PA20 Series
PA40 Series

Logic State Analyzers

User's Manual

DC00201

SofTec Microsystems
via Roma, 1
33082 Azzano Decimo (PN) ITALY
Tel: (+39) 0434 640 729
Fax: (+39) 0434 632 695
E-mail (general information): info@softecmicro.com
E-mail (marketing department): marketing@softecmicro.com
E-mail (technical support): support@softecmicro.com
Web: http://www.softecmicro.com

Notice of Conformity
This product is in conformity with the EN-55022, EN-50082-1, ENV-50140 specifications and therefore it complies with the EMC Directive 89/336/EEC.

Contents

Figures and Tables

1. General Information

Introduction

PA20 and PA40 Series Logic State Analyzers, in combination with a standard personal computer, provide a full featured, high-performance tool for the troubleshooting and performance verification of digital circuits.

A logic analyzer is the digital counterpart of an analog oscilloscope. It allows a number of digital input signals to be sampled and stored sequentially in a high-speed memory or buffer. A logic analyzer can also recognize a condition, or sequence of conditions, on the input data and use that combination of events to trigger data storage. The information acquired is displayed as oscilloscope-like waveforms or as a list of numbers representing a sequence of logic states.

PA20 and PA40 Series Logic State Analyzers can work in two different operation modes: Timing mode and State mode. Timing mode is useful when recording the input data at a constant rate determined by a fixed timebase. As a result, the waveform display represents time in linear form on the X-axis and logical state on the Y-axis.

In State mode, instead, an external sample clock is provided, thus synchronizing sampled data with state transitions that occur in the circuit under test.

PA20 and PA40 Series Logic State Analyzers feature powerful triggering conditions, such as edge trigger, pattern trigger and advanced trigger specification, in order to trigger data storage at the very specific event needed to properly debug the circuit under test.

PA20 and PA40 Series Logic State Analyzers allows you to group input channels into bus items. The user interface will display the acquired data by grouping the input channels as specified. The order with which the input channels build a bus is used to determine the numeric value of bus at each sampling, which is also displayed by the analyzer.

PA20 and PA40 Series Logic State Analyzers, additionally, feature a series of serial data analysis—sets of algorithms that perform a special analysis on the raw acquired data. By defining an item as one of the three available serial analysis functions (asynchronous serial channel, generic synchronous serial channel, I^2C-bus), the user interface will automatically display the appropriate serial character or packet characteristics.

All these features are accessible from a user-friendly yet powerful user interface running under Windows® 95/98/NT. Additionally, an interface library (DLL) is provided (as an option) so that you can interface your own programs directly with the analyzers.

The sampling speed, number of channels and acquisition memory depth for each analyzer of the PA20 and PA40 series is summarized in the table below.

Analyzer Model	Number of Channels	Sampling Speed	Acquisition Memory (per channel)
PA2016A	16	20 MSa/s	128K Samples
PA2032A	32	20 MSa/s	128K Samples
PA4016A	16	40 MSa/s	128K Samples
PA4032A	32	40 MSa/s	128K Samples

PA20 and PA40 Series Logic State Analyzer Summary

System Requirements

PA20 and PA40 Series Logic State Analyzers are controlled by a PC user interface (PA24X) running under Windows.

The following hardware and software is required to run the PA24X user interface.

- An Intel Pentium 90 or better processor running Windows 95, Windows 98, or Windows NT version 4.0 or later.
- 16 MB of random-access memory (RAM) plus 20 MB of available disk space.

Package Checklist

PA20 and PA40 Series Logic State Analyzer are shipped together with the following items:

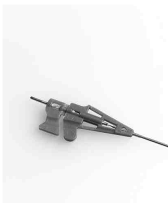
- 237-00131 Logic Analyzer 16-Channel Cable
- 237-00132 AC/DC Switching Power Adapter
- 237-00136 SMD Grabber (set of 17 pcs)
- 237-00139 ISA Bus Bidirectional Parallel Port
- PA24X User Interface Software (3.5” Floppy Disk)
- This manual

Additional 237-xxxxx items can be purchased separately. As an option, a software library (a DLL for Windows) is also available to interface the analyzer with custom developed applications. Furthermore, ultra fine pitch grabbers can be purchased separately.

- 237-00133 Communication DLL (not included in the standard package)
- 237-00138 Ultra Fine Pitch Grabber (not included in the standard package)

Instrument Package (with probes assembled)

SMD Grabber



Optional Fine Pitch Grabber

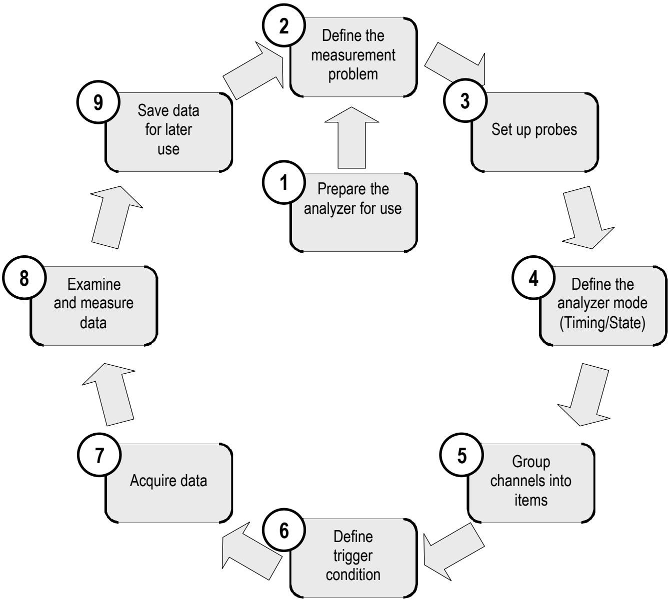
2. Getting Started

Using the Logic Analyzer

When you use the logic analyzer to help test and troubleshoot your systems, you will follow the general process shown below:

1. **Prepare the analyzer** by connecting it to the PC and to power.
2. **Define the measurement problem** by understanding the parameters of the system you wish to test and the expected system behavior.
3. **Set up channel inputs** by connecting the data probes to the appropriate signal and ground nodes in the circuit under test.
4. **Define the analyzer mode**—Timing mode or State mode, depending on whether you want to use the analyzer's internal sampling clock source or an external one.
5. On the user interface, **group input channel into items**, depending on their functions in the system under test. Each item groups one or more input channels into a single, logical entity—a generic channel, a bus, an asynchronous serial channel, a generic synchronous serial channel or an I^2C-bus.
6. **Define the trigger condition** by setting the analyzer configuration to capture only the system events you wish to view.
7. Use the analyzer to **acquire data**, either in continuous or single shot fashion.
8. **Examine the data** and make measurements on it using various analyzer features.
9. **Save the measurement and configuration** for later re-use or comparison with other measurements.

The process is repeated as necessary until you verify correct operation or find the source of the problem.



Using the Analyzer

Installing the PA24X User Interface

The PA24X user interface can be installed by simply running the Setup.exe file provided on the floppy disk. The setup program will copy the required files to your hard drive. Additionally, an uninstall program will be copied, giving you the possibility to uninstall the PA24X user interface at any later time.

To install the PA24X user interface:

1. Insert the installation floppy disk into your computer's floppy disk drive.
2. From Windows, choose **Start > Run**. Click Browse and choose the **Setup.exe** file on the PA24X floppy disk. Click OK in the *Run* dialog box to begin the installation.

   Note: if you are installing the PA24X user interface from Windows NT, you must have logged in as Administrator.
3. Follow the on-screen instructions.

Connecting the Analyzer to the PC

To prepare the analyzer to acquire data, follow the steps indicated below:

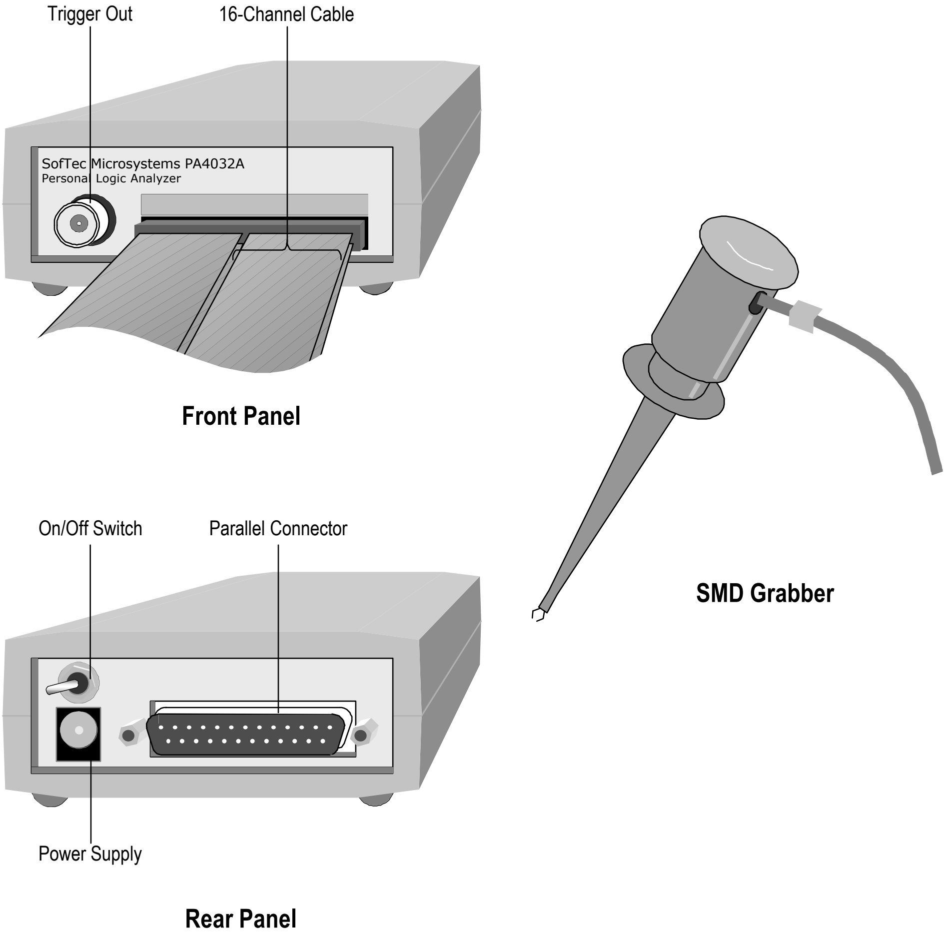
1. Turn off the PC.
2. Insert the male D-Sub miniature connector of the parallel cable into a free PC parallel interface.

   Note: the analyzer communicates with the PC through a standard, bidirectional parallel port. If your parallel port is not fully compatible with the required hardware, the 237-00139 ISA Bus

Bidirectional Parallel Port item, supplied together with the analyzer, can be installed on your PC for proper communication with the analyzer. For more details, see the chapter "Troubleshooting".

3. Insert the female D-Sub miniature connector of the parallel cable into the analyzer parallel connector (located on the analyzer's rear panel).
4. Turn the power switch on the "OFF" position. Connect the output voltage of the AC/DC switching power adapter to the power connector on the analyzer box (located on the instrument's rear panel). Turn the power switch on the "ON" position.
5. Turn on the PC and run the PA24X user interface.

The following figure illustrates where the analyzer's connectors are located.



