

When timing mode is chosen, the LogicPort will sample at a rate specified using one of the configurable controls on the Waveforms tab.

### State Mode (External Sample Clock)

When state mode is chosen, you can select which clock input will be used for sampling, which edge of the clock should be used, and what timing relationship exists between the chosen clock edge and the data you wish to sample. You can also enable qualified sampling to allow the clock input to be qualified by another signal.

Usually you will want to sample data *on* the chosen edge. If the clock signal from the hardware under test is skewed relative to the data, you may want to sample *before* or *after* the chosen clock edge. For example, if the clock signal is obtained from the output of a clock distribution buffer which delays the clock signal, you might want the LogicPort to sample before the clock edge to compensate for this delay. If the data is only valid for a very short time before the clock edge you might need to sample after the clock edge to make sure the LogicPort's setup time requirement is satisfied. If the hardware under test has an excessive amount of clock skew, it may be helpful to sample on the opposite edge. This has the effect of shifting the sample time by an amount equal to one half the clock period (assuming the clock signal is a square wave).

You can enable qualified state mode sampling by clicking in the check-box to select **Sample only when qualifier CLK<sub>n</sub> is**. When qualified state mode sampling is enabled, samples will only be collected when the specified clock edge occurs *while* the qualifier is in its active state. Clock edges occurring while the qualifier is not active will be ignored (except for trigger duration measurements). The state mode clock qualifier will always be the clock input which is not used as the state mode sample clock. For example, if state mode is setup to sample on the rising edge of **CLK1**, then **CLK2** becomes the qualifier input. The clock qualifier's "active" state is specified by selecting **High** or **Low** from the drop down box. Note that the qualifier input always has the same setup/hold requirements as the data inputs.

It is important to have a stable clock for state mode sampling. You can monitor the frequency and stability of the state clock using the LogicPort's internal frequency counter. If the frequency indication is unstable, there may be noise on the clock, or the clock edges may be distorted. The LogicPort's input threshold can be adjusted to eliminate some types of noise from clock signals.

You should specify the rate of the state clock in the state mode settings window. This allows the LogicPort software to accurately calculate displayed time intervals and to set the graticule scale properly. Alternatively, you can select "Show Samples" in the "Options" menu.

## Sample Sequence

The sample sequence consists of three phases: Pre-Fill, Trigger, and Post-Fill. During the Pre-Fill phase the sample buffer is being filled to the level specified by the "Pre-Trigger Buffer" setting. Once buffer prefill is complete, the Trigger phase is entered. During this phase the hardware continues to sample and fill the buffer while waiting for the specified trigger conditions to occur. Because the specified trigger conditions may not occur for some time, the sample buffer may fill and wrap around multiple times while waiting for the trigger event. Once trigger occurs, the Post-Fill phase is entered. During post-fill, the hardware continues to sample and fill any remaining buffer space. If necessary the buffer wraps around, overwriting the oldest data until only the specified amount of pre-trigger data remains. At this point sampling stops and the data is uploaded and displayed by the LogicPort software.

### Trigger Immediate

The current phase of the sample sequence is indicated in Analyzer Status panel of the [status bar](#) at the bottom left corner of the display. You can use the Trigger Immediate function to exit any of the three phases early. For example, during the Pre-Fill phase the Trigger Immediate function will exit the Pre-Fill phase and advance to the Trigger phase. During the Trigger phase, the Trigger Immediate function will force an immediate trigger and advance to the Post-Fill phase. During the Post-Fill phase, the Trigger Immediate function will terminate sampling and cause the acquisition to be completed and displayed.

Note that the Trigger Immediate function is "blacked out" for a short period each time the hardware advances the sample phase. This is to avoid accidentally advancing the sample phase twice if your mouse click happens to coincide with a hardware event.

## Compression

Sample compression is available both for timing mode and state mode. Compression is enabled from the sample mode setup dialog. When compression is enabled, the sample buffer can hold more data. The amount of data which can be compressed into the buffer is

Inversely proportional to the rate of transitions in the sampled data. That is, with fewer transitions in a given period, more samples will be collected. In extreme cases where there are very few transitions (intermittent serial data for instance), it could take hours to fill the sample buffer. This is where the “Pre-Fill Time Limit” and “Post-Fill Time Limit” become useful. These settings allow you to specify time limits for each phase of the acquisition. These limits are approximate because they are based on the PC’s clock.

### **Pre-Fill Time Limit**

Specifies the maximum amount of time you are willing to wait for the pre-trigger buffer to fill. When the time limit is reached, the LogicPort’s trigger phase will be entered regardless of how many samples are in the pre-trigger buffer. Note that this time limit does not *force* a trigger. It only causes the buffer pre-fill phase to be exited, allowing the LogicPort’s trigger logic to *begin* searching for the specified trigger conditions. This setting can effectively override the “Pre-Trigger Buffer” value set on the Waveforms tab if the specified timeout occurs before pre-fill is complete. You can also use the “Trigger Immediate” function to terminate buffer pre-fill early.

### **Post-Fill Time Limit**

Specifies the maximum amount of time you are willing to wait for the acquisition to complete *after* trigger occurs. When the time limit is reached, the acquisition will complete and the sampled data will be displayed regardless of the number of samples in the sample buffer. This setting can effectively override the “Pre-Trigger Buffer” value set on the Waveforms tab when timeout occurs before post-fill is complete. You can also use the “Trigger Immediate” function to terminate buffer post-fill early and complete the acquisition.

The recommended default for both settings is 0.5 second. Specifying a time limit of “None” means never time-out no matter how long it takes. It’s important to understand that trigger events occurring during the pre-fill phase will *not* trigger an acquisition because the trigger logic is not armed until the buffer pre-fill phase completes.

Note that the total acquisition time can be longer than the sum of the pre-fill and post-fill time limits. For example, if the pre-fill time lapses in 1 second, but trigger doesn’t occur until 3 seconds later, and then the post-fill time lapses in 1 second, the total acquisition time could be as much as 5 seconds (depending upon the data). You can use the “Trigger Immediate” feature to *force* a trigger if the expected trigger condition doesn’t occur for a longer than expected period.

There is an exception to the acquisition time limits which can occur in state mode. When sampling in state mode, the LogicPort’s sample and trigger logic are only active when the state mode sample clock is toggling. If the sample clock provided by the hardware under test stops toggling, the LogicPort cannot complete an acquisition regardless of the specified time limits. This is because no data can be acquired without a sample clock. This exception doesn’t apply to timing mode.

### **Compression vs Pre-Trigger Buffer setting**

The portion of the sample buffer specified by the “Pre-Trigger Buffer” setting refers to memory locations *not* samples. Without compression these values would always be equal. With compression enabled, any pair of memory locations could represent as few as 2 or as many as  $2^{34}$  (17,179,869,184) samples. If signal activity before the trigger event is less than signal activity after the trigger event, the pre-trigger buffer may appear larger than specified. Since pre-trigger signal activity is less, more samples are compressed into each memory location of the pre-trigger buffer. Likewise if signal activity is higher before the trigger event, the pre-trigger buffer may appear smaller than specified. You can use the “Transitions” measurement function to get an idea of how active individual signals were before or after the trigger event.

## Trigger Setup

This dialog enables you to specify the conditions under which the LogicPort will trigger and complete an acquisition. Click OK or click the "Trigger Setup" button in the toolbar to close the dialog. The Apply button applies changes without closing the dialog. Clicking Apply also causes all trigger parameters to be validated. For example if you enter an occurrence count or duration value which is out of range, the value will be corrected when you click Apply. You must click Apply or close the dialog before new settings take effect.

### Defining Patterns and Edges

Patterns and edges can be defined and edited at any time before, during or after they are included as terms in the Trigger Setup dialog. Patterns and edges are defined from within the waveform table on the Waveforms tab. You can assign individual bits of a pattern or edge by clicking on a cell which is intersected by the desired signal row and the desired pattern or edge column. As you continue to click on one of these cells, the assignment for the associated signal will cycle through all possible conditions. To make assignments to signals within a group, you must first expand the group. You can do this by clicking on the "+" sign next to its name. When a black dot appears in a group's associated pattern or edge column it indicates that one or more signals within that group contain assignments.

Note that you cannot use the state mode clock signal in trigger settings when sampling in state mode. This is because in state mode the state clock is essentially sampling "itself" on just one of its two edges. This means that the state clock waveform will not be reproduced in the captured data. The state clock signal *can* be used in trigger settings when sampling in timing mode because it functions as a normal sampled channel in timing mode.

