Errata

Document Title: Timing Characterization Using the 16517/18A with Intel Pentium Processor Measurement Examples (AN1261)

Part Number: 5091-8798E

Revision Date: November 1993

HP References in this Application Note

This application note may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this application note copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

About this Application Note

We've added this application note to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent website:

www.agilent.com

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

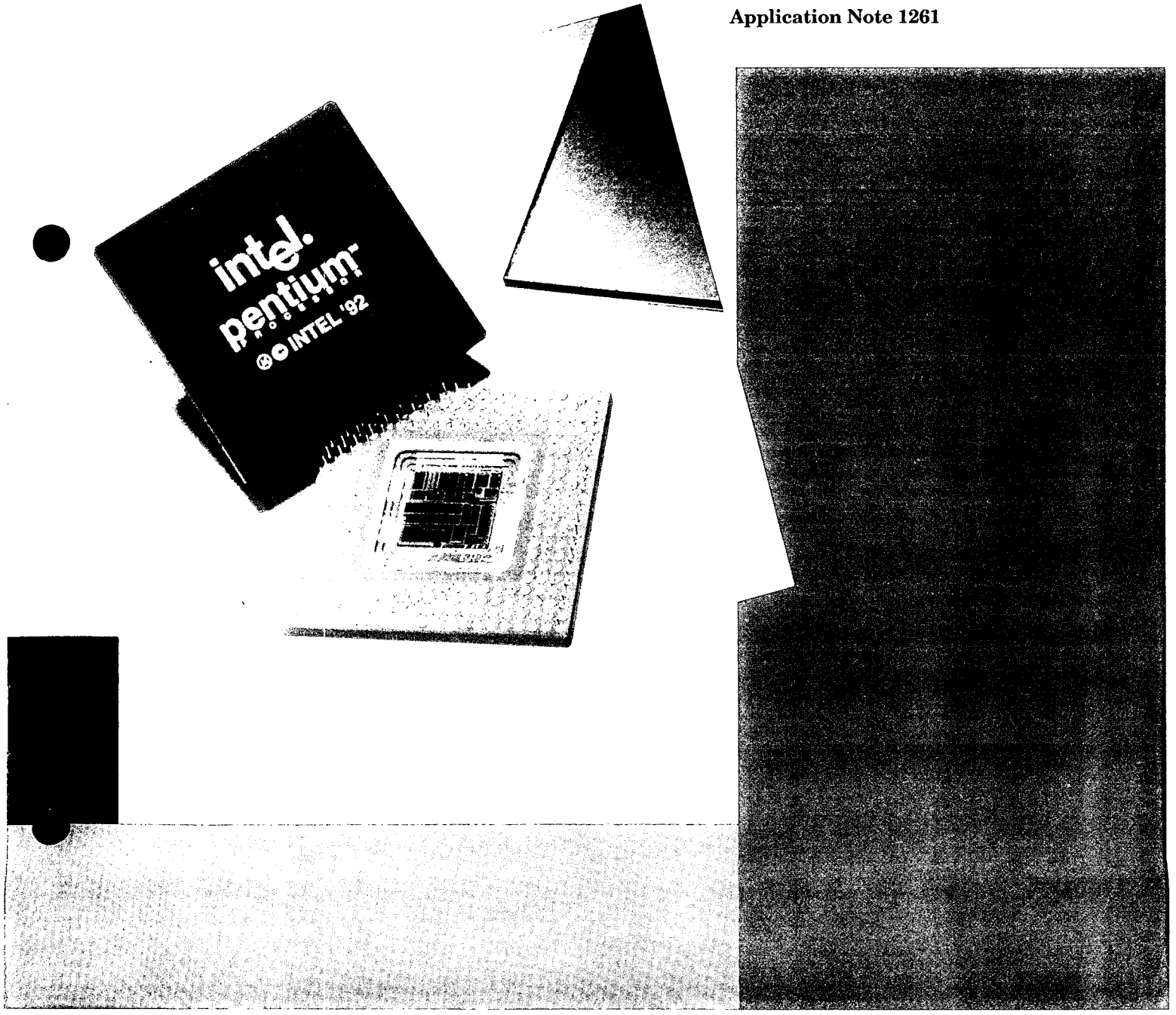
Agilent Technologies

HEWLETT® PACKARD

Timing Characterization Using The HP 16517/18A

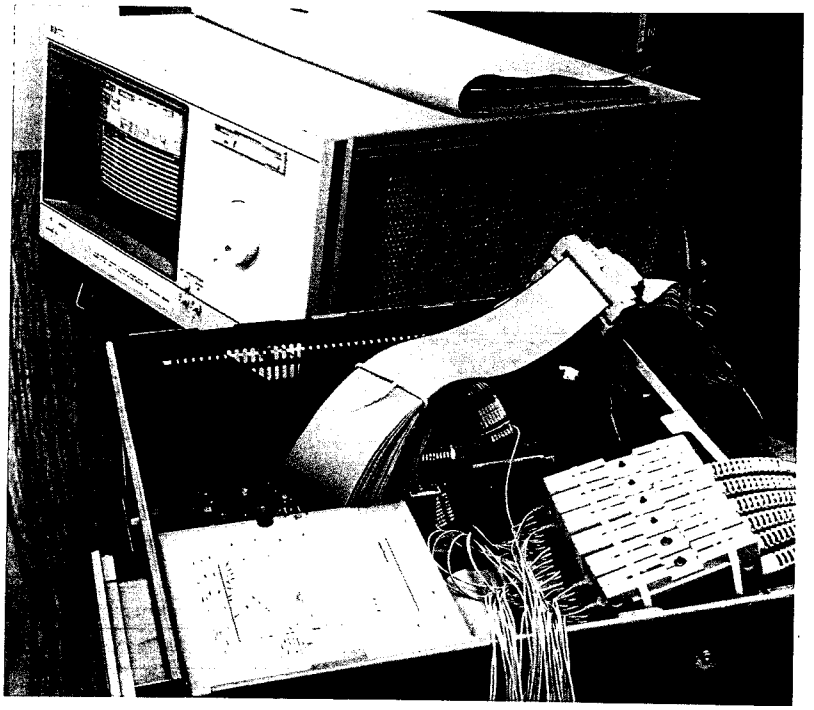
With Intel Pentium™ Processor Measurement Examples

Application Note 1261



Introduction

This application note discusses how the timing characterization of a digital system can be performed more efficiently and effectively with a high-speed logic analyzer. Time-to-market issues limit the time digital designers have to characterize a new design. As a result, some measurements, particularly bus measurements, do not get made. This creates the potential for overlooked timing problems, which then show up during production test or after the product starts shipping. First, the current approach to timing characterization will be discussed. Then, examples of actual measurements made on a Pentium processor running at 50 MHz will be presented. Finally, the method used to probe this dense, load-sensitive IC will be discussed.



