

Time Scales and IEEE 1588

Part 1

BY JOHN C. EIDSON, PH.D., AND DAN PLEASANT, AGILENT TECHNOLOGIES

LXI Class B devices¹ introduce several new capabilities to test and measurement systems not easily achievable with current architectures:

- Peer-to-peer messaging.
- A wider variety of trigger models including LAN- and time-based triggers.
- A system-wide time scale based on IEEE 1588.²
- Flexible, application-specific configuration of devices.

The use of system-wide time scales and IEEE 1588 probably are the least familiar to you but key differentiators of Class B devices.

For system integrators trying to correlate measurements from two or more such devices in rack-and-stack environments, only two techniques are available with current architectures. If two devices must share a common time base, provision often is made for a device to provide a 10-MHz output based on its internal oscillator and accept a similar signal from other devices. These signals typically are distributed using BNC cables and enable consistent frequency-based readings to be made across multiple devices.

However, the absolute accuracy of the time base established in this way is no better than the typical 100-ppm accuracy of the primary oscillator.

If multi-instrument measurement systems also require a precise time reference, the only mechanism currently available is hard-wired triggering. Software-based techniques, such as the group execute trigger in IEEE 488-based systems, are not very precise due to limitations in controller protocol stacks and device message parsing.

Timing is even more problematic in data acquisition systems where the number of devices generally is quite large. The system integrator must either implement the time correlations between multiple sensor readings based on the time of request or receipt of the data at the controller or provide a separate time

distribution scheme to permit timestamps to be generated at the source. In the latter case, time distribution protocols such as IRIG-B often are used.³

The LXI standard section 3.2 requires that all Class A and B devices establish a system-wide time scale based on IEEE 1588.

Continued on page 10

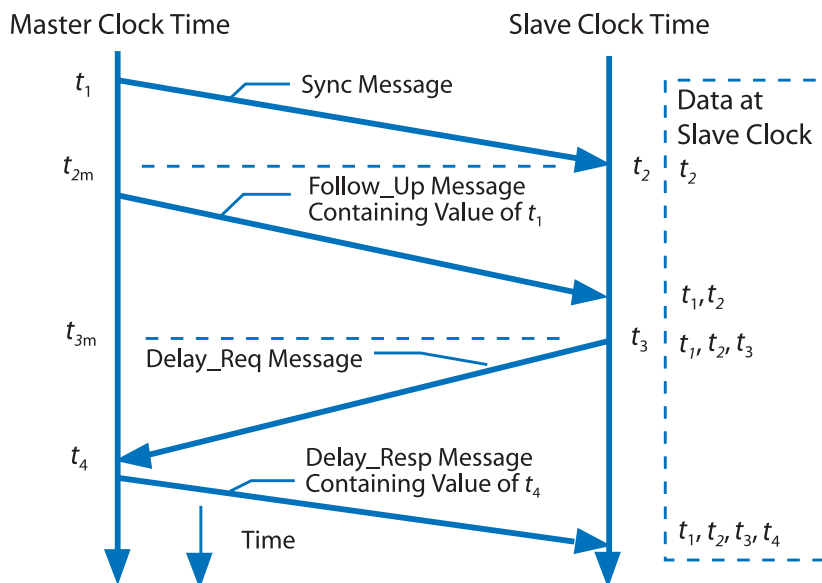


FIGURE 1. IEEE 1588 TIMING MESSAGES

Time has long been recognized as a key variable in test and measurement instrumentation and systems. For many measurement devices, time is one of the measured or controlled variables. Oscilloscopes, logic analyzers, frequency counters, spectrum analyzers, and signal sources all deal directly with time. However, these devices invariably treat time as an internal matter.

The Basics

IEEE 1588 enables the precise synchronization of real-time clocks in devices that are part of a distributed system. This synchronization is achieved by an exchange of timing messages between the clocks over the same network communications media used for other interdevice communications. All LXI implementations of IEEE 1588 use Ethernet as the communications network.

Properly implemented, IEEE 1588 establishes a system-wide self-consistent time scale among the participating device clocks. All clocks will have the same time origin or epoch.

