J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*


# LOGIC ANALYZERS

Integrated digital circuits and digital systems have a large number of inputs and outputs. Thus, when in operation, a large number of signals are involved, most of them interrelated. It is thus natural to require a multichannel data acquisition system when the operation of digital circuits and systems is experimentally assessed. That is the case when troubleshooting a microprocessor-based system for hardware or software errors. However, in general, the logic state of a few signals is enough to check the operation of a digital system. That means that the acquisition system needs only one-level, or at most two-level, detection.

The logic analyzer has traditionally been the instrument chosen to troubleshoot digital circuits and systems according to the two above-mentioned characteristics. It is a multichannel digital storage instrument with one bit of vertical resolution (Fig. 1) and flexible triggering and display modes. One can say that the logic analyzer has been for the inspection of signals in digital systems what the oscilloscope is for time viewing of signals in analog systems.

The operation and performance of a logic analyzer involve and depend on several aspects: (1) the mechanical and electrical connection between the analyzer and the system under test (probing); (2) the memory type, organization, and size; (3) the conditions (triggering and qualification) that *freeze* the analyzer (disconnect its memory from the input sampling process); (4) the data display capabilities. In the section that follows, dedicated to the fundamental aspects of a logic analyzer and based on the typical constitution of a logic analyzer, we address all these issues. The remaining three sections of the article are dedicated to the operation, applications, and specifications of logic analyzers.

## Fundamentals Aspects of a Logic Analyzer

**Constitution.** Nowadays, digital circuits often operate at frequencies higher than 100 MHz. The requirements on equipment to test and troubleshoot such systems are thus extremely demanding. For logic analyzers, scaling up of traditional architectures using higher-speed clocks and logic has proved to be an unsatisfactory solution. New solutions have led to equipment more or less different from the traditional logic analyzer design for low- to medium-frequency applications. Notwithstanding, it is convenient to introduce the operation of the logic analyzer by considering the architecture of the traditional analyzer. We will overview present solutions and future trends in digital-system testing and troubleshooting.

A simplified block diagram of a typical low- to medium-frequency logic analyzer is shown in Fig. 2.

Probes organized in pods assure the interface between the instrument and the system under test. Both the pods and the individual wires on the pods are coded (e.g. by color) so that the various input signals can be more easily tracked. In some cases, probes include level detection and buffer circuits (active probes); in other cases, those circuits are included inside the analyzer (passive probes). The words (data) output by the buffers are clocked into a register at regular intervals until a preset condition (trigger condition) is verified. The clock is either externally provided or internally generated. Depending on the triggering situation, data in the register are stored (or not) in a memory, using the same clock, until either the trigger control detects a stop condition

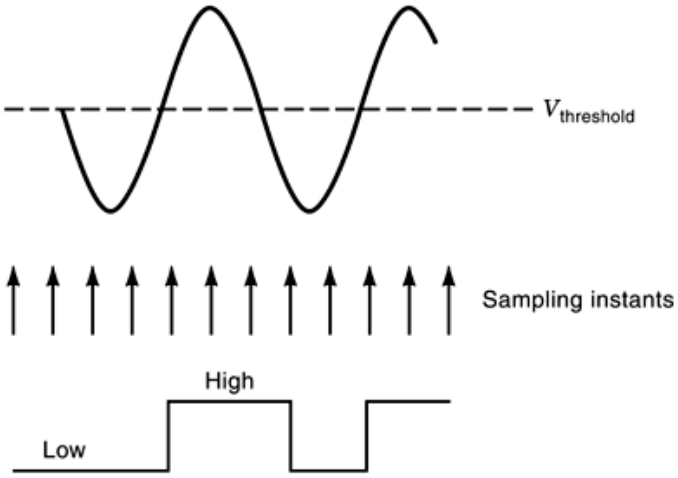



**Fig. 1.** Logic analyzers transform input signals (here a sine wave), into two levels by comparing them with a constant voltage ($V_{threshold}$) at fixed time instants.

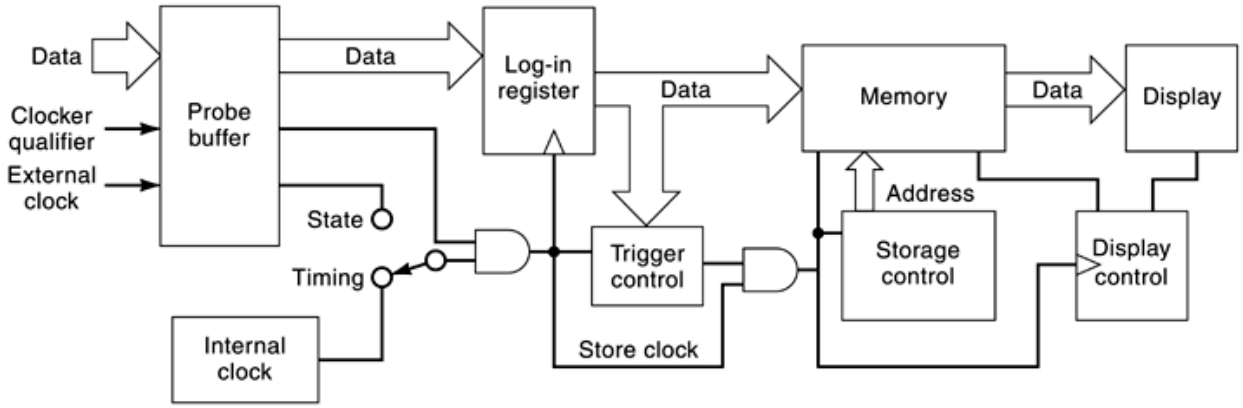



**Fig. 2.** Simplified block diagram of a logic analyzer. The words (data) output by the buffers are clocked into a register at regular intervals until a preset condition (trigger condition) is verified. Depending on the triggering situation, data in the register are stored (or not) in a memory using the same clock until either the trigger control detects a stop condition or the memory is full. When the storage ends, data are shown in a display.

or the memory is full. When the storage ends, data are shown in a display, the two basic formats being timing diagram and state listing.

### Probing.

*Probes.* Probes are the elements of a logic analyzer system that are mechanically and electrically in contact with both the logic analyzer and the system under test. From the mechanical point of view, it is important that probes ensure rugged and reliable connections. Electrically, two aspects are of paramount importance: input impedance and immunity to noise. All these objectives are becoming more and more difficult to achieve due to the increase in number and density of terminals and in clock rates of digital circuits and systems. Pods with probes and accessories for general and dedicated applications are available from both logic-analyzer manufacturers (1) and independent ones (2).