The Intel Pentium system used to make the example measurements included in this application note.

Timing Characterization vs Timing Validation

The measurements presented in this application note may be used to better evaluate the timing performance of a new design. We have called this timing characterization. It might be argued that what we are really doing is timing verification, given the 500-ps time interval accuracy of the logic analyzer. We would not argue with this premise, at least not for the Pentium processor used to make the example measurements. However, most timing measurements made using the techniques presented here will establish that the margin is large enough to preclude a more precise measurement. It will also pin point where margins are questionable or where outright violations occur. A scope can then be used to characterize the specific questionable areas, complementing the logic analyzer.

An overview of timing characterization

Timing characterization is an important part of the development of complex systems or ASICs. It examines the relative timing between signals coming from a chip or system. The primary purpose of timing characterization is to establish the margin between the minimum acceptable specifications and the system's actual performance. How much (or how little) margin a system has provides insight into its reliability, as well as the amount of performance given up in a too conservative design.

Traditionally, high-speed oscilloscopes are used to probe the system under development and to capture timing waveforms. These waveforms are then compared against the timing specifications of the system. State-of-the-art logic analyzers now provide a more powerful and flexible solution over oscilloscope methods. Logic analyzers can be used to trace complete buses using setup/hold violation triggering. Oscilloscopes can still complement this approach by providing measurements for problem areas identified by the logic analyzer.

Every digital system has a set of timing requirements associated with it. These requirements are often specified as propagation delays, setup times, hold times, or maximum/minimum valid delay times. RAMs have access times that correspond to the time it takes to read an internal location and present its contents to a processor. PCs have timing specifications that govern the signals appearing on the interface bus for add-in cards. Systems and components that do not meet their timing specifications often cause failures in the systems they are a part of.

During product development, it is necessary to characterize the system signals. This characterization can be performed at two places: at the system interface and at internal nodes. The system interface is where timing specifications usually apply. In the case of ASICs, the pins of the chip are the system interface, and are the only practical place to make timing measurements. Designers must be able to verify that the timing characteristics of their system meets the original design criteria. In addition to interface measurements, several critical internal nodes of a system may be selected for testing. By measuring the timing of these internal nodes, the inherent timing margin of the system can be checked. An unusually small timing margin may suggest that the system is near functional failure. Also, that the design has little tolerance for process variation in its components. Both system interface and internal signals are often measured as the system is taken to voltage and temperature extremes, ensuring that the system meets its specifications under changing environmental conditions. Characterizing system timing can be easy or difficult, depending on the type of measurement to be made. Verifying the timing relationship of two system outputs is usually a simple measurement. The task becomes more difficult when bus signals, such as a data bus, must be compared against a control signal output. Four-channel oscilloscopes can compare only three of the bus signals at a time against the control signal, requiring tedious changing of probe locations. Among the most difficult timing measurements are setup and hold times. These often require an external stimulus if the signal is an input to the system under test. This input signal can be provided by installing the other components of the complete system, or by using an external pattern generator to provide simulated input signals with controlled transition times. With a pattern generator, the signal edges can be moved with respect to the input clock until a system failure is detected.

Test equipment used for timing characterization should have sufficient timing accuracy and number of channels to accommodate the required measurements. The timing resolution of the equipment should correspond to the speed of the circuitry being tested. High-speed ECL

gates require finer resolution than TTL logic or programmable gate arrays. With the push toward higher performance systems, it is likely that all timing characterization measurements will require time interval accuracy of at least 1 nanosecond. Channel-to-channel timing skew is an important parameter that is often overlooked. Poor timing correlation between channels defeats the purpose of high timing resolution. The skew between channels should be no worse than the sample time, and preferably half that amount. This implies that modern timing characterization test equipment should have skews of 500 ps or less. The number of input channels is another factor that is dependent on the specific test configuration, but it seems that there are never enough. Though two- and four-channel test equipment can be considered to be the bare minimum, sixteen or more channels are required for complex bus-based systems.

High-speed oscilloscopes have often been used for timing characterization. Though they have excellent timing resolution, sometimes 20 ps or less, they are usually limited to four input channels. High-performance logic analyzers, such as the HP 16517A, are often a better solution, with a 250-ps sample period while using eight input channels. For measurements that do not require the full 4 GSa/s timing, the sample rate can be decreased to 500 ps and the number of channels doubled to sixteen. The HP 16500 family of mainframes can hold up to four additional high-speed modules, expanding the input capability of the system to 80 channels (at 2 GSa/s).

Timing Characterization Examples using the HP 16517A and 16518A with the Intel Pentium Processor

The following measurements, while specific to the Pentium processor, can easily be applied to any digital system. The examples will help designers characterize CISC or RISC processors, ASICs, or most any aspect of a new digital design. The system used to make the example measurements is an Intel Express desktop system, with a 50-MHz Pentium CPU module plugged into it. The Pentium processor was prerelease silicon. This, as you will see in figure 7, made our measurements more interesting. The Intel CPU module uses the 82496 Cache Controller and 82496 Cache SRAM for the secondary cache memory module and the main memory interface. The data bus on the CPU module is 64 bits. Main memory is on the mother board of the system and has a 32-bit data bus. This system will be referred to as the target system.