IEEE 1588 implements a master-slave hierarchy of clocks. If the root clock in this hierarchy is synchronized to a standard coordinated universal time (UTC) source such as the GPS, then the IEEE 1588 time scale will be traceable to UTC. As will be seen, this is very easy to do.

Two principal sources of timing errors must be eliminated to provide precise synchronization of clocks. The first is timing errors introduced by instabilities and drift of the local oscillators, and the second is fluctuations in the latency of communications between the clocks. IEEE 1588 addresses the issue of latency fluctuations. Oscillator stability primarily is a component selection issue for implementers.

To see how IEEE 1588 synchronizes clocks, consider two clocks communicating over a communications link with one being the master and one the slave. The master and slave clocks exchange timing messages as illustrated in **FIGURE 1**.

The Sync message is sent from the master to all slaves. As the message is being transmitted, a timestamp, t_1 , is generated by the master as precisely as possible.

In practice, most implementations to date do this by snooping on the media independent interface (MII) within the Ethernet protocol stack as illustrated in **FIGURE 2**. The value of t_1 is communicated to all slaves as a data field in the Follow _ Up messages. When a slave receives the Sync message, it generates a receipt, timestamp t_2 , as shown.

Periodically, the process is reversed with each slave sending a Delay _ Req message to the master. The Delay _ Req messages are timestamped by the slave at transmission, t_3 , and the master upon receipt, t_4 . After the master returns t_4 in a Delay _ Resp message, a slave has all four timestamps.

If you assume that the propagation time from master to slave and from slave to master is the same, then these four timestamps provide sufficient information for the slaves to compute both the mean propagation time and the offset between the two clocks. The slave uses these offsets as the input to a servo that brings the slave clock into alignment with the master clock.

Figure 2 shows block diagrams of a master and a slave clock. To obtain the highest accuracy, an implementation of IEEE 1588 consists of software, running in user space, and specialized hardware consisting of the oscillator, a counter forming the clock, and a packet detector.

The packet detector monitors the MII interface between the PHY and the MAC layers of the protocol stack for both Sync and Delay _ Req messages. When one of these messages is detected, the packet detector takes a snapshot of the 1588 clock to create one of the timestamps.

Since network protocol stacks introduce considerable timing fluctuation, the use of this packet detector eliminates all but the PHY fluctuations, which are quite small. In the slave nodes, the four timestamps are used to adjust the rate and offset of the clock.

The other main source of timing fluctuations in an Ethernet system results from message queuing in switches. IEEE 1588 specifies a special type of switch, the boundary clock, which eliminates the queuing jitter. As illustrated in Figure 2, each boundary clock has a common IEEE 1588 hardware clock, and each port contains a packet detector similar to that in the master or slave clock.

The switch itself is configured to block IEEE 1588 timing messages so that, in effect, the clocks in the end

devices synchronize directly with the clock in the boundary clock. A boundary clock not only eliminates the timing fluctuations that would otherwise be introduced by an ordinary switch, but it also allows clock synchronization to be independent of network traffic.

There are two types of IEEE 1588 clocks: ordinary and boundary. The protocol will automatically configure these into a tree-structured master-slave hierarchy of clocks with the ordinary clocks forming the leaf nodes and the boundary clocks forming the branch points of the tree.

As a result, in Figure 2, the IEEE 1588 clock in the boundary clock is synchronized to the ordinary clock labeled master. Likewise, the

