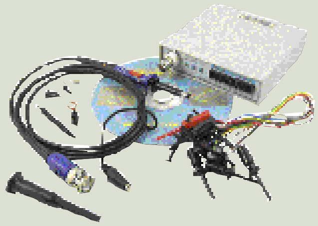
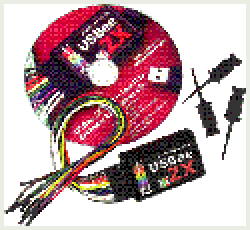
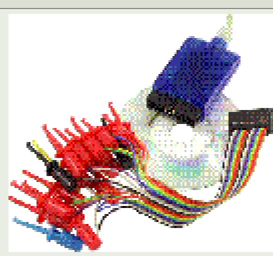
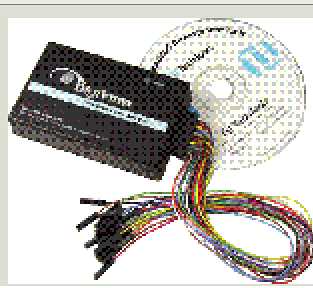
Logic Analyzer		Best Features	Ease of Use
BitScope **BS-50 Pocket Analyzer** **Price: $345** **Phone: +61-2-9436-2955** **http://rbi.ims.ca/4401-557**		• In addition to a logic analyzer, users get a digital storage scope (DSO) and an arbitrary waveform generator (AWG)	• Fairly easy to use, but cursors get "lost"
CWAV **USBee ZX Test Pod** **Price: $895** **Phone: 951-693-3065** **http://rbi.ims.ca/4401-558**		• Easy to use controls • Use of PC's memory provides a large buffer for data • Good triggering options	• Easy to set up and use
Dynon Instruments **ELAB-080** **Price: $495** **Phone: 425-402-4034** **http://rbi.ims.ca/4401-559**		• Multiple functions in one small package • Ability to group signals and display data in several formats	• Fairly good: Better control labels would make this unit easier to understand
Link Instruments **LA-2124-128K** **Price: $800** **Phone: 973-808-8990** **http://rbi.ims.ca/4401-560**		• Solid triggering capabilities as well as capability to trigger on hex, decimal, and ASCII values • Useful cursors make timing measurements a snap	• Easy software setup, but had to find a driver with help of a tech-support person • Overall, this unit gets high marks
NCI **GoLogic-72 System** **Price: from $4300** **Phone: 256-837-6667** **http://rbi.ims.ca/4401-561**		• 72 digital inputs • Lots of triggering capabilities for the most sophisticated user • Excellent manual	• Easy to set up and start using • Groups of eight wires in harnesses makes it easy to identify signals at the wire ends • Good timing information from cursors
Saelig **ANT16 USB Logic Analyzer** **Price: $333** **Phone: 595-385-1750** **http://rbi.ims.ca/4401-562**		• Ease of setup and use • Good triggering options—levels, edges, and sequences	• Very easy to use
TechTools **DigiView** **Price: $499** **Phone: 972-272-9392** **http://rbi.ims.ca/4401-563**		• Long storage buffer that "compresses" data • Fixed sample rate and buffer that uses PC memory	• Easy to set up and use • Pattern and edge triggers easy to set up

Applicability	Limitations	Comments
• With a software upgrade, this unit would provide a handy bench instrument for many engineers	• The use of grabbers limits accessibility to tightly spaced components • Triggers lack edge and sequential capabilities	• The inability to label signals and change their display order shows lack of concern for users • Manual lacks clear information about the logic-analyzer functions
• Useful for tests that will monitor only a few inputs • The unit can count pulses and generate signals, but I did not test these functions	• I had difficulty finding the proper USB driver to make one function operate properly	• Pricey for only eight inputs • Although users can run additional modules in parallel, other suppliers offer higher signal counts at lower cost
• Works well when you don't need more than one level of simple triggers	• Limited triggering capabilities • Never figured out the purpose of the cursors	• Lack of a manual and a "quick-start" guide means you're on your own much of the time
• Adjustable threshold makes this unit applicable in situations that may use logic families that operate at different voltages	• The display does not group signals in a bus and does not display bus data in hex, decimal, or other format • The lack of setting signal edges as triggers may limit the use of this instrument	• Although I like a printed manual, in this case, I didn't need one
• Applicable to the most exacting systems • Can also monitor SPI and I2C signals. Additional capabilities—at extra cost—include disassembly of microprocessor instructions and C/C++ code	• The lack of a quick-start guide may make the manual look overwhelming to new users of logic analyzers; Two or three quick lessons would help users jump into using this capable instrument	• A solid performer that will answer the needs of many engineers • The most expensive instrument reviewed, but it proves the saying, "you get what you pay for"
• Availability of 16 channels makes this unit a good value in labs that develop microcontroller applications	• The lack of a good manual with clear explanations may make some engineers wary of this unit (no explanations of cursors and how to use them, for example)	• The manual needs work: The lack of screen shots and real examples means you're on your own in this academic work
• Solid value for most logic-analysis needs • Availability of 18 inputs makes it useful in 8- and 16-bit systems	• Basic triggering capabilities may not suit sophisticated users	• Fixed input threshold at 1.6V lets this unit work with a variety of logic families