The measurements were made with one HP 16517A and two HP 16518A logic analysis modules in an HP 16500B frame. This setup provided 24 channels at 250-ps resolution, or 48 channels at 500-ps resolution. All measurement examples were made with 250 ps resolution. Both the address and data buses were probed to provide examples of both types of measurements. Therefore, neither bus was fully probed. Normally each bus would be fully probed and characterized before going on to the other.

Maximum/minimum Valid Delay on Address bus

A common problem in system design is excessive loading on one or more of the address lines. The added time delay caused by the load increases the settling time for the address bus, which could lead to intermittent system crashes. One timing specification that would be impacted by excessive address bus load is the maximum/minimum valid delay (t6, table 7-4, Vol. 1 Pentium User's Manual). An interesting note, Intel assumes CL = 0 for all their specifications; therefore, any load that a designer adds to the address bus contributes delay. In particular, a designer who puts together his or her own secondary cache controller is going to want to see what impact it has on the settling time of the address bus.

A Simple Trigger

The timing relationships of the target system clock (CLK), address strobe (ADS#), and address lines A3 through A11 for our Pentium processor can be seen in figure 1. Address lines A0, A1, and A2 do not appear on the Pentium processor, the Byte Enable signals BE0 through BE7 are used instead. Asserting address strobe tells the memory subsystem that the next rising edge of clock will be the beginning of a new memory cycle. In this example, the maximum/minimum valid delay specification will be examined. It is referenced to the rising edge of clock that occurs immediately before the one that starts the memory cycle (see figure 1). Intel guarantees that the address will be valid no earlier than 1.5 ns and no later than 9.0 ns after the rising edge of clock. Again that is with no load on the processor. The goal is to see what the valid delay is with our target's load on the processor.

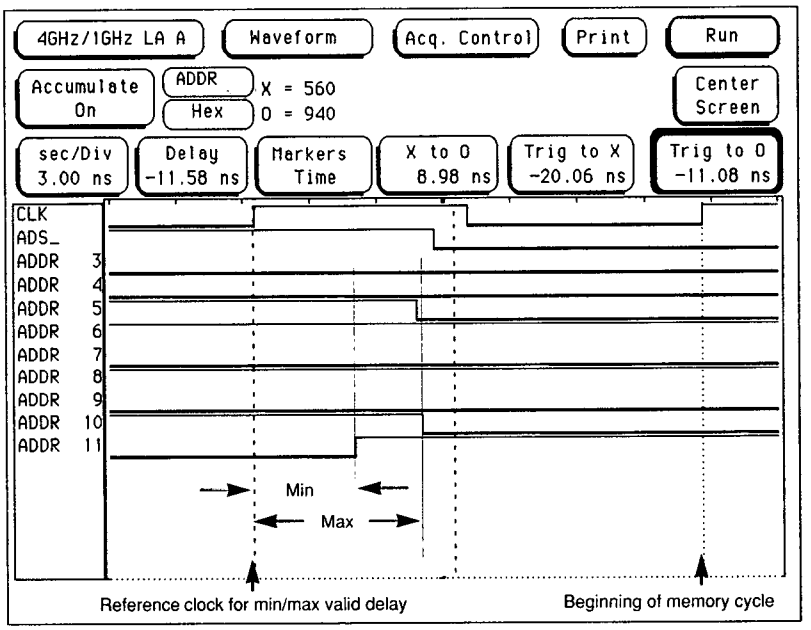


Figure 1 - Trace of Pentium processor clock and address signals.

Using the Accumulate Mode

The trace in figure 1 was obtained by finding the falling edge of ADS#, and then by triggering on the next rising edge of CLK. Using the markers on the measurement in figure 1, the minimum valid delay was found to be 4.5 ns and the maximum to be 7.5 ns. However, a single measurement like this is not sufficient to characterize the target system. The address bus may behave differently in different areas of memory and under different processor cycles. So, the next step is to use an accumulate mode, such as is found in the HP 16517A and 16518A analyzers. This feature allows you to combine a large number of measurements into one display, so you can see the worst case of the timing you are interested in. Ideally, the measurements cover all areas of memory and all methods of accessing memory. The logic analyzer cannot capture every memory cycle, but running the analyzer for a long period of time, while the processor executes a loop that exercises all combinations, may provide a statistically valid measurement. Figure 2 shows such a measurement. Placing the X marker on the slowest signal, the worst-case maximum valid delay time for this group of measurements is 8.50 ns. Therefore, the Intel target system should not have any problems with address settling times. You will see, however, in the next example that this is misleading.

Before going further, implementing such a measurement of the full address bus on a four-channel oscilloscope would take fifteen iterations of changing the probe setup and executing the test. Two of the scope channels would have to be used to monitor the clock and ADS# lines and the remaining two to monitor address lines. If the test procedure includes operating the target system over temperature and voltage variations, the test time could become unreasonably long using an oscilloscope. Therefore, such measurements may not be made during the development of a new system, which could leave timing problems undiscovered until the product starts shipping. You will see an example of this in the next measurement. New high-speed logic analyzers, like the HP 16517A and 16518A, have the precision to make these measurements accurately and the channel count to make them efficiently.

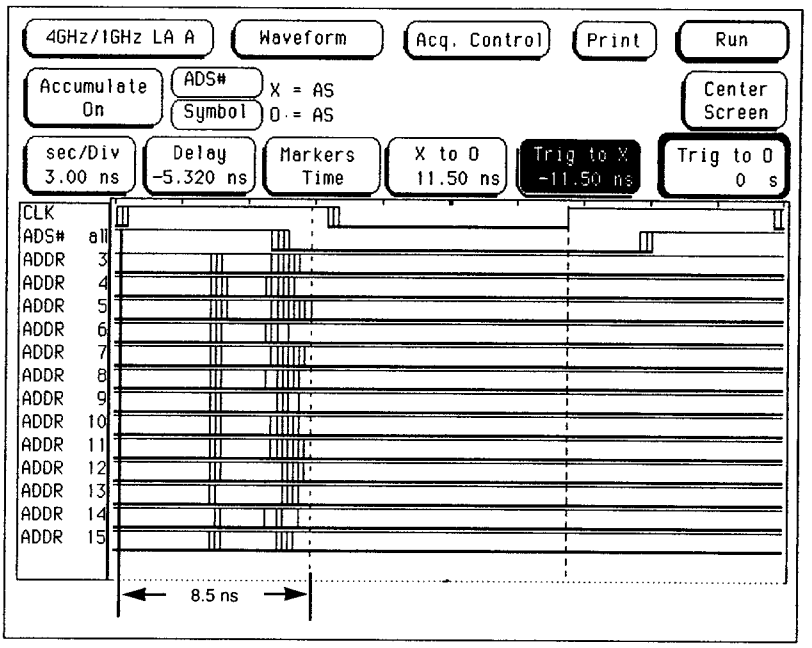


Figure 2 - Pentium processor address bus with accumulate function enabled.