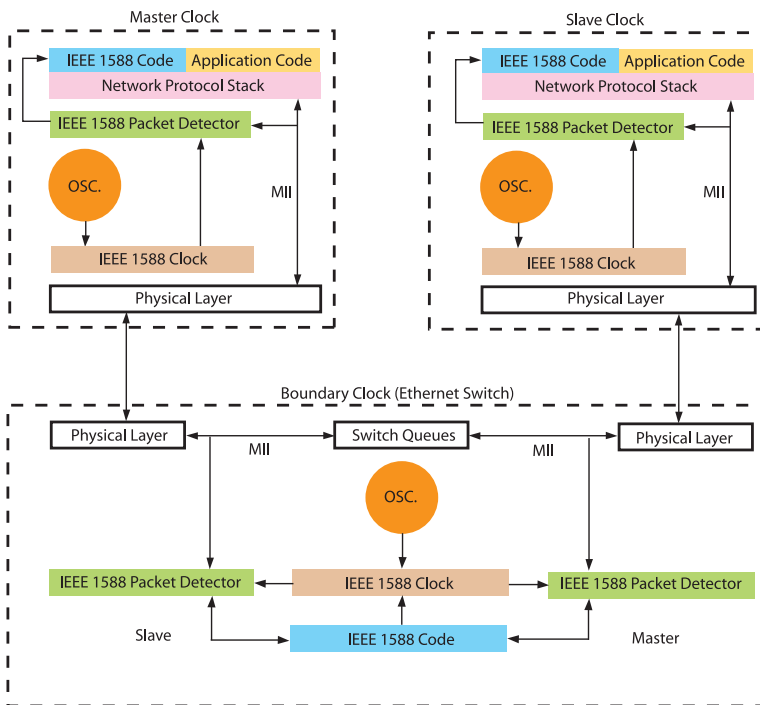


FIGURE 2. IEEE 1588 BLOCK DIAGRAM

ordinary clock labeled slave is synchronized to the IEEE 1588 clock within the boundary clock.

The selection of which clocks are master and which are slave is based on clock characterization information contained in the Sync messages. Each clock executes an independent copy of an algorithm called the best master clock algorithm that compares the characterization information in the Sync messages to internal characterization information.

If the local clock is judged to be inferior to the clock described in the timing message, then the local clock will cease transmitting Sync messages and become a slave. If the local clock is judged to be superior, then it transmits Sync messages. The situation is slightly more complex in a boundary clock, but the result is that after a few Sync messages the master-slave hierarchy is established with the best clock in the system, the grandmaster clock, appearing at the root of the tree.

The best clock is determined by the best master clock algorithm, which is based on the following parameters:

- Preferred Status: Allows a system integrator to optionally designate a collection of devices from which the grandmaster clock will be selected.
- Stratum: Characterizes the source of time. For example, a clock linked to a GPS and UTC is judged better than a free-running clock.
- Identifier: Characterizes the accuracy of the clock. For example, a clock linked to UTC via a GPS is judged more accurate than a clock set to UTC by hand from a user interface.
- Variance: Characterizes the precision of the clock and is a measure of the fluctuations of this clock when compared to a perfect clock.
- Universally Unique Identifier (UUID): Serves as a tiebreaker.

IEEE 1588 currently is being revised, and version 2 is expected to be available in early 2007. New features and modifications in version 2 will greatly ease the implementation of high-accuracy clocks. Other features will enable the construction of fault-tolerant and secure systems that may be required in some critical applications.

Performance

Since the publication of the IEEE 1588 standard in November 2002, there have been numerous independent implementations. Some were pure software implementations with timestamps generated in interrupt service routines while others used the hardware packet detectors. Performance figures for synchronization between a master and a slave for various configurations of an Ethernet network are shown in **TABLE 1**.

Pure software implementations depicted as configuration 1 in Table 1 are less precise due to the increased fluctuations introduced by the lower levels of the Ethernet protocol stack. However, if the required accuracy is in the low microsecond range and network traffic is well controlled, software implementations can be quite useful.

The effect of including a non-boundary clock switch in the synchronization path is evident in configurations 2 and 3. Network loading by ordinary commercial off-the-shelf (COTS) switches can severely degrade performance. However, under the controlled conditions illustrated by configuration 2, quite satisfactory performance can be achieved for many applications.

For synchronization, configurations 4 and 5 are better than ordinary switches due to the absence of queuing. Most test and measurement applications will make use of boundary clocks per configuration 6 to achieve good accuracy independent of network loading.

Configuration 7 indicates that very good accuracy is achievable if proper attention is paid to all sources of fluctuation and bias. High-accuracy implementations will require good oscillators, clock resolution at or below 1 ns, and careful attention to PHY selection and clock design.⁶

Implementation

To date, most implementations of IEEE 1588 have used FPGAs for the packet detectors and the clock. The associated software is executed as an application-level task in the processor of the host device. While this is clearly feasible, supporting silicon will shorten the design cycle and simplify the task of the instrument designer. Likewise, the test and measurement industry will require boundary clocks for the more demanding applications.