Contributing writer Jon Titus can be reached at jontitus@comcast.net

Dynon Instruments ELAB-080
Price: $495
Woodinville, WA
Phone: 425-402-4034
www.dynoninstruments.com

The Dynon Instruments ELAB-080 provides several functions; a 16-channel logic analyzer (LA), a 2-channel digital oscilloscope, and a 1-channel arbitrary waveform generator (AWG). The package included two 60-MHz 1X/10X scope probes. I confined my investigations to the logic-analyzer portion of the unit and its software. The package received here included the main instrument box, a large plug-in power supply, and connection leads. The LA leads connect to a header that furnishes eight signal leads and a ground for each--a nice feature. Labels CH1, CH2, and so on mark the individual leads so you don't have to rely solely on the color-coded wires for identification at your system under test.

Unfortunately, the package lacks a CD-ROM, manual, and quick-start guide. Instead, a slip of paper asks users to visit the Dynon web site to download the latest software. I downloaded the software onto a USB memory stick and moved it over to my isolated lab PC. The setup went well and the installed software displayed small windows for the logic-analyzer waveforms, oscilloscope signals, and spectrum display. I closed all but the logic-analyzer window and a control window that displayed timing and setup information.

Controls made it easy to set a sample rate (1 kHz--100 MHz in 1-2-5 steps) and a sample size (1 ksamples--32 ksamples in 2n steps). To test the logic analyzer functions, I ran Test 1 so I had some data to capture. Clicking a the green arrow button in the control window started an acquisition, but I could not see any changes in the data and no matter how I adjusted the sample rate, the display timing remained fixed.

Off to one corner, the control window displays a group of two buttons labeled (DSO/LA Zoom). Yes, the label is in parentheses. The buttons carry labels S and nS. Clicking the buttons zoomed the display in or out so I could see more or less of the data. The S button (seconds?) zooms out and the nS button (nanoseconds?) zooms in. After zooming appropriately, I could see the eight logic-zero signals. (Why not label this button group ZOOM, with IN and OUT controls?)

Clicking on the "LA" (logic-analyzer) tab in the control window opened a small sub-window that let me quickly label my signals, designate a signal's source, and change a signal's displayed color. Changing a name did not change the signal's channel number, a nice feature that makes it easy to name signals and still know what points they come from on a circuit. A "Buses" tab lets users group signals and display bus information in octal, binary, decimal, or hex format.

A "Trigger" tab let me select an optional trigger, but trigger choices appears limited. I could select to trigger only on patterns that involved four channels 00--03. And the trigger-setting window would accept any binary pattern except for 00002. The unit's help file explained that exception arises from a hardware limitation. The analyzer may accept an external trigger signal on a separate pin, labeled TRIG, but nothing in the trigger window provided a clue about using this input line. I tried several trigger combinations, all of which worked as expected. When I removed the check mark from the Enable Trigger" option, the analyzer simply acquired data as soon as I clicked on the green arrow button.

The display provides two timing cursors, A and B that in my opinion have little value. Although I could place them anywhere in the display, no information about their location appeared on the screen, nor did information about the time "distance" between them. Without an easy way to make time measurements, I didn't attempt to measure the regular interrupt period from the Rabbit's BIOS software.

The ELAB-080 software will display numeric timing information in a separate window so you can view state changes as they relate to sample times (timing analysis). Unfortunately, the display shows only 19 time-stamped samples simultaneously and you cannot expand the size of this window. I searched in vain for a color change or mark that would indicate a cursor position in the timing-analysis window. The timing-analysis and state-analysis (waveform) windows have no link, so scrolling through one does not move the other. The capability to lock the two would prove helpful in some analyses.