Logic Analyzer Hardware

Probing a Circuit

Probe Leads Arrangement

Before making any connection, an understanding of the probe leads arrangement is required. The probe leads are arranged in groups: A, B, C, D (C and D groups are only present in 32-channel analyzer models). Each group contains 8 data channels; for example, group A contains channels A0, A1, …, A7. Groups are paired together: groups A and B form a 16-channel cable, and groups C and D form another 16-channel cable. Each 16-channel cable contains a ground lead, colored black.

To each cable is attached a colored label for an easy identification of the associated input channel, as specified in the following table.

Input Channel	Label Color
0	Black
1	Brown
2	Red
3	Orange
4	Yellow
5	Green
6	Blue
7	Violet

Input Channel Label Colors

To learn how to configure the logic analyzer input channels in the PA24X user interface, see the chapter *"Setting Up the Analyzer"*.

Probing Tips

1. Turn off the power supply to the circuit under test.

   Turning off power to the circuit under test prevents damage that might occur if you accidentally short two lines together while connecting probes. You can leave the analyzer powered on because no voltage appears at the probes.
2. Connect the probe cable to the analyzer.
3. Connect a grabber to one of the probe leads.
4. Connect the grabber to a node in the circuit you want to test.
5. Connect the ground probe grabber to the circuit ground.
6. Repeat steps 3 through 5 until you have connected all test points of interest.
7. Turn on the power supply to the circuit under test.
8. You are now ready to make measurements using the PA24X user interface.

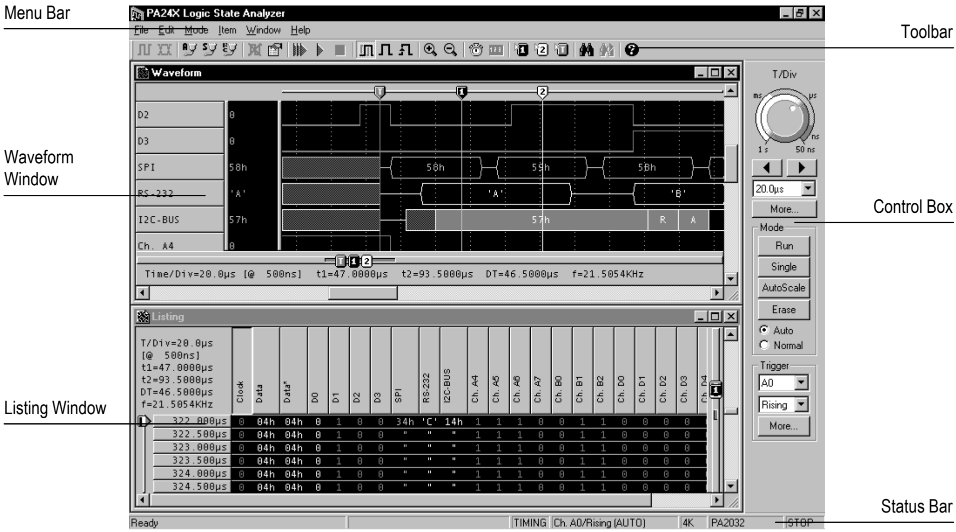
User Interface Overview

The PA24X user interface consists of several elements, organized in a way that gives you the maximum benefits both in easy-to-use and productivity terms.

- **A menu bar.** The menu bar groups all of the user interface commands.

- **A toolbar.** The toolbar shows the most frequently used commands—it's a subset of the menu bar commands.
- **A status bar.** The status bar shows analyzer's status information, such as acquisition memory setting, operation mode (State/Timing), edge trigger settings, acquisition mode (Run/Stop), etc.
- **A Control Box Panel.** The Control Box panel groups the main acquisition commands (such as Run and Stop) as well as the main acquisition settings (such as Timebase, trigger condition, etc.).
- **A *Waveform* window.** The *Waveform* window displays the acquired data in a graphical fashion. Input channels are grouped up into customizable items for meaningful representation.
- **A *Listing* window.** The *Listing* Window provides an alternative presentation for sampled data. The sampled data is displayed as a list of numerical logic values instead of a waveform graphical representation.

The toolbar, the status bar, the Control Box panel, the *Waveform* window and the *Listing* window can be turned on or off; the *Waveform* window and the *Listing* window, additionally, can be arranged any way you like. The following figure illustrates the PA24X user interface.



PA24X User Interface

The *Waveform* Window

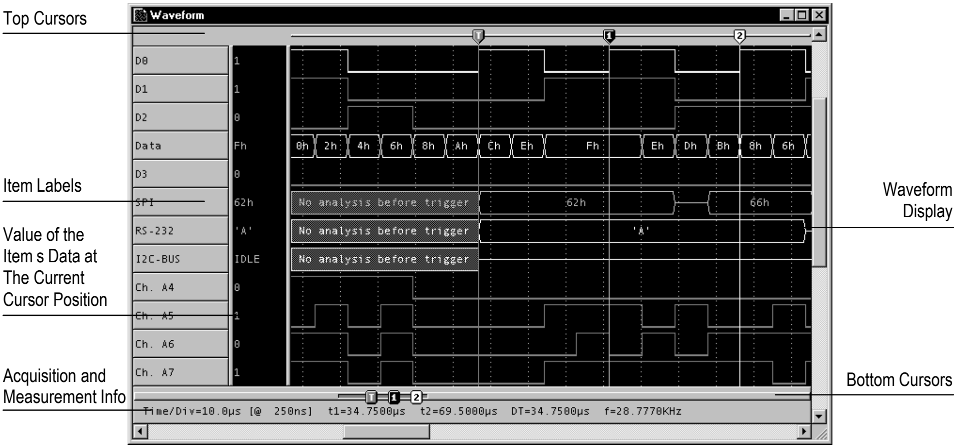
The *Waveform* window represents the acquired data graphically. To open the *Waveform* window, choose **Window > Waveform** from the menu bar.

In the *Waveform* window, each item occupies a row in the waveform display—the item's label, the value of the item's data at the current cursor position and the graphical representation of the item's data are displayed. (For more information on items, see the chapter *"Setting Up the Analyzer, Defining Items"*.)

Two cursors (Cursor 1 and Cursor 2) are available to make data measurements. Furthermore, the trigger position can be adjusted via the trigger cursor. Cursors position can be adjusted either via the arrowed cursors at the top of the *Waveform* window or via the rectangular cursors at the bottom. The former allow fine-grained adjustment within the data displayed in the window width; the latter allow quick adjustments within the whole acquisition memory.

Note: top cursors and bottom cursors are synchronized: moving the top Cursor 1, for example, causes the bottom Cursor 1 to slide, and vice versa.

The horizontal red bar represents the amount of the acquisition memory that is actually rendered in the waveform display. The following picture illustrates the various parts of the *Waveform* window.



The Waveform *Window*

At the very bottom of the *Waveform* window is displayed information about the acquired data, depending on the analyzer mode. In Timing mode, the following information is displayed:

- the T/Div setting;
- the sample period used in the last acquisition;
- the position of Cursor 1 (t1) with respect to the trigger cursor (time difference between trigger and t1);
- the position of Cursor 2 (t2) with respect to the trigger cursor (time difference between trigger and t2);
- the Dt (t2 – t1);
- the frequency corresponding to Dt.

Instead, in State mode, the following information is displayed:

- the external clock source and edge;
- the position of Cursor 1 (s1) with respect to the trigger cursor (number of samples between trigger and s1);
- the position of Cursor 2 (s2) with respect to the trigger cursor (number of samples between trigger and s2);
- the Ds (s2 – s1);

To pan across the acquisition memory, use the horizontal scrollbar; to pan across items, use the vertical scrollbar.

The *Listing* Window

The *Listing* window represents an alternative representation for acquired data; instead of a waveform graphical representation, data is displayed as a list of numerical logic values. To open the *Listing* window, choose **Window > Listing** from the menu bar.

The *Listing* window is organized in a way similar to the *Waveform* window; however, in the *Listing* window, each item occupies a column—consequently, cursors slide vertically rather than horizontally.

Additionally, a tag is attached to each sample (on the left of each row) indicating its "distance" from the trigger cursor. In Timing mode, this value is expressed as a time; in State mode, this value is expressed as a number of samples. Samples acquired before the trigger event have a negative distance; samples acquired after the trigger event have a positive distance. You can think of these values as if Cursor 1 or Cursor 2 were to assume the position of the corresponding sample.

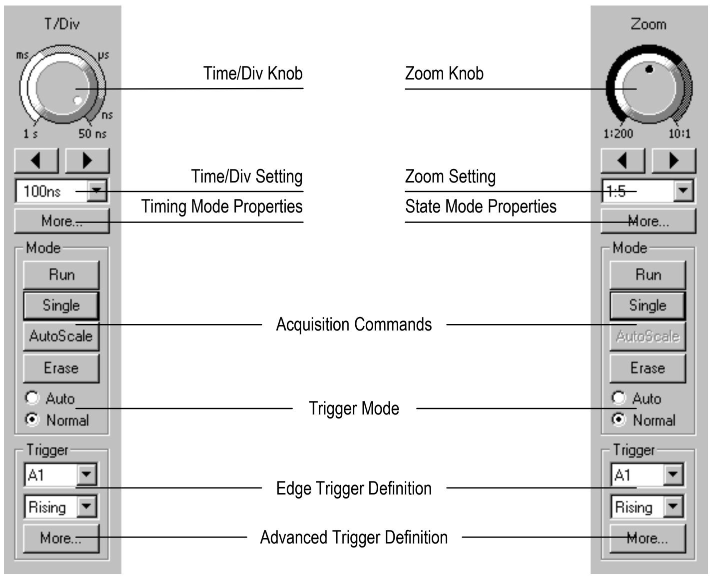
To pan across the acquisition memory, use the vertical scrollbar; to pan across items, use the horizontal scrollbar.

Note: the Waveform *window and the* Listing *window are synchronized—changing the position of the cursors in one window results in the update of the cursors position in the other. Furthermore, data displayed in one window is always the same data displayed in the other window: the* Waveform *window and the* Listing *window are just two different views of the very same acquired data.*

The Control Box Panel

The Control Box panel is placed at the rightmost side of the analyzer's user interface main window and contains the most commonly used commands, such as the T/Div or Zoom knob, the Run/Stop/Single commands, the trigger mode selection (Auto/Normal), the edge trigger definition, etc.

The Control Box appears slightly different as to whether the analyzer is in Timing or State mode. Refer to the figure below for a description of the Control Box commands.



Control Box for Timing Mode **Control Box for State Mode**

The Control Box *Panel*

The Toolbar

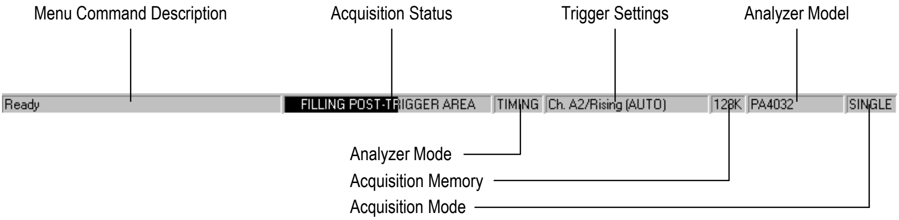
The toolbar shows the most frequently used commands—it's a subset of the menu bar commands.