Please see [Sample Mode Setup](#) for information regarding the acquisition cycle and sequencing related to trigger events.

### Choosing Trigger Levels

The "Trigger" drop-down list at the top of the Trigger Setup dialog specifies which trigger levels (if any) are active, and how they should be logically combined. When a trigger level is not active, its "terms" window will be grayed out. There are six possible combinations of the two trigger levels. They are:

#### Immediately when acquisition starts

Trigger will occur immediately with no conditions. This is similar to clicking the "Trigger Immediate" button on the toolbar. This mode is much like the "Auto" trigger mode available on most oscilloscopes.

#### When Level A is satisfied

Trigger will occur when all selected terms for level A have been satisfied.

#### When Level B is satisfied

Trigger will occur when all selected terms for level B have been satisfied.

#### When either level A or B is satisfied

Trigger will occur when all selected terms for level A have been satisfied, or when all selected terms for level B have been satisfied. This does not logically OR terms between the two levels. It only specifies that level A or level B can cause a trigger.

#### When levels A and B have been satisfied in any order

Trigger will occur once all selected terms for level A have been satisfied, and all selected terms for level B have been satisfied. Either level can be satisfied first. This does not logically AND terms between the two levels, and the two levels do not need to be satisfied simultaneously.

#### When level A is satisfied and then level B is satisfied

Trigger will occur once all selected terms for level A have been satisfied, and then all selected terms for level B have been satisfied. Level B will not be armed until Level A is satisfied. This does not logically AND terms between the two levels, and the two levels do not need to be satisfied simultaneously.

### Trigger Terms

Each trigger level has a set of terms which may be selected by clicking in the check-box next to it. Terms which are not selected are grayed-out indicating that they are not part of the trigger scheme. Some terms are mutually exclusive. For example the pattern "Entered" condition is not compatible with the edge term (you would normally use the pattern "True" condition with the edge term instead). When a selected term precludes the use of another term, the unavailable term will have its check-box grayed-out. If no terms are selected, the trigger level is unconditionally satisfied. The following terms are available for each trigger level:

#### Edge

This term is satisfied when the selected edges occur the specified number of times. Edges are selected in the waveform table. Multiple edges may be selected. If multiple edges are selected, they will be logically OR'd. Multiple edges occurring within one sample period will be counted as a single occurrence. Note that if this term is enabled but no edges are selected in the waveform window, trigger cannot occur.

### **Pattern**

This term is satisfied when the sampled data meets the associated pattern's specified conditions. Patterns are specified in the waveform table. Note that if all bits of the associated pattern are set to "X", and "Prequalify Pattern/Value Terms" is selected, trigger cannot occur since the pattern term cannot be false.

### **Value**

This term is satisfied when the value of the sampled data for the selected group is within the specified range. The group can be selected by clicking in the text box. The data format for the group can be specified by right-clicking one of the specified values (note this also changes the data format in the waveform table and state list).

### **Duration**

This term is satisfied when all of the other selected terms have been satisfied for the specified duration. Note that the duration can be specified in samples or seconds. The range is 1 to 1,048,575 samples (or the equivalent time period at the current sample rate). Scientific notation can be used to enter these values. When sampling at 500MHz, duration values are constrained to an even number of samples, and there is a +/-1 sample uncertainty for each value because the trigger generator is sampling at its maximum of 250MHz.

## **Consecutive vs Cumulative**

"Consecutive" means each occurrence must be qualified or the counter resets and begins counting again from zero. For example, consider the following trigger scheme:

"Edge A occurs 3 time(s) Consecutive while Pattern A is True"

If the second occurrence of edge A happens while pattern A is false, then the occurrence counter will reset. There would have to be three more consecutive edges with pattern A true before a trigger would occur. It's OK if pattern A goes false between edges as long as it is true each time edge A occurs.

"Cumulative" means every qualified occurrence is counted, even if there are unqualified occurrences in-between. For the given example, the second occurrence of edge A (with pattern A false) would not increment the counter, but it would not reset the counter either.

## **Prequalify Pattern/Value Terms**

When selected, at least one of the enabled "Pattern" or "Value" terms of each active trigger level must be false before the associated trigger level is armed. For example, consider the following trigger scheme:

"Edge A occurs 1 time(s) Cumulative while Pattern A is True"

With "Prequalify Pattern/Value Terms" selected, trigger level A will not be armed until pattern A is false. If pattern A is false when acquisition is enabled, then trigger level A is armed immediately, and trigger will occur as soon as edge A occurs with pattern A true. In contrast, if pattern A is *true* when acquisition is enabled, then trigger level A will not be armed until a transition from true to false occurs for pattern A.

This feature is useful when you want to make sure that the hardware under test is not in the middle of a sequence of interest when acquisition is enabled. For example, consider the following trigger scheme:

"Edge A occurs 5 time(s) Consecutive while Pattern A is True"

Say once per second pattern A goes true for 100 occurrences of edge A. Say you have chosen this trigger scheme because you want to trigger on the 5<sup>th</sup> occurrence of edge A at the beginning of one of these cycles. If you happen to enable acquisition when the hardware is in the middle of a cycle on the 50<sup>th</sup> occurrence of edge A, the trigger would occur on the 55<sup>th</sup> occurrence of edge A. To prevent this, you can enable "Prequalify Pattern/Value Terms". Since trigger level A won't arm until pattern A is false, the trigger can now only occur on the 5<sup>th</sup> occurrence of edge A at the *beginning* of a full cycle.

Use this advanced feature with caution. Many times it doesn't matter if a pattern or value term is already true when acquisition starts. This feature can be a great benefit in certain trigger schemes, but it can be a source of frustration if enabled when it shouldn't be.

When in doubt you should disable (uncheck) the "Prequalify Pattern/Value Terms" option.

## **Constrained Settings**

Each trigger level has a 20 bit counter which can be used to count edges, pattern entry/exit, or duration. Each counter has a range of 1 to 1,048,576. Depending upon how the counter is used, the allowed range may vary by one count at either limit. If you enter a trigger setting which is outside the range of the counter, the value will be adjusted to a number within the counter's limits, and you will get a message in the status panel indicating that the trigger setting is "constrained". For example, If you are sampling at 10MHz, the maximum "Greater than" duration range is 1,048,575 x 100 nanoseconds, or 104.8575 milliseconds. If you enter a duration setting of

200 milliseconds, it will be adjusted to 104.8573 milliseconds, and displayed as a (rounded) value of 104.9 milliseconds. You will then get a message indicating that the setting is constrained. This can also happen for very small duration values. For example, if you are sampling at 1KHz, and enter a "Greater than" duration setting of 100 microseconds, the value will be constrained to 1.000 milliseconds. The minimum "Greater than" duration setting is 1 sample, which is a period of 1 millisecond for a 1KHz sample rate.

When using the trigger counters for duration measurements the software has to recalculate the appropriate counter values as you adjust the sample rate. If you change the sample rate and a duration value within +/-2% of the specified value cannot be achieved, you will get a status indication that the value is constrained. The 2% tolerance applies only to duration measurements. If you specify the duration value as an integer multiple of the sample period, then the tolerance is zero +/- the timebase accuracy of 50ppm. When the counters are used to measure edge or pattern occurrences, the tolerance is zero.

## Measurement Setup

This dialog enables you to choose the four measurements which will be displayed in the [status bar](#). Selected channels are identified by their wire names instead of signal names to keep the display format constant. The following measurement types are available:

### Frequency From Hardware Counter

Measures and displays the frequency of a selected channel. The frequency is measured and updated in real-time using the LogicPort's built in hardware frequency counter. Select the channel using [configurable controls](#) in the Waveforms tab.

### Period From Hardware Counter

Measures and displays the period of a selected channel. The period is measured and updated in real-time using the LogicPort's built in hardware frequency counter. Select the channel using [configurable controls](#) in the Waveforms tab.

### Interval <X> to <Y>

Measures and displays the interval between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. The interval will be displayed as time or samples depending on which format is selected in the "Options" menu.

### Rate <X> to <Y>

Measures and displays  $(1 / \text{Interval})$  between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. The interval will be displayed as a frequency based on the acquired sample period in timing mode, or based on the specified sample period in state mode. Note that since rate relates to time, it is not available if "Show Samples" is selected in the "Options" menu.