I can understand the manufacturer's desire to have users download the latest software, so the lack of a CD-ROM isn't bothersome. But, the lack of a manual and a quick-start guide troubled me. A user new to logic analyzers and other PC-based instruments needs some hand holding that this product doesn't provide. The software comes with a help file that includes some explanations of controls and displays, but the information about the timing cursors, for example, simply explains they exist and that a user can move them.

Bottom line: The instrument provides more functions that you would expect in one package, which may make it attractive in some lab situations. The lack of triggering options and cursor-timing information, however, will limit the use of this instrument as a logic analyzer. When compared with the other instruments I reviewed, this one failed to deliver the performance I expect from an instrument that costs close to $500.

TechTools DigiView
Price: $499
Garland, TX
Phone: 972-272-9392
www.tech-tools.com

Although the software setup instructions didn't follow the printed user's guide, poking through menus led me to the right choices: CDWeb-->More Links: Software-->d_dv1. The latter program started the installation of the DigiView code. Connection of the module via a USB cable and installation of the DigiView driver went flawlessly, though.

The small logic-analyzer module simplifies connections by providing access to 18 input channels and two ground signals through color-coded wires. Each wire comes with a female adapter that connects directly with 0.025-in square pins like those on my test board. The DigiView package included small clips, too, so users can grab component leads.

As the code started, it produced a colorful display of sample information on which I could try operations. But why take the simple path? I guessed and clicked on Config, which popped open a window that let me select signals by their color-coded wires, assign them a name, choose polarity (invert the displayed logic level), and designate the displayed signal's color. In addition, I easily grouped the eight test signals to form a bus.

The Config menu also assigns trigger conditions--combinations of logic levels on input lines followed by the detection of signal edges. This type of trigger is not as sophisticated as it is on other analyzers, but for many situations it will do nicely. I set up several trigger condition for the eight inputs.

After choosing and arranging my signals, I turned to the display, which presents signals in a clean and colorful format. Two waveform displays show the same information at different zoom levels.

Running the DigiView logic analyzer requires only a click on the Run button. The module acquires data at a fixed 100 Msamples/sec (10 nsec) rate. Unlike other analyzers that store information, the DigiView saves data in compressed form, the equivalent of a 90 Gbyte file, according to the company. The software uncompresses only the data a user wants to examine on the PC's display. Here's the key point. If the module sees no changes on its inputs, it doesn't save anything. Thus, when signals do not change rapidly, the "length" of the saved data record expands. So, if you need to monitor sequences of infrequent changes, this unit fills the bill.

After acquiring data from the test board, I "unlinked" the two displays so I could investigate the digital information independently. Linking the signals, though, lets you home in on an area of interest and then zoom in on it in the other display. Because I defined a bus that combined the eight values, they appeared as one logic trace with superimposed hexadecimal values, as space permits. Bus traces include a small plus sign, which when clicked expands the display to show individual bus signals.

The DigiView software also includes a sophisticated search feature that lets users find specific conditions, such as signals at a specified level, times above or below a limit and so on. When the software finds a condition, it can attach a cursor mark to it. Search criteria you plan to use often can reside in the Search Manager window. The software comes with several examples that show how to apply a search.

The display includes six cursors, A--D, X and Y, which you can move along traces to indicate events and measure relative times. The display notes the period between cursors, which makes them handy for determining the times between events. Because the cursor can "stick" to a given trace, you can easily measure times between events on different signals. Cursors will also "snap" to a signal transition.

The display of Test 2 data showed the timing anomaly expected due to the regular interrupt from the real-time clock on the Rabbit processor board. Placing the X and Y cursors at the start of two "glitches" showed a period of 489 µsec.

Bottom line: An excellent small logic analyzer that offers sophisticated display and analysis options. Easy to set up and use. I recommend it to anyone involved with analyzing logic signals in an embedded system. This unit provides a fixed 1.6-V threshold, which makes it suitable for logic families that operate between 2 and 5V logic. A solid value for the money.

Link Instruments LA-2124-128K Logic Analyzer
Price: $800
Fairfield, NJ
Phone: 973-808-8990
www.linkinstruments.com

Link Instruments provides a family of logic analyzers that operate with a PC over either a parallel-port or USB-port connection. I chose the least-expensive model, which uses the parallel port. Users who have another device connected via the parallel port may opt for a USB-compatible model with a faster sampling rate and more channels, but at higher cost.