Add Generic Channel Item	Trigger Definition
Add Bus Item	Zoom In
Add Asynchronous Serial Channel Item	Zoom Out
Add Generic Synchronous Serial Channel Item	Timing Mode Properties
Add I^2C-Bus Item	State Mode Properties
Remove Item	Go To Cursor 1
Item Properties	Go To Cursor 2
Run (Continuous Acquisition)	Go To Cursor Trigger
Single Shot Acquisition	Find
Stop Acquisition	Find Next
Auto Trigger Mode	Help
Normal Trigger Mode	

Toolbar Buttons

The Status Bar

The status bar shows the analyzer's status information, such as acquisition memory setting, operation mode (State/Timing), edge trigger settings, acquisition mode (Run/Stop), etc. The various fields of the status bar are illustrated below.



The Status Bar

Shortcut Keys

Most of the PA24X User Interface commands are quickly accessible via keystrokes. The following table summarizes the available shortcut keys.

Command	Shortcut Key	Command	Shortcut Key
Go To Cursor 1	Ctrl + 1	Item Properties	Ctrl + R
Go To Cursor 2	Ctrl + 2	Item Select Previous	Left Arrow/ Up Arrow
Go To Cursor Trigger	Ctrl + T	Item Select Next	Right Arrow/ Down Arrow
Run	F5	Item Move Backward	Shift + Left Arrow/ Shift + Up Arrow
Stop	F6	Item Move Forward	Shift + Right Arrow/ Shift + Down Arrow
Single	F7	Decr. Timebase/Zoom	F3
Erase	F8	Inc. Timebase/Zoom	F4
Trigger Mode Auto	F9	Scroll Backward	PgDwn
Trigger Mode Normal	F10	Scroll Forward	PgUp
Trigger Definition	F11	Waveform Zoom In	+
Item Add Generic	Ctrl + H	Waveform Zoom Out	-
Item Add Bus	Ctrl + B	Find	Ctrl + F
Item Add Async. Serial	Ctrl + A	Find Next	Ctrl + N
Item Add Sync. Serial	Ctrl + S	Environment Settings	Ctrl + E
Item Add I^2C-Bus	Ctrl + I	Copy To Clipboard	Ctrl + C
Item Remove	Del		

Shortcut Keys

3. Setting Up the Analyzer

Overview

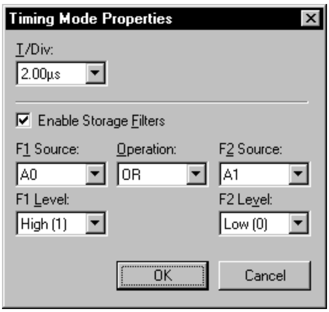
PA20 and PA40 Series Logic State Analyzers provide powerful options for acquiring data: the possibility to work either in Timing mode or in State mode, the possibility to use storage filters to memorize only desired data and the possibility to group input channels into items for better and advanced data analysis.

This chapter explains how to set up the various options and how to operate the PA24X user interface to achieve the best measurement results. For further information about Timing mode, State mode, and other acquisition topics, refer to the next chapter, "Acquiring Data".

Timing Mode (Internal Clock) Parameters

Timing mode is one of the two modes the analyzer can operate in (the other is State mode). Timing mode (or internal clock mode, or synchronous mode) is typically used for timing analysis—input signals are asynchronously sampled at a constant rate (determined by the timebase setting). Thus, the waveform window represents time on the x-axis in linear fashion and logical state on the y-axis.

To work in Timing mode, just select **Mode > Analyzer Mode > Timing** from the menu bar. To set up Timing mode properties, select **Mode > Timing Mode Properties**. The following dialog box will appear.



The Timing Mode Properties *Dialog Box*

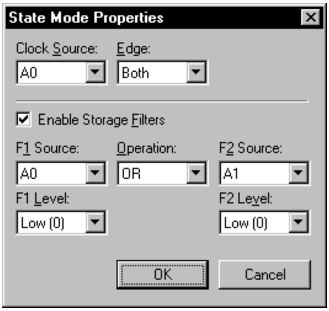
The T/Div setting represents the timebase value—that is, the amount of time represented by one grid division. The T/Div setting can also be quickly changed from the T/Div Knob on the Control Box panel. For more information about the scope of the T/Div setting, see the chapter *"Acquiring Data, Adjusting the Timebase/Zoom Setting"*.

Storage filters are explained later in this chapter.

State Mode (External Clock) Parameters

State mode is the other mode the analyzer can operate in. In State mode (or external clock mode, or asynchronous mode), you define which analyzer channel will be used as a sampling clock source. With a microprocessor, for example, instruction execution can be monitored by connecting the processor's data bus to the analyzer input channels, and connecting another input channel to a control signal which indicates when signals are valid on the data bus. The latter input channel can thus be usefully defined as the analyzer's sampling clock source.

To work in State mode, just select **Mode > Analyzer Mode > State** from the menu bar. To set up State mode properties, select **Mode > State Mode Properties**. The following dialog box will appear.



The State Mode Properties *Dialog Box*

In State mode, you define which channel supplies the sampling clock—additionally, the sampling edge must be specified as well.

After acquiring, the magnification of the *Waveform* window can be adjusted via the Zoom knob in the Control Box panel—again, please refer to the chapter *"Acquiring Data, Adjusting the Timebase/Zoom Setting"*.

Storage Filters

Storage filters, if enabled, allow the analyzer to memorize data only if the specified storage condition is met. Two storage filters (F1 and F2) are available, and both of them consist of a source channel and level specification (see the figures above). You can use one storage filter, two storage filters combined through a "OR" or "AND" specification, or none (default).
When no storage filter is used, all of the sampled data is stored on the analyzer's acquisition memory. When one or both storage filters are enabled, the sampled data is stored only if the filter condition is true.

Note that storage filters do not affect the operation of the trigger detector; the analyzer will look for the trigger event even if, due to the storage filters specification, data is not being stored. Storage filters only tell the analyzer whether to store the sampled data or not. Storage filters are available both is Timing and in State mode—however, they are typically used in State mode only.

Defining Items

What is an Item?

An item is a logical grouping defined to associate input channel(s) with their functions in the system under test. Each item groups one or more input channels into a single, logical entity—a generic channel, a bus, an asynchronous serial channel, a generic synchronous serial channel or an I^2C-bus. Special analysis is performed on serial items, in order to display the data they transfer rather than the logical level of their lines. Each item must be assigned a label and a color.

In the *Waveform* window, items occupy rows (item labels are displayed at the leftmost position on the window) while in the *Listing* window items occupy columns (item labels are displayed at the top of the window).

Note: only source channels belonging to items are displayed in the Waveform *and* Listing *windows.*

Adding/Removing/Ordering Items

To add an item, select the type of item you want to add from the **Item > Add** menu. The new item will be added at the bottom of existing items. If an acquisition took place before the item is added, the item's content will reflect the data sampled during the last acquisition.

To remove an item, select it by clicking on its label and then choose **Item > Delete**.

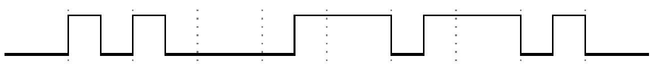
To change the position of an item with respect to other items, just drag it to the new desired location.

To review/edit item properties, select the desired item and choose **Item > Properties**.

Note: the Item menu can be displayed by simply right-clicking over an item's label.

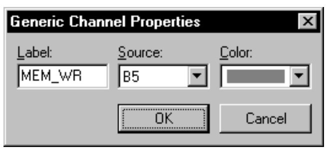
Generic Item Definition

A *generic channel item* is an item consisting of only one input channel. A generic channel item is displayed in the *Waveform* window as a standard logic waveform (to improve readability, the logic "0" level is rendered with a broad line).



Example of a Generic Item Rendering

To insert a generic channel item into the *Waveform* and *Listing* window, choose **Item > Add > Generic Channel**. The following dialog box will appear asking you to fill in the item's properties.



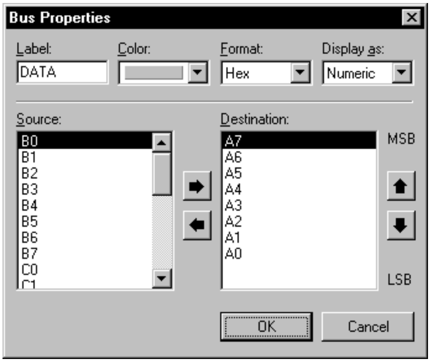
The Generic Channel Properties *Dialog Box*

Bus Item Definition

Bus definition allows you to group one or more input channels into an item. The analyzer will display the acquired data by grouping the input channels as specified. The order with which the input channels are grouped into a bus is used to determine the numeric value of the bus at each sampling, which is also displayed by the analyzer.

Bus items are best used to analyze microprocessor's data and address buses—and in general, when a set of inputs must be held together for meaningful representation.

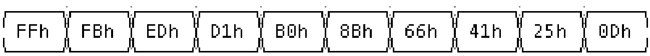
To insert a bus item into the *Waveform* and *Listing* window, choose **Item > Add > Bus**. The following dialog box will appear asking you to fill in the item's properties.



The Bus Properties *Dialog Box*

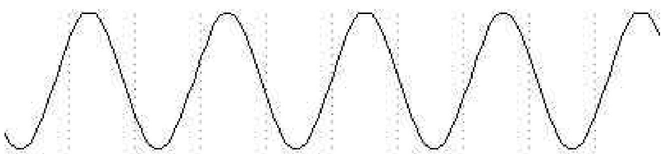
The "Display as" property determines how data is rendered on the *Waveform* window.

- When "Numeric" is selected, source channels are grouped as specified and the values of the bus are displayed as a character string using the specified output format (binary, hexadecimal, octal, decimal or ASCII).



Example of a Numeric Bus Item Rendering

- When "Graphic" is selected, the values the bus assumes at each sample are connected by straight lines. This representation is useful, for example, when you are monitoring the output of a parallel A/D converter.



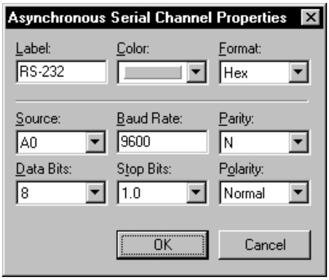
Example of a Graphic Bus Item Rendering

Asynchronous Serial Channel Item Definition

Asynchronous serial channel definition allows you to turn on asynchronous serial analysis for an input channel. You can use an asynchronous serial channel item to debug, for example, an RS-232 line.

Defining an item as an asynchronous serial channel causes the analyzer to perform a special analysis on that item (based on baud rate, data bits, parity bit and stop bits and signal polarity settings you specify).

To insert an asynchronous serial channel item into the *Waveform* and *Listing* window, choose **Item > Add > Serial Analysis > Asynchronous Channel**. The following dialog box will appear asking you to fill in the item's properties.



The Asynchronous Serial Channel Properties *Dialog Box*

Valid values for the Baud Rate setting are between 110 and 115200 BPS. The Data Bits setting ranges from 4 to 12, while the Stop Bits setting can assume the values 1.0, 1.5, 2.0.

Note: the analysis is performed starting from the trigger position, and as result each serial character detected is displayed. Starting from the trigger position, the analyzer will look for a start bit, followed by the specified number of data bits, followed by the specified number of stop bits. Each bit must respect the appropriate timing, determined by the Baud Rate setting.

Example of an Asynchronous Serial Channel Item Rendering

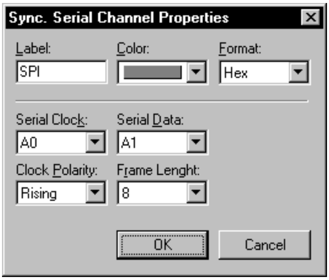
Note: in order for the analyzer to perform the analysis, the sampling period must be sufficiently low as to discriminate each bit of the serial channel. Also note that asynchronous serial analysis in only available in Timing mode.