### Transitions <X> to <Y>

Measures and displays the number of transitions on a selected channel between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. Rising and falling edges both count as transitions. Select the channel using [configurable controls](#) in the Waveforms tab.

### Cycles <X> to <Y>

Measures and displays  $\text{integer}(\text{transitions} / 2)$  for a selected channel between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. Rising and falling edges both count as transitions. Any two transitions count as a complete cycle. Select the channel using [configurable controls](#) in the Waveforms tab.

### Duty <X> to <Y>

Measures and displays the Duty Cycle (fraction of time which the signal is high) for cycles found on a selected channel between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. The software will look for the first rising edge on the specified channel as it searches from the first reference toward the second reference. It will also look for a rising edge as it searches from the second reference toward the first reference. The software then searches the data between these two edges and calculates the duty cycle. If more than one cycle is found then the indicated duty cycle will be the average for all cycles. The duty cycle will always be a value between 0 and 1 unless no complete cycles are found, in which case it will indicate N/A. Select the channel using [configurable controls](#) in the Waveforms tab.

### 1-Duty <X> to <Y>

This is simply one minus the duty cycle (see Duty above). It is a measure of the fraction of time which a signal is low. The value will always be between 0 and 1 unless no complete cycles are found, in which case it will indicate N/A.

### +Width <X> to <Y>

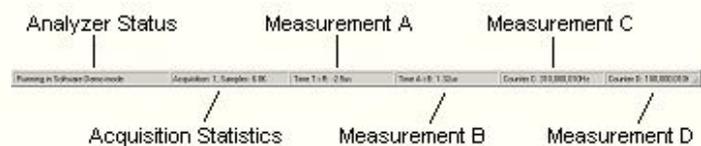
Measures and displays the width of positive going pulses found on a selected channel between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. The software will look for the first rising edge on the specified channel as it searches from the first reference toward the second reference. It will also look for a falling edge as it searches from the second reference toward the first reference. The software then searches the data between these two edges and calculates the pulse width. If more than one pulse is found then the indicated pulse width will be the average for all pulses. The pulse width will be displayed as time or samples depending on which format is selected in the "Options" menu. If no pulses are found then the indication will be N/A. Select the channel using [configurable controls](#) in the Waveforms tab.

### -Width <X> to <Y>

Measures and displays the width of negative going pulses found on a selected channel between any two of the following reference points: Trigger sample, Reference position, Cursors A, B, C, D, E, F. The software will look for the first falling edge on the specified channel as it searches from the first reference toward the second reference. It will also look for a rising edge as it searches from the second reference toward the first reference. The software then searches the data between these two edges and calculates the pulse width. If more than one pulse is found then the indicated pulse width will be the average for all pulses. The pulse width will be displayed as time or samples depending on which format is selected in the "Options" menu. If no pulses are found then the indication will be N/A. Select the channel using [configurable controls](#) in the Waveforms tab.

## Status Bar

The status bar is located at the bottom of the application window. The following items are displayed in individual panels:



## Interpreter Overview

Interpreters enable the LogicPort software to post-process acquired data and then display the translated result as high-level information. For example, RS232 data can be interpreted to show ASCII values, and I<sup>2</sup>C bus data can be interpreted to show bus transactions along with the associated address and data values.

An interpreter can be thought of as a special type of [group](#). These special groups have specific requirements and capabilities. Interpreters are created, edited and displayed in a fashion very similar to normal groups. Like normal groups, interpreters can be expanded to show their constituent signals. This feature makes it possible to simultaneously show the acquired waveforms along with interpreted results.

To instantiate an instance of an interpreter, follow these steps:

1. Right-click on the "Signal" cell in the waveform table or the "Relative to ..." cell in the state list.
2. From the popup menu select "Add Interpreter..."
3. From the "Select Interpreter(s)" dialog, click to select one of the existing interpreter instances or click "Create..." to create a new one.
4. Once a new or existing interpreter has been selected, click OK to close the dialog and add the interpreter to the display.

Once an interpreter has been added to the display, you can edit its settings at any time by right-clicking on its name, then selecting "Edit Interpreter..." from the popup menu.

**Tip:** To clone an existing interpreter, simply right-click on a selected interpreter from within either the "Edit Interpreters" or "Select Interpreters" dialog, then choose "Copy Selected...".

Multiple instances of each type of interpreter can be created and displayed. This enables the software to interpret multiple channels of a particular protocol. For example, with the LogicPort's 34 channels you could acquire and simultaneously interpret up to 17 separate I<sup>2</sup>C buses. The LogicPort can monitor multiple channels of I<sup>2</sup>C, SPI and RS232 data simultaneously. Be sure to use meaningful names for your interpreters. For example, if you are using three instances of the I<sup>2</sup>C interpreter, you could name them something like I<sup>2</sup>C-U3, I<sup>2</sup>C-U21, and I<sup>2</sup>C-U27.

Note that interpreter settings are specific to each interpreter instance. For example, if you have two instances of the I<sup>2</sup>C interpreter named I<sup>2</sup>C-1 and I<sup>2</sup>C-2, editing the settings for I<sup>2</sup>C-1 will have no effect on I<sup>2</sup>C-2.

You can conveniently create and edit multiple interpreters by opening the "Edit Interpreters" dialog using **Setup>Interpreters...** . Right-click within the selection window to view more options. Interpreters can also be edited from the "Select Interpreter(s)" dialog, which is accessible from context-sensitive menus within the waveform and state list tabs.

Since interpreters are post-processors of acquired data, it is possible to add an interpreter to a project before *or after* the data has been acquired. It should be apparent from this fact that interpreter settings have no effect on data acquisition or triggering. Interpreter settings are used exclusively to post-process and correctly display the interpreted data.

Some interpreted data can be displayed as ASCII coded characters. When ASCII is specified as the data format, codes 33 through 126 will be displayed as characters. In addition, the following codes will be displayed using the listed mnemonics:

9	Tab	TB
10	Line Feed	LF
13	Carriage Return	CR
32	Space	SP

All remaining codes will be displayed as three digit decimal values.

Note that the State List becomes especially useful when interpreters are being utilized. Setting the "State List Format" to "Unique States" will often allow convenient display of much more data than would be possible in the waveform window.

## I<sup>2</sup>C Bus Interpreter

The I<sup>2</sup>C bus interpreter post-processes acquired data and displays the resulting transactions as defined by the Philips Inter-Integrated Circuit (I<sup>2</sup>C) bus standard.

Note: If you have not yet reviewed the [Interpreter Overview](#), please be sure to read that section before trying to setup interpreters.

The following settings apply to I<sup>2</sup>C bus interpreters:

### Name

Specify a meaningful name for this particular instance of the interpreter. This name will be displayed in the waveform and state list displays.

### SDA Signal

Specify the channel which is connected to the SDA signal in your hardware. Note that from within the selection dialog you can also right-click to edit signal names.

### SCL Signal

Specify the channel which is connected to the SCL signal in your hardware.

### Glitch Filtering

Specify a glitch filter. The available filters correspond to the internal filters present in various speed-grade I<sup>2</sup>C devices. Note that glitch filtering applies only to interpreter results. It does not modify the acquired waveforms.

### Data Format

Specify a format for the display of raw data transactions.

### Interpret Command Bytes

If enabled, commands are translated to high level transactions as defined in the I<sup>2</sup>C bus specification. If not enabled, commands are shown as raw numeric values for troubleshooting purposes.

### Command Format

Specify a display format for the data portion of interpreted command bytes.

**Frame Synchronization** is the method by which the I<sup>2</sup>C bus interpreter recognizes the beginning of the first complete bus transaction. There are two framing methods available:

### I<sup>2</sup>C Start Condition

This is the preferred synchronization method. It should be used when the initial I<sup>2</sup>C start condition is contained within the LogicPort's acquired data, or if there is a repeat I<sup>2</sup>C start condition present before the data of interest. When this method is used, the interpreter will automatically find the first I<sup>2</sup>C start condition on the bus and use it for frame synchronization just as I<sup>2</sup>C hardware would do.

### Cursor Position

This method is useful if the acquisition began after the initial I<sup>2</sup>C start condition had already occurred. This method allows you to "show" the interpreter the beginning of a valid data byte using a cursor. From this initial starting point the interpreter will automatically track all remaining bus transactions.