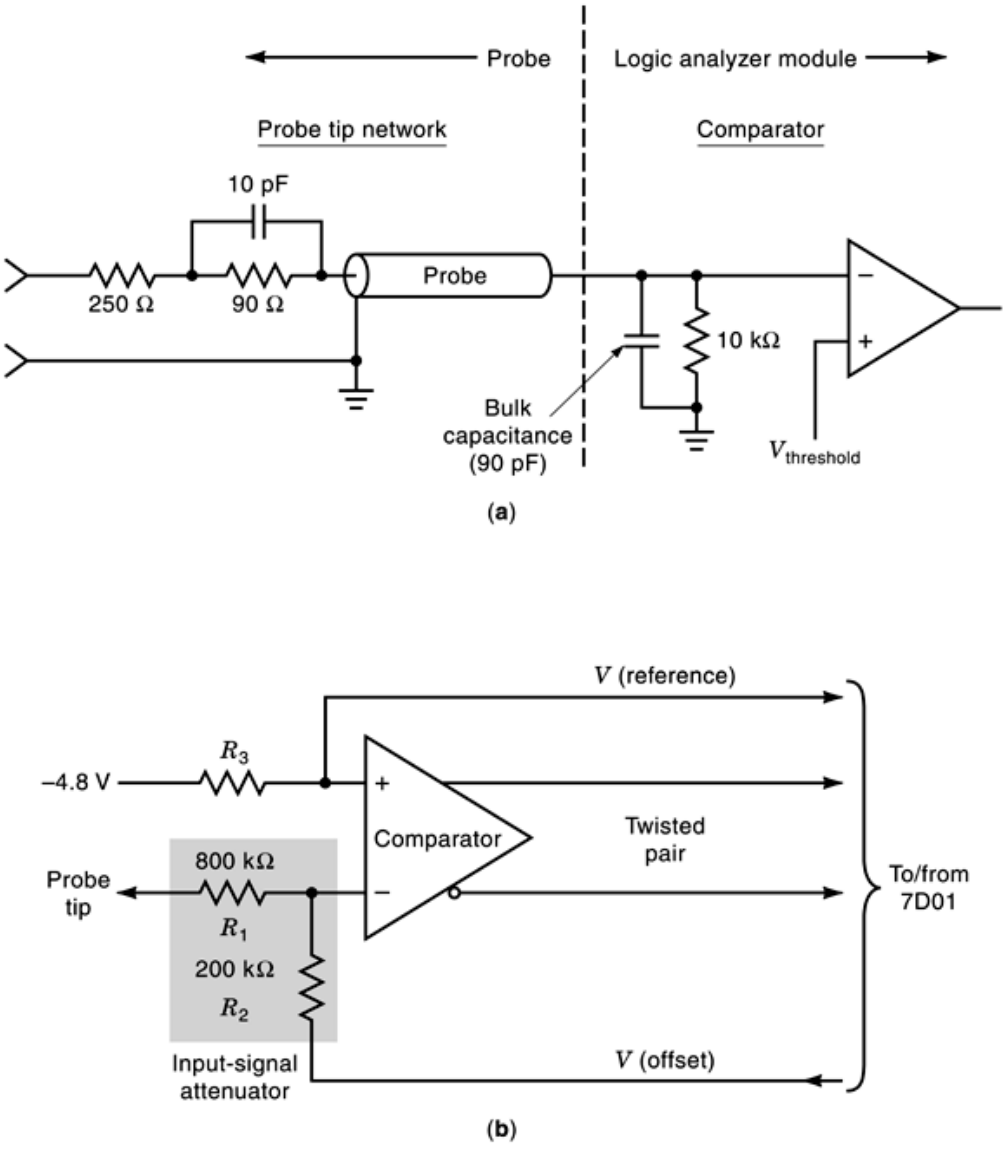



**Fig. 3.** Attenuator probes for logic analyzers: (a) passive probe; (b) active probe. Probes are the elements of a logic analyzing system that are mechanically and electrically in contact with both the logic analyzer and the system under test.

Probes for logic analyzers are voltage probes. According to their internal constitution, they are classified into two categories: passive, when they include only passive electrical elements, and active, when they incorporate active electronic components. Both categories may include attenuating or nonattenuating types of probes. Logic-analyzer probes were traditionally passive attenuator probes. These are less susceptible to transient noise, and even nowadays some manufacturers prefer them. Figure 3(a) shows a schematic diagram of a conventional passive attenuation probe and logic-analyzer input.

## 4 LOGIC ANALYZERS

To overcome the limited input impedance of passive probes, particularly their undesirable capacitive components, active solutions are introduced. Input resistance values greater than 100 kΩ and capacitance values between 0.4 and 2 pF are achieved. At frequencies higher than several megahertz, the input impedance of the probe is dictated mainly by its capacitance. The lower the capacitance, the higher the input impedance and the lower the delay introduced in signals (for fast signals, 8 pF produces about 1 ns timing error). For this reason the values obtained with active attenuator probes are particularly significant. Other positive features of active attenuator probes include (3): (a) high signal bandwidth, up to 350 MHz; (b) performance less dependent on the capacitance of the cable that connects the probe to the analyzer and on stray capacitances; (c) good input timing characteristics, with low rise times, due to the high bandwidth and impedance (low circuit loading).

Figure 3(b) represents the equivalent circuit for one channel of an active data-acquisition probe (Tektronix P6451 used with a Tektronix 7D01 logic analyzer). In the present case, the probe includes only a voltage comparator. An offset voltage on the bottom of the probe input signal attenuator allows a change of the comparison voltage of the probe comparator. This feature is essential when the logic analyzer is used with components and systems of different logic families. Note, however, that this feature does not impose a peculiar characteristic on the logic analyzer, since most analyzers allow the adjustment of the threshold voltage ($V_{\text{threshold}}$).

*Preprocessors Pods.* An important accessory for the test of a microprocessor is a preprocessor, also called a disassembler pod. A preprocessor is a special adapter that typically plugs into a microprocessor socket, with the microprocessor plugging into the preprocessor. It is not only a fast and reliable way to connect to the circuit under test (with the advantage of reducing probing loading effects on high-speed processors), but it can also provide clocking and demultiplexing circuits to capture the signal to and from the processor. It is common to find preprocessors not only for the general microprocessors, but also for standard buses and interfaces such as IEEE 488, RS 232, VXI, and SCSI.

**Triggering.** Logic analyzers with the layout shown in Fig. 2 are mainly built with circular memories that store data continuously, storing each new sample in the next available location in memory. That is one of the reasons why it is so important to establish a condition that causes memory to *freeze*, or be disconnected from the input sampling process. Such is the function of triggering.

In digital systems, namely in microprocessor-based ones, software problems are usually discovered long after they have been created. Debugging them requires sophisticated triggering and also large-capacity memories. On the other hand, in hardware failures the effect and its cause are normally closer in time, and so triggering by the cause is recommended.

The number and flexibility of triggering modes an analyzer has is one of its most important performance-defining parameters, since troubleshooting of both hardware and software problems often heavily depends on them. The flexibility includes the possibility of using more than one mode to trigger the analyzer. The priority is then either predefined or user-defined (using, for instance, logic conditions). Among the several triggering modes available in logic analyzers one can mention the following: (a) data transition in a channel; (b) external signal; (c) slew rate; (d) glitches; (e) pulse width; (f) timeout; (g) setup–hold violation; (h) runt pulse; (i) word recognizer; (j) delay by events; (k) delay by words. Word recognizer, or data word triggering, is the basic trigger mode, and low-cost general-purpose logic analyzers have not only that mode but others such as (a), (b), and (d), as well as the possibility of combining them to produce a trigger condition. Triggering modes, such as (c), (f), or (g), are only available in high-quality analyzers or in instruments that combine a logic analyzer and a digital storage oscilloscope (*DSO*).

*Data Transition in a Channel (Edge Triggering) and External Signal Triggering Modes.* These two modes are very similar in that the end of data storage in memory is conditioned by the activity of a single signal (Fig. 4). In the first case, the signal inputs the analyzer through one of the probes and is extracted from the system under analysis; in the second case, the signal is external to that system.