Generic Synchronous Serial Channel Item Definition

Generic synchronous serial channel definition allows you to turn on synchronous serial analysis for a group of input channels. Defining an item as a generic synchronous serial channel causes the analyzer to perform a special analysis on that item (based on serial clock source, frame length and other settings you specify).

Generic synchronous serial channel items are useful to test, for example, SPI communication lines. Two input channels are used for a generic synchronous serial analysis: a serial clock source, together with its edge specification; and a serial data source. The clock source defines when serial data are valid.

To insert a generic synchronous serial channel item into the *Waveform* and *Listing* window, choose **Item > Add > Serial Analysis > Generic Synchronous Channel**. The following dialog box will appear asking you to fill in the item's properties.



The Generic Synchronous Serial Channel Properties Dialog Box

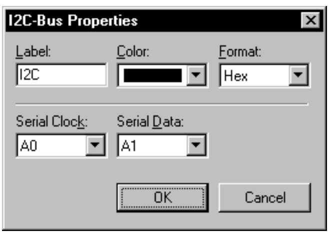
Note: the analysis is performed starting from the trigger position, and as result each serial character detected is displayed. The Frame Length setting determines how many bits build up a serial character.

Example of a Generic Synchronous Serial Channel Item Rendering

I^{2}C-Bus Item Definition

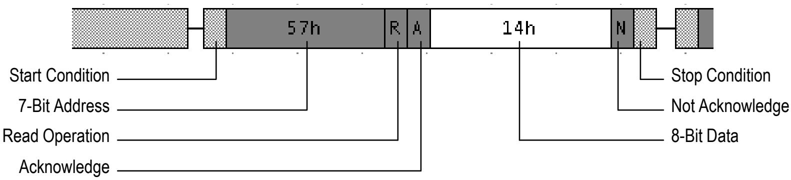
I^{2}C-bus definition allows you to turn on I^{2}C-bus analysis for a group of input channels. Defining an item as an I^{2}C-bus item causes the analyzer to perform a special analysis on that item (based on serial clock source and other settings you specify). The analysis is performed starting from the trigger position, and as result each I^{2}C-bus packet is automatically detected and displayed. Every component of an I^{2}C-bus packet (start and stop conditions, acknowledgement and read/write signals, address and data fields) is recognized and rendered appropriately.

To insert I^{2}C-bus item into the *Waveform* and *Listing* window, choose **Item > Add > Serial Analysis > I^{2}C-Bus**. The following dialog box will appear asking you to fill in the item's properties.



The I^{2}C-Bus Properties *Dialog Box*

Note: the analysis is performed starting from the trigger position, and as result each I²C-bus packet detected is displayed.



Example of an I^2C-Bus Item Rendering

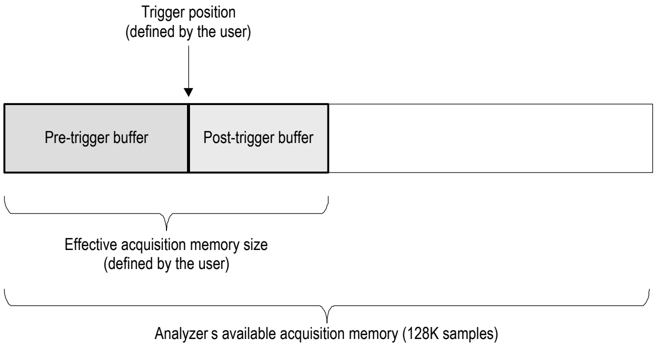
For more information about the I^2C-bus, please refer to *"Appendix C: I^2C-Bus Specifications"*.

4. Acquiring Data

Acquisition Memory, Pre- and Post-Trigger Buffer

The analyzer's acquisition memory is 128K samples deep; however, you can specify the amount of memory that will be effectively used during acquisition: 1K, 2K, 4K, 8K, 16K, 32K, 64K and 128K samples can be specified. Note that the deeper the acquisition memory, the slower the acquisition time (since more samples have to be acquired).

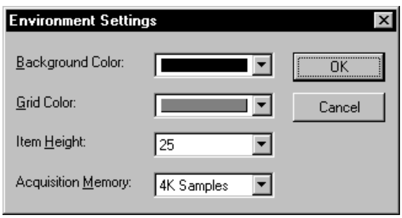
The figure below shows how the effective acquisition memory of the analyzer is divided into two variable parts: a pre-trigger buffer and a post-trigger buffer. The size of the pre-trigger buffer (and, consequently, that of the post-trigger buffer) is determined by the position of the trigger event.



Acquisition Memory, Pre- and Post-Trigger Buffer and Trigger Position

Defining Acquisition Memory and Trigger Position

The amount of acquisition memory can be specified in the *Environment Settings* dialog box (**Edit > Environment Settings**).



The Environment Settings *Dialog Box*

The trigger event position can be easily set by moving the trigger cursor in the *Waveform* window or in the *Listing* window. As a result, the new pre-trigger buffer and post-trigger buffer sizes are calculated and used when a new acquisition takes place.

Note: the trigger position affects how serial analysis is performed. In particular, PA24X performs serial analysis starting from the trigger position—before it, no analysis is performed. By changing the position of the trigger event the serial analysis results are updated to reflect the new trigger position, even if the analyzer is in Stop.

Timing Mode vs. State Mode

PA20 and PA40 Series Logic State Analyzers can work in two different operation modes: Timing mode and State mode. Depending on the system you are testing, and on the bug you are looking for, each mode has its pros and cons. It is then important to understand the differences between Timing mode and State mode in order to take advantage of them.

Timing Mode

In Timing mode (also called *asynchronous* sampling mode, or internal clock mode), the analyzer uses its internal clock generator, and the result is that it samples data with a different clock to that of the system under test. Timing mode is useful when you need to record the input data at a constant rate determined by a fixed timebase. As a result, the waveform display represents time in linear form on the X-axis and logical state on the Y-axis. This representation allows you to verify that the succession and timing of the signals in the circuit under test is that which you expected; furthermore, it is possible to find out any unwanted signal (*glitches*) which may lead to system malfunctions.

State Mode

In many cases (for example, when analyzing states in a state machine or when examining a microprocessor system) it's useful that the analyzer samples data *synchronously* with an external clock—i. e., works in State mode (also called *synchronous* sampling mode, or external clock mode). When sampling with an external clock, the analyzer turns off its internal clock and only uses the clock signal supplied at a given input channel. As a result, the waveform display still represents successive sampled data along the X-axis, but without timing information—since the supplied external signal may not have a constant period.

Even though in State mode you loose information about the time interval between samples, you have the benefit that the data you acquired is consistent with the external clock signal. This is useful when testing a synchronous system: typically, only the signal states concomitant with a master clock edge are meaningful; in State mode you can sample only those signals, without worrying about what happens elsewhere.

To change the analyzer mode, choose **Mode > Analyzer Mode** from the menu bar.

Note: when you change the analyzer mode, you loose the results of the last acquisition.

Starting and Stopping an Acquisition

During an acquisition, the logic analyzer examines the input voltage at each input probe at a sampling rate defined by the timebase setting (if in Timing mode) or by an external clock (if in State mode). At each sample, it compares the input voltage to the logic threshold. If the voltage is above the threshold, the analyzer stores a “1” in the acquisition memory; otherwise, it stores a “0”. When the acquisition memory is full, the PA24X user interface reconstructs the input waveform from the pattern of bits stored in the acquisition memory.

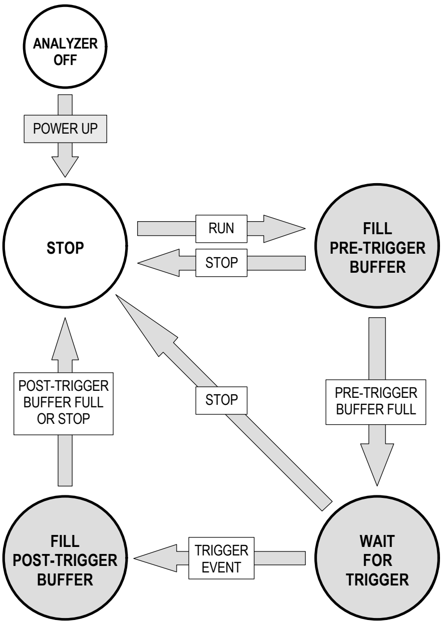
The PA24X user interface provides several ways to control the acquisition process. You can:

- Perform continuous acquisition by pressing the “Run” button;
- Make one acquisition, then stop, by pressing the “Single” button;
- Erase the results of the last acquisition by pressing the “Erase” button.

If you press the “Stop” button while the analyzer is acquiring, the following conditions may occur:

- If the pre-trigger buffer is not full, the acquisition is stopped but no data is displayed.
- If the pre-trigger buffer is full, but the trigger condition has not been found yet by the analyzer, you force a trigger condition and the acquisition stops. Only data before the trigger cursor are valid, and the analyzer displays them.
- Finally, if the pre-trigger buffer is full and the trigger condition has been found by the analyzer, the acquisition stops and the available acquired data is displayed.

When acquiring, the analyzer’s internal states evolve according to the flow diagram depicted below.



Analyzer Acquisition Loop

Note: the way the analyzer performs data acquisition depends on the trigger mode. For more information, see “Normal Mode vs Auto Mode” later in this chapter.

Autoscale

To quickly configure the analyzer, just press the “Autoscale” button in the Control Box panel. The Autoscale function will look for the appropriate Timebase/Zoom setting, so that the acquired data will be displayed properly. The Autoscale function is only available in Timing mode.

When you press the “Autoscale” button the analyzer will perform the following steps:

1. *The acquisition memory is set to 1K Samples; the trigger mode is set to “Auto”; storage filters are disabled.*
2. *The analyzer will perform an acquisition using a window of 50 ms, thus allowing it to recognize signals slower than 50 Hz for autoscaling.*
3. *If no channel activity is detected, the analyzer will exit the Autoscale function restoring previous settings.*
4. *Autoscaling attempts to find a timebase setting such that the slowest signal has less than 10 periods displayed; if such a signal is present, the analyzer will exit the Autoscale function—the analyzer will be set in RUN mode (continuous acquisition), storage filters will be disabled, trigger mode will be set to “Auto”, and all of the channels with detected activity will be displayed. Otherwise, if the minimum value of timebase was used, the analyzer will exit the Autoscale function restoring previous settings.*
5. *A smaller value of timebase will be used and the analyzer will return to step 4.*

Adjusting the Timebase/Zoom Setting

Adjusting the Timebase Settings

When the analyzer is in Timing mode, the T/Div knob (located on the Control Box panel) allows you to change the timebase value, that is the amount of time per division. The timebase ranges from 1 s to 50 ns in 1, 2, 5 increments.

Note: when the analyzer is in RUN mode, changing the T/Div setting causes a new acquisition to take place immediately with the new sampling period. If the T/Div setting is set to a higher value (slower sweep speed) the sample period is also lengthened, and it will take longer before a complete screen full of data is displayed. If the analyzer is in STOP mode, changing the T/Div setting causes the waveform to zoom in/out without acquiring any new data.