The software setup went quickly until Windows asked me to install a device driver. The company's technical support person said the driver exists within the installed software directory (C:/la2124) not on the supplied CD-ROM. He also suggested downloading the latest version (1.34) of the software, which went without any problems.

Hardware setup went easily, too, although I dislike pushing individual wires onto small pins. I'd prefer a cable connected to a single header. The analyzer came with 30 color-coded E-Z-Hook clips, though, so I could have attached the clips to my signal pins and the clips to the analyzer's wires. This model accommodates as many as 24 logic inputs and it provides for an external clock and trigger signal.

After connecting eight wires plus ground, I ran the LA-2124 software and set up Test 1 on the Rabbit test board. A click on the displayed GO button caused the unit to acquire data. The software acquires data in three modes: Single, Normal, or Auto. In single or normal mode, the analyzer waits for a trigger and then acquires data. The Single mode needs no explanation. Normal operation acquires a new buffer's worth of data after each trigger. Auto mode acquires data regardless of the trigger settings.

After adjusting the sample rate with a "+" and a "-" button, I could see useful test results. (The sample rate control offers 14 1-2-5 steps from 5 ksamples/sec to 100 Msamples/sec.) A right click on the signal name column let me change the designation for each signal and its color. I labeled the eight lines Data 0--Data 7 and changed the colors to match the wire colors.

Trigger setup involved clicking on Trigger and then on Trigger word. I set up a pattern of logic levels and don't-care conditions (logic-0 or logic-1) for my group of eight signals. I could select a trigger-true or trigger-false condition and the software let me establish triggers in binary (my choice), hexadecimal, ASCII, or decimal formats. The trigger menu also provides a threshold setting that ranges from -1.15V up to 2.80V. I left the 1.4-V setting unchanged.

You set the relative size of the pre- and post-trigger record lengths by dragging the red trigger cursor along a time line that represents the data-buffer's depth. A small box, or timing window, on this line represents the amount of information in the display window below it.

Four cursors, A--D, let you mark positions in the data, and you can use cursors A and B to measure the time between two points or between each cursor and the trigger point. The BIOS interrupt period, as measured by averaging eight values, came to 488 µsec.

I found the cursors easy to use and helpful. You can track a cursor by its color, and a check mark in a small display also lets you know which cursor you have under control of the mouse. Cursor movements involve "grabbing" one with the mouse or you can select one in the upper-left cursor box and use six buttons to move it left or right in time increments. The cursors also appear along the time line, so you always know their locations. Menu choices let you quickly move the cursors into the display area. I never lost track of them.

In addition to state information, the display includes a timing list so you can use the information for timing analysis as well. The display produced the state data as either hexadecimal, binary, decimal, or ASCII characters. You can assign your own values to the display and can include a translation table. The latter lets you establish op-codes for a processor so you can examine instructions and data as they flow over buses in a system. By adding extra columns of data in different formats, I examined state information as binary and hexadecimal codes.

The software does not "collapse" a group of signals into a bus and it doesn't display a hex, decimal, or other code for bussed signals. But, the state window provides some of those capabilities. The state window and the timing window are not linked, although you can coordinate cursors between them. You can insert a "spacer," essentially a colored bar, between groups of signals to separate them from other signals.

Bottom line: There's a lot to like about this instrument. The built-in capability to handle 24 inputs as well as an external clock and trigger will make this a useful instrument for embedded-systems designers. The capability to add on modules makes it easy to expand digital inputs. I found the menus generally intuitive, although I had to ask a tech-support person how to remove a spacer. (It's Insert Spacer, but Delete Line.) If the analyzer lacks anything, it's the ability to set up edges as triggers and to establish sequential trigger conditions. I give this unit high marks, and in this case, the lack of a printed manual caused no problems.

CWAV USBee ZX Test Pod
Price: $895
Temecula, CA
Phone: 951-693-3065
www.usbee.com

This module provides eight digital inputs, a ground line and external trigger (TRG) and clock (CLK) lines. The built-in wires come with female adapters that slide onto 0.025-in. square pins. The unit includes microclips that can grab component leads. Sampling of digital signals takes place at nine rates, from 1 Msamples/sec to 24 Msamples/sec. Because the module uses a PC's internal memory for storage, acquisitions can run from 1 Msamples up to 810 Msamples, depending on the PC memory available. Users can add modules to acquire data from more than eight signals at a time. (Each module provides a 5-V and a 3.3-V output.)