*Slew Rate.* Slew-rate triggering (Fig. 5) is an extension of edge triggering that allows stopping acquisition when edges on signals are either too slow or too fast. This mode is useful for detecting reduced time

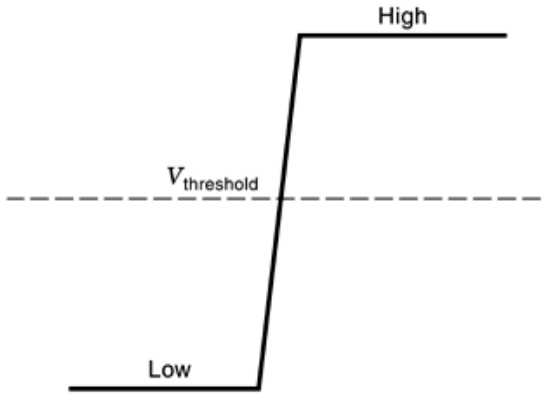



**Fig. 4.** Data transition triggering. The end of data storage in memory is conditioned by the activity of a single signal.

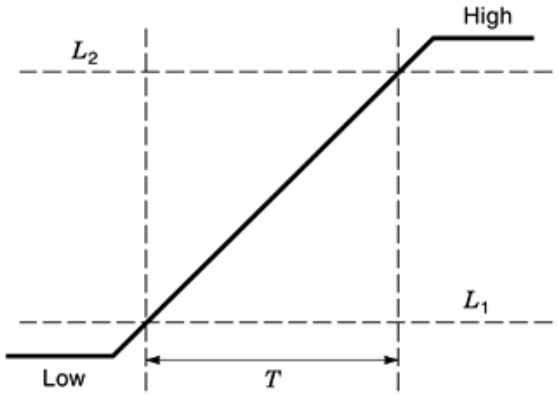



**Fig. 5.** Slew-rate triggering. Acquisition stops when edges on signals are either too slow or too fast.

margins due to slow transitions and possible sources of radiated interference due to transitions faster than required.

In this mode of triggering, the user selects two voltage levels $L_1$ and $L_2$, the edge (rising or falling) of the signal in a channel, and a time value $T$ that defines either the upper or the lower limit of the time between the two defined voltage levels.

*Glitches.* A *glitch* can be defined as a transition on a signal that crosses the threshold voltage more than once between samples (Fig. 6). To detect or capture such unwanted and sometimes infrequent events, it is necessary (a) to include in the probe a high-performance glitch circuit capable of detecting transitions in either direction within a sample period, and (b) to implement a special triggering mode. Such a mode, particularly useful when glitches violate the minimum-pulse-width specification or when they occur on an otherwise constant signal, is set with the definition of a threshold voltage level, the channel to analyze, and a time value $T$ defining the minimum or maximum duration of the glitch.

*Pulse Width.* The pulse-width triggering mode is similar to the glitch triggering mode, but instead of a time value, a time range for the pulse duration is specified (Fig. 7). Thus, it is possible not only to monitor a channel continuously until the occurrence of a pulse whose duration is outside the preset range value, but also to make time-margin studies by varying the range limits.

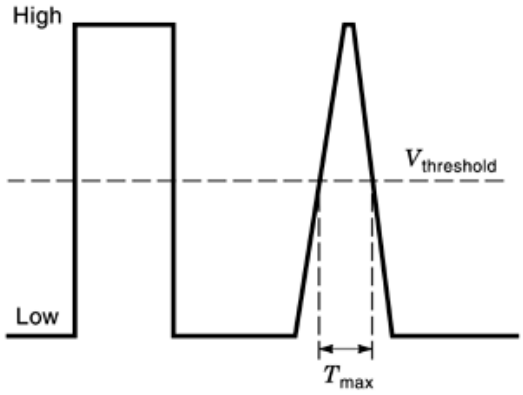



**Fig. 6.** Glitch triggering. A glitch can be defined as a transition on a signal that crosses the threshold voltage more than once between samples. To set the glitch triggering mode it is necessary to select a threshold voltage level, the channel to analyze, and a time value $T$ defining the minimum or maximum duration of the glitch.

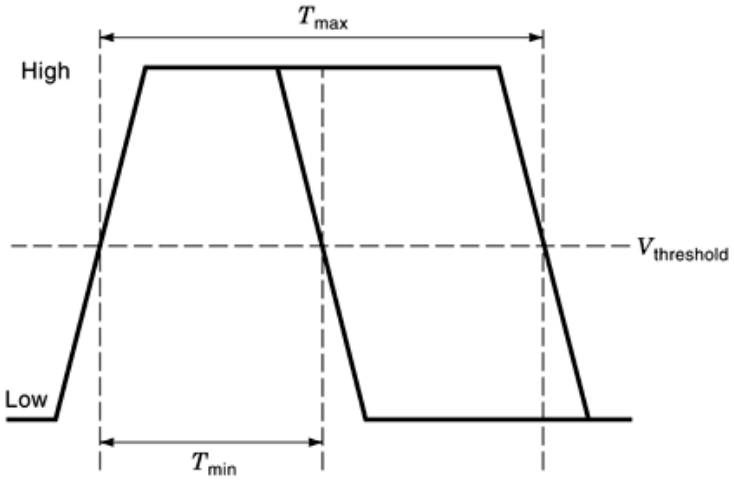



**Fig. 7.** Pulse-width triggering. The pulse-width triggering mode is similar to the glitch triggering mode, but instead of a time value a time range for the pulse duration is specified.

*Timeout.* Sometimes, signals remain in the active state when a failure occurs. To troubleshoot such situations a special triggering mode is required, since all the time-qualified triggering modes mentioned before fail to produce a trigger condition when the signal does not change level. In timeout triggering (Fig. 8) this problem is overcome, since a trigger condition is produced when a preset time interval, $T_{UL}$, elapses regardless of whether the signal in a selected channel remains in the active state or returns to the inactive state.

*Setup–Hold Violation.* This mode of triggering allows a deterministic detection of setup- and hold-time transgressions in synchronous data signals (Fig. 9). Upon selection of the clock channel, clock edge, data channels, and threshold voltages for both clock ($V_{thres1}$) and data channels ($V_{thres2}$), limit values for hold time ($T_1$) and setup time ($T_2$) must be specified. The analyzer triggers when the data channels cross $V_{thres2}$ within the specified setup and hold time limits relative to $T$, the instant that the selected clock edge crosses $V_{thres1}$.

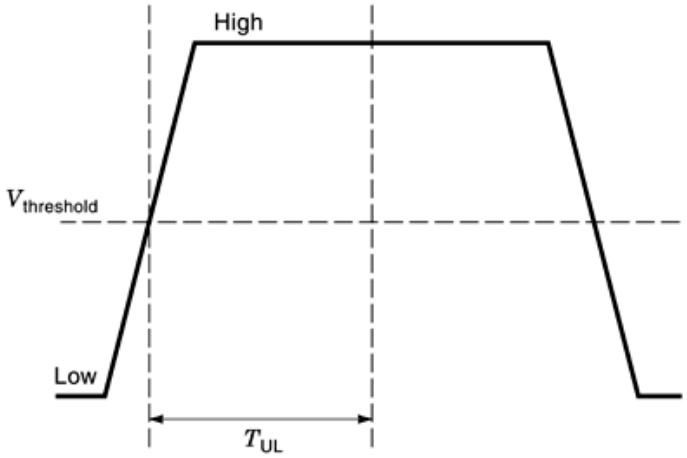



**Fig. 8.** Timeout triggering. A trigger condition is produced when a preset time interval $T$ has elapsed, regardless of whether the signal in a selected channel remains in the active state or returns to the inactive one.