Note: the current timebase value affects other instrument operating parameters. For example:

- *The sampling period is a function of the T/Div setting (see table below), but changes only when a new acquisition begins.*

T/Div	Sample Period	T/Div	Sample Period	T/Div	Sample Period
1 s	25 ms	1 ms	25 ìs	1 ìs	50 ns (25 ns*)
500 ms	12.5 ms	500 ìs	12.5 ìs	500 ns	50 ns (25 ns*)
200 ms	5 ms	200 ìs	5 ìs	200 ns	50 ns (25 ns*)
100 ms	2.5 ms	100 ìs	2.5 ìs	100 ns	50 ns (25 ns*)
50 ms	1.25 ms	50 ìs	1.25 ìs	50 ns	50 ns (25 ns*)
20 ms	500 ìs	20 ìs	500 ns	* for 40 MSa/s analyzer models	
10 ms	250 ìs	10 ìs	250 ns		
5 ms	125 ìs	5 ìs	125 ns		
2 ms	50 ìs	2 ìs	50 ns		

Timebase Setting and Sampling Interval

- *To avoid missing fast events or short pulses, the analyzer automatically enables glitch capture once the T/Div setting is 5 ìs (on PA20 Series analyzer models) or 2 ìs (on PA40 Series analyzer models), or slower.*

Adjusting the Zoom Setting

When the analyzer is in State mode, the Control Box shows the Zoom knob instead of the T/Div knob. Since in State mode there is no information about the time interval between samples, it makes no sense to adjust the amount of time per division—instead, a zoom value can be modified to zoom in/out of the acquired data. Zoom values range from 1:200 (1 screen pixel corresponds to 200 samples) to 10:1 (10 screen pixels corresponds to 1 sample).

Normal Trigger Mode vs. Auto Trigger Mode

Normal Trigger Mode

In Normal trigger mode, the analyzer begins filling the pre-trigger buffer with data. As soon as the buffer is full, the analyzer will begin searching for the trigger event. While searching for the trigger, the analyzer overflows the pre-trigger buffer.

When the trigger event is found, the analyzer will fill the post-trigger buffer and display the acquisition memory. If the acquisition was initiated by the “Run” button, the process repeats.

Auto Trigger Mode

In Auto trigger mode, the analyzer fills the pre-trigger buffer, then searches for the trigger event for a predetermined interval. If no trigger is found, the analyzer forces a trigger and displays the data as though a trigger has occurred. If the acquisition was initiated by the “Run” button, the process is repeated.

To change the Trigger mode, check the “Auto” or “Normal” check box on the Control Box panel.

Defining the Trigger Event

PA20 and PA40 Series Logic State Analyzers allow you to synchronize the analyzer display to the actions of the circuit under test by defining a trigger condition. The analyzers offer three types of triggering, allowing you to match the complexity of the trigger to that of the data you want to capture. These trigger types are as follows:

- Edge trigger
- Pattern trigger
- Advanced trigger

Note: changes to the current trigger specification are handled in real-time. If the acquisition is stopped when you change a trigger specification, the analyzer uses the new specification as soon as you press the "Run" button or the "Single" button. If the analyzer is performing an acquisition when you change a trigger specification, it immediately begins a new acquisition using the new trigger definition.

Edge Trigger

In the Edge trigger, you either define a single raising or a falling edge that must be recognized on an input channel to satisfy the trigger condition. Edge trigger is best when there is a unique waveform edge that defines the event you wish to capture.

To define an Edge trigger:

1. Open the *Trigger Definition* dialog box, by selecting **Mode > Trigger Definition** command from the menu bar.
2. Check the "Edge Trigger" radio button.
3. In the "Source" drop-down list, select an input channel as the trigger source. In the "Edge" drop-down list, select the edge condition.

Note: when the "Edge Trigger" radio button is checked in the Trigger Definition *dialog box, the trigger source and edge can be specified directly from the Control Box, in the "Trigger" section.*

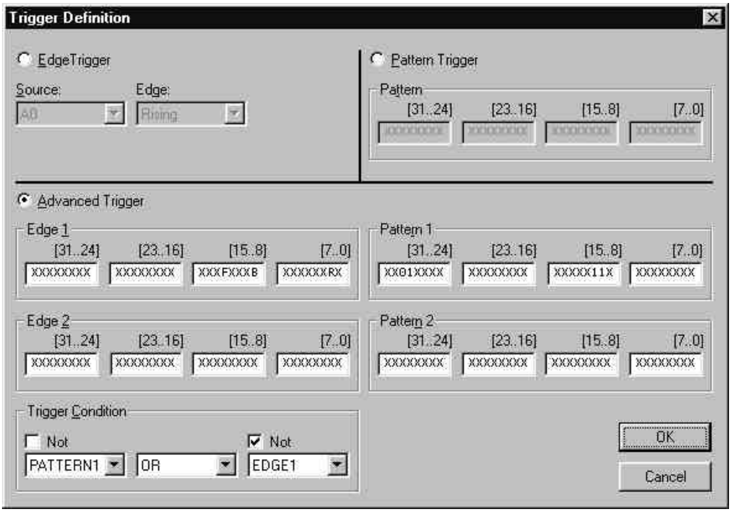
Pattern Trigger

Pattern trigger allows you to define a pattern of highs, lows and don't care inputs that must be recognized across the input channels during any given input sample. Pattern trigger is best when there is a unique pattern that occurs across a group of signals, and the pattern defines the events you wish to capture.

To define a Pattern trigger:

1. Open the *Trigger Definition* dialog, by selecting **Mode > Trigger Definition** command from the menu bar.
2. Check the "Pattern Trigger" radio button. By default, to each input channel is assigned a "don't care" level value ("X").
3. To specify a different logic level for a particular input channel, select the corresponding level character and type in the new desired level, either "0" or "1". To restore the "don't care" condition for a particular input channel, type "X" instead.

Note: all settings within the pattern are logically ANDed; that is, all conditions on the pattern must be specified before the analyzer will trigger.



The Trigger Definition *Dialog Box*

Advanced Trigger

Advanced trigger allows you to define up to two pattern and edge sources that are combined with a trigger operator to form the complete trigger specification. Advanced trigger is best when the events you wish to capture are defined by a complex series of waveform events in the system, and neither Pattern mode nor Edge mode are capable alone of clearly resolving the necessary sequence.

To define an Advanced trigger:

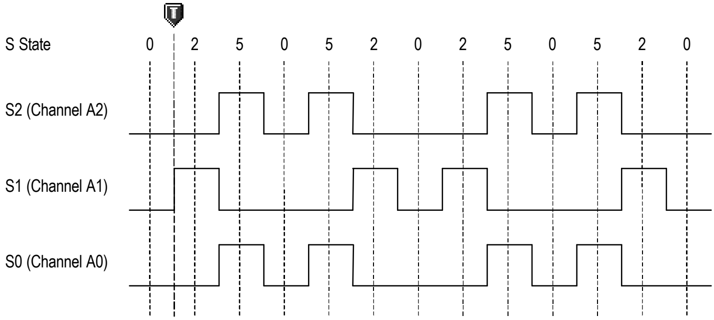
1. Open the *Trigger Definition* dialog box, by selecting **Mode > Trigger Definition** command from the menu bar.
2. Check the “Advanced Trigger” radio button.
3. Choose the trigger operator in the “Trigger Condition” section. Available operators are “OR”, “AND”, “THEN” and “OCCURRED”. The “OR”, “AND” and “THEN” operators require two sources (sources are explained below); the “OCCURRED” operator requires a source and a number. Additionally, no operator can be selected (“---”, thus allowing you to specify only one source).
4. Choose the source(s) for the trigger operator. A source can be EDGE1, EDGE2, PATTERN1 or PATTERN2. Source selection is done via the drop-down list boxes at the left and at the right of the trigger operator. Additionally, each source can be logically inverted by checking the “Not” check box.
5. Set up the source(s) by editing the content of the source(s) you specified in the “Trigger Condition” section. If, for example, you specified “PATTERN1 AND EDGE2” as trigger condition, you need to edit the content of both “Pattern1” and “Edge2” edit boxes. For pattern sources, allowed values are “0”, “1” and “X” (don’t care). For edge sources, allowed values are “R” (rising edge), “F” (falling edge), “B” (both edges) and “X” (don’t care).

Note: settings within pattern terms are logically ANDed. Settings within edge terms are logically ORed.

Advanced Trigger—Usage Examples

The following examples show how to take advantage of the advanced trigger capabilities of PA20 and PA40 Series Logic State Analyzers.

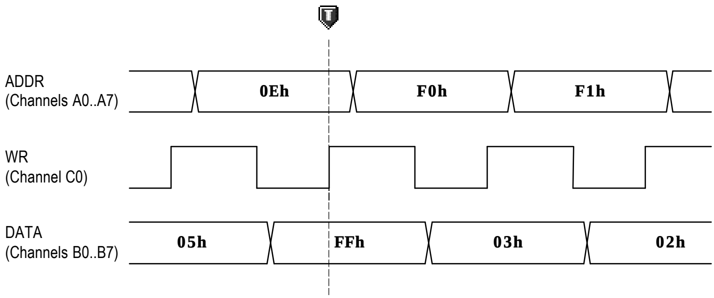
- Advanced trigger example: sequential output from a state machine. If you trigger on any of the single output patterns 0, 2, or 5, the display will not be stable. However, you can use the "THEN" operator to trigger on the sequence of patterns "0h" followed by pattern "2h".




```
Advanced Trigger
Pattern 1 = XXXX XXXX XXXX XXXX XXXX XXXX XXXX X000
Pattern 2 = XXXX XXXX XXXX XXXX XXXX XXXX XXXX X010
Trigger Condition = Pattern 1 THEN Pattern 2
```

Advanced Trigger Example: Sequential Output From a State Machine

- Advanced trigger example: on a micocontroller-based system, you want to trigger on a specific RAM write event, for example when a write cycle is performed on the address 0Eh. For this purpose, you can use the "AND" operator to trigger on the pattern "0Eh" and, simultaneously, the WR signal edge is raising.




```
Advanced Trigger
Pattern 1 = XXXX XXXX XXXX XXXX XXXX XXXX 0000 0111
Edge 1    = XXXX XXXX XXXX XXXR XXXX XXXX XXXX XXXX
Trigger Condition = Pattern 1 AND Edge 1
```

Advanced Trigger Example: RAM Write Event Capture

Trigger Notes

In either Auto or Normal mode, the trigger may be missed completely under certain conditions. This is because the analyzer will not recognize a trigger event until the pre-trigger buffer is full. If, for example, you are working in Timing mode and set the T/Div knob to a slow speed, such as 500 ms/div, and the trigger condition always occurs before the analyzer has filled the pre-trigger buffer, the trigger will never be found.

Some measurements you wish to make will require you to take some action in the circuit under test to cause the trigger event. Usually, these are single-shot acquisitions, where you will use the “Single” button. On the PA24X user interface status bar, a progress indicator shows you the analyzer status—in particular, it shows you when the pre-trigger buffer is full and the analyzer is ready to recognize the trigger event. If you wait for this condition before causing the action in the circuit, the analyzer will always find the trigger condition correctly.

Trigger Output

PA20 and PA40 Series Logic State Analyzers provide a feature that helps extend their triggering capabilities and allows you to use them with other instruments: the Trigger Output.

The Trigger Output signal is generated when the analyzer’s trigger condition is satisfied, and it is available at the BNC connector on the analyzer’s front panel; thus, it can be used to trigger an external measurement system or other device. For example, you may want to use the Trigger Output signal to trigger an oscilloscope. The Trigger Output signal cannot be viewed on the waveform display.

A rising edge of the Trigger Output signal indicates that the trigger condition was satisfied.

