Coordination of time-of-day clocks among multiple systems

by Noshir R. Dhondy
Richard J. Schmalz
Ronald M. Smith, Sr.
Julian Thomas
Phil Yeh

The IBM Enterprise Systems Architecture/390™ External-Time-Reference (ETR) architecture facilitates the synchronization of time-of-day (TOD) clocks to ensure consistent time-stamp data in an installation with multiple systems. The ETR architecture also provides a means by which the TOD clocks can be set automatically, without human intervention, to an accurate standard time source. This paper reviews the design considerations involved in providing these functions—along with "clock integrity" and continuous operation—as a consistent extension of the System/370™ TOD-clock architecture. The paper also provides a functional description of the IBM 9037 Sysplex Timer™, which is an implementation of the sending unit of the ETR network.

Introduction and background

There is a long-standing requirement for accurate time and date information in data processing. As single systems have been replaced by multiple, coupled systems, this need has evolved into a requirement for both accurate and consistent clocks among the systems. (Clocks are said to be consistent when the difference or offset between them is sufficiently small. An accurate clock is one that is consistent with a standard time source.) This paper summarizes these requirements, discusses alternative approaches to the coordination of time across multiple systems, and describes the External-Time-Reference (ETR) architecture and some implementation details.

In the context of the IBM Enterprise Systems Architecture/390™ (ESA/390™) architecture, each "system" is called a central processing complex (CPC); it consists of one or more central processing units (CPUs) and associated hardware units (such as main and expanded storage, TOD clocks, and channels) that can be configured to operate under the control of a single operating system. A configuration of coupled CPCs that are cooperating to process a common workload is called a sysplex.

- *Time concepts*

Historically, the most important requirement for highly accurate time was for navigational purposes. For applications such as very precise navigation and satellite tracking, which must be referenced to the earth's rotation, a time scale that is consistent with the earth's rotation must be used. Today, this time scale is known as Universal Time 1 (UT1) [1, 2]. UT1 does not advance at a

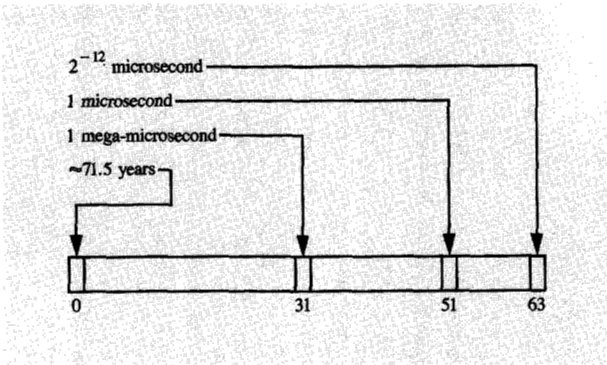

Figure 1

Stepping rate for selected TOD-clock bit positions.

fixed rate, but speeds up and slows down with the earth's rotation rate. UT1 is computed using astronomical data from observatories around the world in order to correct for variations in the rotational axis ("wobble") of the earth, and UT1 is consistent with civil, or solar, time.

Until 1967, the second was defined on the basis of UT1. Since 1967, the internationally accepted definition of the second has been "9 192 631 770 periods of the radiation corresponding to the transition between two hyperfine levels of the ground state of the cesium-133 atom" [2]. In 1967, this definition was 1000 times more accurate than that achievable by astronomical methods. The atomic definition of the second is primarily intended to provide an accurate measure of time intervals. When this was instituted, however, the need for an accurate time-of-day measure was also recognized.

This led to the adoption of two basic scales of time:

- International Atomic Time (TAI), based solely on an atomic reference, provides an accurate time scale that is increasing at a constant rate with no discontinuities.
- Coordinated Universal Time (UTC) is derived from TAI and is adjusted to keep reasonably close to UT1. UTC is the official replacement for (and is generally equivalent to) the better-known "Greenwich Mean Time" (GMT).

Since January 1, 1972, occasional corrections of exactly one second—called "leap" seconds—have been inserted into the UTC time scale, whenever needed, to keep UTC time within ±0.9 second of UT1 at all times. These leap seconds, which have always been positive (in theory, they can be positive or negative) are coordinated under international agreement by the International Time Bureau (BIH) in Paris. This adjustment occurs at the end of a UTC month, normally on June 30 or December 31. The last minute of a corrected month can, therefore, have either a positive adjustment to 61 seconds or a reduction to 59 seconds. As of July 1, 1992, 17 positive leap seconds had been introduced into UTC [3].

Note that the effect of a leap second is the introduction of an irregularity into the UTC time scale, so exact interval measurements may be made with UTC only if the leap seconds are included in the calculations. After every positive leap second, the difference between TAI and UTC increases by one second.

- *TOD clock*

The TOD clock was introduced as part of the System/370™ architecture [4] to provide a high-resolution measure of real time, suitable for the indication of date and time of day. It is a 64-bit unsigned binary counter with a period of approximately 143 years. The value of the TOD clock is directly available to applications programs by use of the STORE CLOCK (STCK) instruction, which stores the value of the clock into a storage location specified by the instruction.

Conceptually, the TOD clock is incremented so that 1 is added into the low-order bit position (bit 63) every 2^{-12} microsecond (1/4096 microsecond). Actual TOD-clock implementations may not provide a full 64-bit counter, but maintain an equivalent stepping rate by incrementing a higher-order bit position at a correspondingly lower rate. **Figure 1** shows the stepping rate (rate at which the bit positions change) for selected TOD-clock bit positions. The architecture requires that the TOD-clock resolution be sufficient to ensure that every value stored by a STCK instruction is unique, and that consecutive STCK instructions always produce increasing values.