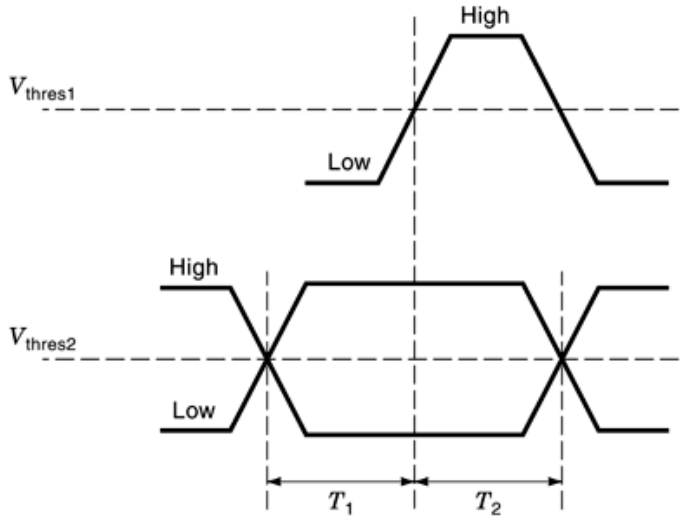



**Fig. 9.** Setup–hold violation triggering. Upon selection of the clock channel, we must specify the clock edge, data channels, and threshold voltages for both clock ($V_{thres1}$) and data channels ($V_{thres2}$), as well as limit values for hold time ($T_1$) and setup time ($T_2$). The analyzer triggers when the data channels cross $V_{thres2}$ within the specified setup and hold time limits relative to $T$, the instant that the selected clock edge crosses $V_{thres1}$.

*Runt Pulse.* A runt pulse is a pulse that fails to reach the high-level state, that is to say, a pulse that crosses one threshold voltage but not both (Fig. 10). Such unwanted pulses may occur in a digital system having an insufficient noise margin. Clock runt pulses can easily occur in microprocessor-based systems or in other synchronous systems when the distribution path of the clock signal is long.

In logic analyzers having the ability of runt-pulse detection, a runt-pulse trigger mode is provided. To initiate that mode, threshold voltage levels for both low ($L_1$) and high ($L_2$) logic states must be specified. Two other parameters that may be selectable are the pulse polarity and its minimum time duration $T$. If all the

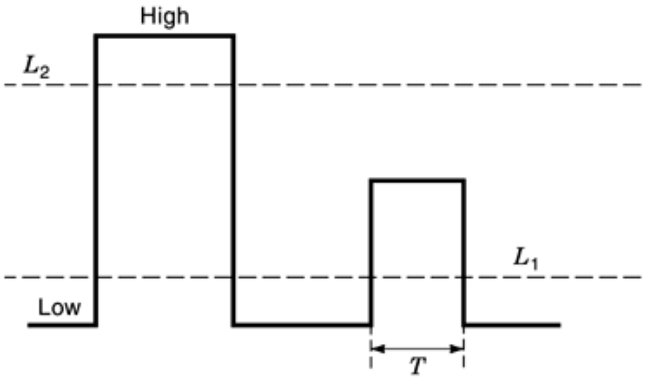



**Fig. 10.** Runt-pulse triggering. A runt pulse is a pulse that fails to reach the high-level state, that is to say, a pulse that crosses one threshold voltage but not both. To initiate that mode, threshold voltage levels for both high and low logic states must be specified. The logic analyzer triggers only when the input signal has a pulse with the specified polarity that crosses the lower threshold level but does not cross the higher one.

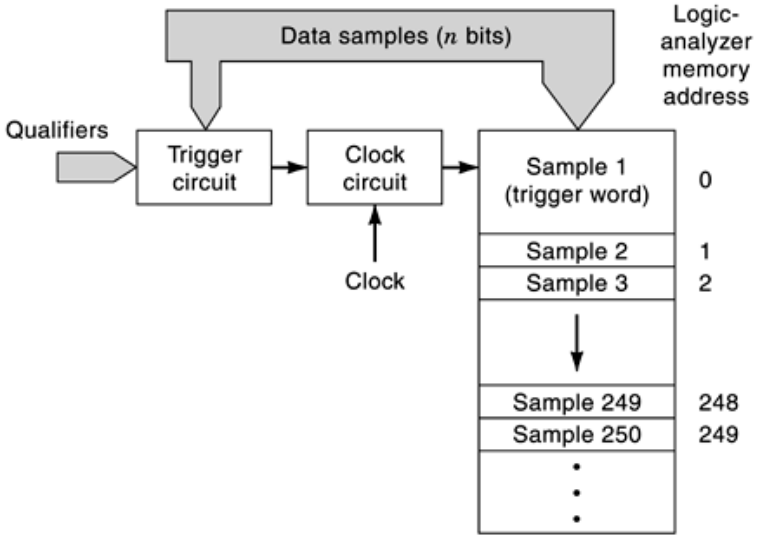



**Fig. 11.** Word-recognizer triggering. The logic analyzer triggers when a preset parallel data word whose size may reach the number of inputs of the logic analyzer is detected.

above four parameters are specified, the logic analyzer triggers only when the input signal has a pulse with the specified polarity that crosses the lower threshold level but does not cross the higher one. It is usual to have the possibility of triggering on any such pulse or only on those having duration greater than $T$.

*Word Recognizer.* A preset parallel data word whose size may reach the number of inputs of the logic analyzer is sometimes a convenient means of triggering. Figure 11 illustrates this trigger mode that is perhaps the basic mode for logic analysis. In the situation depicted the memory is totally filled with data that occurred after trigger.

*Delay by Events.* Sometimes, an error occurs in a digital system only after several repetitive activities have taken place. This is particularly true for software errors. It is common for a subroutine to be executed

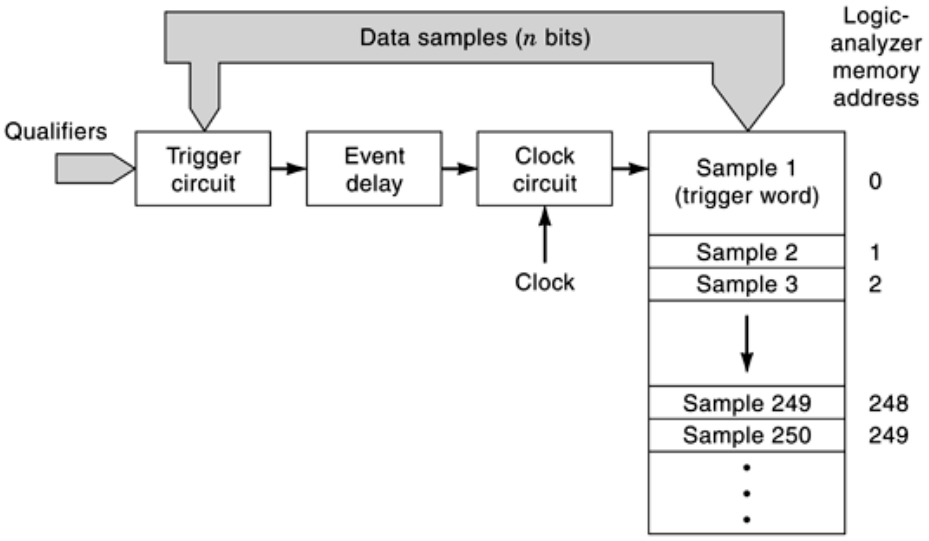



**Fig. 12.** Delay-by-events triggering. A trigger is produced only when a selected event has occurred a defined number of times.