The display window shows all eight color-coded signals, which users can name as they like. Four columns near the signal names let you select sequential trigger conditions based on logic-1, logic-0, or don't-care states. If you do not set a trigger condition, the module starts to acquire data as soon as you click on the "Acquire" button in the acquisition area. The trigger levels proved easy to manipulate and the module acquired data from the Test 1 and Test 2 program runs without any difficulty.

The display area includes two cursors, X and O that you can move freely through the data. Three windows show the time position of the cursors as well as the time difference between X and O. You can set either X, O, or T (the trigger condition) as the reference for time measurements. Measuring the times between the expected glitch in the data during Test 1 provided an average time of 485 µsec from eight consecutive measurements. I found the cursors and the display controls easy to use. An additional measurement, called Insta-measure, lets you Select, Width, Frequency, Period, or Byte. Select one of these measurements and it appears next to the cursor as you move it among the waveforms. The Byte "measurement" displays the hexadecimal byte value for all eight bits at the cursor's corresponding time-axis location.

Because the module comes with what looks like an external trigger input (TRG), I decided to use it to trigger the logic analyzer. Although the USBee ZX software includes help files that exist as video segments, the triggering options described in the triggering video segment did not match the latest version of the logic-analyzer software. A printed manual might have helped.

How hard can it be to configure a simple trigger input line? I set up conditions to try and clicked on the Acquire button. A blue Waiting-for-Trigger indicator flashed for longer than what I deemed appropriate for the signal I selected as the trigger source. So, I clicked on the Stop button to abort the run. That action sent the software into the "Not Responding" condition and in this state, the Windows Task Manager could not shut down the program. To double check the operation, I tried similar external trigger actions several times and all locked up the program. (Left for long enough, the program eventually shuts down.)

Tim Harvey at USBee at first remarked I had uncovered an unknown bug in the software, which should otherwise let users gracefully "back out" of a condition in which the software waits for a trigger and a clock signal. Harvey later explained I probably needed to upgrade my USB-port drivers because early drivers had a problem that prevented certain timing operations from working properly. Although he provided guidance on finding the latest USB drivers on the Microsoft Web site, never managed to locate them. I keep my lab computer isolated and disconnected from all networks so it will not affect other equipment should an experiment go awry. So, I hesitated to connect it to my router for Microsoft updates. I did manage to update some files, but the USB drivers remained elusive. I'll Take Harvey's word about the need for new drivers.

Bottom line: The logic-analyzer functions work well and the ability to cascade USBee ZX modules in parallel makes the logic-analyzer functions very useful. This unit will appeal to designers who develop small embedded systems and who need basic logic-analyzer functions. Although the USBee ZX modules and software offer capabilities such as signal generation, pulse counting, and so on, I did not experiment with them.

BitScope BS-50 Pocket Analyzer
Price: $345
St. Leonards, New South Wales, Australia
Phone: +61-2-9436-2955
www.bitscope.com

This small module provides logic-analyzer and digital storage oscilloscope (DSO) functions, and it also includes a single-channel arbitrary waveform generator (AWG). Setup went smoothly, although engineers may find it easier to work with a 26-pin header than with individual wires. The color-coded wires that accompany the analyzer provide a grabber on one end and a 0.025-in. female connector on the other. The grabbers crowded the I/O pins on my test board, but they fit snuggly and held in place during testing. (The unit I received for testing came with a small power module. The newest version, the BS-50U derives power from its USB-port connection.)

The BitScope DSO 1.2 software started easily, but it took a moment to realize I had to click the POWER button to "start" the instrument. The software's initial screen looks like a DSO, but a click on the LOGIC button on the right-hand menu produces the eight logic signals the module can acquire. When the software changes to logic-analyzer mode, the upper-left TRIGGER area displays each logic signals and an associated buttons that let me select a logic-0, logic-1, or don't-care condition for triggering. The software does not provide for edge triggering or for a sequence of events that trigger acquisition of signals.

I ran the Test1 code on the Rabbit board and set triggering for a logic-1 on D0 and a logic-0 on D1. The display seemed upside down. This display puts D7 at the top of the display and D0 at the bottom. The software provides no way to change the color of traces or to name the traces. Those capabilities would help users keep track of the bit order and signal actions.