In System/370 architecture, when more than one TOD clock exists within a shared-storage multiprocessor (a single CPC), the stepping rates are synchronized, so that all TOD clocks are incremented at exactly the same rate, and the architectural requirement for unique and increasing TOD-clock values still applies. In the case in which simultaneous STCK instructions are issued on different CPUs, uniqueness may be ensured by inserting CPU-specific values in bit positions to the right of the incrementing position.

A carry out of bit 32 of the TOD clock occurs every 2^{20} microseconds (1.048576 seconds). This interval is sometimes called a "mega-microsecond" ($M\mu s$). This carry signal is used to start one clock in synchronism with another, as part of the process of setting the clocks. The carry signals from two or more clocks may be checked, to ensure that all clocks agree to within a specified tolerance.

The use of a binary counter, such as the TOD clock, for time of day requires the specification of a time origin, or epoch; that is, the time at which the TOD-clock value would have been all zeros. The System/370 architecture

established the epoch for the TOD clock as January 1, 1900, 0 a.m. GMT.

- *TPF time synchronization*

The IBM Transaction Processing Facility (TPF) [5] is a specialized control program for multiple CPCs coupled by means of shared disk storage that cooperate to process transactions for a shared database. The design of this program requires that the TOD clocks of the multiple CPCs be consistent—to ensure the integrity of transaction data (i.e., time stamps accurately reflect the sequence of events). The remote TOD-clock synchronization facility[1], available on most large IBM CPCs (3033, 308X, 3090™), provides this capability by substituting a single 1.0-MHz TOD-clock-stepping signal (from a designated "master" CPC) for the individual TOD-clock-stepping signal oscillators in each CPC. This eliminates variations caused by differences in TOD-clock-stepping rates. A "sync" signal every Mμs from the "master" CPC enables starting one TOD clock in each CPC in synchronism with the master system, as well as continuously checking that this synchronism is being maintained. The connection topology of this facility is complex and limits the number of CPCs that can be connected.

- *Time coordination in network-coupled systems*

Today, many conglomerations of systems[2] linked by networks exist, most using Internet Protocol (IP) [6]. Software techniques, such as Network Time Protocol (NTP) [7], have evolved for time distribution and consistency among the individual systems.

The individual systems of a conglomeration observe a common transmission protocol (e.g., IP). They are generally workstations, widely dispersed, under multiple administrative controls, produced by different manufacturers, and with different architectures and operating systems. Usually the interactions between systems are only casual in nature.

Differences in time between the systems are often measured only to the second; this is consistent with typical network data-transmission times (as seen by applications). Time in this environment is usually expressed in terms of civil time, so the leap-second discontinuities introduced in UTC are rarely a significant factor (the NTP time scale is effectively redefined at each leap second).

ESA/390 time-coordination requirements

A number of requirements had to be considered in the design of a time distribution and coordination facility for ESA/390 systems:

- The facility had to be a compatible and evolutionary extension of the System/370 TOD architecture. It had to maintain the TOD format and epoch, and the facility had to be accessible by using the STCK instruction.
- Time values had to be accurate in relation to standard, or civil, time. There could be no dependency on a human operator to enter time and date information at every CPC initialization. Following are some examples of the operational problems and other consequences of erroneous CPC date and time settings:
  - Passwords expire prematurely, so that terminal users cannot log on, or a system security facility blocks access to data.
  - Retention dates pass, causing tape or disk files to be scratched.
  - System programs discard "old" system management data.
  - Jobs are erroneously started (or missed) by automatic job-starting routines.
  - Hours are spent by individuals in determining how to "back out" or reprocess transactions properly.
  - A user looks at his watch and requests an action at a time a few minutes in the future. Because the system time has already passed that time, the request is discarded.
- Time consistency had to be maintained among CPCs. The justification for consistency between TOD clocks in coupled CPCs can be illustrated by the following scenario:
  1. CPC A executes a STCK instruction (time stamp x), which places the clock contents in storage.
  2. CPC A then signals CPC B.
  3. On receipt of the signal, CPC B executes STCK (time stamp y).

  For time stamps x and y to reflect the fact that y is later than x, the two TOD clocks must agree within the time required to send the signal. The consistency required is limited by the time required for signaling between the coupled CPCs and the time required by the STCK instruction itself.

  Consider a transaction-processing system in which the recovery process reconstructs the transaction data from log files. If time stamps are used for transaction-data logging, and the time stamps of two related transactions are transposed from the actual sequence, the reconstruction of the transaction database may not match the state that existed before the recovery process. This is just one example of the problems associated with processes that appear (as observed at one CPC) to end before they have started (as observed at another CPC).

[1] Available through the IBM "Request for Price Quotation" (RPQ) process used for ordering special functions.
[2] In this section, we use "system" rather than "CPC," since this is the term most frequently used in the context of such conglomerations.

Table 1 Relationship of UTC and ETR time scales.