Using a Violation Trigger

One problem with the accumulate approach is instrument dead time. This is the time the instrument spends, after it triggers, pulling in the data and updating the display. This is a problem for both scopes and logic analyzers. Even an instrument that is capable of taking 200 traces per second has a lot of dead time. If you consider a processor running at one million cycles per second, a lot of cycles do not get looked at if you only trigger on 200 per second. To get a statistically valid set of measurements, you would have to run the instrument for a very long time.

Fortunately, there is an alternative to the accumulate method which is violation triggering. The HP 16517A and 16518A are capable of triggering on violations of timing specifications. This is a particularly powerful approach since it allows the analyzer to look for a timing violation in every address cycle that occurs. Only when the analyzer triggers is there any dead time. The HP 16517A and 16518A have trigger macros that include three setup/hold violation variations (figure 3). These macros only trigger the analyzer when a violation of the specified setup or hold time has occurred.

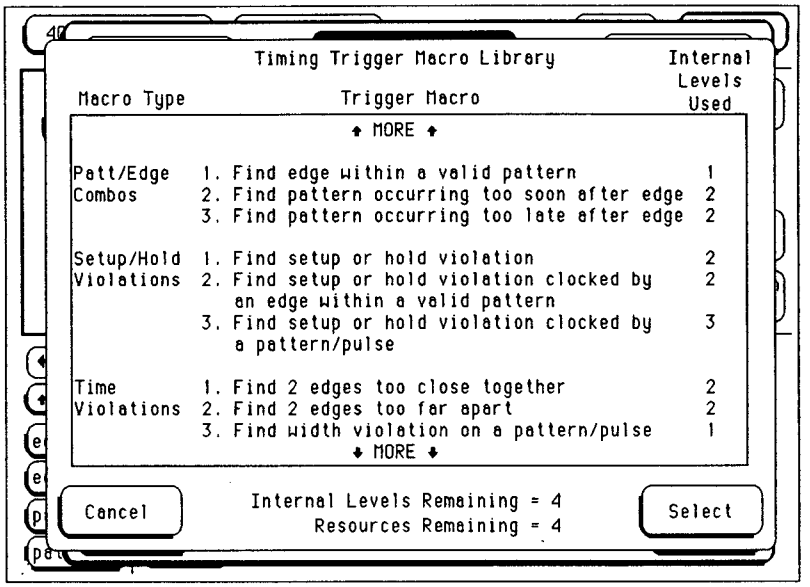


Figure 3 - Timing trigger macros on the HP 16517A and 16518A.

Let's take a look at how a setup/hold trigger macro could be used to catch a maximum valid delay violation. Minimum valid delay was not examined since there is not normally much interest in this measurement. Figure 4 illustrate the calculation necessary to do this. First, the rising edge of clock must be used when ADS# is low as reference for the measurement. The macro needs a qualifier that occurs at the time of the specified reference edge (rising clock). Second, the 9-ns maximum valid delay is subtracted from the 20-ns clock period to come up with 11 ns of setup time. So, a violation of the 11-ns setup time is a violation of the 9-ns maximum valid delay time. This assumes the clock period is stable, which given Intel's specifications on the Pentium's clock, is a reasonable assumption. It is also possible to create a custom trigger specifically for the maximum/minimum valid delay.

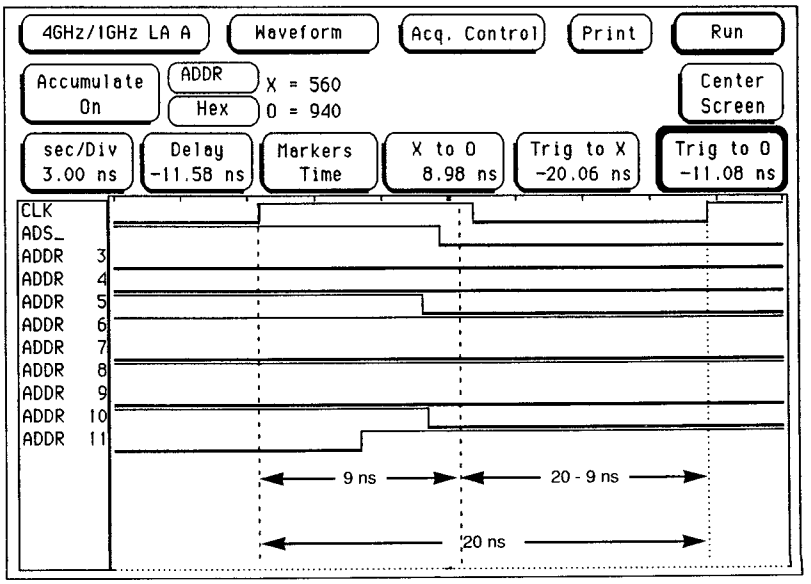


Figure 4 - How to calculate maximum valid delay for setup violation triggering.

Figure 5 shows the "Find setup or hold violation clocked by an edge within a valid pattern" macro parameter entry screen. It shows that the analyzer will look for transitions on any of the lines of the label ADDR, which is the label for the portion of the address bus that was probed. It will use CLK_RISE as the reference edge, but only when ADS_LOW is true. In addition, the setup time has been set to 2 ns and the hold time has been set to 8 ns. In other words, if transitions occur on the address bus within 2 ns before the rising edge of clock and ADS# low, or if a transition occurs within 8 ns after the rising edge of clock and ADS# low, the analyzer will trigger. Figure 6 shows the trigger term definitions that went into creating this trigger.