Glitch Detection

In digital system design, a *glitch* is an unintentional or unexpected signal transition, which may or may not pass through the logic threshold. The analyzer provides support for capturing glitches during acquisition. In particular, the analyzer considers a glitch to be any set of two or more edges that pass through the logic threshold and fall between logic analyzer samples.

Why Glitch Capture?

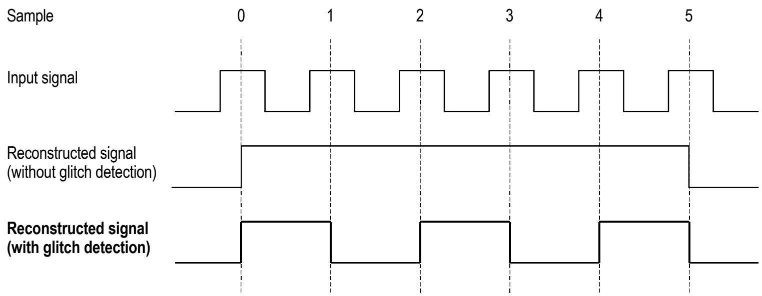
When the analyzer is sampling at its maximum rate (that is, with a sampling period of 50 ns or 25 ns, depending on the analyzer model), all pulses within the bandwith of the probes will be captured by the analyzer. As the sweep speed is decreased, the sample period is increased to make best use of acquisition memory. A longer sample period increases the probability that a pulse will fall between samples, and will therefore be missed.

To prevent missing pulses, the analyzer automatically enables glitch capture once the T/Div setting is 5 ìs (on PA20 Series analyzer models) or 2 ìs (on PA40 Series analyzer models), or slower. Thus, because of the glitch capture circuitry, the analyzer can capture pulses as narrow as 50 ns (on PA20 Series analyzer models) or 25 ns (on PA40 Series analyzer models).

Please note that glitch capture only makes sense in Timing mode: in State mode, the analyzer cannot recognize signal transition between samples.

Aliasing and Glitch Capture

In a sampled device, aliasing occurs when the same set of sampled data could be used to reconstruct many different waveforms, because of insufficient data. In the example below, because of the insufficient sampling rate, the reconstructed waveform (without glitch detection) would not contain information about the input signal transition. Using glitch detection, however, helps preventing aliasing by recognizing additional transitions that occur between samples. The third waveform in the figure below indicates how the analyzer will reconstruct the input signal. In addition, the analyzer will turn on a "•" mark on the corresponding item's label, indicating that aliasing occurred. This information tells you that the input signal has been acquired at an insufficient sampling rate; you should decrease the T/Div setting and acquire the signal again.

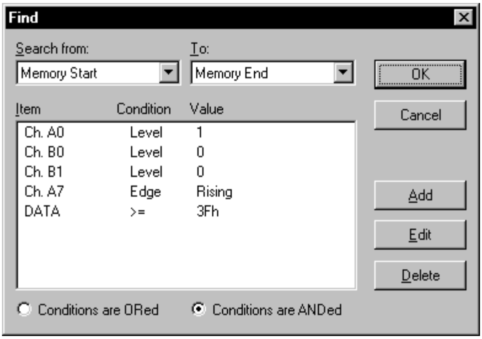


Effects of Glitch Detection on Reconstructing an Input Signal

Searching for a Specific Event

PA20 and PA40 Series Logic State Analyzers provide a powerful search engine that helps locate particular events after data has been acquired. The following dialog box illustrates an example of search conditions.

To search for a specific event, open the *Find* dialog box (**Edit > Find**) and specify one or more search condition by clicking the "Add" button. You will be asked for a source item and a search condition. Depending on the type of item you specify as a source for your search (generic channel, bus or serial item), the appropriate search conditions will display accordingly.



The Find *Dialog Box*

The *Find* dialog box allows you to group all of your search condition through an “OR” or “AND” operator, as well as to specify the scope of the search.

When you are finished specifying your search conditions, click the “OK” button. The analyzer will look for the first match for your search criteria. To look for successive matches, select **Edit > Find Next** from the menu bar.

When a match is found, Cursor 2 jumps to its exact position.

5. Printing/Exporting Acquired Data

Printing the Acquired Data

The content of the *Waveform* and *Listing* window can be printed via the **File > Print** command. The **File > Print Preview** command shows a preview of what will be printed.

Exporting the Waveform Display to Other Applications

The **Edit > Copy to Clipboard** command places a copy of the current *Waveform* or *Listing* window (whichever is the current window) content into the Windows clipboard, in bitmap format. From within other applications you can paste this image and include it in your documents.

Exporting the Acquired Data in ASCII Format

PA24X user interface allows you to export the acquired data into an ASCII format output file (via the **File > Export as ASCII Data** command). The file contents are similar to the contents of the *Listing* window. The resulting file can be used by a user application to perform user-specific operations on the acquired data.

The item properties (source channel grouping and display format) are preserved when data is exported. The output file has a ".wtd" extension.

Saving/Loading the Acquired Data and Environment Settings

PA24X user interface automatically saves environment settings (including window position, analyzer settings, etc.) when the application is closed. However, by default, the acquired data is not saved.

The **File > Save Waveform Data As** command allows you to save the acquired data along with all of the other analyzer and user interface settings. You can therefore save interesting acquisition sessions for later analysis. To reload a previously saved session, just select **File > Load Waveform Data** from the menu bar.

6. Troubleshooting

Installation Issues

Installing the PA24X User Interface Under Windows NT

In order to install the PA24X under Windows NT, you must have logged in as *Administrator*.

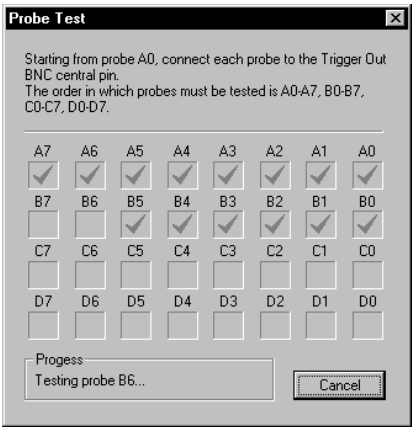
Parallel Port Topics

The analyzer communicates with the PC trough a standard, bidirectional parallel port. To check if your parallel port is compatible with the analyzer, reboot your PC, enter the BIOS utility and make sure that the parallel mode is set to SPP/bidirectional.

If your parallel port is not fully compatible with the required hardware, the 237-00139 ISA Bus Bidirectional Parallel Port item, supplied together with the analyzer, can be installed on your PC for proper communication with the analyzer.

Probe Test

PA20 and PA40 Series Logic State Analyzers include a probe test to verify the electrical continuity of the probe sets. To start the test, choose **Edit > Probe Test** from the menu bar. The following dialog box will appear.



The Probe Test *Dialog Box*

During the probe test, a special signal is present at the Trigger Output BNC. Connect in turn (in the correct order) each grabber to the central pin of the BNC, and the PA24X user interface will check for the probe cable integrity.

Error Messages

Most of the PA24X user interface error messages come together with an explanation and a suggestion on how to solve the problem. In case you can't still solve your problem, please contact our Technical Support.

Getting Technical Support

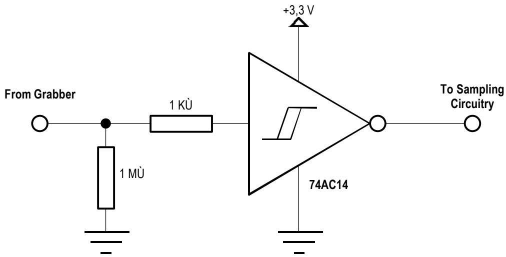
Technical assistance is provided free to all customers. For technical assistance, documentation and information about products and services, please refer to your local SofTec Microsystems partner.

SofTec Microsystems offers its customers a free technical support service at *support@softecmicro.com*.

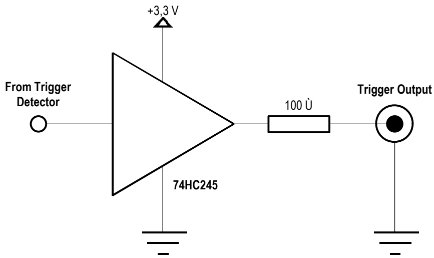
Appendix A: Technical Specifications

	PA2016A	PA2032A	PA4016A	PA4032A
General Specifications				
Number of Channels	**16**	**32**	**16**	**32**
Sampling Speed (MSa/s)	**20**	**20**	**40**	**40**
Acquisition Memory (K Samples per channel)	**128K**	**128K**	**128K**	**128K**
PC Connection	**Parallel Port**	**Parallel Port**	**Parallel Port**	**Parallel Port**
Acquisition and Analysis				
User Interface	**Windows 95/98/NT**	**Windows 95/98/NT**	**Windows 95/98/NT**	**Windows 95/98/NT**
Timing Mode				
State Mode				
Storage Filters				
Autoscaling				
Glitch Capture				
Bus Grouping				
Asynchronous Serial Channel Analysis				
Generic Synchronous Serial ChannelAnalysis				
I^2C-Bus Analysis				
Advanced Trigger				
Trigger Output Connector				
Electrical Characteristics				
Input Threshold Low (V)	**1.1**	**1.1**	**1.1**	**1.1**
Input Threshold High (V)	**2.0**	**2.0**	**2.0**	**2.0**
Input Hysteresis (V)	**0.9**	**0.9**	**0.9**	**0.9**
Input Leakage Current (mA)	**1**	**1**	**1**	**1**
Input Absolute Maximum Voltage (V)	**+/-20**	**+/-20**	**+/-20**	**+/-20**
Power Supply Voltage (V DC)	**9 to 15**	**9 to 15**	**9 to 15**	**9 to 15**
Power Supply Current (mA)	**500**	**500**	**500**	**500**
Dimensions				
Dimensions L x W x H (mm)	**190 x 100 x 40**	**190 x 100 x 40**	**190 x 100 x 40**	**190 x 100 x 40**
Weight Without Probes (g)	**300**	**300**	**300**	**300**
Probe Cable Length (mm)	**300**	**300**	**300**	**300**
Accessories Included in the Standard Package				
Logic Analyzer 16-Channel Cable				
AC/DC Switching Power Adapter				
SMD Grabber (set of 17 pcs)				
ISA Bus Bidirectional Parallel Port				
PA24X User Interface Software				
User's Manual				
Optional Accessories				
Communication DLL				
Ultra Fine Pitch Grabber				

PA20 and PA40 Series Logic State Analyzers Specification Summary



Probe Input Circuit



Trigger Output Circuit

Appendix B: Interfacing the Analyzer With Your Application

PA20 and PA40 Series Logic State Analyzers can be controlled without the PA24X user interface. A Communication DLL (Dynamic Link Library) for Windows 95/98/NT (order code 131-00137) is available from SofTec Microsystems. The Communication DLL allows you to set up the instrument and perform data acquisition from within your application. The Communication DLL contains C written routines, and can be used to interface the analyzer from within, for example, a Microsoft® Visual C® or Visual Basic® application, as well as any other programming language that supports the DLL mechanism.

The Communication DLL contains:

- A .dll file (the main DLL file);
- A .lib file (the library file for Visual C applications);
- A .h file (the include file for Visual C applications);
- A .bas file (the include file for Visual Basic applications)
- Complete documentation in Adobe® Acrobat® PDF format.