It is important to have an adequate sample rate for the I<sup>2</sup>C bus interpreter to function properly. A sample rate of 20MHz is normally high enough for all I<sup>2</sup>C devices. If you suspect that your hardware is malfunctioning because of clock glitches, you might want to increase the sample rate and turn off the interpreter's glitch filter. This will cause the interpreter to be more critical of the signals, and may help find the problem. Keep in mind that most I<sup>2</sup>C devices do contain some form of glitch filtering.

When commands are interpreted, they are shown as mnemonics in the waveform and state list displays. If you hover the mouse cursor over a displayed mnemonic, a more detailed description is shown. The following is a complete list of mnemonics:

<b>S</b>	<b>Start</b>
<b>Sr</b>	<b>Start (repeated)</b>
<b>P</b>	<b>Stop</b>
<b>A</b>	<b>Acknowledge</b>
<b>N</b>	<b>Not Acknowledge</b>
<b>W</b>	<b>Write with 7 LSBs of address</b>
<b>R</b>	<b>Read with 7 LSBs of address</b>
<b>Wt</b>	<b>Write with 3 MSBs of a 10 bit address</b>
<b>Rt</b>	<b>Read with 3 MSBs of a 10 bit address</b>

<b>GC</b>	<b>General Call</b>
<b>HW</b>	<b>Hardware General Call</b>
<b>SB</b>	<b>Start Byte</b>
<b>HS</b>	<b>High Speed Mode</b>
<b>RP</b>	<b>Reset and Program command</b>
<b>PR</b>	<b>Program command</b>
<b>CB</b>	<b>CBUS Transaction</b>
<b>RV</b>	<b>Reserved Value</b>
<b>IV</b>	<b>Illegal Value</b>

Please refer to the Philips I<sup>2</sup>C bus specification for more information.

## Synchronous Serial / SPI Interpreter

The Synchronous Serial/SPI interpreter post-processes acquired data and displays the resulting transactions in a user-defined format. The default settings for this interpreter are targeted for SPI analysis. In addition to SPI, this interpreter has the flexibility to interpret Microwire, 3-Wire, PS/2 Interfaces and other synchronous serial protocols. It can also be used for general purpose serial to parallel conversion.

Note: If you have not yet reviewed the [Interpreter Overview](#), please be sure to read that section before trying to setup interpreters.

The following settings apply to Synchronous Serial/SPI interpreters:

### Name

Specify a meaningful name for this particular instance of the interpreter. This name will be displayed in the waveform and state list displays.

### Data Signal

Specify the channel which is connected to the Serial Data signal in your hardware. For SPI, this would be MOSI or MISO. Note that from within the selection dialog you can also right-click to edit signal names.

### Clock Signal

Specify the channel which is connected to the Serial Clock signal in your hardware. For SPI, this would be SCK.

### Enable Signal

Specify the channel which is connected to the Enable signal in your hardware. For SPI, this would be SS. If no enable signal is available, or if it is not used, see the next setting below.

### Enable is <Active High/Low / Not Used>

Specify the active state for your Enable signal. For SPI this should be "Active Low". If no enable signal is available, specify "Not Used" for this setting.

### Data is transferred <LSB/MSB First>

Specify whether your data is transferred LSB or MSB first. SPI data is typically transferred MSB first.

### Interpret <nn> bits as <format>

Specify the number of bits which should be interpreted for each displayed data value, along with the desired data format. The range is 1 to 96 bits. Note that longer data streams can be interpreted as individual bytes by specifying a length of 8 bits.

### Frame Synchronization Method

Specify the frame synchronization method (see detailed description below).

### Data Shift Offsets

Allow position offsets to be specified for the serial data (see detailed description below).

### Mode

Specify the sample mode. Sample mode numbers correspond to the SPI specification. The SPI controller register bit settings are shown in parenthesis for reference.

### Glitch Filter

When glitch filtering is enabled, the interpreter can ignore glitches on the clock signal which are narrower than the specified minimum.

### Glitch Filter Minimum Duration

Specify the minimum clock pulse width which should be considered a valid clock pulse by the glitch filter.

The Motorola SPI standard is a formal description for what is otherwise, a generic synchronous serial interface. The SPI standard describes all possible sample clock phase/polarity relationships for SCK. The SPI standard specifies the enable signal SS as active low. It also specifies that data will be sent MSB first. These are the default settings for the LogicPort's Synchronous Serial/SPI interpreter.

This interpreter offers the ability to specify the enable signal as being active high or low, and the bit ordering as MSB or LSB first. It also allows frame offsets to be applied to sampled data. As a result, it can be used for a variety of synchronous serial protocols as well as general purpose serial to parallel conversion. For example, if your hardware contains a 4094 shift register you can monitor its Data, Clock, and Strobe inputs. The interpreter can convert the serially shifted data into its corresponding 8 bit values.

It is important to have an adequate sample rate for the Synchronous Serial/SPI interpreter to function properly. A sample rate of at least

4 times the shift clock rate is recommended. If you suspect that your hardware is malfunctioning because of clock glitches, you might want to increase the sample rate and adjust or turn off the interpreter's glitch filter. This will cause the interpreter to be more critical of the signals, and may help find the problem.

**Glitch Filtering** allows the interpreter to "ignore" clock pulses which are too narrow. Note that the glitch filter looks at both the positive *and* negative half-cycles to determine whether or not a clock cycle is valid. The recommended minimum duration setting is 10% of the hardware's clock period. If your hardware's clock has a very low *or* very high duty cycle, you may need to specify a shorter minimum duration.

**Frame Synchronization** is the method by which the Synchronous Serial/SPI interpreter recognizes the beginning of a shift sequence. This is important because if acquisition happens to begin while a shift sequence is already in progress, some bits will have already been missed. Without some method of frame synchronization the interpreted data would be shifted and incorrect. There are three frame synchronization methods plus one additional qualifier available:

#### **Enable Signal enters active state**

This tells the interpreter that a new frame begins each time the enable signal transitions to the active state. This is the preferred method when available. This method can't always be used because some hardware sets the enable signal permanently active, or toggles it high and low during partial frames.

#### **Clock Break**

This tells the interpreter that a new frame begins with the first clock pulse after the clock signal has been inactive for at least the specified period. This method can only be used if your hardware gates the shift clock between frames. Some hardware uses a continuous shift clock with no breaks.

#### **Cursor position**

In the unlikely event that neither of the other methods can be used, you can use a cursor to "show" the interpreter where a new frame begins.

#### **Start bit**

The interpreter can be set to search for a start bit signifying the beginning of each frame. This setting is not exclusive of, and can be applied in addition to the other three methods. When enabled, *each* interpreted data value must begin with the specified start bit (not just the first frame). Unlike asynchronous protocols, these synchronous start bits only need to be valid *on* the specified clock edge. They do not have any other associated timing requirements. This option is not normally applicable to SPI data.

In most cases frame synchronization is only needed to find the beginning of the *first* frame. The exceptions are protocols which use a start bit instead of an enable signal to frame their data (such as the PS/2 keyboard interface). No interpreted data will be displayed until a valid frame synchronization event has been found. From that point forward, the interpreter can remain synchronized according to the specified shift format. If additional frame synchronization events are available, the interpreter will take advantage of them when possible. These additional frame synchronization events are helpful when the hardware under test is malfunctioning (sending the wrong number of bits for example).

**Data Shift Offsets** allow the interpreter to discard up to 999 bits before and/or after interpreting a data value. For example, if you are analyzing an SPI interface which has two daisy-chained parts, an 8 bit part followed by a 12 bit part, you can create an instance which will interpret only the data targeted to the 8 bit part, and a separate instance which interprets only data targeted to the 12 bit part. This eliminates the need to mentally separate the data for the two parts from a 20 bit interpreted value. For this particular example, you would set the 8 bit interpreter to discard the first 12 bits, and interpret only 8 bits. The 12 bit interpreter would be set to interpret only the first 12 bits, then discard 8 bits. This example assumes that you are monitoring MOSI at the input of the first part. You can create similar instances to interpret the data shifted out of MISO from the second part. Please see the example project files included with the software installation for sample interpreter setups.

The default setting is to discard the specified number of bits before/after *each* interpreted value. There may be times when you will only want to discard bits on the first value in a frame. For example, maybe you are analyzing an interface which sends a 10 bit command word followed by a burst of 16 bit data words, all in a single frame. You can create an instance of the interpreter which discards only the 10 bit command word, then interprets each of the data words. You can also create an instance which interprets only the 10 bit command word, then discards all remaining data in the frame.