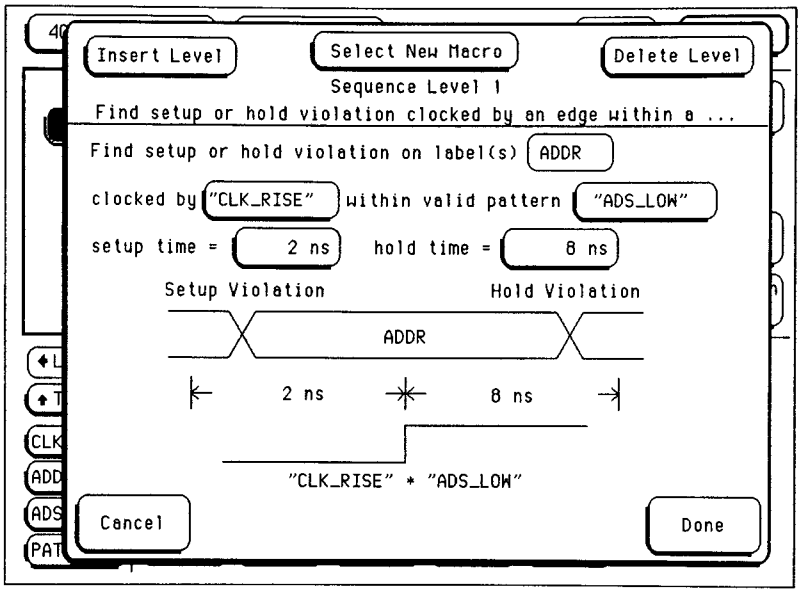


Figure 5 -Parameter entry screen for setup/hold violation trigger macro.

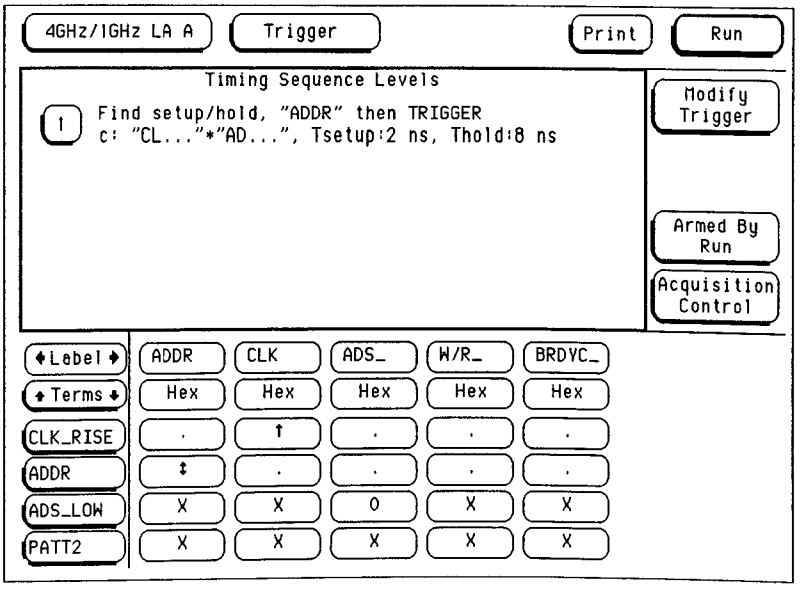


Figure 6 - Trigger resources used in the setup/hold violation trigger.

Figure 7 shows a violation captured by this violation trigger. This is a significant violation; the address bus does not become valid until 12 ns after the rising edge of the clock. In other traces using this trigger, the delay has been measured to be as long as 17 ns. This definitely has the potential to interfere with memory setup time. This problem seemed to occur very infrequently and only during certain phases of the boot process of our Pentium target system. Since the Intel target contained a processor with prerelease silicon, the problem could be attributed to the processor rather than to the target system's design. However, had this been a hidden timing problem caused by a memory module and this measurement had not been made, the problem occurred so infrequently that it might not have shown up until the product started shipping. Fixing it then would have been very expensive.

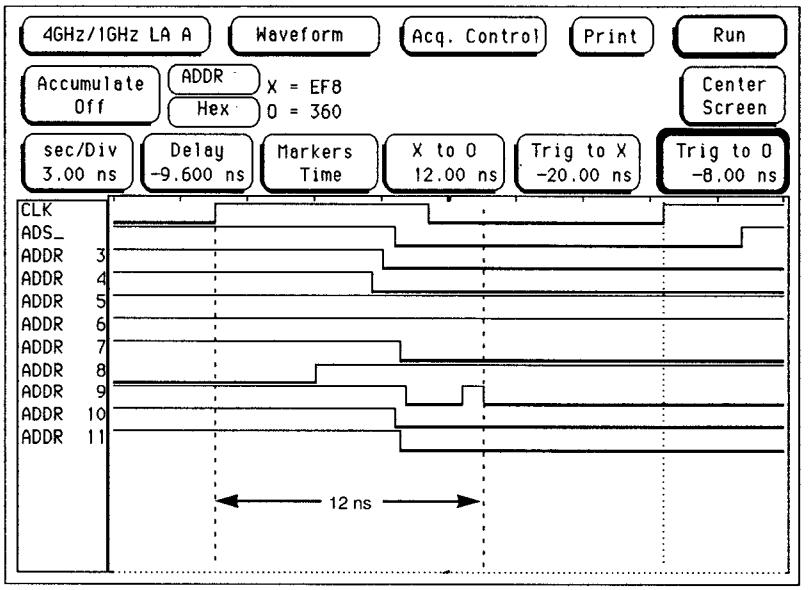


Figure 7 - A captured violation of the maximum valid delay.

Getting the Processor's Context

It would be useful to see the context in which this violation occurred, that is, what cycle the processor was in, what instruction it was executing and the full address it was at. Viewing all this is possible by using the Intermodule Bus of the HP 16500B frame. By adding two HP 16550A modules with an HP E2443B Pentium preprocessor to the frame, you can get context. The HP 16517A and 16518A modules can arm the HP 16550A modules, which will then take a state trace of the Pentium processor at the time of the violation. The trace is automatically inverse assembled, which makes it easier to read. Both the violation and the listing can be displayed together on the screen, time correlated. Figure 8 shows the violation and the listing from the Pentium processor. It appears that the violation always occurred after a memory write. By collecting enough traces, you may be able to determine the address range that the violation occurs in.