Early adoption of IEEE 1588 in the industrial automation industry has resulted in products that can be used by instrument designers and system integrators. For example, Intel® has introduced the IXP465® chip that incorporates hardware support for IEEE 1588.⁷ This network processor has been adopted by Rockwell Automation for use in future control products.

Boundary clocks targeted at industrial automation but useful in test and measurement have been introduced by Hirschmann Automation and Control⁸ and Westermo OnTime⁹. At the PlugFest during the 2004 IEEE 1588 Conference in Zurich, three companies demonstrated IEEE 1588 grandmaster clocks linked to GPS.

Applications

At the present time, there are numerous field trials and prototype application tests under way. In telecommunications, field trials have demonstrated frequency transfer over metropolitan area Ethernet networks using IEEE 1588.¹⁰ It is being used in a new generation of industrial robots to coordinate multiple robot controllers.¹¹

The largest fielded application to date is in the latest line of General Electric controllers for the large turbines used in power plants.¹² It is very much like large data acquisition applications in test and measurement with the added feature of a few very stringent control and timing requirements. In many respects, the new GE controller design represents the same sort of architectural shift represented by LXI Class B instrumentation relative to the GPIB architecture.

The original GE controllers used a centralized architecture based on the VME backplane. Signals from various sensors and actuators were routed to signal-conditioning circuitry on the backplanes, which was, in turn, controlled by computers also resident on the backplanes. Increases in the number of I/O points and bandwidth limitations on the backplanes caused GE to move to a LAN-based, distributed architecture in the new controllers.

In the new design, small local microprocessor-based controllers and signal conditioners interact directly with the sensors and actuators. These local controllers are on 3" × 4" boards and com-

Continued on page 12

municate with supervisory controllers in the main control racks using Ethernet LAN.

In a large installation, there can be up to 3,000 of these sensors distributed over hundreds of meters. Most of the application consists of acquiring sensor data for logging and inputting to local and supervisory controllers. Tight control loops are executed in the local controllers under supervision of the main controllers. It is very important not to lose data but even more critical not to delay reaction to critical events such as a loss of grid load.

To meet these communications and timing constraints, GE uses the simplest Ethernet switches to eliminate overhead and queuing introduced by the higher-level network switching protocols such as quality of service (QOS), spanning tree, and others normally found in Ethernet environments. Each of the local controller boards and the supervisory controllers executes IEEE 1588 to allow precise timestamps to be assigned at the source to all sensor data. The observed timing accuracy is on the order of a few hundred nanoseconds, a value consistent with the data associated with configuration 2 of Table 1.

A comparison with configuration 3 in the table indicates that this accuracy only can be achieved by eliminating the queuing jitter in the switches. This was done by using the IEEE 1588 time base to enforce admission control to the network at each local controller. The admission control mechanism is a time-slotted protocol in which each local controller is assigned a periodic time slot during which it is permitted to use the network, eliminating queuing in the switches.

Time-slotted protocols are almost always used in safety-critical systems to provide predictable execution. This same mechanism

Row	Measurement Condition	LSB (ns)	Peak-to-Peak Jitter (ns)	Standard Deviation (ns)	Reference
1	Via a boundary clock (end devices were software only)	1,000	10,000	1,000	4
2	Via a COTS Switch (low load)	10	375	49	5
3	Via a COTS Switch (10% random + burst load)	10	2,500,000	360,000	5
4	Via a Repeater (low load)	10	214	31	5
5	Via a Repeater (high load)	10	260	31	5
6	Via a Boundary Clock (independent of load)	10	120	15	5
7	Direct Clock-Clock (high-accuracy hardware)	1	4	0.7	6

TABLE 1. CLOCK SYNCHRONIZATION JITTER

EXCEPT FOR ROW 1, ALL CLOCKS USE HARDWARE PACKET DETECTORS.

guarantees that time is available to communicate critical control information during normal operation and abnormal events such as a loss of load which requires a response within a known and bounded time.

The time-slotted protocol also permits the use of user datagram protocol (UDP) rather than the more resource and bandwidth intensive transmission control protocol (TCP) by eliminating queue

overflow as a cause of packet loss. In fact, GE found that in this application UDP actually is more reliable than TCP.