Note that separate instances of this interpreter must be created to simultaneously interpret both MOSI and MISO (slave device Shift In and Shift Out). Likewise, you will need to setup separate instances to view Commands and Data for a 3-Wire interface. Once you have setup one instance, you can easily clone those settings using the "Copy Selected..." menu item from within the "Edit Interpreters" dialog. Click to highlight an existing interpreter, then right-click to open the menu.

The datasheet for the MC68HC11D3 makes an excellent reference for the SPI standard.

## Asynchronous Serial / RS232 Interpreter

The Asynchronous Serial/RS232 interpreter post-processes acquired data and displays the resulting values in a selected format such as ASCII or Hex. This interpreter can be used with most asynchronous protocols, including RS422 and RS485 as long as appropriate signal levels are available.

Note: If you have not yet reviewed the [Interpreter Overview](#), please be sure to read that section before trying to setup interpreters.

The following settings apply to Asynchronous Serial/RS232 interpreters:

### Name

Specify a meaningful name for this particular instance of the interpreter. This name will be displayed in the waveform and state list displays.

### Signal

Specify the channel which is connected to the Serial Data signal (TX or RX) in your hardware. Note that from within the selection dialog you can also right-click to edit signal names.

### Logic Sense

Specify the data signal's logic sense. If the voltage level representing a "Mark" (logical 1) is more negative than for a "Space" (logical 0) then the signal's logic sense is negative. This is typical for true RS232 signals. The logic inputs and outputs of RS232 drivers and receivers often use positive logic sense instead.

### Baud Rate

Specify the baud rate for your hardware. Any value in the range of 1 to 20,000,000 is allowed.

### Use Glitch Filter

When enabled, the interpreter will ignore "False" start bits caused by glitches. This mimics the behavior of a typical UART.

### Data Bits

Specify the number of data bits.

### LSB/MSB First

Specify whether data is sent MSB or LSB first. RS232 data is normally sent LSB first.

### Parity

Specify the parity mode (see detailed description below).

### Stop Bits

Specify the number of stop bits.

### Data Format

Specify the desired display format for interpreted data.

It is important to have an adequate sample rate for the Asynchronous Serial/RS232 interpreter to function properly. A sample rate of at least 20 times the baud rate is recommended. If you suspect that your hardware is malfunctioning because of glitches, you might want to increase the sample rate and turn off the interpreter's glitch filter. This will cause the interpreter to be more critical of the signal, and may help find the problem.

The parity mode can be set to any of the following options:

### None

Means no parity bit is present in the parity position.

### Odd

Means the parity bit is present, and the sum of individual data and parity bit values must be an odd number.

### Even

Means the parity bit is present, and the sum of individual data and parity bit values must be an even number.

### Mark

Means the parity bit is present, and must always be a "Mark" (logical 1).

### Space

Means the parity bit is present, and must always be a "Space" (logical 0).

### **Ignore**

Means the parity bit is present, but should be ignored. This is useful for troubleshooting when you know that your hardware is not transmitting parity bits properly.

### **Frame Synchronization**

(See description below).

If a parity error is found, it will be indicated by "PE" in the interpreted data. Note that specifying the incorrect parity mode can also cause framing errors because the timing of the stop bit(s) may be wrong. For example, if the data is sent with no parity bit, but you tell the interpreter to use even parity, the actual position of the stop bit(s) would be one bit sooner than expected.

A frame synchronization error will occur any time the stop bits are not found in the expected position. Framing errors are indicated by "FR" in the interpreted data. The interpreter requires a frame break to resynchronize the next frame after a framing error.

A frame break is defined as a period having no transitions on the data signal for a duration of at least (total frame bits x (1/ baud rate)). For example, if the baud rate is 300, and the data format is 7 data bits, even parity, and 1 stop bit, the total number of bits in a frame is 10 (remember to include the start bit). In this case a frame break is a "quiet" period of at least  $10 \times (1/300)$ , or 33.3 milliseconds.

**Frame Synchronization** is the method by which the Asynchronous Serial/RS232 interpreter recognizes the beginning of the first full frame of captured serial data. There are two framing methods available:

#### **Frame Break**

This is the preferred synchronization method. It should be used when the beginning of the serial data stream is contained within the LogicPort's acquired data, or if there is a frame break present before the data of interest. When this method is used, the interpreter will automatically find the first start bit in the serial stream and use it for frame synchronization just as a UART would do.

#### **Cursor Position**

This method is useful if acquisition began in the middle of a serial frame. This could happen if you've captured just part of a long serial data stream which doesn't contain any frame breaks. This method allows you to "show" the interpreter the beginning of a valid frame using a cursor. From this initial starting point the interpreter will automatically track all remaining serial data.

The Asynchronous Serial/RS232 interpreter begins interpreting data from the first acquired sample in the buffer. If the acquisition happened to begin in the middle of a frame, a frame synchronization error is likely to occur. In this case, the interpreter will resume interpreting data after the next frame break. If your data doesn't include frame breaks then you must make certain that the acquisition begins before the first frame of transmitted data.

Note that separate instances of this interpreter must be created to simultaneously interpret both TX and RX. Once you have setup one instance, you can easily clone those settings using the "Copy Selected..." menu item from within the "Edit Interpreters" dialog. Click to highlight an existing interpreter, then right-click to open the menu.

## CAN Bus Interpreter

The CAN bus interpreter post-processes acquired data and interprets it according to the Bosch CAN 2.0A / 2.0B specification. It will automatically detect and display frames with either 11 or 29 bit identifiers.

Note: If you have not yet reviewed the [Interpreter Overview](#), please be sure to read that section before trying to setup interpreters.

The following settings apply to each instance of the CAN bus interpreter:

### Name

Specify a meaningful name for this particular instance of the interpreter. This name will be displayed in the waveform and state list displays.

### Signal

Specify the channel which is connected to your CAN hardware. Note that from within the selection dialog you can also right-click to edit signal names.

### Rate

Specify the bit rate for your CAN hardware. Any value in the range of 1 to 20,000,000 bits per second can be specified.

### Dominant Level

Specify the signal level which represents dominant bit states in your hardware (as described in the CAN specification). For a typical differential CAN driver, CANL will drive LO for dominant bits, while CANH will drive HI for dominant bits. Note that only one of these two differential signals is necessary for the interpreter. CANL is recommended because its logic levels are typically compatible with the logic threshold settings required by other logic. If the logic sense of your signals matches that of the example waveforms in the Bosch CAN 2.0A / 2.0B specification, then you should select LO for this setting.

### Field Format

Specify a data format for the display of interpreted fields such as ID and Data Length Code.

### Data Format

Specify a format for display of the data byte portion of interpreted frames.

### Display

There are three display formats available for interpreted fields:

**Interpreted Data** Displays the highest level of interpreter output. This is the default setting.

**Raw Data** Displays dominant/recessive bit states as "0"/"1" respectively, and marks stuffed bits with "S". Dominant and recessive bits are marked with the same convention as that used in the CAN specification. That is, dominant bits are represented as "0", while recessive bits are represented as "1". Note that when the interpreter's "Dominant Level" is specified as "HI", these indications will have *opposite* logic sense from the acquired waveform.

**Bit Sequence** Displays the numerical sequence of bits within a frame, and marks the stuffed bits with "S". Note that stuffed bits do not increment the sequence count. This allows you to match the bit sequence to expected patterns without the need to calculate/remove stuffed bits.

It is possible to create multiple instances of the CAN interpreter, each with a different display setting. This allows the interpreted output, bit stuffing, bit sequence, and raw captured waveform to all be displayed simultaneously. See [Interpreter Overview](#) for information about cloning interpreters.

It is important to have an adequate sample rate for the CAN bus interpreter to function properly. A sample rate of at least 10 times the bit rate is recommended. Note that the CAN interpreter uses the same style of glitch filtering as that found in typical CAN hardware. Additionally, its bit sample point is resynchronized with each signal transition in the same fashion as CAN hardware.