UTC date (yyyy.mm.dd)	*UTC time* (hh:mm:ss)	*ETR time since epoch* (s)	*ETR date* (yyyy.mm.dd)	*ETR time* (hh:mm:ss)
...	...	...	...	...
1971.12.31	23:59:59	2,272,060,799	1971.12.31	23:59:59
1972.01.01	00:00:00	2,272,060,800	1972.01.01	00:00:00
1972.01.01	00:00:01	2,272,060,801	1972.01.01	00:00:01
...	...	...	...	...
1972.06.03	23:59:59	2,287,785,599	1972.06.30	23:59:59
1972.06.30	23:59:60*	2,287,785,600	1972.07.01	00:00:00
1972.07.01	00:00:00	2,287,785,601	1972.07.01	00:00:01
...	...	...	...	...
1972.12.31	23:59:59	2,303,683,200	1973.01.01	00:00:00
1972.12.31	23:59:60*	2,303,683,201	1973.01.01	00:00:01
1973.01.01	00:00:00	2,303,683,202	1973.01.01	00:00:02
...	...	...	...	...
1992.06.30	23:59:59	2,918,937,615	1992.07.01	00:00:15
1992.06.30	23:59:60*	2,918,937,616	1992.07.01	00:00:16
1992.07.01	00:00:00	2,918,937,617	1992.07.01	00:00:17
...	...	...	...	...

*= A positive leap second. (Not all leap seconds are shown.)

Note that when two clocks are both reasonably accurate with reference to a common standard, they will also be reasonably consistent.

- Continuous availability of the time facility was required. If coupled CPCs are cooperating to provide continuous availability of an application function (not limited by the availability characteristics of an individual CPC), the time facility must have availability characteristics that exceed those of the individual CPCs. This implies that the time reference could not be integrated into any one CPC, since it would become unavailable when that CPC was being maintained or upgraded or during reconfiguration operations.
- Clock integrity had to be maintained. That is, any failure that might cause a lack of clock consistency had to be made known to programs that depended on that consistency.
- The system environment had to permit multiple CPC model types (and not be limited to like models).
- Time-distribution distances had to be as great as I/O-connection distances (which are no longer limited to a single machine room). In addition, there had to be some provision for coordination of time between different locations.
- Operating system support of the time reference facility could not require constant inter-CPC coordination and could not introduce significant overhead.
- The time facility had to be adequate for the needs of the future as well as the present. In particular, the allowable offset between two clocks in different CPCs is limited by the minimum inter-CPC signaling time, which can be expected to diminish for future generations of systems.

ETR architecture and implementation

The ETR architecture provides a means of synchronizing TOD clocks in different CPCs with a centralized time reference, which in turn may be set accurately on the basis of an international time standard. The architecture defines a time-signal protocol and a distribution network, called the ETR network, that permit accurate setting and maintenance of consistency of TOD clocks. This section presents major architectural features, including the aspects of fault tolerance and clock integrity. The rationale of some design decisions is discussed, and certain hardware elements are described.

- *ETR time*

In defining an architecture to meet ESA/390 time-coordination requirements, it was necessary to introduce a new kind of time, called ETR time, reflecting the evolution of international time standards, yet remaining consistent with the original TOD definition. Until the advent of the ETR architecture, the TOD-clock value had been entered manually, and the occurrence of leap seconds had been essentially ignored. Introduction of the ETR architecture has provided a means whereby TOD clocks can be set and stepped very accurately, on the basis of an external UTC time source, so the existence of leap seconds cannot be ignored.

Several requirements influenced the definition of ETR time:

- Since the TOD clock is directly available to application programs, the definition of ETR time must be consistent with the current definition of the TOD clock.

- The TOD-clock format was designed to be suitable for performing arithmetic; that is, subtracting two TOD-clock values must provide an accurate measure of time interval. To meet this requirement, ETR time must be strictly monotonic, with no discontinuities. It is defined in terms of atomic seconds.
- The TOD-clock value is defined to represent the number of atomic seconds since the epoch (originally defined in terms of GMT). To avoid the necessity of determining the number of atomic seconds that have occurred between 0 a.m. January 1, 1900 GMT and 0 a.m. January 1, 1972 UTC (when the process of adding leap seconds began), the TOD epoch was redefined as January 1, 1900, 0 a.m. ETR time. ETR time was defined to be equal to UTC on January 1, 1972, 0 a.m.

ETR time can be computed from UTC time by adding the number of accumulated leap seconds between 1972 and the time that is to be converted:

ETR time = UTC + leap seconds.

Table 1 illustrates the relationship between the UTC and ETR time scales. See [3] for the full schedule of leap-second insertion.

- *ETR signals*

The ETR architecture defines three signals for clock synchronization, which are sent to each attached CPC, in a single serial bit stream:

- *ETR oscillator signal.* This signal is used by each CPC as a common TOD-clock-stepping signal for all TOD clocks in the CPC. It ensures that all clocks step at the same rate, so that once they are set to ETR time, they will stay consistent with ETR time.
- *ETR on-time signal.* This signal is used by the clock-setting process as the reference time instant. An ETR on-time signal occurs every Mμs and corresponds to the carry from bit 32 of a TOD clock.
- *ETR data signal.* The ETR data include the ETR-time value, local-time-zone and leap-second offset information, and link-connection status. For fault tolerance, the same ETR data are transmitted several times. Time-offset information is automatically made available to the system control programs in the CPCs for use in time-conversion algorithms. It does not affect the TOD-clock-setting value.