$n$ times with success before it leads to trouble. The possibility of delaying the triggering of the logic analyzer until the occurrence of the error is in such cases of great interest, especially for saving memory space. Figure 12 illustrates the delay-by-events mode, also for a situation where the memory is totally filled with data that occurred after trigger. The event delay block is basically a counting element that delays the output of the trigger circuit by a predefined number of trigger events.

*Delay by Words.* Delay by words, or clock delay, is the trigger mode in which data storage only takes place a preselected number of clock pulses after the trigger condition is fulfilled (Fig. 13). Usually delay by events has priority over delay by words when both modes are selected, which means that the first must be satisfied before the second starts.

**Qualification.** In practical applications, once the trigger mode is selected, it is usually useful to increase the number of the conditions that control the storage of words in a logic analyzer's memory. The signals that are not stored but are used to assert these additional conditions are called *qualifiers*. Qualifiers may be used not only to enable but also to disable data, which permits the recording of only the desired sets of data (blocks) to conserve memory and to make problem areas more visible. One example: if one detects that a problem occurs in a microprocessor when a specific word is written in memory, the word-recognizer trigger mode should be complemented with a one-bit identifying memory write mode (using the read/write line of the processor).

The most common qualifiers are for the clock and for the word recognizer. Dedicated external digital signals input to the logic analyzer by means of specific inputs are used (clock qualifier, external qualifier). In both cases, the selection of the logic state of the signals defines the additional storage condition. In the example given before, storage would occur only when the problematic word was present in the data lines and the read/write line of the processor was in the logic state corresponding to write.

When using the delay-by-events trigger mode, the delay refers to trigger events and thus takes into consideration both trigger conditions and qualifiers. Qualification does not affect recognition of trigger words but does affect the trigger delay; a word included in the trigger sequence will be recognized regardless of whether or not qualification allows it to be recorded.

In microprocessor-based logic analyzers, qualification capabilities can be greatly increased. In the general-purpose Philips PM 3632 logic analyzer, for instance, data qualification permits determination of an individual

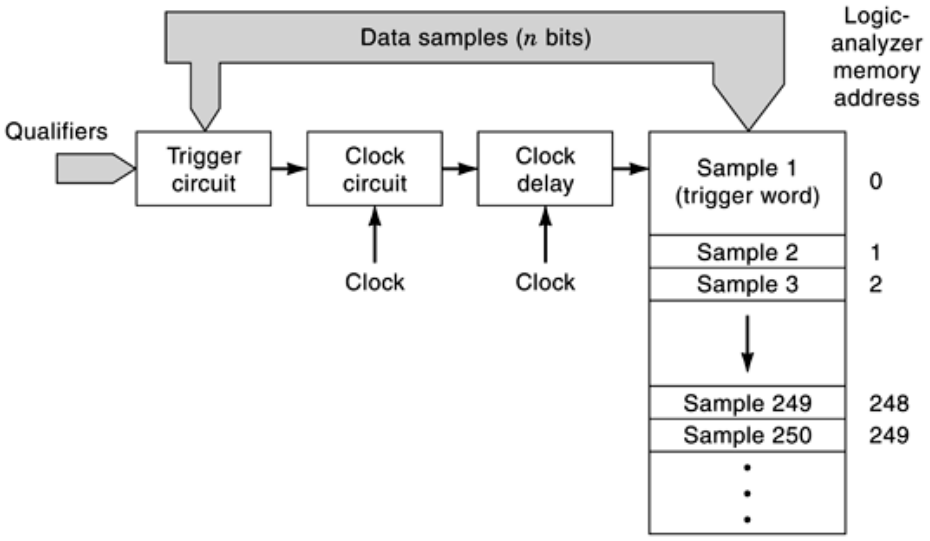



**Fig. 13.** Delay-by-words triggering. Acquisition is stopped a defined number of words after the trigger.

(combinational data qualification) or sequential (state data qualification) basis as to whether or not any given word (data sample) will be recorded.

**Memory.** Words that meet triggering and qualification conditions must be stored for later display and manipulation. As already mentioned, logical analyzers with the classical layout shown in Fig. 2 have circular memories of the first-in, first-out type (*FIFOs*). Such solution allows the display of data words before and after the trigger but is rather limited in the triggering possibilities and the length of the time window displayed.

With the introduction of microprocessors in logic analyzers, memory tends to be of the addressable type (*RAM*), and its management is usually the responsibility of the processor. The integration of a microprocessor as a building block of a logic analyzer was also instrumental in the implementation not only of advanced triggering modes (some of which introduced in a previous section), but also of techniques that allow better use of the available memory and thus larger-time-window visualization.

A logic analyzer's memory is often organized in two separate banks, one used for temporary storage and the other for permanent storage. The permanent storage bank is adequate to keep reference data for comparison and upon command; it is loaded with data previously stored in the temporary bank.

**Display.** Data stored in memory can be displayed in three major ways: timing diagrams, state lists, or tables and assembly mnemonics.

In a timing diagram, each trace represents the waveform of one input signal in binary form (Fig. 14). Depending on the position of the trigger word in memory, the display may include only posttrigger data (posttrigger display), pre- and posttrigger data (centered display), or pretrigger data (pretrigger display). Input channels can be grouped together into buses and viewed in several formats: ASCII, hexadecimal, decimal, binary, or user-defined mnemonics. Cursors permit the identification of the memory position of the trigger word and of the status of the different channels in each sampling instant. As mentioned in the following section, timing diagrams are adequate when the logic analyzer operates in asynchronous mode, but of lesser interest when in synchronous operation. In this case, state-list display is much more elucidative. As in timing-diagram display, channels can be organized into groups and displayed on screen in ASCII, binary, decimal, hexadecimal, and user-defined mnemonics. The cursors have the same functionality as before.