When the various fields of a frame are interpreted, they are shown as mnemonics in the waveform and state list displays. If you hover the mouse cursor over a displayed mnemonic, a more detailed description is shown. The following mnemonics are used for normal frame fields:

<b>D:n</b>	<b>Data Frame, ID = n</b>
<b>R:n</b>	<b>Remote Transmit Request, ID = n</b>
<b>L:n</b>	<b>Data Length Code = n</b>
<b>C:n</b>	<b>Checksum = n</b>

<b>A</b>	<b>Acknowledged</b>
<b>N</b>	<b>Not Acknowledged</b>
<b>EF</b>	<b>End Of Frame</b>
<b>IM</b>	<b>Intermission</b>
<b>OL</b>	<b>Overload Frame</b>

The CAN interpreter also detects and reports errors in the frame (such as those caused by a node on the bus transmitting an error frame in response to a detected error). When an error is detected, the error reporting field will begin at the bit position of the detected error, and will continue until the frame is terminated by either an error frame delimiter or by normal end of frame and intermission fields. The raw data for any incomplete field leading up to the error is also displayed. In the example below, a bit stuffing error was detected at bit 11 of the checksum field. As a result, the 10 bits of raw data leading up to the error are displayed, followed by a field with the mnemonic SE beginning at bit position 11.



The following mnemonics are used for error reporting:

<b>SE</b>	<b>Bit Stuffing Error</b>
<b>CE:n (m)</b>	<b>Checksum Error, Calculated (Reported) = n (m)</b>
<b>FE:CD</b>	<b>Form Error: Checksum delimiter bit invalid</b>
<b>FE:AD</b>	<b>Form Error: Acknowledge delimiter bit invalid</b>
<b>FE:EF</b>	<b>Form Error: EOF bits invalid</b>
<b>FE:IM</b>	<b>Form Error: Intermission bits invalid</b>

Please refer to the Bosch CAN 2.0A / 2.0B bus specification for more information. Other good sources of information include datasheets for the Philips SJA1000 and Microchip MCP2515.

## Exporting Data

The data export dialog enables you to save a copy of the sampled data to a specified file. The data is exported as comma-separated ASCII values. Note that when sample compression is enabled, only those channels displayed in the waveform window are actually acquired. This means you cannot export channels which were not included in the waveform window at the time the acquisition occurred. For example, if an acquisition occurs with D[7..0] displayed in the waveform window, and you add D8 to the list *after* the acquisition, D8 cannot be exported (and no waveform will be displayed). When compression is disabled, all channels are acquired, even if they are not displayed.

The following format options are available when exporting data:

### **As Formatted in the Waveform Window**

Exported data will be formatted exactly as it is shown in the waveform window, including the duplicate signals shown for expanded groups.

### **As a Single Group of <Binary|Hex> Digits**

All displayed/acquired channels (whether individual signals or contained within groups) will be exported as a single word value with the specified radix. Channels are not duplicated, regardless of how many times they appear in the waveform window.

### **As Individual Channels Bits**

Each displayed/acquired channel will be exported as an individual comma-separated value. Channels are not duplicated, regardless of how many times they appear in the waveform window.

The first two lines of the exported file form a header which indicates the sample period, trigger sample, and compression formatting. The third line is a legend for the exported data. Note that when "CompressedData" is "True", two or more identical consecutive samples have been skipped to reduce file size (which could otherwise be terabytes under certain circumstances). You can determine how many samples were skipped by comparing the "SampleNumber" for each sample. When samples are skipped, it means their values were equal to the preceding sample.

## Save On Acquisition

This feature allows the LogicPort to monitor your hardware for hours, days or even weeks, waiting for the specified trigger conditions to occur. When a trigger event occurs, an acquisition completes and the software can export the acquired data, or save a copy of the project including acquired data. This feature is especially handy for finding random or intermittent problems.

**Tip:** Be sure that your PC's power management features are setup properly so that your PC doesn't shutdown or go to standby while you are waiting for an acquisition to be saved.

After setting up this feature, you must also enable it using **Acquisition>Save On Acquisition>Enable**. Be sure to disable this feature when not in use to avoid saving unnecessary files.

When "Save a copy of the project" is specified, the software will save a copy of the project in the same directory as the current project file. If the directory is no longer available (for example if a network drive is disconnected) the software will save the file in the LogicPort's installation directory instead.

When "Export the acquired data" is specified, the software will export the data according to the settings specified in the **File>Export Data...** dialog. If a valid "Target File Path" has not been specified, or if the path becomes unavailable, the exported file will be saved in the LogicPort's installation directory instead.

When a file is saved, its name is appended with a time stamp indicating the date and time of the acquisition. The format for a project file named MyProject.LPF would be as follows:

MyProject-YYMMDD-HHMMSS-CC.LPF

YYMMDD is the date, HHMMSS is the time, and CC is a counter. The first file created within a given 1 second interval will have a count of 01. If additional files are created during the same second, the counter will be incremented for each file.

You should specify the maximum number of files to save. The default for new projects is 1. Each project file requires approximately 160K of storage. The size of an exported file depends upon the export settings and data. The file counter resets each time acquisition is halted. This means that if you specify a maximum of 2 files, the software will save up to 2 files *each* time an acquisition is initiated from the Halt state.

You can also specify the "Holdoff Time" or minimum time to wait between file saves. This is useful if you expect that trigger events may come in bursts of activity. For example if you expect the trigger events to occur in bursts of 100 events over a period of a few minutes, but only expect a burst once per hour, you could set the holdoff time to 3000 seconds. This will insure that no more than one file will be saved for each burst of activity.

Note that this function does *not* continuously stream data to disk. That job is reserved for data loggers with much slower sample rates. The LogicPort is capable of capturing data at rates of up to 17 gigabits per second (500MHz x 34 channels)! This is much faster than any current-technology PC interface is capable of transferring data. As a result, the LogicPort must buffer its data during acquisition, then transfer it to the host PC at a much slower rate *after* acquisition.

## Specifications and Characteristics

**Sampled channels:** 34

**Timing mode sample rate:** 1KHz to 500MHz, standard 1-2-5 sequence

**State mode sample rate:** 0 to 200MHz

**Sample buffer:** 34 x 2048 samples

**Maximum sample compression:**  $2^{33}$  to 1 (sample rates to 200MHz)

**Trigger sequencer:** 250MHz max (4ns minimum pulse width)

**Trigger event counters:** 20 bits (range 1 to 1,048,576) for each trigger level

**Input impedance:** 200k $\Omega$ , paralleled by <5pF

**Frequency counter range:** 300MHz with 10Hz resolution, 4 channels

**Threshold range:** adjustable from +6 to -6 volts with 50mv resolution

**Threshold accuracy:** +/- (100mV + 5% of setting)

**Channel to channel skew:** 0.6ns typical, 1.0ns max

**State mode Setup/Hold times:** +2.0/0ns, adjustable +/-2.5ns

**Input sensitivity:** 0.25Vpp @50MHz, 0.5Vpp @150MHz, 0.8Vpp @250MHz

**Input dynamic range:** 10 volts peak-peak

**Maximum input:** +/-40 volts DC, 15 volts peak-peak AC

**Timebase accuracy:** +/-0.005% over full temperature range

**Power supply current:** 200 milliamps max (supplied by USB)

**Temperature range:** operating +5 to +50C, storage -10 to +65C

### Minimum system requirements:

Pentium II or better (400MHz or faster recommended)

32MB available memory

20MB available disk space

800 x 600 (1024 x 768 recommended) standard DPI display

USB 1.1 or 2.0 port

The LogicPort is compatible with Windows 98, Windows 98 Second Edition, Windows ME, Windows 2000, Windows XP Home, Windows XP Pro and Windows Vista (32 bit versions).

**Specifications subject to change without notice**

## Frequently Asked Questions

### How much does sample compression really help?

The best way to answer this question is to look at the actual acquired data which is included with the LogicPort software demo. You can see that it's not uncommon to collect many times the size of the sample buffer. The success of sample compression depends entirely upon the nature of the data. If the sampled data has many transitions occurring at or near the sample rate, there may be little benefit. If the sampled data is sporadic or has few transitions, the benefit can be enormous. The time between signal transitions is almost "free", so in some cases it could literally take hours to fill the LogicPort's sample buffer!

### How many full-speed channels does the LogicPort support?

The LogicPort will simultaneously sample all 34 channels at rates up to 500MHz in timing mode. This is in direct contrast with many logic analyzers which only sample at full speed on a fraction of their available channels. When sampling in state-mode one channel is used as a state-clock input, allowing the remaining 33 channels to be sampled at the state-clock rate.