- *ETR network*

An ETR network consists of the following three types of elements configured in a network with star topology: ETR sending unit, ETR link, and ETR receiving unit. The ETR sending unit is the centralized external time reference, which transmits ETR signals over dedicated ETR links. It provides a means by which ETR time can be accurately maintained with respect to external standard time services. The ETR receiving unit in each CPC, which receives the ETR signals, includes the means by which the TOD clocks are set and maintained consistent with ETR time. **Figure 2** shows a typical ETR network, which connects the ETR sending unit to CPCs in an installation. The ETR network may comprise one or more sysplexes and CPCs not belonging to a sysplex.

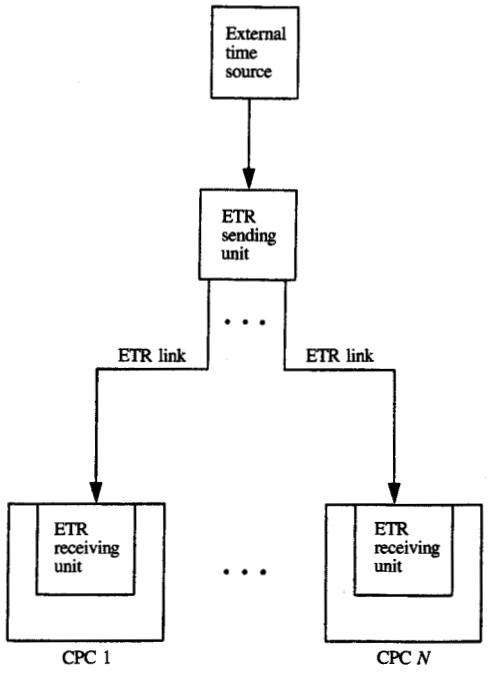

Figure 2

Typical ETR network.

A fault-tolerant configuration can be provided by coupling and synchronizing two ETR sending units with each other, so that they transmit consistent ETR timing information. **Figure 3** shows a typical fault-tolerant ETR network. The ETR receiving unit at each CPC has two ports; each port is normally connected to a different ETR sending unit of a coupled pair in the same network. This fully duplicated structure minimizes the probability that a single failure can adversely affect ETR network capability.

It is likely that a large installation may have more than one ETR network, in which case it is important that all CPCs within a sysplex be attached to the same ETR network. The ETR data include network ID information, which is verified by the system control programs running

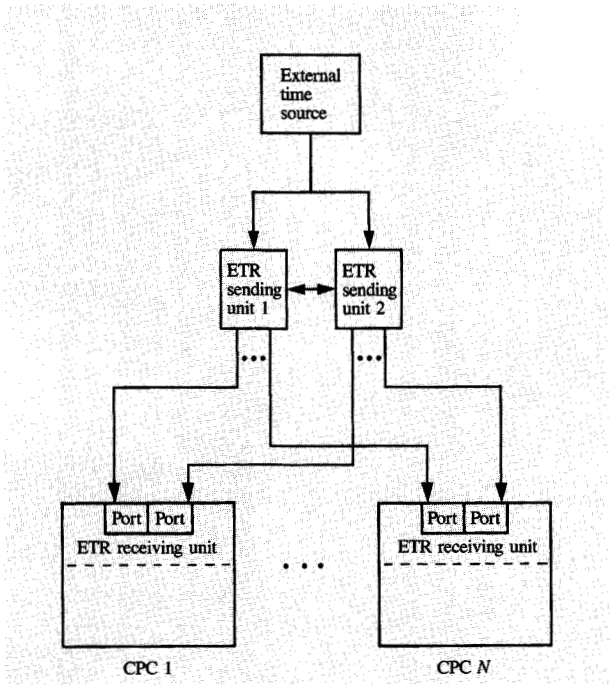

Figure 3

Fault-tolerant ETR network.

in the attached CPCs to ensure true consistency of all TOD clocks within a sysplex.

ETR sending unit

The IBM 9037 Sysplex Timer™ unit [8], shown in **Figure 4**, is an implementation of the ETR sending unit. It can transmit to up to 16 ETR receiving units attached to its ports by means of fiber optic cables, which serve as the ETR links. The 9037 unit transmits the ETR signals, described previously, and each ETR receiving unit echoes back the received signals, which allows the 9037 unit to check the condition of each link and compensate for the propagation delay through the fiber optic cables.

A console (an IBM PS/2® computer or equivalent) communicates with the 9037 and is used to enter and display initialization data, such as time and offsets, and to collect error information from the 9037 units. The time for the 9037 unit can be set from an external time source (such as a low-frequency time-code radio receiver) attached to the console. This allows the 9037 unit to keep its time in close agreement with available time services. [Reference [8] provides more details on external time source configurations supported by the 9037 unit, and Reference [9] discusses available National Institute of Standards and Technology (NIST) time services in the United States.] If no external time source is available, the time for the 9037 unit can be entered manually as a part of the 9037 installation process.

The following time offsets must be entered at the 9037 console during installation:

- Leap-second offset (the number of accumulated leap seconds since January 1, 1972, 0 a.m.).
- Time zone (the difference between local standard time and UTC. For example, UTC − 5 hours = Eastern Standard Time).
- Daylight Savings Time (if in effect).

These offsets are first used to calculate the correct time value (ETR time) to be set in the 9037 TOD clock. The 9037 unit then transmits its TOD-clock value and the offsets, as separate entities, to the attached CPCs.