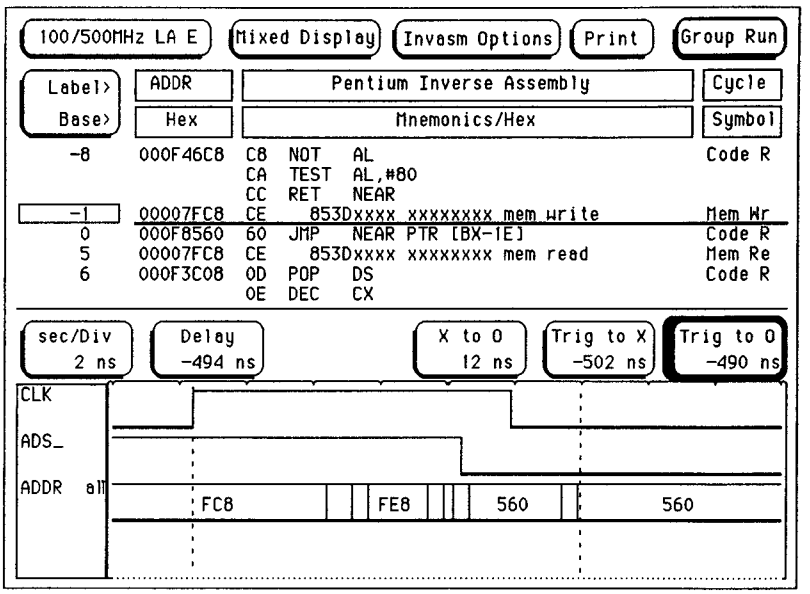


Figure 8 - Code flow at the violation of maximum valid delay timing.

Setup/hold for Data bus

Figure 9 show the timing of a secondary cache read cycle on the Pentium data bus. The first rising edge of CLK after BRDYC# goes low is when the processor clocks in data. Intel specifies the setup/hold times (t34/t35, table 7-4, Vol. 1 Pentium User's Manual) for the data bus. The specified πsetup time is 4.5 ns and hold time is 2 ns. This timing is of interest to designers who have developed their own secondary cache modules. However, this measurement is also not made very often because of the difficulty of doing it with a 4-channel scope. Three channels are required for the CLK, BRDYC#, and W/R# lines. That leaves one channel, which must be moved 64 times in order to capture the entire data bus.

Using a Simple Trigger

The measurement in figure 9 was made by finding the falling edge of BRDYC#, and then by triggering on the rising edge of CLK, while W/R# is low. Setup/hold timing only applies to data reads; therefore, the measurement must be qualified as a read cycle. For this particular measurement, hold time is no problem (off the screen) and the margin for setup time is 7.75 ns. However, as noted in the last example, a single measurement does not characterize a system.

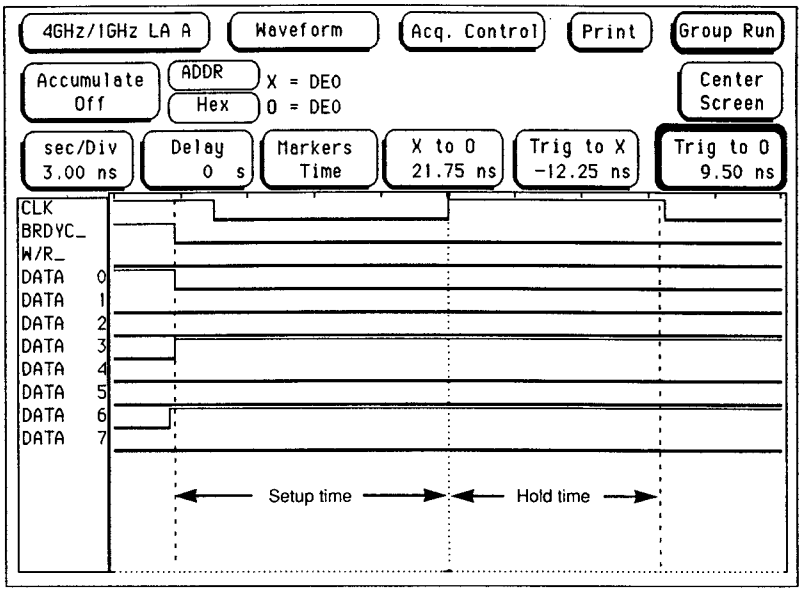


Figure 9 - Setup/hold timing for data reads from secondary cache.

Using a Violation Trigger

Violation triggering will provide a better measurement. The focus here will be on the hold time, since the measurement in figure 9 really did not give a feel for what the margin might be. Figure 10 shows a trace obtained by using the setup/hold violation trigger macro, with the hold time set to 8 ns. Setup time was set to 2 ns, so no setup violation trigger would occur. Before the measurement in figure 10 was made, however, the hold time was set at 2 ns to confirm there were no violations of the Intel specification. Since there was no trigger, you know with great certainty that the target system is not violating the specification. With violation triggering, the analyzer examines every secondary cache read cycle for a violation, as long as the analyzer is running. Violation triggers like this are next to impossible to set up on an oscilloscope.

Next, the hold time was set to 4 ns, and then to 6 ns (the resolution of the analyzer's sequencer is 2 ns) with no resulting triggers. Finally, it was set to 8 ns, and the trace in figure 10 was captured. From these measurements, you can see that the hold margin is definitely no less than 4 ns and that it could be as much as 5.75 ns.