The prototypes of the available functions included on the Communication DLL are summarized below, in alphabetical order.

```
BOOL PA_CloseCommunication (void);
BOOL PA_GetAcquisitionBufferSize (unsigned long *size);
BOOL PA_GetAddressOnTheFly (unsigned long *addr);
void PA_GetAnalyzerError (unsigned char *errors);
BOOL PA_GetAnalyzerMode (int *mode);
BOOL PA_GetAnalyzerStatus (unsigned char *status);
BOOL PA_GetExternalClock (int *ch, int *edge);
BOOL PA_GetGlitch (unsigned long *glitch);
BOOL PA_GetInstrumentID (unsigned char *id);
unsigned int PA_GetLPTBaseAddress (unsigned char nLPT);
BOOL PA_GetNumAcquiredSamples (unsigned long *nsamples);
int  PA_GetNumLPTPorts (void);
BOOL PA_GetPreTriggerBufferSize (unsigned long *size);
BOOL PA_GetStorageFilter (PA_FILTERS *f);
BOOL PA_GetTimebase (unsigned long *time_ns);
BOOL PA_GetTrigger (PA_TRIGGER *trg);
BOOL PA_InitInstrument (unsigned int boardaddr, unsigned char ramtest,
      PROGRESSPROC *callback);
BOOL PA_ReadBuffer (unsigned long start_addr, unsigned long *data_buf,
      unsigned long size);
BOOL PA_SetAcquisitionBufferSize (unsigned long size);
BOOL PA_SetAnalyzerMode (int mode);
BOOL PA_SetExternalClock (int ch, int edge);
```

```
BOOL PA_SetPreTriggerBufferSize (unsigned long size);
BOOL PA_SetStorageFilter (PA_FILTERS *f);
BOOL PA_SetTimebase (unsigned long time_ns);
BOOL PA_SetTrigger (PA_TRIGGER *trg);
BOOL PA_StartAcquisition (void);
BOOL PA_StopAcquisition (void);
BOOL PA_TestTrgOut (unsigned char mode);
```

Appendix C: I^2C-Bus Specifications

For detailed information, refer to Philips Semiconductors, "The I^2C-Bus Specification", v. 2.0, December 1998.

Introduction

The I^2C-bus is a serial bus for 8-bit oriented digital control applications. Two wires, serial data (SDA) and serial clock (SCL), carry information between the devices connected to the bus. Each device is recognized by a unique address (whether it's a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. Obviously an LCD driver is only a receiver, whereas a memory can both receive and transmit data. In addition to transmitters and receivers, devices can also be considered as masters or slaves when performing data transfers (see the table below). A master is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

Term	**Description**
Transmitter	The device which sends data to the bus
Receiver	The device which receives data from the bus
Master	The device which initiates a transfer, generates clock signals and terminates a transfer
Slave	The device addressed by a master
Multi-master	More than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	Procedure to ensure that, if more than one master simultaneously tries to control the bus, only one is allowed to do so and the winning message is not corrupted
Synchronization	Procedure to synchronize the clock signals of two or more devices

Definition of I^2C-Bus Terminology

The I^2C-bus is a multi-master bus. This means that more than one device capable of controlling the bus can be connected to it. As masters are usually micro-controllers, let's consider the case of a data transfer between two microcontrollers connected to the I^2C-bus. The transfer of data would proceed as follows:

1. Suppose microcontroller A wants to send information to microcontroller B:
   - microcontroller A (master), addresses microcontroller B (slave)
   - microcontroller A (master-transmitter), sends data to microcontroller B (slave-receiver)
   - microcontroller A terminates the transfer
2. If microcontroller A wants to receive information from microcontroller B:
   - microcontroller A (master) addresses microcontroller B (slave)
   - microcontroller A (master-receiver) receives data from microcontroller B (slave-transmitter)
   - microcontroller A terminates the transfer. Even in this case, the master (microcontroller A) generates the timing and terminates the transfer.

The possibility of connecting more than one microcontroller to the I^2C-bus means that more than one master could try to initiate a data transfer at the same time. To avoid the chaos that might ensue from such an event—an arbitration procedure has been developed. This procedure relies on the wired-AND connection of all I^2C interfaces to the I^2C-bus. If two or more masters try to put information onto the bus, the first to produce a 'one' when the other produces a 'zero' will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line.

Generation of clock signals on the I^2C-bus is always the responsibility of master devices; each master generates its own clock signals when transferring data on the bus. Bus clock signals from a master can only be altered when they are stretched by a slow-slave device holding-down the clock line, or by another master when arbitration occurs.

Both SDA and SCL are bi-directional lines, connected to a positive supply voltage via a current-source or pull-up resistor. When the bus is free, both lines are HIGH. The output stages of devices connected to the bus must have an open-drain or open-collector to perform the wired-AND function.

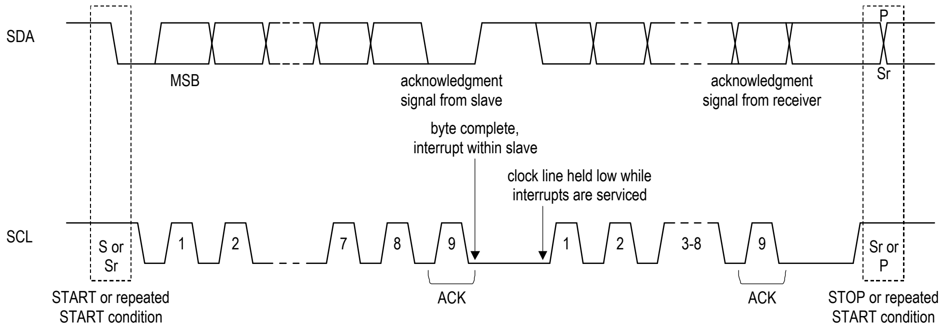
Transferring Data

Byte Format

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL. In some cases, it's permitted to use a different format from the I^2C-bus format (for CBUS compatible devices for example). A message that starts with such an address can be terminated by generation of a STOP condition, even during the transmission of a byte. In this case, no acknowledge is generated.

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. Of course, set-up and hold times must also be taken into account. Usually, a receiver which has been addressed is obliged to generate an acknowledge after each byte has been received, except when the message starts with a CBUS address When a slave doesn't acknowledge the slave address (for example, it's unable to receive or transmit because it's performing some real-time function), the data line must be left HIGH by the slave. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer. If a slave-receiver does acknowledge the slave address but, some time later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave generating the not-acknowledge on the first byte to follow. The slave leaves the data line HIGH and the master generates a STOP or a repeated START condition. If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave-transmitter must release the data line to allow the master to generate a STOP or repeated START condition.



Data Transfer on the I²C-Bus

Arbitration and Clock Generation

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I²C-bus. Data is only valid during the HIGH period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedure to take place. Clock synchronization is performed using the wired-AND connection of I²C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and, once a device clock has gone LOW, it will hold the SCL line in that state until the clock HIGH state is reached. However, the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. The SCL line will therefore be held LOW by the device with the longest LOW period. Devices with shorter LOW periods enter a HIGH wait-state during this time. When all devices concerned have counted off their LOW period, the clock line will be released and go HIGH. There will then be no difference between the device clocks and the state of the SCL line, and all the devices will start counting their HIGH periods. The first device to complete its HIGH period will again pull the SCL line LOW. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition within the minimum hold time of the START condition that results in a defined START condition to the bus. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output stage because the level on the bus doesn’t correspond to its own level. Arbitration can continue for many bits. Its first stage is comparison of the address bits. If the masters are each trying to address the same device, arbitration continues with comparison of the data-bits if they are master-transmitter, or acknowledge-bits if they are master-receiver. Because address and data information on the I²C-bus is determined by the winning master, no information is lost during the arbitration process. A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration. As an Hs-mode master has a unique 8-bit master code, it will always finish the arbitration during the first byte. If a master also incorporates a slave function and it loses arbitration during the addressing stage, it's possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode. Since control of the I^2C-bus is decided solely on the address or master code and data sent by competing masters, there is no central master, nor any order of priority on the bus. Special attention must be paid if, during a serial transfer, the arbitration procedure is still in progress at the moment when a repeated START condition or a STOP condition is transmitted to the I^2C-bus. If it's possible for such a situation to occur, the masters involved must send this repeated START condition or STOP condition at the same position in the format frame. In other words, arbitration isn't allowed between:

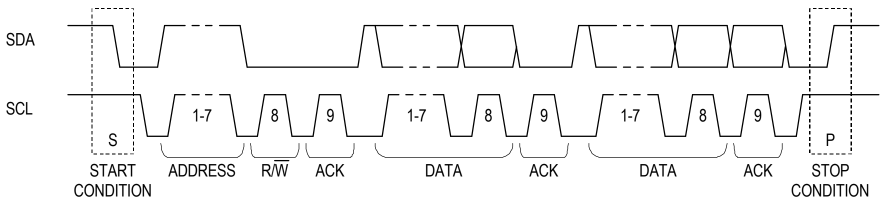
- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition. Slaves are not involved in the arbitration procedure.

Use of the Clock Synchronizing Mechanism as a Handshake

In addition to being used during the arbitration procedure, the clock synchronization mechanism can be used to enable receivers to cope with fast data transfers, on either a byte level or a bit level. On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. Slaves can then hold the SCL line LOW after reception and acknowledgment of a byte to force the master into a wait state until the slave is ready for the next byte transfer in a type of handshake procedure. On the bit level, a device such as a microcontroller with or without limited hardware for the I^2C-bus, can slow down the bus clock by extending each clock LOW period. The speed of any master is thereby adapted to the internal operating rate of this device. In Hs-mode, this handshake feature can only be used on byte level.

Formats With 7-Bit Addresses

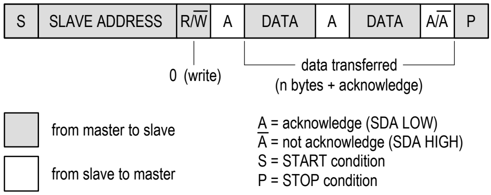
Data transfers follow the format shown in the figure below. After the START condition (S), a slave address is sent. This address is 7 bits long followed by an eighth bit which is a data direction bit (R/W)—a 'zero' indicates a transmission (WRITE), a 'one' indicates a request for data (READ). A data transfer is always terminated by a STOP condition (P) generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition (Sr) and address another slave without first generating a STOP condition. Various combinations of read/write formats are then possible within such a transfer.



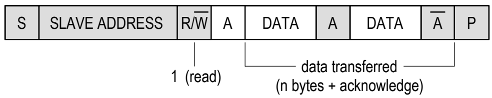
A Complete Data Transfer (7-Bit Addresses Format)

Possible data transfer formats are:

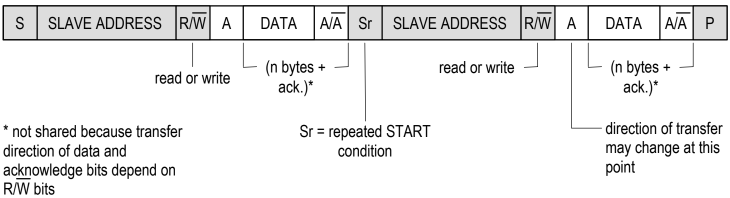
- Master-transmitter transmits to slave-receiver. The transfer direction is not changed.
- Master reads slave immediately after first byte. At the moment of the first acknowledge, the master-transmitter becomes a master-receiver and the slave-receiver becomes a slave-transmitter. This first acknowledge is still generated by the slave. The STOP condition is generated by the master, which has previously sent a not-acknowledge (A).
- Combined format. During a change of direction within a transfer, the START condition and the slave address are both repeated, but with the R/W bit reversed. If a master receiver sends a repeated START condition, it has previously sent a not-acknowledge (A).