To acquire data, users click REPEAT to continuously acquire and display data whenever the analyzer "sees" the trigger condition and is not busy acquiring data. A click on TRACE, though, initiates a one-shot capture. The timebase controls let users select a timebase (25 nsec--200 µsec) and a zoom factor (10 mX--20X). The display shows a time-delay value (TD), the timebase value (TB), and the sample-rate (FS). I chose a 1 µsec timebase and a 1X zoom factor and the data generated by Test 1 appeared in the display area. Although the display triggered properly, I could not related the TD value to anything I had set or could control. It turned out this value shows the displacement of the currently display data from the trigger point. That's helpful information, but the manual doesn't explain how to set this value, where it comes from, or how to relate it to other measurements.

The display provides an x-axis cursor and a display of the x cursor's time relative to the trigger. But good luck if can't remember where you placed the cursor last time you used it. No subdisplay shows the location of the cursor relative to your current view of the acquired data. And you can't click within a display of data to retrieve the X cursor from a distant or forgotten position, you must hunt for it.

I performed the BIOS interrupt timing measurements on the Pocket Analyzer's data and calculated an average period of 488 µsec.

Bottom line: Before I'd consider using this analyzer in the lab, the manufacturer would have to upgrade its software. The inability to easily move a timing cursor and the lack of signal-name assignments and color codes makes the instrument difficult to use. Because the manual concentrates on DSO measurements, logic-analyzer operations get scant coverage. Also, the lack of any Help files means the manual--which could go missing--provides the only local documentation.

Saelig ANT16 USB Logic Analyzer
Price: $333; 16-channel clip-lead set and cable, $39
Pittsford, NY
Phone: 585-385-1750
www.saelig.com

The ANT16 analyzer I received came with an analyzer module, clip-lead set, getting-started guide and mini-CD-ROM. The equipment provides a basic logic analyzer module and software that communicate through a USB connection. Software setup took little time, although Windows XP displayed a caution message about using software that had not passed "Windows Logo" testing. I ignored the message and proceeded to start the logic-analyzer software. The unit provides sample steps in 21 increments, from 100 Hz to 500 MHz, plus a synchronous mode controlled by an external clock input.

Setup with my test board took little time, although placing eight signal clips and two ground clips on a row of 0.025-in. header pins made for a crowded space. Users may want to make up their own color-coded cables with smaller clips from a supplier such as E-Z-Hook. Test leads that push directly onto square pins would help, too.

After I connected the analyzer and the test board, and started the Ant16 software, a small display at the bottom of display indicated signal activity on all the inputs I had connected. I like this feature because it indicated the module "sees" activity on its connections.

The logic analyzer had no difficulty displaying the results of my standard two test, although figuring out the triggering modes caused some head scratching. The manual's academic descriptions of how trigger conditions work provide no hands-on explanations of how to make the triggers work. Without screen shots, references to menus, and step-by-step instructions about using triggers, the manual has little value and it diminishes the value of this product.

Triggering involves setting levels, signal edges, and don't-care conditions for the signals in use near each displayed signal. Once I figured out how to set a simple trigger condition to match a pattern or detect a signal change, I had no difficulty capturing data. A slider on top of the display lets users set the pre-trigger record from 10% to 90% of the display in 10% increments. I would have liked more control over this setting, but that's a personal preference rather than a criticism.

Don't expect the ANT16 to operate like an analog scope, though. The display doesn't get refreshed until the software detects a trigger condition and acquires all the data. In continuous mode, I thought I'd see a constant flow of digital information, but instead, the software constantly updates the entire display after each trigger. It takes a bit of getting used to.

Although the cable provides a connection for an external trigger signal or clock input, the manual lacks information about how to set up and use this line. Likewise, a separate output line, labeled Trigger Out, may serve a useful purpose, but no documentation describes it.

The display provides a cursor that shows the hexadecimal and decimal value for an 8-input group of signals. I found no way, though to change the display to octal or to group other signals and display their numeric value. The display also includes two cursors (red and blue). The red one moves, but the blue remains fixed. The manual dismissed these cursors with one unhelpful sentence. Clicking on the logic signals moves the red cursor and provides timing information, which I found helpful. The purpose of the blue cursor remains unknown.

While running the high-speed binary-counter program (Test 2), the display showed the expected odd timing patterns. I used the red cursor to extract the time stamps for the observed "glitches," although I had to subtract sequential values to get time differences. Positioning two user-controlled cursors to compute a time difference would have sped the process. After measuring seven periods between glitches and averaging them, I came up with a 488 µsec period between them. That value matches the period expected from the test board.