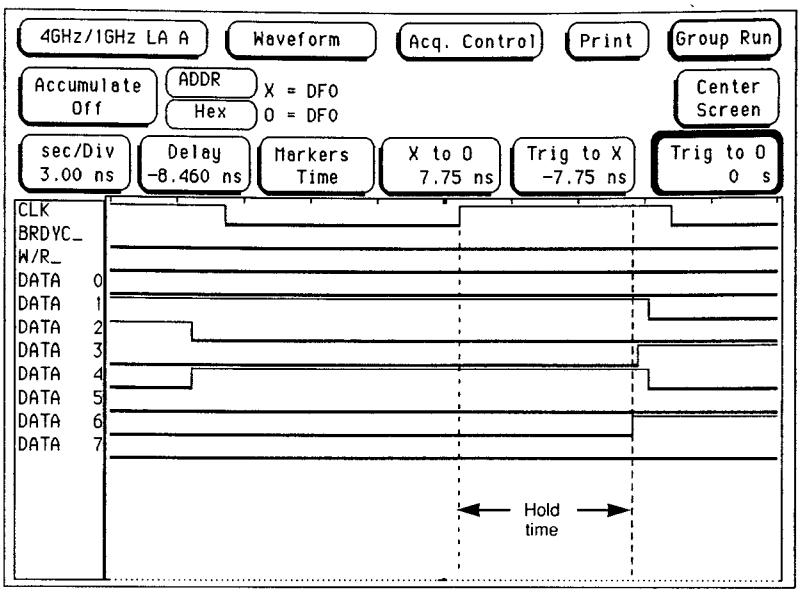


Figure 10 - Timing trace captured using violation triggering.

Using Statistics

The statistics mode of the HP 16517A and 16518A can be used to further refine this measurement. The X marker is set to lock onto the rising edge of CLK just before the trigger occurs. This provides an automatic measurement of the hold time. Putting the analyzer in statistics mode and running it repetitively produces the trace in figure 11. You can see from the resulting statistics that the worst-case hold time was 7.5 ns, which gives a margin of 5.5 ns. Note that the statistics we have obtained here are for hold times that have violated the timing specified in the trigger, they are not statistics for hold times in general.

When running repetitively, the violation trigger is better than just triggering on each secondary cache read cycle. It allows the analyzer to spend less time processing data and more time looking at cycles. Fewer runs are then required to obtain a statistically valid result.

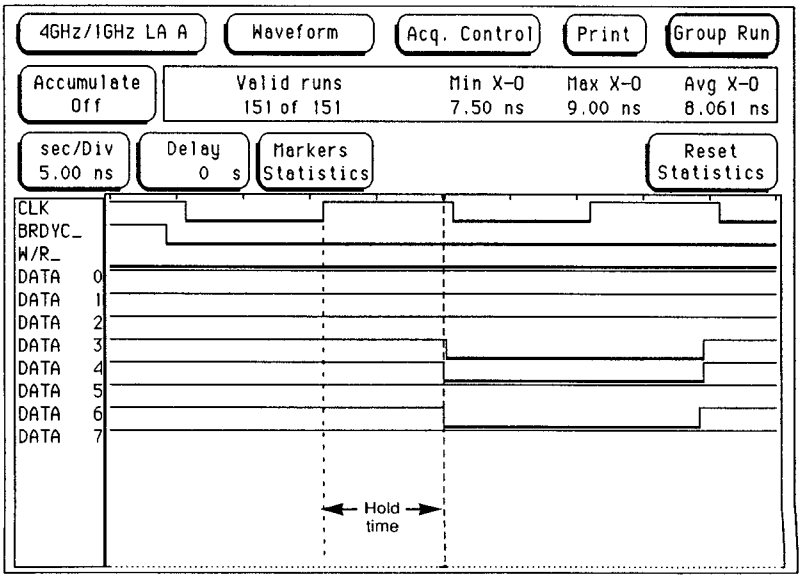


Figure 11 - statistics mode helps to get worst case.

Time interval measurements in manufacturing test

The manufacturing environment sometimes uses timing characterization for final test and process control. Many kinds of assemblies and systems can benefit from thorough testing. Final production test for many systems is merely a functional test, but high performance systems often require timing measurements to insure compliance with the specification. Another use of timing characterization in the production environment is for process control. Rather than simple go/no-go measurements during final test, detailed timing information is recorded and stored in a central database. Analysis of this data can point to process deviations before they adversely affect production yields.

Digital oscilloscopes are often used for timing characterization during production testing. RF switches and multiplexers are used to route signals into the inputs of the oscilloscope, and the test procedure is repeated several times for each configuration. Software to process the waveform and determine switching points is another task that takes time from the already short schedule. High-speed logic analyzers are a better solution than scopes. The large number of inputs available on a logic analyzer eliminate the need for expensive RF switches. The software required for processing timing data from a logic analyzer is simpler, since logic thresholds are set with hardware.

Many modern logic analyzers have the ability to include pattern generator or oscilloscope modules as part of the measurement system. This setup combines all of the benefits of a high-speed logic analyzer with the ability to record analog waveforms and to provide system stimulus. Also, you can dramatically reduce the software development time required for a production test station since a programmer doesn't need to learn the peculiarities of several different instruments. The combination of reduced hardware and software development time results in significant cost savings. The capabilities of a logic analyzer for timing characterization can reduce the time required to build and deploy production test stations, which is often a critical factor in quick turn-around projects.

The Mechanics of the Measurements

The Pentium processor is located on a CPU module mounted perpendicular to the mother board of the target system. The module is positioned 2.5 inches (6.35 cm) from the power supply, limiting access to the processor (see figures 12 and 13). Any probing must maintain a low profile, as well as keeping the load to a minimum. The processor was probed in two ways, simultaneously. An HP E2443B preprocessor was used to provide state traces of the processor. Its buffering significantly reduces its load on the processor The collection of ribbon cables bent away from the floppy disk drive come from the HP E2443B, and it represents 160 separate connec tions. A socket was wired up to provide the signals for the high-speed analyzer. The collection of six pods sitting on top of the power supply come from the HP 16517A high-speed logic analyzer. The analyzer is used in the half-channel mode, so this represents 24 connections. Figure 13 shows a close up of the HP E2443B preprocessor; the board with the 8-position DIP switch; the wired socket, plugged into the preprocessor; and the Pentium processor, located under the miniature fan.