A Master-Transmitter Addressing a Slave Receiver With a 7-Bit Address



A Master Reads a Slave Immediately After the First Byte



Combined Format

7-Bit Addressing

The addressing procedure for the I^{2}C-bus is such that the first byte after the START condition usually determines which slave will be selected by the master. The exception is the ‘general call’ address that can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge. However, devices can be made to ignore this address. The second byte of the general call address then defines the action to be taken.

Definition of Bits in the First Byte

The first seven bits of the first byte make up the slave address. The eighth bit is the LSB (least significant bit). It determines the direction of the message. A ‘zero’ in the least significant position of the first byte means that the master will write information to a selected slave. A ‘one’ in this position means that the master will read information from the slave. When an address is sent, each device in a system compares the first seven bits after the START condition with its address. If they match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter, depending on the R/W bit. A slave address can be made-up of a fixed and a programmable part. Since it’s likely that there will be several identical devices in a system, the programmable part of the slave address enables the maximum possible number of such devices to be connected to the I^2C-bus. The number of programmable address bits of a device depends on the number of pins available. For example, if a device has 4 fixed and 3 programmable address bits, a total of 8 identical devices can be connected to the same bus. The I^2C-bus committee coordinates allocation of I^2C addresses. Two groups of eight addresses (0000XXX and 1111XXX) are reserved for the purposes shown in the table below. The bit combination 11110XX of the slave address is reserved for 10-bit addressing.

Slave Address	**R/W Bit**	**Description**
0000 000	**0**	General call address
0000 000	**1**	START byte
0000 001	**X**	CBUS address
0000 010	**X**	Reserved for different bus format
0000 011	**X**	Reserved for future purposes
0000 1XX	**X**	Hs-mode master code
1111 1XX	**X**	Reserved for future purposes
1111 0XX	**X**	10-bit slave addressing

Definition of Bits in the First Byte

Glossary

Acquisition is the process by which the logic analyzer collects data. The analyzer acquires data by sampling the voltage on each input channel and comparing it to the threshold voltage to determine whether the input is high or low. In Normal trigger mode, the analyzer begins filling the pre-trigger buffer with data. When that buffer is full, the analyzer will begin searching for the trigger event. In Auto trigger mode, the analyzer fills the pre-trigger buffer, then searches for the trigger event for a predetermined interval. The acquisition process also depends on the analyzer's mode: in Timing mode, the sampling clock is generated internally, while in State mode an external clock is used. Additionally, storage filters can be defined to control under which condition sampled data is effectively stored into the acquisition memory.

Acquisition memory stores the acquired sample. The acquisition memory is divided into a pre-trigger buffer and a post-trigger buffer; the trigger position can be freely defined.

Acquisition memory depth is 16/32 bits wide (depending on the analyzer model being used) and up to 128K samples deep. The memory deep can be adjusted depending on the characteristics of the acquisition to be performed, from a minimum value of 1K samples to a maximum value of 128K samples.

Advanced trigger consists of two pattern and two edge sources combined with a variety of operators to form the complete advanced trigger specification. Advanced trigger is best used when the events you want to capture are defined by a complex series of waveform events in the system, and neither edge trigger nor pattern trigger modes are capable of clearly resolving the necessary sequence.

Asynchronous serial channel definition allows you to turn on asynchronous serial analysis for an input channel. Defining an item as an asynchronous serial channel causes the analyzer to perform a special analysis on that item (based on baud rate, data bits, parity bit, stop bits and idle state settings you specify). The analysis is performed starting from the trigger position, and as result each serial character detected is displayed.

Auto trigger mode causes the analyzer to fill the pre-trigger buffer, then searches for the trigger event for a predetermined interval. If no trigger is found, the analyzer forces a trigger and displays the data as though the trigger had occurred.

Bus definition allows you to group one or more input channels into an item. The analyzer will display the acquired data by grouping the input channels as specified. The order with which the input channels build a bus is used to determine the numeric value of bus at each sampling, which is also displayed by the analyzer.

Control Box panel is the panel placed at the rightmost side of the analyzer's user interface main window. The Control Box panel contains the most commonly used commands, such as the T/Div or Zoom knob, the Run/Stop/Single commands, the trigger mode selection (Auto/Normal), the edge trigger definition, etc.

Cursors are used to measure either time (or number of sampled data) between particular points on the display or numeric values of the currently displayed waveforms. Cursors are also used when searching for a specific pattern on the acquisition memory. There are three cursors available: the trigger cursor (which defines the position of the trigger event in the acquisition memory), and the cursors t1 and t2, which are used to perform measures on the acquired data.

Data format defines how the analyzer displays item's data. For each type of item (but the generic channel item) you can choose whiter to display data in hexadecimal, decimal, binary, octal or ASCII format.

DLL (dynamic link library) is a library of functions that uses dynamic linking. This allows an executable module to include only the information needed at run time to locate the executable code for a DLL function.

Edge trigger is a single rising or falling edge (or both) that must be recognized on an input channel to satisfy the trigger condition.

Generic channel is an item consisting of only one input channel. A generic channel item is displayed in the *Waveform* window as a standard logic waveform.

Glitch is, in a digital system, an unintentional or unexpected signal transition. The analyzer considers a glitch to be any set of two or more edges that pass through the logic threshold and fall between logic analyzer samples.

Glitch mode is indicated by "GL" on the status line. To prevent missing glitch pulses, the analyzer automatically enables glitch capture at T/Div settings greater than 5 ìs (on PA20 Series analyzer models) or 2 ìs (on PA40 Series analyzer models). The analyzer can capture pulses as narrow as 50 ns (on PA20 Series analyzer models) or 25 (on PA40 Series analyzer models). If at least one glitch has been detected on an item, an indicator is turned on that item's label.

Hold time is the time for which a data input to a clocked device must remain stable after the active edge of the clock occurs. Violating the hold time specification of a device can lead to unstable circuit operation.

Generic synchronous serial channel definition allows you to turn on synchronous serial analysis for a group of input channels. Defining an item as a generic synchronous serial channel causes the analyzer to perform a special analysis on that item (based on serial clock source, frame length and other settings you specify). The analysis is performed starting from the trigger position, and as result each serial character detected is displayed.

I^2C-bus definition allows you to turn on I^2C-bus analysis for a group of input channels. Defining an item as an I^2C-bus causes the analyzer to perform a special analysis on that item (based on serial clock source and other settings you specify). The analysis is performed starting from the trigger position, and as result each I^2C-bus packet is automatically detected and displayed. Every component of an I^2C-bus packet (start and stop conditions, acknowledgement and read/write signals, address and data fields) is recognized and rendered appropriately.

Input channels are the analyzer's inputs. Input channels are labeled A0…A7, B0…B7, C0…C7, D0…D7 (C and D groups are only available on 32-channel analyzer models). Input channels can be grouped into items for useful data representation and analysis.

Items are logical groupings defined to associate input channel(s) with their functions in the system under test. Each item groups one or more input channels into a single, logical entity—a generic channel, a bus, an asynchronous serial channel, a generic synchronous serial channel or an I^2C-bus. Special analysis is performed on serial items, in order to display the data they transfer rather than the logical level of their lines. Each item can be assigned a label and a color.

***Listing* window** provides an alternative presentation for sampled data. The sampled data is displayed as a list of numerical logic values instead of a waveform graphical representation.

Memory bar is the horizontal, red line positioned below the waveform window (and at the right side of the listing window) which shows the amount of acquisition memory that is displayed, and the position of the display with respect to acquisition memory. When the analyzer is stopped, you can change the size of the display with respect to acquisition memory using the T/Div (if you are working in Timing mode) or Zoom (if you are working in State mode) control knob, which allows you to zoom in on a specific portion of memory.

Normal trigger mode causes the analyzer to begin filling the pre-trigger buffer with data. When the buffer is full, the analyzer will begin searching for the trigger event. When the trigger event is found, the analyzer fills the post-trigger buffer and displays the acquisition memory. Until the analyzer finds the trigger, the acquisition will not be displayed, and the analyzer will search for the trigger indefinitely.

Occurrence trigger operator is used to specify the number of times an edge term or a pattern term occurs. Occurrence operators are available in advance trigger mode.

Pattern trigger defines a set of pattern of highs, lows, and don't care inputs that must be recognized across the input channels during any given input sample. The pattern trigger occurs when the pattern is entered.

Sample period is the interval (in Timing mode) at which the input is sampled. The analyzer acquires data by sampling the voltage on each input channel and comparing it to the threshold voltage to determine whether the input was a logic high or logic low.

Serial data analysis is a set of algorithms that allows the analyzer to perform a special analysis on the raw acquired data. By defining an item as one of the three available serial analysis functions (asynchronous serial channel, generic synchronous serial channel, I2C-bus), the analyzer will automatically display the appropriate serial character or packet characteristics.

Setup time is the time for which a data input to a clocked device must remain stable before the active edge of the clock occurs. Violating the hold time specification of a device can lead to unstable circuit operation.

Single shot acquisition indicates that the analyzer performs a single acquisition and then stops. When the Run command is invoked, the analyzer performs an acquisition, displays the acquired data, performs another acquisition, etc. Contrariwise, the Single command instructs the analyzer to only perform a single acquisition.

Status indicators are at the bottom of the waveform and listing windows and display information about the sampling interval, trigger position, cursor positions, glitch mode, etc.

State mode is one of the two modes the analyzer can operate in (the other is Timing mode). In State mode (or external clock mode), you define which analyzer channel will be used as a sampling clock source. With a microprocessor, for example, instruction execution can be monitored by connecting the processor's data bus to the analyzer input channels, and connecting another input channel to a control signal which indicates when signals are valid on the data bus. The latter input channel can thus be usefully defined as the analyzer's sampling clock source.

Storage filters control input data storage. They work both in Timing mode and State mode and can be used to control under which condition sampled data is effectively stored into the acquisition memory. When one or both of the storage filters is enabled, data will be stored by the analyzer only if the signal connected to the enabled filter is logically true. Storage filters do not affect the operation of the trigger detector; they only control the data memorization.

Sweep speed (timebase) is the amount of time per division on the waveform window (when the analyzer is in Timing mode). To adjust the timebase setting, use the T/Div knob on the Control Box panel.

Timing mode is one of the two modes the analyzer can operate in (the other is State mode). Timing mode (or internal clock mode) is typically used for timing analysis—input signals are asynchronously sampled at a constant rate (determined by the timebase setting). Thus, the waveform window represents time on the x-axis in linear fashion and logical state on the y-axis.

Trigger is the point that causes the analyzer to begin storing acquisition data. The trigger can be a signal edge, a pattern, or a combination of edges, patterns, or occurrences.

Trigger mode can be set to Normal or Auto.

Trigger output is an output signal used to trigger an external measurement system or other device. This output is pulsed when the analyzer's trigger condition is satisfied and the analyzer begins acquiring data.

***Waveform* window** displays the acquired data in a graphical fashion. Input channels are grouped up into customizable items for meaningful representation.

Zoom is the magnification amount of the waveform window (when the analyzer is in State mode). In State mode, all timing information about the acquired data is missing, so it doesn't make sense to speak of sweep speed; instead, the concept of *zoom* is introduced to refer to the magnification of the waveform. To adjust the timebase setting, use the Zoom knob on the Control Box panel.