Since this is a critical control application, the most recent data is more valuable than a missed data point in the past, which makes the preservation of the time-slotted protocol critical. The time required to reestablish a TCP connection in the face of failure is inconsistent with maintaining the protocol timing.

More detail on the General Electric and other applications including a more in-depth discussion of IEEE 1588 can be found in reference 13.

Conclusions

The LXI specification mandates the use of IEEE 1588 in Class B devices to produce a system-wide time scale. IEEE 1588 is easily implemented in typical test and measurement devices and components, and infrastructure devices are now appearing on the market that support system-wide implementation of the IEEE 1588 protocol.

Part 2

Applications in industrial automation and power have proved the worth of this technology and provide important lessons for the test and measurement community. Part 2 of this article will explore the use of IEEE 1588, system-wide time scales, and other Class B features in test and measurement applications. It will appear in the October 2006 issue of LXI ConneXion.

References

1. LXI Specification, Revision 1.0, Sept. 23, 2005.
2. IEEE 1588-2002, *Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, Nov. 8, 2002.
3. Inter-Range Instrumentation Group, <http://www.jcte.jcs.mil/sitemap.htm>
4. Correll, K. et. al, "Designing Software-Only IEEE 1588 Systems," *Proceedings of the 2005 IEEE 1588 Conference*.
5. Mohl, D., "Experiences With IEEE 1588 in Higher Cascaded Ethernet Networks," *Proceedings of the 2005 IEEE 1588 Conference*.
6. Vook, D. et al, "Update on High Precision Time Synchronization," *Proceedings of the 2005 IEEE 1588 Conference*.
7. Sharma, P., "Hardware Assisted IEEE 1588 Implementation in a Next Generation Intel Network Processor," *Proceedings of the 2004 IEEE 1588 Conference*.
8. Mohl, D., "Implementation Results of an IEEE 1588 Boundary Clock," *Proceedings of the 2004 IEEE 1588 Conference*.
9. Holmeide, O., "Boundary Clock Implementation," *Proceedings of the 2003 Workshop on IEEE 1588*.
10. Tonks, D., "IEEE 1588: PAR Work and Field Trial Results With

Reference to G.pactiming,” Meeting of the ITU, Study Group 15, Working Party 3, Question 13/15, May 2005.

11. Gerstenberger, M. et. al, “Application of IEEE 1588 to Synchronize Multiple Robot Controllers,” *Proceedings of the 2005 IEEE 1588 Conference*.

12. Shepard, M., “Implementation of IEEE 1588 in a Networked I/O Node,” *Proceedings of the 2003 Workshop on IEEE 1588*.

13. Eidson, J.C., *Measurement, Control, and Communication Using IEEE 1588*, 2006.

ABOUT THE AUTHORS

John C. Eidson, Ph.D., received a B.S. and an M.S. from Michigan State University and a Ph.D. from Stanford University, all in electrical engineering. He held a postdoctoral position at Stanford for two years, spent six years with the Central Research Laboratory of Varian Associates, and joined the Central Research Laboratories of Hewlett-Packard in 1972. When HP split in 1999, he transferred to the Central Research Laboratory of Agilent Technologies. Dr. Eidson was heavily involved in IEEE



1451.2 and IEEE 1451.1 and is the chairperson of the IEEE 1588 standards committee and a life fellow of the IEEE. e-mail: john_eidson@agilent.com



Dan Pleasant holds a B.S.E.E. from the University of Colorado and an M.S.E.E. from Stanford University. After a seven-year stint at MIT Lincoln Laboratory, Mr. Pleasant joined the CAE group (now known as Agilent EEsof) at HP in 1989. He has been a member of the DoD ATML Working Group and the LXI Consortium where he is the co-chair of the Timing and Synchronization Subcommittee. e-mail: dan_pleasant@agilent.com

Agilent Technologies, 395 Page Mill Rd., Palo Alto, CA 94306, 800-829-4444.

FOR MORE INFORMATION

on literature on products, services, and pointers to IEEE 1588 enter this rsleads URL www.rsleads.com/615ee-176