Subsequent changes to either the leap-second offset (when the date of a leap-second insertion becomes known [3]) or local-time-zone offset (most probably because of a Daylight Savings Time adjustment) can be scheduled in advance at the 9037 console. At the scheduled time, the new offset information is sent to the CPCs. Note that the ETR time and therefore the TOD-clock value do not need any corrections. As mentioned above, the time-offset information may be used by the system control program to convert ETR time to civil time.

Fault-tolerant configuration

As mentioned above, a fault-tolerant ETR network configuration can be provided by coupling and synchronizing two ETR sending units with each other, so that they transmit consistent ETR timing information. This is accomplished by installing the Expanded Availability feature in each 9037 unit. The resulting configuration, shown in **Figure 5**, is called an Expanded Availability configuration.

The 9037 unit uses a quad clock design in order to meet the requirements of synchronization and fault tolerance. A detailed description of the quad clock design of the 9037 unit can be found in [10]. The clocks are fully connected to one another, as illustrated in Figure 5. The packaging is essentially dual; each 9037 unit contains two elements of the quad clock packaged together. Phase locking of voltage-controlled crystal oscillators (VCXO) in each clock source is used to achieve synchronism. The connections between 9037 units are duplicated in order to provide redundancy, and critical information is exchanged between the two 9037 units every Mμs, so that if one of the 9037 units fails, the other 9037 unit will continue transmitting to the attached CPCs.

The following error-handling rules implemented within each 9037 unit ensure single-point fault tolerance:

- If any internal 9037 failure that can potentially affect the integrity of the ETR timing information is detected,

transmission of ETR signals from all ports of the failing 9037 is terminated. The other 9037 unit continues to transmit.

- If any failure on one of the links between the two 9037 units is detected, transmission of ETR signals from all ports of one of the 9037 units is terminated (an internal algorithm determines which 9037 unit stops transmission). The other 9037 unit continues to transmit.
- If any internal 9037 failure is detected that affects only a specific port or ports but does not affect ETR timing integrity, transmission of ETR signals from the failing ports is terminated.

The ETR receiving units are responsible for properly processing ETR signals, for detecting ETR-signal errors, and for switching from a failed ETR signal to a good ETR signal.

An Expanded Availability configuration is fault-tolerant to a power outage affecting only one 9037 unit. However, a power outage that affects both 9037 units terminates the transmission of ETR signals to all attached CPCs. An internal battery-powered clock module in each 9037 unit automatically maintains critical configuration information and continues to update the date and the time within each 9037 unit. When power is restored to either 9037 unit or to both of them, the operating conditions that existed before the power outage are restored, and data transmission, using the updated time, is resumed to the attached CPCs. There is no need to reenter initialization data at the 9037 console. If errors are detected during the restoration process, the affected 9037 unit is not allowed to go on-line (transmit to the attached CPCs). Note that the 9037 console is not required in order to resume normal transmission to the CPCs.

Tracking of precision time sources

As mentioned before, both time consistency and time accuracy in relation to standard, or civil, time outside the CPC are required to meet ESA/390 time coordination requirements. The 9037 Expanded Availability feature provides the capability of tracking the 9037 time to a precision time source (i.e., maintaining consistency). This is referred to as steering. In an Expanded Availability configuration, the ETR timing signals from each 9037 unit are derived from the VCXO time bases, which are part of the quad clock design. VCXOs are primarily designed to provide a wide pull range (the range over which the output frequency may be adjusted by a control voltage signal) and provide only modest accuracy, in the range of several tens per million. Therefore, it is necessary to steer the VCXO time bases to a reference time base.

Each 9037 unit has a fixed-frequency temperature-compensated crystal oscillator (TCXO), with a nominal accuracy of one part per million, equivalent to 32 seconds

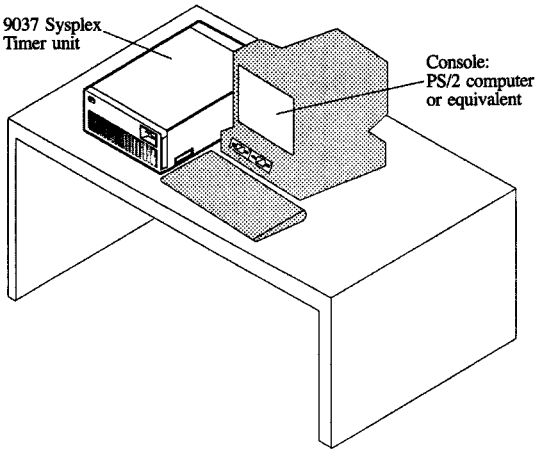

Figure 4

IBM 9037 Sysplex Timer unit.

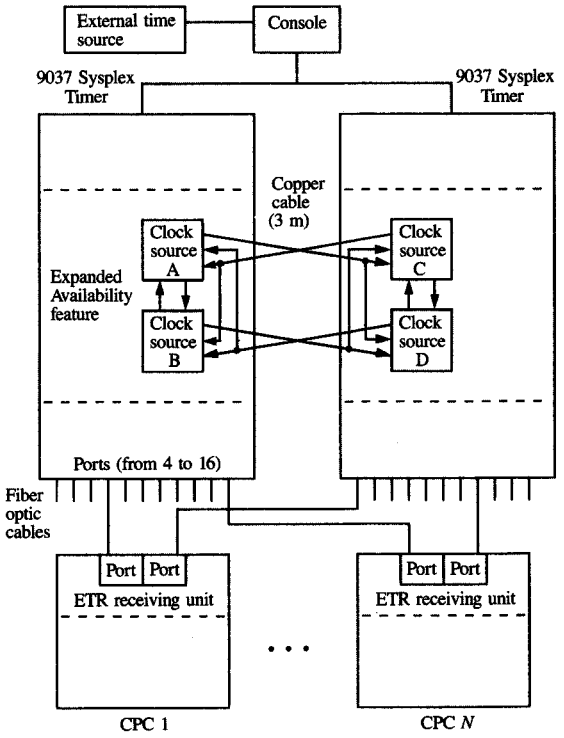

Figure 5

Expanded Availability configuration.