Bottom line: Although this analyzer's logic-signal capture works well, I cannot recommend it until the manufacturer, USB Instruments, revises its manual to include how-to information and examples that show specific settings and operations. The software may provide many capabilities beyond those I discovered, but who knows. And I'd still like to know what the blue cursor does.

NCI GoLogic-72 System
Price: from $4300
Huntsville, AL
Phone: 256-837-6667
www.nci-usa.com

The GoLogic-72 System provides for as many as 72 digital inputs and eight clock inputs, which makes it powerful tool when testing and debugging embedded systems that require monitoring of several buses, I/O ports, and control signals. The unit I received came with all 72 connections, split into two groups of 32 each. Each group of eight labeled connections passes through a small harness that makes it easy to handle them all simultaneously. This feature simplifies setup when attaching connections to a board. The wires each furnish a female connector that pushes onto 0.025-in. square pins, which is how I used the logic analyzer. NCI also provides micro-grippers ($4 each) and nano-grippers ($7 each) to connect with small and tiny component leads.

The GoLogic system comes with a CD-ROM and it took no time to install the software and get it running. The module connects to a power cube (supplied) and to a PC through a USB cable. The comprehensive manual provides basic information on how the analyzer works and then goes into details about sampling, clocking, state analysis, timing analysis and triggering. Unfortunately, the manual looks formidable and no quick-start tutorial exists to take me on a tour through the many options and menus.

Lacking such a tutorial, I plunged in and connected the lines labeled A0--A7 plus two ground lines to my test board. A setup menu let me select a sample rate, memory depth, and the type of sampling. The latter provides nine options that include I2C and SPI modes--ideal for troubleshooting serial communications in embedded systems.

Next, I selected the channel groups, which showed the wire color codes and indicated signal activity on lines A0--A7. The threshold for my signals was set for +1.58V, but I could reset it for ECL signals, or adjust it between -4.90 and +5.27 for other logic families. That's a nice feature when you plan to use the GoLogic analyzer with different logic devices.

After getting antsy to start something, I downloaded Test 1 to the Rabbit test board to make information available to the analyzer. If you need to perform more than basic triggering, plan to spend time learning about the various triggering modes by experimenting with them. To start acquisition of data for Test 1, I wanted to trigger or a falling edge on signal A0. Clicking on the Setup tab let me select a simple edge trigger, but I couldn't figure out how to select the A0 signal. After I gave up and switched back to the Waveforms display I noticed small gray squares associated with each signal. Clicking on these let me select a rising, falling, rising-or-falling, or unused triggering attribute. It's unfortunate this display lacks a trigger label for these squares. Their use becomes clearer after using the software for a while. You can scan through the trigger settings you used previously, which makes it easy to keep several settings at hand for complex tests.

Once set to trigger on a falling edge for A0, the software acquired and displayed data. (Note the edge symbols in the gray squares to the right of the colored signal labels.)

Displays that include at least one channel (signal) include three pairs of cursors you can move with the left or right mouse button to measure times between events. Luckily, you don't have to drag a cursor from place to place. Clicking on an area of the waveform display moves the selected cursor to the clicked-on position. The cursor notations need some enhancements, though. I found it difficult to tell which set of cursors was active and which mouse button controlled a given cursor. Often, I meant to move one cursor and ended up moving the other.

Moving on to the binary-count test (Test 2) let me set up a pattern trigger for FF (hexadecimal) and I set the duration of the trigger for at least 8 µsec, which put it about in the middle of an LSB=1 level. That trigger setup worked fine and the display provided the expected data. In this test, the gray trigger squares indicated logic levels rather than transitions. You can click on these squares to change trigger conditions without going back to the Setup menu.

The logic-analyzer software also provides a numeric display of information, sample by sample. So, if you work with data values on a bus and want to monitor those rather than waveforms, the software lets you. You can still use the cursors to mark values and determine the time between them. If you have a waveform and a numeric display open simultaneously, moving around in one display also moves your view in the other. Thus, you can easily track logic levels and numeric data.