### Can I still sample at 500MHz if my computer is only 400MHz?

Yes. The speed of the PC's CPU and USB port have no relation to the LogicPort's sample rate. The LogicPort can acquire data at up to 17 gigabits per second (34 x 500MHz). This is much faster than a USB interface can transfer data. As a result, the LogicPort hardware must first acquire data in its internal memory at the chosen sample rate, then stop sampling and transfer data to the PC at a slower rate. Once the transfer is complete, the LogicPort hardware begins a new acquisition while the LogicPort software processes and displays the most recently acquired data. This process is virtually identical to what occurs in a stand-alone analyzer, except that the acquisition and display functions happen to be in separate enclosures. The speed of the PC's CPU only effects how fast the software can process the acquired data. Running the LogicPort on a slower PC will reduce the number of times the display is refreshed each second, but the sample rate during acquisition will not be affected. On an average 400MHz Pentium desktop PC the LogicPort can refresh the displayed data about 4 times per second. With faster PCs it is possible to refresh the data 15 times per second or more.

### How does the LogicPort connect to my circuit?

See [Probing Options](#) in the appendix for complete details.

### Are the LogicPort's ground leads isolated from earth ground?

Not usually. The LogicPort's ground leads connect to your PC's chassis through the USB cable. This means that in most cases, there is a direct connection to earth ground as well. One notable exception is the use of a battery-powered PC having no other connections (such as monitor, network or modem). Be careful not to let the LogicPort's ground leads contact power in the circuit under test.

### Can I analyze captured data on another computer?

Yes. You can install the LogicPort software on as many computers as you like. Simply run the software in demo mode to view files captured on another PC.

### Can I export the sampled data to another program?

Yes. The sampled data may be exported as comma-separated ASCII values. A variety of formats are available. You can import this data directly into programs such as Microsoft Excel.

### Can I run the LogicPort from a bus powered USB hub?

Yes. As long as the current required by the hub plus all connected devices does not exceed 500 milliamps (per USB specification). The LogicPort's low power consumption makes this condition easy to meet.

### Can I use more than one LogicPort to acquire additional channels?

Not easily. It is possible to run more than one LogicPort from a PC, but each unit must have its own instance of the software. It's not really practical to use multiple units in timing mode because there is no way to synchronize the sample clocks between separate units. In state mode multiple units are naturally synchronized because they are all sampling based on a clock provided by your hardware. It's possible to acquire data this way, but the data from each unit can only be displayed by the instance of software controlling that particular unit. You could however, export the data from each software instance to be combined/utilized by your external application. If you are considering using multiple LogicPort units in this fashion, please consult with tech-support for more information.

**Input connector pin assignments**

Pin Number	Wire Name	Wire Color	Contact Color
1	D0	Black	White
2	D1	Brown	White
3	D2	Red	White
4	D3	Orange	White
5	D4	Yellow	White
6	D5	Green	White
7	D6	Blue	White
8	D7	Violet	White
9	Ground	Gray	Black
10	Ground	N/A	N/A
11	D8	Black	Blue
12	D9	Brown	Blue
13	D10	Red	Blue
14	D11	Orange	Blue
15	D12	Yellow	Blue
16	D13	Green	Blue
17	D14	Blue	Blue
18	D15	Violet	Blue
19	Ground	Gray	Black
20	CLK1	White	Blue
21	D16	Black	Yellow
22	D17	Brown	Yellow
23	D18	Red	Yellow
24	D19	Orange	Yellow
25	D20	Yellow	Yellow
26	D21	Green	Yellow
27	D22	Blue	Yellow
28	D23	Violet	Yellow
29	Ground	Gray	Black
30	Ground	N/A	N/A
31	D24	Black	Green
32	D25	Brown	Green
33	D26	Red	Green
34	D27	Orange	Green
35	D28	Yellow	Green
36	D29	Green	Green
37	D30	Blue	Green
38	D31	Violet	Green
39	Ground	Gray	Black
40	CLK2	White	Green

**Caution:** Ground leads [connect to the PC's chassis](#) through the USB cable.

## Probing Options

There are several options for connecting the LogicPort to your circuit.

Each lead of the color-coded probe supplied with the LogicPort has a female contact compatible with standard 0.025-0.030" round or square pins. Typically these leads are connected directly to the headers found on many circuit boards. The probe includes four ground leads. See the [Connector Pin Assignments](#) section of the appendix for probe color codes.

The LogicPort probe contacts will connect directly to grabbers such as the EZ-Hook XKM, and Pomona 5360. These grabbers have 0.025" pins designed for this purpose. XKM grabbers are available in our online store. Pomona 5360 grabbers are available from Newark Electronics and others.

You can also connect the probe leads directly to PLCC clips such as the 3M 923675-84 and DIP-clips such as the 3M 923690-40. These clips are available from Digi-Key and others, and allow direct connection to PLCC and DIP IC packages.

The LogicPort input connector is a standard 40 pin IDC compatible header. You can use inexpensive IDC connectors and ribbon cable to connect directly to a header on your circuit board, or to create your own custom test leads. When using ribbon cable, we recommend keeping the leads as short as possible. It is also best to split the leads as much as possible to reduce crosstalk.

You can also create test-points by soldering short sections of 22-24ga solid bus wire to your board. The LogicPort's probe contacts will connect directly to these test points.

Whichever method you choose, be sure to hook up as many of the ground leads as possible. This can greatly improve sampling accuracy with fast signals.

If adding a dedicated test header to your circuit board at design time, consider placing a small resistor in series with each test pin. Resistors in the range of 22-75 ohms can improve pulse response and sampling accuracy for signals with very fast edges. Oscilloscopes and other test equipment can benefit from this as well.

**Caution:** The LogicPort's ground leads have a direct connection to your PC's chassis through the USB cable. This means that in most cases, there is also a direct connection to earth ground. Be careful not to let the LogicPort's ground leads contact power in the circuit under test. Damage to the circuit could result. We stress this fact because while most stand-alone test equipment has this same connection to earth ground, it may be less obvious that this connection exists with PC-based test equipment. Some laptops have isolated chassis when running from the AC line but many do not. A PC may have other paths to earth ground as well, such as through the cables connecting monitors and other equipment. If the circuit you're testing is isolated from ground, the LogicPort may become its sole connection to ground through the USB cable. If sufficient current flows through this connection, USB signals can become corrupted causing USB errors. In extreme cases (many amps), damage to the USB circuitry may be possible.

## Understanding Timing and State modes

Timing mode is based upon a logic analyzer's internally generated sample clock. One sample is stored in the sample buffer on each rising edge of this internal clock. The rate of this clock is selectable in finite increments, and is not synchronized to the hardware under test. This mode is typically used when we want to see the timing relationships between signals such as "WR", "ADDRESS" and "DATA".

State mode uses an external clock supplied by the hardware under test to synchronously sample the data. One sample is stored in the sample buffer on each (specified) edge of this external clock. State mode is useful when we are interested in the data values rather than their timing relationships. For example, we could use state mode to capture data coming from an A/D converter on each rising edge of its sample clock. The resulting sampled data would have all of the actual values generated by the A/D converter without the intermediate values created as the individual bits of the data bus settled. This data could then be viewed as an analog waveform in the LogicPort software, showing a clean representation of the A/D's conversions. If timing mode were used, this waveform might appear noisy because of the intermediate data values produced as individual bits settled.

A variation on state mode is "Qualified" state mode. When the LogicPort's qualified state mode is enabled, data is only sampled when the external clock's specified edge occurs *while* the qualifier input is in its specified active state. This advanced feature is useful for demultiplexing address/data buses or for capturing data directed toward a specific device. For example you could use the ALE signal from an Intel processor to qualify the sample clock, thereby capturing only the *addresses* from the multiplexed address/data bus. You could also use the CE signal for a particular device to capture only the data which is directed toward that device.

Timing mode is the most commonly used mode. Some logic analyzers (even some expensive ones) don't support state mode.

Most logic analyzers can sample at higher rates in timing mode than in state mode. The LogicPort is no exception.

When capturing data in timing mode, it takes at least two samples for each clock cycle of the hardware under test. This is an absolute minimum per Nyquist theory. At this minimum sample rate the acquired data would be difficult to interpret, and probably incomplete since the sample clock and circuit clock would rarely be in phase. In real-world applications, we are usually interested in the timing relationships between signals. This means we usually need far more than two samples per clock cycle. A good rule of thumb is to sample at least four times the fastest toggle rate in the hardware under test. Capturing subtle hardware timing relationships requires sampling at even higher rates.