per year. (Accuracy may vary with the effects of component aging and the environment.) It is used to step a TOD counter, called the Reference TOD-clock. This Reference TOD-clock and its associated TCXO provide the reference time source required for steering. An internal algorithm and the coupling protocol between the two 9037 units select one of the two 9037 reference time bases for steering and maintain consistency of Reference TOD-clock values in both 9037 units. This allows uninterrupted steering operation, in the event of a failure of either 9037 unit. Every Mμs, the Reference TOD-clock value is compared to the TOD-clock value of the counters stepped by the VCXOs. A steering factor is calculated and is then applied simultaneously, as either a positive or negative bias for all the VCXOs in the quad clock. This results in the stepping rate of the oscillator signal transmitted to the CPCs being speeded up or slowed down without any accompanying discontinuity in the TOD-clock value transmitted to the CPCs.

When the external time source function is used, 9037 accuracy can be maintained to within ±0.005 second of a stable external time source. The 9037 console queries the external time source periodically for new time values and updates the Reference TOD-clock, still being stepped by the TCXO. The steering mechanism, described above, allows the 9037 unit to maintain consistent time with an external time source. If the external time source function is not used, the 9037 accuracy is based on the TCXO accuracy (typically within ±32 seconds per year from the initial time setting).

During normal 9037 operation, a time adjustment up to ±4.999 seconds can be made. The time adjustment is either entered manually at the 9037 console, or it may be the result of a periodic automatic comparison with an external time source, as is described above. In either case, the Reference TOD-clock is updated with the new value, and a steering factor is applied to the VCXOs every Mμs until the desired time adjustment is reached.

ETR links

Fiber optic cables are used to connect the ETR sending unit ports to the ports of the ETR receiving units. Cable lengths may be up to 3 km (1.87 miles) for 62.5/125-μm fiber and 2 km (1.24 miles) for 50/125-μm fiber. The fiber optic components used are the same as those used in other ESCON™ products [11], to avoid a need for a special ETR cable type.

The 9037 unit automatically compensates for propagation delay through the fiber optic cables. This virtually eliminates cable length as a source of TOD-clock inconsistency between CPCs that are different distances from the 9037 unit. For this compensation to be effective, the length difference between the two fibers constituting a fiber optic duplex cable should not exceed 10 meters. (The two fibers are used to send data in opposite directions.)

ETR receiving unit

The following describes the major features of an ETR receiving unit, with particular emphasis on availability and clock integrity. Program actions required for certain events are also discussed.

Port selection

To avoid a single point of failure, each ETR receiving unit contains two ports, but only one port is active at any given time. The control program running on the CPC designates the active port. The ETR oscillator signal received at the active port is used by the CPC as the common clock-stepping signal. When the active port becomes nonoperational, the hardware automatically selects the other port as the new active port, and there is no loss of synchronization. If neither port is operational, the TOD clocks of the CPC are stepped by a local oscillator in the CPC, and the clocks lose consistency with ETR time.

Port-connection verification must be performed at system-initialization time and whenever a port is reconnected. To verify a port connection, the control program reads data from that port. If the port to be verified is found to be connected to an ETR network different from that connected to the active port or that connected to the rest of the sysplex, it is unsafe to step the clocks with the signal received from the port being verified. The control program can disable a port to prevent it from being eligible for automatic switchover; this is necessary when a port is scheduled for service or when port verification fails. Port verification requires that the control program be able to read the ETR data from both ports while stepping the clocks with the signal from the active port. For simplicity, the design provides only one ETR-data buffer shared between two ports, so the CPC can read data from either port, under program control.

Synchronizing TOD clocks

Clock synchronization to ETR time may be performed at system-initialization time or during certain recovery actions. The control program first reads the ETR-time value (part of the ETR data signal) from the active port. This value corresponds to the time at the next ETR on-time signal. The clocks are set to this value and are placed in the stopped state. Upon the occurrence of the next ETR on-time signal, the clocks begin to be incremented, stepped by the common clock-stepping signal (the ETR oscillator signal received from the active port).

Time-stamp register

On some ESA/390 models, there is a 64-bit time-stamp register for each TOD clock in the CPC. Whenever an

ETR on-time signal is received at the active port, the contents of all TOD clocks are stored into their associated time-stamp registers. The contents of a time-stamp register can be used to determine the time difference between the associated TOD clock and the ETR time. This difference is needed for some recovery actions. This register also facilitates exception handling and clock setting in the Processor Resource/Systems Manager™ (PR/SM™) mode, as explained later. (PR/SM is a hardware feature that allows the resources of a CPC to be dynamically shared among multiple, independent partitions. Each partition can run a system control program, and all partitions can run simultaneously [12].)