The logic analyzer detected the anomaly caused by the Rabbit board's BIOS and measuring the time over 10 occurrences provided a period of 489 µsec. The display also showed two "glitches" on the A7 and A6 lines where the other signals change state (see Figure). The period of these glitches always equaled the sampling period; 8 nsec at 125 Msamples/sec and 20 nsec at 50 MHz. So it seemed unlikely the glitches existed--they would not expand and shrink in sync with the sampling clock. But glitches might arise from simultaneously switching so many loads (the LEDs), so I wrote another test program that alternated rapidly between the values 7F and OO at the processor's output port. In effect, the software switched outputs A6--A0 simultaneously on or off without affecting output A7. I monitored A7 with my Tektronix 465B oscilloscope (P6105 probe) and saw no activity of A7.

Do the glitches observed with the logic analyzer, but not with the scope, actually exist or does the logic analyzer "see" something that doesn't exist? I'd need a storage scope to tell.

Bottom line: Although priced much higher than any other analyzer, the wealth of triggering options provides a strong justification for buying this unit. The display options and the cursors make it a solid analytical tool. Plan to spend time learning about the triggering modes and options. Unfortunately, lack of time prevented me from exercising many capabilities, such as searching data for patterns and triggering a storage scope and displaying analog data simultaneous with digital information. The analyzer will link with disassemblers and will display C/C++ source code so you can observe op-code-by-op-code operation of a program. Software add-ins that provide these functions cost extra.

How We Ran the Tests

To test the logic analyzers described in this series of hands-on reviews, I used an older Rabbit-2000 development kit from Rabbit Semiconductor. (The Rabbit processors expand upon the capabilities of the popular Z-80 instruction set.) The two-board set includes a Jackrabbit processor-and-memory board and a Development Board that provides the usual LEDs, push-buttons, and rows of contacts. These contacts provide access to the Rabbit processor's built-in I/O ports. This board served as a test setup for previous experiments, so I had several simple C-language programs on hand that would readily exercise one of the processor's general-purpose 8-bit parallel port. (See Notes.)

Software development relied on Rabbit's Dynamic-C and Dynamic-C Premier integrated development environments (IDEs). Software downloading took place over an RS-232C connection between my Dell Dimension 8300 lab PC and the Rabbit processor board. The board provides both SRAM and FLASH memory. But because programming--and reprogramming--the FLASH memory could take extra time, I opted to run compiled programs from SRAM. The two test programs provided the following functions (See Notes):

Test 1: Shift a logic zero individually through all eight bits.
Test 2: Standard 8-bit binary count.

Although the tests noted above seem simplistic, they let me test the logic-analyzers' abilities to measure time intervals. Rabbit board includes a basic I/O system (BIOS) program that handles start-up and shut-down operations as well as a watch-dog timer and a real-time clock (RTC). So, instead of observing only the operation of the test programs, I expected the logic analyzers to show the effect of cycles "stolen" by the BIOS code to capture and process interrupts or to handle other house-keeping tasks. According to Rabbit's documentation, the RTC causes a periodic interrupt every 16 clock cycles (32.768 kHz clock source), or every 488 µsec. Interrupt servicing involves incrementing several memory-based counters. Figure X shows the effect of the BIOS interrupt on a 5-bit binary counter. Note the stretched clock periods in the least significant bit (top waveform).

Note:

1. Digi International (Minnetonka, MN) acquired Rabbit Semiconductor, Inc., formerly Z-World, in May 2005. Although Rabbit Semiconductor no longer produces the Jackrabbit board I used, the company still offers a wide range of processor boards and development kits that provide comparable functions.

2. Test Code:

```
//  Test 1--Shift a logic zero through all eight bits at output port A.

main()
{
	int j;
	int code;
	int data;

	// Write 84h to slave port control register
	// Set Port A as an output port

	WrPortI(SPCR, &SPCRShadow, 0x84);
	WrPortI(PADR, &PADRShadow, 0xff);

	while(1) {  // begin endless loop

		//WrPortI(PADR, &PADRShadow, 255);  // Turn all bits off
		code = 255 - data;
		WrPortI(PADR, &PADRShadow, code);

		data = data * 2;
		if (data == 0) ( data = 1);
		if (data >= 256) (data = 0);

	}	// end while loop

} // end program

==========


//  Test 2--Produce an 8-bit binary count at port A.

main()
{
	int j;
	int code;
	int data;

	WrPortI(SPCR, &SPCRShadow, 0x84);

	// Set all 1's at port A

	WrPortI(PADR, &PADRShadow, 0xff);
	WrPortI(PADR, &PADRShadow, 255);
	code = 0;
	while(1) {  // begin endless loop

		WrPortI(PADR, &PADRShadow, code++);

	}	// end while loop

} // end program
```