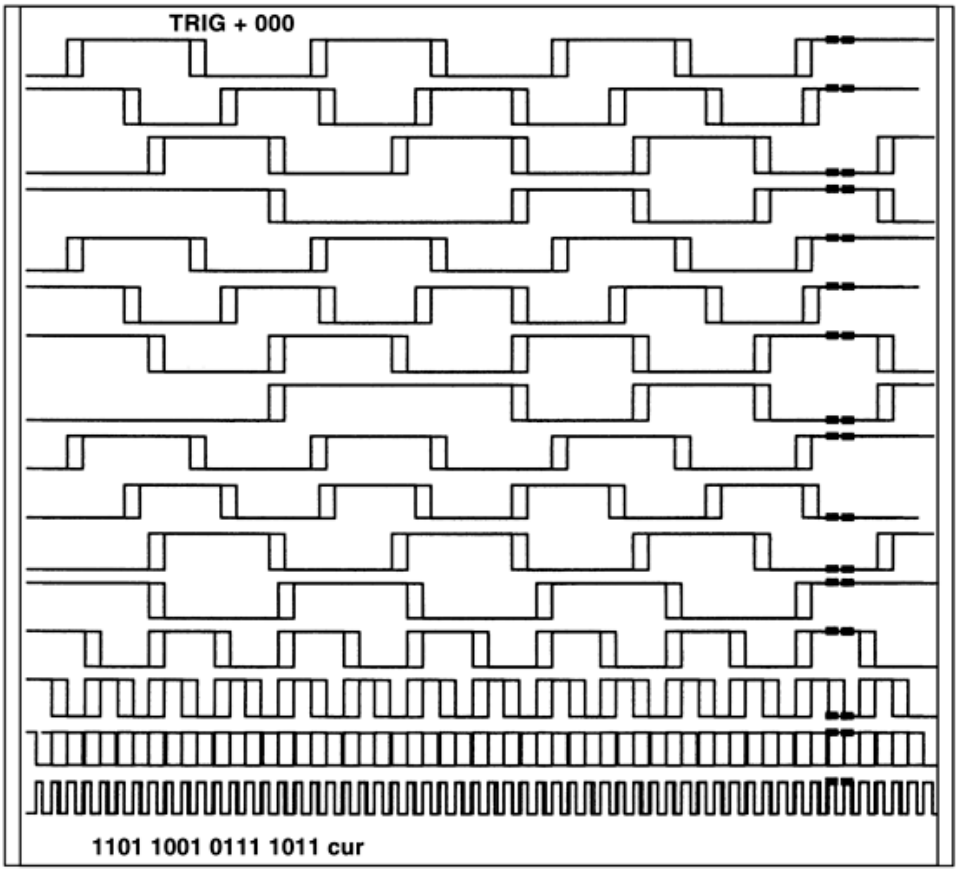
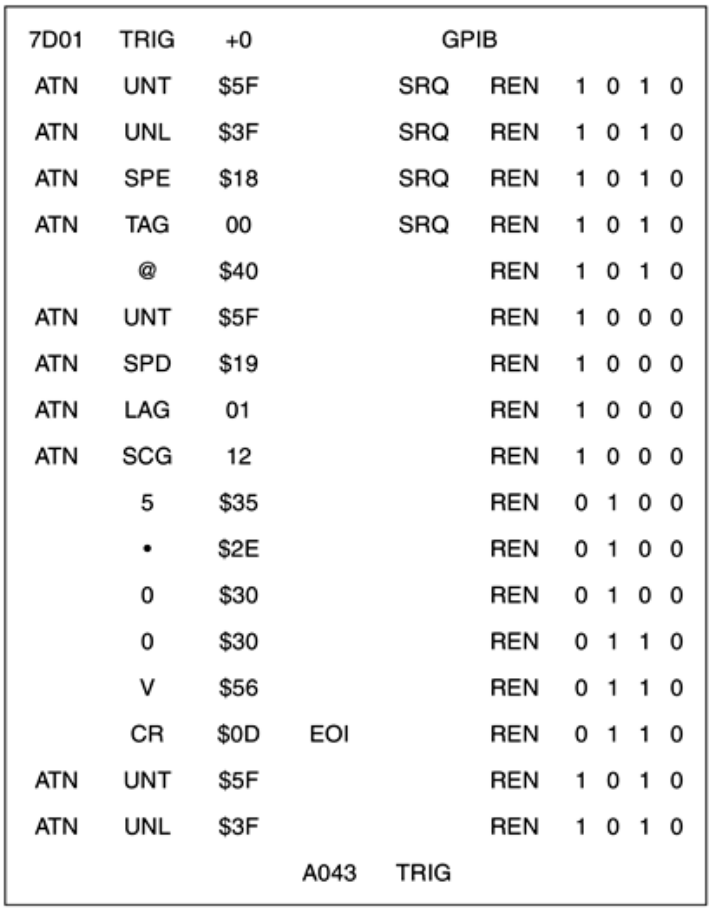



**Fig. 14.** Timing diagrams. In this mode of display, each trace represents in binary format one input signal. The upper trace corresponds to channel 0, and the lower trace to channel 15. The cursor (cur) identifying the current word, which in this case is also the trigger word, is intensified.

When the analyzer is able to display more than one state table at the same time, it becomes easier to identify differences that occur when the system under test performs the same tasks. It is then usual to use one of the tables as a reference table. Upon selection, the analyzer displays the differences in some special way.

Due to the difficulty in the interpretation of the information stored in the memory of the logic analyzer, especially when the number of input channels is very large, some manufacturers provide inverse assemblers (or *disassemblers*). A disassembler is a program that runs in the analyzer and that interprets the data captured and formats them in an appropriate way (assembly, mnemonics, etc.) for displaying. Disassemblers are particularly useful in conjunction with preprocessors for testing microprocessors, buses, and interfaces. Each microprocessor, bus, or interface requires a specific preprocessor and also a specific disassembler. Figure 15 exemplifies the display obtained using a GPIB (IEEE 488) disassembler. By using different versions of these two components, a logic analyzer can be used for testing several processors, buses, or interfaces.

Some logic analyzers, such as the Tektronix 7D01/DF2 set, can display the contents of memory as $x$–$y$ dots on the display. In this mode (Map), the vertical axis represents the most significant half of the data word, and the horizontal axis represents the least significant half.



| 7D01 | TRIG | +0 | | GPIB | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ATN | UNT | $5F | | SRQ | REN | 1 | 0 | 1 | 0 |
| ATN | UNL | $3F | | SRQ | REN | 1 | 0 | 1 | 0 |
| ATN | SPE | $18 | | SRQ | REN | 1 | 0 | 1 | 0 |
| ATN | TAG | 00 | | SRQ | REN | 1 | 0 | 1 | 0 |
| | @ | $40 | | | REN | 1 | 0 | 1 | 0 |
| ATN | UNT | $5F | | | REN | 1 | 0 | 0 | 0 |
| ATN | SPD | $19 | | | REN | 1 | 0 | 0 | 0 |
| ATN | LAG | 01 | | | REN | 1 | 0 | 0 | 0 |
| ATN | SCG | 12 | | | REN | 1 | 0 | 0 | 0 |
| | 5 | $35 | | | REN | 0 | 1 | 0 | 0 |
| | • | $2E | | | REN | 0 | 1 | 0 | 0 |
| | 0 | $30 | | | REN | 0 | 1 | 0 | 0 |
| | 0 | $30 | | | REN | 0 | 1 | 1 | 0 |
| | V | $56 | | | REN | 0 | 1 | 1 | 0 |
| | CR | $0D | EOI | | REN | 0 | 1 | 1 | 0 |
| ATN | UNT | $5F | | | REN | 1 | 0 | 1 | 0 |
| ATN | UNL | $3F | | | REN | 1 | 0 | 1 | 0 |
| | | | A043 | TRIG | | | | | |

**Fig. 15.** Typical GPIB display showing disassembled instructions in IEEE Standard 488-1975 message mnemonics.

Logic analyzers having raster-type screens can easily accommodate two other display modes: accumulative and overlay. In accumulative mode the display is not erased between successive acquisitions, which allows easier detection of changes occurring in the different lines. In overlay mode, multiple channels are represented in a single display line, which permits easier identification of time differences between transitions of signals in the channels.

High-quality (PC-based) logic analyzers have the ability of displaying memory contents in both timing-diagram and state-list forms. For that purpose the screen is split in two halves. Such a display mode requires either special probes or input accessories. It can also be used for simultaneous testing of two circuits or systems.