Exception monitoring and handling

New categories of processor interruption conditions are provided by the ETR architecture to report the following events and exceptions:

- ETR sync check, which indicates that the TOD clocks in the CPC and the ETR time are not in synchronism. The normal recovery action performed by the system control program is to resynchronize all TOD clocks to the ETR time.
- Switch to local, which indicates that the CPC has lost ETR signals from both ports and started using the local oscillator to step TOD clocks. When this condition occurs, all programs that require TOD clocks to be synchronized to the ETR time immediately initiate an orderly shutdown process.
- Port-availability change, which indicates that a port status has changed from operational to nonoperational, or vice versa. This interruption causes the system control program to perform error logging (when the port status changes to nonoperational) or port-connection verification on reconnection (when the port status changes to operational).
- ETR alert, which indicates that certain information, such as time offsets or ETR sending unit status, has changed in the ETR data. This informs the system control program to either update the parameters for time-conversion algorithms or perform error logging.

PR/SM-support considerations

The PR/SM support for ETR was complex and difficult because of the need to control the use of physical resources in the real-time environment. The following discussion of some of the design issues, alternatives, and decisions illustrates some of these problems.

- *Basic design considerations*

One approach that was considered for supporting ETR under PR/SM was to hide the ETR functions from the PR/SM partitions as much as possible. This approach was not taken for the following reasons:

- If a partition is operating with other CPCs in a sysplex, the partition must be able to read the ETR-network ID to determine whether it is connected to the same ETR network as the rest of the sysplex.
- Error situations, such as switch to local, must be reported to all partitions, since in such situations the control program running in each partition must be prepared to take special emergency action—for example, to shut down applications that require synchronized clocks.
- PR/SM must not introduce a discontinuity in the TOD-clock value used by a partition without reporting this to the partition. A discontinuity might occur, for example, if PR/SM attempted to synchronize the physical TOD clock with the ETR after the CPC had been in the local mode.

  It would have been possible to define a new signal to the partitions indicating that the TOD clock had been resynchronized with the ETR and that there was a possibility of discontinuity. However, it was decided that a better solution was to provide the same ETR information and functions to the partitions, rather than define another interface.

Under PR/SM, a TOD-clock-offset register is provided for each partition. When a STCK instruction is issued by a partition, the contents of the physical TOD clock associated with the partition are adjusted by the contents of the TOD-clock-offset register, and the result is returned to the partition. With this TOD-clock-offset register, the clock in a partition and the associated physical clock can have different values. When PR/SM sets physical TOD clocks to synchronize the clocks with the ETR time as part of certain recovery actions, the TOD-clock-offset register for each partition is adjusted by the appropriate amount, so that no discontinuity is viewed by the partition. The time-stamp register provides a convenient mechanism to determine the appropriate change in the offset. The same error indication that caused PR/SM to set the physical clocks is then reported to all partitions by PR/SM. Those partitions not aware of, or not using, ETR are not affected by changes in the physical TOD clocks due to ETR and can ignore the report. Those partitions that desire to use the ETR must then, in turn, set their TOD clocks to be in synchronism with the ETR time. When partitions synchronize their clocks with the ETR time, PR/SM simply sets the contents of their TOD-clock-offset registers to zeros.

- *Port-selection constraints*

Although a function is provided for the control program to select a port as the active port, this function cannot be supported under PR/SM, because different partitions could make conflicting selections. This problem was solved by

defining an additional port state in the architecture, indicating that a port is available for reading ETR data but not available for providing the ETR oscillator signal. PR/SM selects one port as the active port and reports to all partitions that the other port is unavailable as a source of the ETR oscillator signal. Should a partition attempt to select the other port as active, the condition is treated as if the partition had switched to local stepping mode.

Conclusion

In a sysplex environment, the allowable offset between TOD clocks in different CPCs is limited by inter-CPC signaling time, which is very small (and is expected to become even smaller in the future). Some environments require that TOD clocks be accurately set to an international time standard. Software techniques, such as Network Time Protocol, cannot meet these requirements. The ETR architecture satisfies these requirements by providing an accurate clock-setting process, a common clock-stepping signal, and an optional capability for attaching an external time source.

A primary goal of the architecture is to provide a time facility whose availability exceeds the availability of any of the individual sysplex elements. The ETR architecture ensures that consistent timing information will always be available by the use of extensive redundancy and recovery mechanisms that far exceed those of the TPF facility.

It is also essential that the integrity of this timing information be ensured. This is accomplished by extensive error detection and correction and by high-priority interruptions for situations where there is a loss (or possible loss) of ETR synchronization. These interruptions alert the system control programs in all participating systems that they must initiate immediate recovery or an orderly shutdown to maintain data integrity.

Acknowledgments

Although the design and implementation of a time-synchronization mechanism may appear to be a simple task, nothing could be further from the truth. Many people from planning, architecture, engineering, programming, and other disciplines and specialities have contributed substantial technical effort to take the ESA/390 time-coordination requirements from concept to product. The authors would have liked to acknowledge each contributor by name and contribution. However, such a list would be too long for this publication. Therefore, the authors simply express their thanks to all those who have been involved, over the years, on the ETR project.