State mode makes more efficient use of sample memory because there is only one sample for each clock cycle of the hardware under test. Qualified state mode makes even more efficient use of memory because little or no sample memory is consumed when the qualifier is inactive.

State mode is often used for software debugging because it shows the actual values written to or read from memory without all of the "noise" in-between caused when different bits of the bus settle at different times. This is when the state list becomes especially useful.

Note that data sampled in either mode can be viewed as waveforms, or as numerical values in the state list. When the "Unique States" format is selected for the state list, it can often display more data than the waveform window.

## Getting the most from Sample Compression

Sample compression allows the LogicPort to pack more data into its sample buffer. The compression occurs in real-time as samples are being collected. The basic idea is that if signals are not changing, there is no reason to keep filling the sample buffer with copies of identical data. Instead, the LogicPort keeps track of how many sample periods elapse with no change in the data. The count is then stored in one of the memory locations which otherwise would have contained a copy of the data. The LogicPort software then “unpacks” the compressed samples and displays the data as it actually occurred. This form of compression has been referred to as “Transitional Sampling” because samples are only collected when there is a transition on at least one signal.

From this description you can see that the level of compression is determined by the number of transitions which occur in the data over a given period of time. Note that the LogicPort’s compression logic is only “sensitive” to signals and groups currently displayed in the waveform table. Signals which are not displayed are also not sampled, so transitions on these signals will be ignored by the compression logic. For this reason, it is advisable to remove signals from the waveform table if they are not needed for triggering, or are no longer of interest.

Some circuits such as microprocessor memory interfaces have an abundance of narrow glitches which occur during bus transactions. Because these signals are narrow, and often not full amplitude, it is possible to eliminate some of the glitches by adjusting the LogicPort’s input threshold. This decreases the overall number of transitions in the data and will improve the compression level.

Clock signals naturally have a large number of transitions. Consider using the clock signal to sample in state mode instead. This can greatly increase the level of compression for some types of data. State mode sampling often makes better use of sample memory with or without compression. Qualified state mode sampling makes even better use of memory, and is the best choice when possible.

## Warning Messages

Under certain circumstances, warning messages may appear in the status bar with a highlighted background. Warning messages do not necessarily indicate errors. The messages are intended to make you aware that a particular setting, or combination of settings may be conflicting or limiting performance in some fashion. The following list explains various warning messages, along with tips on how to eliminate them if desired.

### "Buffer depth constrained by Time Limit settings"

This warning occurs when the Pre-Fill Time Limit or the Post-Fill Time Limit is reached during an acquisition. Under these circumstances, only a portion of the sample buffer has actually been utilized for the acquisition. This is often desirable in cases where very high sample compression ratios cause excessively long acquisition times. This warning may indicate that more data could be acquired if the time limits were increased. For more information see [Sample Mode Setup](#). Note that during a recurring acquisition sequence this warning only occurs for the *first* constrained acquisition.

### "Trigger Level {A|B} Range value(s) constrained"

This warning occurs if you have specified an illegal range for the Value term of the indicated trigger level. For example, assume that the group "Data[7..0]" is an 8 bit bus and that you have specified the following for the Value term:

**Value of Data[7..0] is Equal to 257**

You would get a warning message because the maximum possible unsigned value for an 8 bit group is only 255. The software will adjust the value to bring it within legal limits, and then display this warning.

### "Trigger Level {A|B} Edge count constrained"

This warning occurs if you have specified an illegal value for the Edge term counter. The trigger event counters are 20 bits. This means the legal Edge count range is 1 to 1,048,576. If you specify a value outside that range, the software will adjust the value to bring it within legal limits and then display this warning. You can extend the Edge count range by cascading trigger levels using the "Trigger When level A is satisfied and then level B is satisfied" option.

### "Trigger Level {A|B} Pattern count constrained"

This warning occurs if you have specified an illegal value for the Pattern term's Entry/Exit counter. The trigger event counters are 20 bits. This means the legal Entry/Exit count range is 1 to 1,048,576. If you specify a value outside that range, the software will adjust the value to bring it within legal limits and then display this warning. You can extend the Entry/Exit count range by cascading trigger levels using the "Trigger When level A is satisfied and then level B is satisfied" option.

### "Trigger Level {A|B} Duration value(s) constrained"

This warning occurs if you have specified a value outside the range of the Duration term's sample counter. The trigger event counters are 20 bits. This means the possible Duration range is 1 to 1,048,576 samples (or sample periods if specified in seconds). If you specify a value outside that range, the software will adjust the value to bring it within limits and then display this warning. The constrained value will be displayed next to the specified value in the setup dialog. You can extend the Duration period by decreasing the sample rate if required. You can also extend the Duration range by cascading trigger levels using the "Trigger When level A is satisfied and then level B is satisfied" option.

## USB Driver Reinstallation

If the LogicPort hardware is connected to your PC *before* the LogicPort software is installed or if you cancel the driver installation, Windows may not install the USB drivers properly. If this happens, the LogicPort software will not recognize the hardware. To solve this problem, you should reinstall the USB drivers properly using this procedure:

### **First, uninstall the LogicPort hardware/USB driver following these steps:**

1. Plug in the LogicPort.
2. Press the <Windows>+<Break> keys simultaneously to bring up "System Properties".
3. Click on the "Hardware" tab.
4. Click "Device Manager".
5. Look in "Universal Serial Bus controllers".
6. Right click on "LogicPort Logic Analyzer".
7. Click "Uninstall".
8. When the "Confirm Device removal" dialog appears click "OK".
9. Unplug the LogicPort.

### **Follow these steps to properly install the device drivers (note that you should not browse for the driver files. Windows will find them automatically):**

1. Reinstall the LogicPort software to make sure that the driver files are present.
2. Plug in the LogicPort.
3. Wait until the "Found New Hardware Wizard" appears. This could take up to 2 or 3 minutes on XP machines.
4. If given the option to search the internet for drivers, select "Not this time", and click "Next".
5. Select "Install the software automatically (Recommended)", and click "Next".
6. If the unsigned driver warning dialog appears, click "Continue Anyway".
7. When "Found New Hardware Wizard" dialog returns click "Finish".
8. Exit the LogicPort software if it's running.
9. Run the LogicPort software to verify that the LogicPort hardware is recognized.

## Copyright Intronix 2004-2007

No part of this document may be reproduced or transmitted in any form or by any means without the express written consent of Intronix.

### Warranty

Information and specifications contained in this document are provided "as is", and are subject to change without notice. To the extent permitted by law, Intronix disclaims all warranties, either express or implied, with regard to the information contained in this document, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

### Federal Communications Commission (FCC) Statement

Intronix Model LA1034 LogicPort Logic Analyzer

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult an Intronix representative for help.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emissions limits. Proper cables and connectors are available from Intronix. Intronix is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

### Trademark Acknowledgments

Intronix is a registered trademark of Intronix Test Instruments, Inc. USA. All other referenced marks, whether claimed or registered, are for identification purposes only and are the exclusive property of their respective owners.

## **Analog Waveforms**

Groups of digital signals can be collectively displayed as an analog waveform. When the analog waveform style is selected for a group row, sequential data values for the group are translated to a representative analog waveform. This is equivalent to connecting the selected digital signals to a D to A converter, and then viewing the resulting analog waveform on an oscilloscope. This is especially useful for DSP development, and can also be used for quick visual profiling of software performance.

The digital data can be interpreted as unsigned values, or as 2's compliment signed values. The scale is automatically set according to the number of bits in the group. The height of the waveform can be adjusted by dragging the lower border of the cell containing the group's name.

## Software Updates and Internet Access

The LogicPort software includes an option to automatically check for software updates at startup. You can find this option in the **Help** menu, under **Software Updates**. When enabled, the software will check the Intronix website for availability of a newer software version each time the program is started. If a newer version is found, you will receive a prompt indicating its availability.

Note that the software will not automatically download new versions. It will simply make you aware of their availability. You will need to visit the Intronix website to download the latest version. You can do this conveniently by using the **Visit Website...** option in the **Help** menu. This option opens the default browser on your system, and navigates to the Intronix home page.