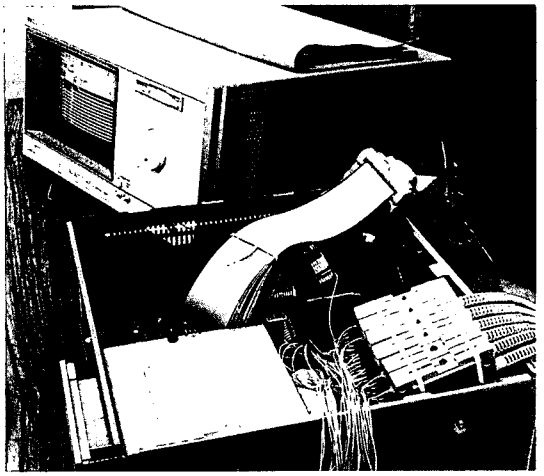


Figure 12 - Pentium target system with HP 16500B logic analysis frame.

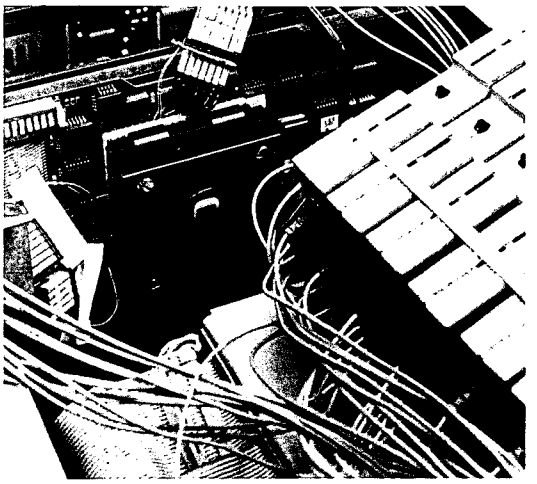


Figure 13 - Pentium processor plugged into wired socket and preprocessor.

Several options were available to provide the connections for the high-speed logic analyzer. Wires could have been soldered to the back of the CPU module. This would have presented the lowest load to the target system. However, for the designer who wants to measure more than one board, this would be impractical. Since the target system seemed to be able to tolerate the extra socket, soldering wires to the socket seemed the best approach. Another possible approach would be to use a bug Katcher from Emulation Technology, Inc. The Pentium Bug Katcher is part number BC2-273-PGA14-PENTIUM and BC2-273-PGA14-PENTIUMZ with a zero insertion socket. This approach would present the highest load to the processor, due to the long lead length, but it is also the easiest to implement. The Pentium Bug Katcher was not available at the time this application note was being developed.

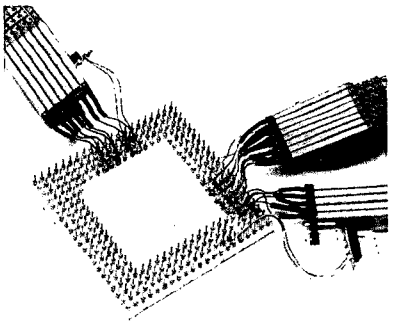


Figure 14 - Wired socket for connecting the Pentium processor to the analyzer.

The wired socket actually worked quite well. It was robust enough to survive a world tour that included 5 different stops. The socket was attached and removed from the processor at each stop. It was relatively nonintrusive to the target, both electrically and mechanically. Electrical loading was kept down by using short lead lengths on the wirewrap wire and a high quality, gold plated socket. The following techniques were used to increase the robustness of the socket. Signals were grouped on 0.1-inch berg strips. This made it easier to keep track of control signals, data and address buses. It also gave the analyzer probes better organization. Shrink tube was used to strain relive the point where the wires connected to the burg strips. This prevented the wires from flexing and breaking from metal fatigue. You may note in figure 14 that the ground wires do not have heat shrink tubing on them. This proved to be a problem and it was added later on. Wires connected to pins on the outer edge of the socket received special treatment. They were wrapped around one of the pins alongside them and then brought off the socket. This prevented the stripped part of the wire from extending over the edge of the socket, where it would eventually break from metal fatigue.

Summary

In this application note, the characterization of new digital systems with a high-speed logic analyzer has been explored. The 500-ps time interval accuracy makes it possible to consider the logic analyzer, rather than the scope, for this purpose. The large channel count allows setup/hold measurements to be made across the wide buses encountered in today's microprocessors. The advanced triggering allows the logic analyzer to find violations much more quickly and easily than possible with a scope. With these advantages, measurements can now be made that were once considered too involved for the time available. As a result, fewer undiscovered timing problems will make it into production, and designers will have greater confidence in there digital designs.

HEWLETT® PACKARD

For more information, call your local HP sales office listed in your telephone directory or an HP regional office listed below for the location of your nearest sales office.

United States:
Hewlett-Packard Company
4 Choke Cherry Road
Rockville, MD 20850
(301) 670-4300

Hewlett-Packard Company
5201 Tollview Drive
Rolling Meadows, IL 60008
(708) 255-9800

Hewlett-Packard Company
1421 S. Manhattan Ave.
Fullerton, CA 92631
(714) 999-6700

Hewlett-Packard Company
2000 South Park Place
Atlanta, GA 30339
(404) 980-7351

Canada:
Hewlett-Packard Ltd.
6877 Goreway Drive
Mississauga, Ontario L4V 1M8
(416) 678-9430

Europe:
Hewlett-Packard
European Marketing Centre
P.O. Box 999
1180 AZ Amstelveen
The Netherlands
(31) 20 547 9999

Japan:
Yokogawa Hewlett-Packard Ltd.
3-29-21 Takaido higashi
Suginami-ku
Tokyo 168, Japan
(813) 3335 8192

Latin America:
Latin American Region Headquarters
Monte Pelvoux No. 111
Lomas de Chapultepec
11000 Mexico, D.F.
(525) 202-0155

Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia (A.C.N. 004 394 763)
(03) 895-2895

Far East:
Hewlett-Packard Asia Ltd.
22/F EIE Tower, Bond Centre
89 Queensway, Central
Hong Kong
(852) 848-7070

5091-8798E
10/93
Printed in U.S.A