References

1. A. G. Mungall, "Frequency and Time—National Standards," *Proc. IEEE* **74**, No. 1, 132–136 (1986).
2. *Time and Frequency Users' Manual*, NIST Special Publication 559 (revised 1990), Time and Frequency Division, National Institute of Standards and Technology, Boulder, CO 80303.
3. *Time and Frequency Bulletin*, Time and Frequency Division, National Institute of Standards and Technology, Boulder, CO 80303.
4. *IBM Enterprise Systems Architecture/390 Principles of Operation*, Order No. SA22-7201; available through IBM branch offices. (The TOD-clock mechanisms are the same in System/370 and System/390.)
5. *IBM Transaction Processing Facility General Information*, Order No. GH20-7521; available through IBM branch offices.
6. Douglas Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1988.
7. David L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Trans. Commun.* **39**, No. 10, 1482–1493 (1991).
8. *Planning for the 9037 Sysplex Timer*, Order No. GA23-0365; available through IBM branch offices.
9. *NIST Time and Frequency Services*, Time and Frequency Division, National Institute of Standards and Technology, Boulder, CO 80303.
10. T. Basil Smith, William A. Moorman, and Thao Dang, "The IBM S/390 Sysplex Timer," *Proceedings of Fault-Tolerant Computing: The Twenty-First International Symposium*, IEEE Catalog No. 91CH2985-0, June 1991, pp. 144–151.
11. S. A. Calta, J. A. deVeer, E. Loizides, and R. N. Strangwayes, "Enterprise Systems Connection (ESCON) Architecture—System Overview," *IBM J. Res. Develop.* **36**, 535–551 (1992, this issue).
12. T. L. Borden, J. P. Hennessy, and J. W. Rymarczyk, "Multiple Operating Systems on One Processor Complex," *IBM Syst. J.* **28**, No. 1, 104–123 (1989).

Received January 7, 1991; accepted for publication May 18, 1992

Noshir R. Dhondy *IBM Enterprise Systems, Neighborhood Road, Kingston, New York 12401 (DHONDY at KGNVMF, ndhondy@vnet.ibm.com).* Mr. Dhondy is an Advisory Engineer in the Advanced Technology Systems organization of the Mid-Hudson Valley Development Laboratory in Kingston. He joined IBM Kingston in 1968 as an associate engineer in the Electromagnetic Compatibility Department. Since 1978, he has been involved with various aspects of hardware design, including gate array design, card design, and reliability analysis. He was the lead designer of the 9037 Sysplex Timer and was responsible for its high-level design. He has published three invention disclosures, and in 1991 received a divisional president's award. Mr. Dhondy received his B.Tech degree in electrical engineering from the Indian Institute of Technology, Bombay, India, and his M.S. degree in electrical engineering from the University of Pittsburgh.

Richard J. Schmalz *7 Edge Hill Drive, Wappingers Falls, New York 12590 (retired).* Mr. Schmalz was a Senior Planner in the Processor Architecture and System Structure Department of the Mid-Hudson Valley Development Laboratory in Poughkeepsie. He joined IBM in 1956 in St. Louis, Missouri, and held numerous marketing, programming, and engineering positions throughout his IBM career. Mr. Schmalz received three IBM Outstanding Innovation Awards, for his contribution to ESA/370™, for his work on Expanded Storage, and for Hiperspace™. He also received an IBM Second-Level Invention Achievement Award, and holds five issued patents. Mr. Schmalz received his B.S. in mathematics from Washington University, St. Louis, Missouri.

Ronald M. Smith, Sr. *IBM Enterprise Systems, P.O. Box 950, Poughkeepsie, New York 12602.* Mr. Smith is a Senior Technical Staff Member in the Enterprise Systems Central Architecture Department of the Mid-Hudson Valley Development Laboratory in Poughkeepsie. He received his B.E.E. degree in electrical engineering from The Ohio State University in 1957 and joined IBM at the Endicott Laboratory that same year, moving to Poughkeepsie in 1961. He worked on assignments in circuit design, central processor design, and programming before joining Central Systems Architecture in 1966. Mr. Smith has twelve patents, six patent applications on file, and thirteen published invention disclosures. He has received an IBM Outstanding Contribution Award, an IBM Outstanding Innovation Award, and an IBM Sixth-Level Invention Achievement Award.

Julian Thomas *IBM Enterprise Systems, Neighborhood Road, Kingston, New York 12401 (JT at KGNAIX11, jt@donald.aix.kingston.ibm.com).* Mr. Thomas is a Senior Engineer in the AIX/ESA™ Architecture Department of the AIX® High End Systems organization. He joined IBM in 1962 at Poughkeepsie to work on microcode (and microcode-tools) development for the System/360™ Model 50. Recently, he has participated in 3090 and ESA architecture and system design projects, including access register definition and the ESA integrated cryptographic facility. He has also participated in the system design of the ESA ETR facility, including concept, detailed architecture, and specification of 3090 ESA ETR processor attachment designs. Mr. Thomas has eight issued patents and has received an IBM Second-Level Invention Achievement Award and two IBM Outstanding Technical Achievement Awards. He received an A.B. in mathematics (*cum laude*) in 1954 and an A.M. in applied mathematics in 1957, both from Harvard University.

Phil C. Yeh *IBM Enterprise Systems, P.O. Box 950, Poughkeepsie, New York 12602.* Dr. Yeh is a Senior Engineer in the Enterprise Systems Central Architecture Department of the Mid-Hudson Valley Development Laboratory in Poughkeepsie. He received an M.S. degree in computer science and a Ph.D. in electrical engineering from the University of Illinois at Urbana–Champaign in 1977 and 1981, respectively. In 1981, he joined IBM at Poughkeepsie, where he has worked on several assignments in architecture. He has five issued patents and three patent applications on file. He has also published several technical papers and has received an IBM Outstanding Innovation Award. Dr. Yeh is a member of the ACM and the IEEE Computer Society.