## Logic-Analyzer Operation

At first the interface between the operator and the analyzer was through front or rear panel controls and switches. Nowadays logic analyzers use menus and user prompts to help in their configuration. The configuration, executed after the analyzer has been connected to the system under test using the several wires of the pods, includes among others the following choices: (a) clock source and polarity and (for internal clock)

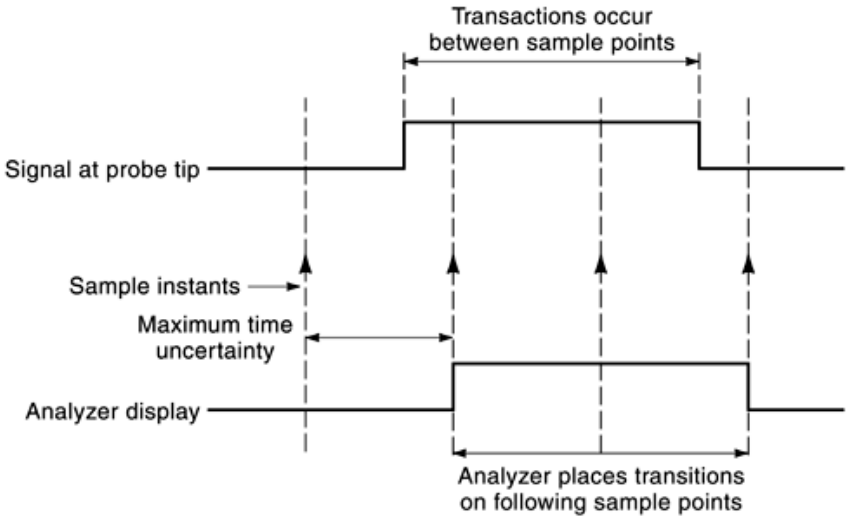



**Fig. 16.** Asynchronous mode: timing analysis. The maximum time uncertainty is equal to the clock period.

clock interval; (b) threshold voltage; (c) trigger conditions; (d) qualifiers; (e) memory occupation (pretrigger, posttrigger, or centered); (f) data presentation (timing diagram, state list, disassembled code, or other).

The choice of clock source should be the first to be made, since some of the other parameters to configure depend on it. When a timing-diagram display like the one obtained on an oscilloscope better suits the problem under study, the interval between samplings should be smaller than the time interval over which system's signals change (system's clock). In such a case, it is convenient to use an internal clock, available in the analyzer, as the sampling clock. The analyzer then operates asynchronously with the system under test, that is to say, the sampling clock and the clock of the system under test are not related (internal-clock, asynchronous mode of operation). When state analysis or disassembled codes are the best means to troubleshoot a system, the analyzer must sample the input signals synchronously with the clock of the system. This clock must thus be used as the analyzer's sampling clock (external-clock, synchronous mode of operation).

**Asynchronous Mode—Timing-Analyzer Mode.** Although the logic analyzer is primarily meant for functional testing, the timing mode of operation is particularly suited to parametric testing, such as to verify proper timing, including setup and hold times and propagation delay timing.

The maximum time uncertainty in timing mode is the clock period (Fig. 16). For this reason, the clock used should be such that at least two or three samples of the channels are taken during both high and low states. This usually leads to a ratio of 5 to 10 between the rates of the sampling clock and system's clock. Due to the fixed size of the analyzer's memory, a compromise exists between the sampling-clock rate (time resolution) and the length of data displayed: the more time resolution one selects, the smaller the time interval one can view. High-end logic analyzers are equipped with a costly sampling technique, called *transitional sampling*, that can partially overcome this compromise.

**Transitional Sampling.** In this mode of sampling, when an input line changes state (transition), a sample of all input channels is taken along with the clock time (Fig. 17). Since the individual clock pulses are assigned a reference time, it is possible to fully reconstruct the timing waveform. With this technique no redundant data are stored, which leads to significant memory saving when the input signals occur, for instance in bursts.

**Synchronous Mode—State-Analyzer Mode.** Synchronous mode uses a clock generated by the system under test and is primarily dedicated to software troubleshooting. In this mode the logic analyzer operates

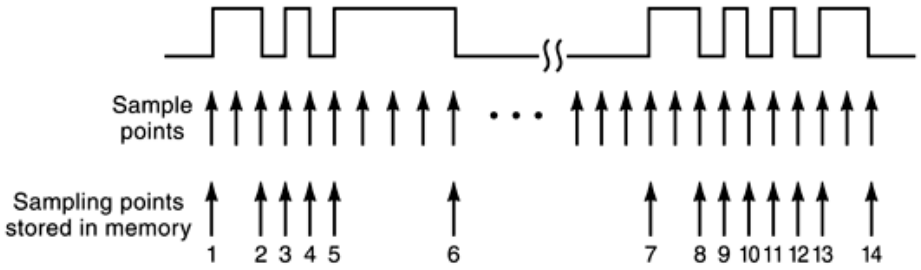



**Fig. 17.** Transitional sampling. A sample of all input channels is taken, along with the clock time, when an input line changes state (transition). No redundant data are stored, which leads to significant memory saving when (e.g.) the input signals occur in bursts.

in the data domain, and often relevant signals do not occur periodically. For this reason, rather than continuous monitoring of signals in the system, it is necessary to trigger the logic analyzer in such a way that only the pertinent states are displayed. The capabilities of data qualification are then decisive. When troubleshooting a microprocessor, a bus, or an interface, it is also very important to have access to an adequate disassembler to interpret and display the stored data in a comprehensive way.

When the system under test has a clock rate that approximates the logic analyzer's performance limits (namely, its setup and hold time and channel-to-channel skew), incorrect data may be stored. The channel-to-channel skew error is the time difference between the signal at the probe tip and at the comparator output. Time differences between the clock and signals in data channels are particularly important for systems operating at high rates. To reduce such problems, those types of logic analyzers include delay and skew compensation circuits at the input.

It should be emphasized that, either in synchronous or in asynchronous mode, the time window that can be displayed and analyzed depends heavily on the trigger mode and qualifier selections. A judicious selection can lead to the storage of only pertinent data and thus to a maximization of the size of the time window displayed and analyzed.

## Logic-Analyzer Applications: State of the Art And Future Trends

Logic analyzers are basic tools for troubleshooting digital systems, including new systems under development. Both the inner operation of systems and some types of buses and digital communications between systems (e.g. IEEE 488) can be tested.

The great majority of digital systems that nowadays require the use of a logic analyzer are microprocessor-based ones, and the most common problems occur in its software, hardware, interface, or input/output ($I/O$) components. Software problems, resulting usually from programming errors, are better traced using state analysis and thus the synchronous mode of operation of the analyzer (external clock). Hardware, interface, and I/O problems are better identified using timing analysis. In some cases, I/O problems involving communications between systems are easier to isolate using state analysis and appropriate disassemblers.

Logic-analyzer design and performance have naturally followed testing needs. Those needs have been essentially dictated by the technical evolution of components and systems. A brief discussion of the problems that have arisen and will go on arising and some solutions to overcome them follows.

**Number of Signals to Analyze.** The number of signals to analyze has significantly increased (e.g., because of wider microprocessor data and address buses). As a consequence, larger memories and smaller probing devices are required. The mechanical probing requirements have been particularly demanding due

to the difficulty in accessing signals, the size reduction of electronic components, and the increase in the number of pins, which contributes to a higher density of test points. Neither of these requirements presently poses technical problems. High-end commercial logic analyzers with tens or hundreds of input channels and memories per channel up to some tens of megabytes (4) and probes and accessories for different components and systems, (1) are currently available.

**Clock Rate.** The clock rate of digital systems increased more than ten times in the last few years of the twentieth century. For logic analyzers this increase has had several consequences. First of all, probing devices had to be upgraded. Probe capacitance had to be reduced from some 8 pF that was typical of passive attenuator probes to some 2 pF, in order to make the time errors smaller than the clock period. The delay introduced in a high-speed CMOS gate by probe capacitance is approximately 0.15 ns/pF, and 10 times that value for a standard CMOS gate. On the other hand, such an increase of clock rates also meant faster signals with faster transitions and thus more electromagnetic interference. Electromagnetic compatibility also became more difficult to ensure, due to the size reduction of probes and wires. The extensive use of preprocessors, the redesign of attenuator probes, and more careful wiring have been able to limit to a reasonable level the probing problems due to high clock rates.

With regard to the internal structure of the logic analyzer, scaling up of old logic-analyzer architecture using faster logic proved to be inadequate or too expensive for several reasons (5), particularly because it is extremely difficult to tune clock and data delays for clock rates higher than 100 MHz. Among the solutions that proved to be successful is the introduction of the oversampling technique (5). That technique, already developed for oscilloscopes, uses one internal clock of 250 MHz to provide 2 gigasample per second (GS/s), 500 ps resolution acquisition on all channels of the analyzer. To take advantage of the possibilities opened by that technique, important changes in the overall architecture of the logic analyzer (triggering and qualification, memory, and display) also took place.

**Complexity of Troubleshooting.** The increases in clock rate, number of signals, and functions of digital systems mean more complex systems and thus more difficult troubleshooting, with the following identifiable consequences:

(1) Increase of the importance of simultaneous display of timing diagrams and state lists or assembly mnemonics. This problem is solved with the use and management of split screens.
(2) Need for better amplitude resolution for signal analysis. This led to the integration of a digital storage oscilloscope and a logic analyzer in one instrument by some manufacturers (4).
(3) Need for finer isolation of data to analyze (that is to say, improved triggering and qualification capabilities) and for better analysis tools.

The integration of logic analyzer and digital storage oscilloscope functions in one microprocessor-based instrument allows the versatility required to answer these needs nowadays and for at least the near future (6). Note that: (1) some of the trigger modes referred to in a previous section are characteristic of digital storage oscilloscopes and can be used for logic-analyzer triggering through intermodule triggering; (2) microprocessor support eases the implementation of triggering modes that can evaluate triggering conditions simultaneously instead of one at a time (advanced triggering) (6). The remote use of a logic analyzer, with the help of an instrumentation network, a local network, or even the Internet, optimizes the possibilities of manipulation of the data stored in the instrument memory and thus their analysis.

One final word about some other issues that have conditioned the evolution of logic analyzers.

Logic analyzers are used either at the project stage of a digital system or later for its maintenance or repair. They are dedicated standalone instruments and thus are individually used on short-term basis. High-end logic analyzers are, however, extremely expensive and most of the time rather difficult to use, due to their extensive capabilities. For these reasons, one solution is currently gaining popularity: analyzers based

on personal computers (PCs). Advanced data analysis and remote control required in many test situations nowadays are at the origin of the development of network logic analyzers.

**PC-Based Logic Analyzers.** PC-based or PC-hosted logic analyzers are plug-in cards or external devices that plug into the PC to convert it into a logic analyzer. They became an attractive solution particularly after operating systems, programming languages, and portable PCs reached current levels of performance. State-of-the-art operating systems and programming languages allow user-friendly interfacing, which is particularly important to reduce the time required to learn or relearn how to use the instrument. The processors and monitors of current high-end portable PCs are compatible with the performance required for a general-purpose logic analyzer, and though more expensive than nonportable PCs, they provide important convenience. The capabilities of PC-based logic analyzers are usually limited to make them easier to use, but the acquisition performance achieved is already good: 34 state–timing channels, 100 MHz state analysis, 250 MHz timing analysis, 128 kbyte (timing) and 64 kbyte (state) memory across the 34 channels in the case of Agilent LogicWave E9340A.

**Network Logic Analyzers.** Although most logic analyzers are still operated in local mode, interfacing with the operator via the display and front-end controls, keyboard, and mouse, there are situations where they need to be remotely operated as a data acquisition system. On the other hand, the analysis tools available even for high-end logic analyzers are becoming more and more insufficient for some applications in engineering development (e.g. verification, validation, characterization) and manufacturing (e.g. quality assurance and failure analysis). Both situations point to the need for connecting the analyzer to a computer. Still more interesting is the possibility of including the logic analyzer in a network linking several users and providing access to wider resources. Instrumentation interfaces, such as IEEE 488 (GPIB), can be a solution, but more interesting now and in the future are connections using *LANs* (local area networks) and *WANs* (wide area networks). The Tektronix TLA 700, for instance, is standardly equipped with an interface between the analyzer application and other user applications according to a client–server protocol architecture (7).

## Specifications of Logic Analyzers

The most important specifications of a logic analyzer are the number of channels, maximum internal clock rate (timing analysis), maximum external clock rate (state analysis), trigger and qualification options, memory size, and support for microprocessors and analysis tools. The values that follow are for a high-end logic analyzer that can be used for timing and state analysis, single-processor and single-bus analysis, real-time instruction trace analysis, source-code debugging, performance analysis, multiprocessor and multibus analysis, digital stimulus and control, digital signal quality analysis, and system validation:

Number of channels per mainframe: 680
Timing analysis: 250 MHz
State analysis: 100 MHz, 400 MHz maximum, half channels
Time resolution: 500 ps
Trigger states: 16
Glitch trigger and storage
Setup–hold violation trigger
Setup–hold window: 2 ns adjustable
Edge, pulse-width, timeout, runt, slew-rate, and logic-pattern triggering
Memory size per channel: 16 Mbyte
Transitional sampling (storage)
Simultaneous state and timing analysis through the same probe

Data analysis tools: waveform, listing, histogram, performance-analysis, and source-code standard data window types, waveform zoom and search, processor/bus support, remote control with Microsoft COM/DCOM, symbol extraction from a variety of object file formats, remote user interface, offline data analysis.

## BIBLIOGRAPHY

1. Probing Solutions for Agilent Technologies Logic Analysis Systems [online]. Available http://www.agilent.com
2. Corelis [online]. Available http://www.corelis.com
3. Tektronix Active Attenuator Probing [online]. Available http://www.tektronix.com
4. *Tektronix, Measurement Products Catalog 2000*, pp. 126–143
5. *MagniVu™ Technology Provides 500 ps Timing Resolution*, Tektronix Technical Brief [online]. Available http://www.tektronix.com
6. Advanced Triggering with the TLA 700 Series Logic Analyzers [online]. Available http://www.tektronix.com
7. *TLA 700 Remote Control with Support for Advanced Data Analysis*, Tektronix Technical Brief [online]. Available http://www.tektronix.com

## READING LIST

H. M. Berlin F. C. Getz, Jr. *Principles of Electronic Instrumentation and Measurements*, Columbus, OH: Merryll, 1988.
D. Buchla W. McLachlan *Applied Electronic Instrumentation and Measurement*, New York: MacMillan, 1992.
R. A. Witte *Electronic Test Instruments—Theory and Applications, Hewlett- Packard Professional Books*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
*Basic Concepts of Logic Analysis*, Beaverton, OR: Tektronix, 1989.
*Powerful, affordable logic analysis*, Publication 5954-2686, Palo Alto, CA: Hewlett-Packard Company, 1989.

PEDRO M. B. SILVA GIRÃO
ANTÓNIO M. CRUZ SERRA
HELENA M. GEIRINHAS RAMOS
Instituto Superior Técnico