Copyright © 1981 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 6, July–August 1981
Printed in U.S.A.

No. 4 ESS:

Network Clock Synchronization

By R. METZ, E. L. REIBLE, and D. F. WINCHELL

(Manuscript received August 4, 1980)

Accurate clock control is a necessity for direct digital transmission of voiceband data between No. 4 ESS offices. This paper describes synchronization of the No. 4 ESS clock, which consists of phase-locking the clock oscillators to an external frequency reference. The operation performed is a second-order digital phase-lock with time constants of 1.4 hours and 3.12 days. In addition, a fast-start mode is provided with time constants of 2.6 minutes and 8.31 minutes. These characteristics provide tracking of frequency shifts due to daily propagation delay variations on transmission facilities, while filtering out higher-frequency jitter components of the references. They also guarantee stable convergent operation, even with the unlikely worst-case of linear oscillator drift. Implementation consists of a unit containing two matched and synchronous microprocessors and a microprogrammed controller. While the phase detector is in hardware, the microprocessors provide the remainder of the loop. Firmware performs the filtering algorithm, oscillator control, unit diagnostics, as well as extensive defensive operational checks. A great deal of effort is made in both hardware and firmware to ensure the integrity of data written to the oscillators. Finally, experimental results show the unit operation tracking design predictions closely.

I. INTRODUCTION

The No. 4 ESS is a digital toll switching system, whose time-division network routes standard 8-bit PCM signals.[1] Digital interfacing of the network to T-carrier facilities is provided by the Digital Interface Frame (DIF), which converts and concatenates a number of T1 facilities into higher-speed serial bit streams for the switch, and vice versa. Thus, the basic timing for the switching network is the 8-kHz frame rate typical of T-carrier facilities. Timing for the network is provided by the Network Clock (NCLK) frame, which distributes a 16.384-MHz square-wave pulse train to each of the network frames. The 8-kHz framing information is transmitted as a missing pulse in the 16.384-MHz signal once every 125 μs. Synchronization of the network consists of controlling the frequency of the NCLK such that No. 4 ESS offices that are digitally connected run as close as possible to the same frequency. This is done by phase-locking the clock oscillators to the externally supplied Bell System Reference Frequency (BSRF) or to a T1 line from another No. 4 ESS. Thus, the No. 4 ESS is part of an overall system timekeeping plan consisting of a master-slave hierarchical timing structure.[2] The BSRF is the master timing source, distributed throughout the country to clusters of digitally interconnected No. 4 ESS switches. One switch in each cluster is designated as a master and is phase-locked to the BSRF, while the other switches are slaved in a tree-like structure to the master via timing carried in the digital interconnections.

II. SYNCHRONIZATION REQUIREMENTS

When two No. 4 ESS offices are directly digitally interconnected by a T-carrier facility, data arrives at a DIF at a rate determined by the source NCLK, and is read out of buffers into the network according to the local NCLK. Thus, differences in clock frequencies between the two offices result in DIF buffer overflow or underflow. This is compensated for by the loss or repetition of a frame of data, and is called a slip. The impairment to PCM voice is negligible for slip rates as high as several per second, while the effect on voiceband data is more drastic. Any slip is an undesirable loss of information, and thus, the end-to-end slip rate objective for the Switched Digital Network has been set at one slip in 5 hours.[3] One-half of the objective (one slip in 10 hours) is allocated to digital transmission facilities. The remaining half of the objective is allocated to local digital switching systems. Thus, essentially no slips are allowed for digital toll switching systems.

The drift rate of the No 4 ESS NCLK is somewhat better than one part in 10^{10} per day, which means that oscillator adjustment is necessary.

With the synchronization unit described here, the No. 4 ESS experiences an essentially zero slip rate, and if the No. 4 ESS is forced by trouble to run for two weeks without an external reference, it still experiences less than one slip every 20 hours.

III. NO. 4 ESS NETWORK CLOCK CHARACTERISTICS AND ARCHITECTURE

The source of the 16.384-MHz clock frequency is a set of four double-oven, quartz-crystal, voltage-controlled oscillators, connected in an analog master-slave phase-locked arrangement.[4] (See Fig. 1.) One oscillator is designated master, and the other three are phase-locked slaves.

Phase error detectors and voting circuits are designed to implicate failing oscillators and reconfigure the master-slave arrangement as necessary. In addition to the analog frequency-control input, each oscillator has a 14-bit digital control, capable of adjusting the frequency a maximum of ± four parts in 10^7, with a nominal least-significant-bit sensitivity of 5×10^{-11}. Finally, the oscillators each provide two phase bits, indicating whether the oscillator, if it is a slave, is lagging or leading the master in phase by more than 0.5 degree.

The Network Clock Synchronization Unit (NCSU) controls the clock frequency via the 14-bit oscillator inputs. While the master is being

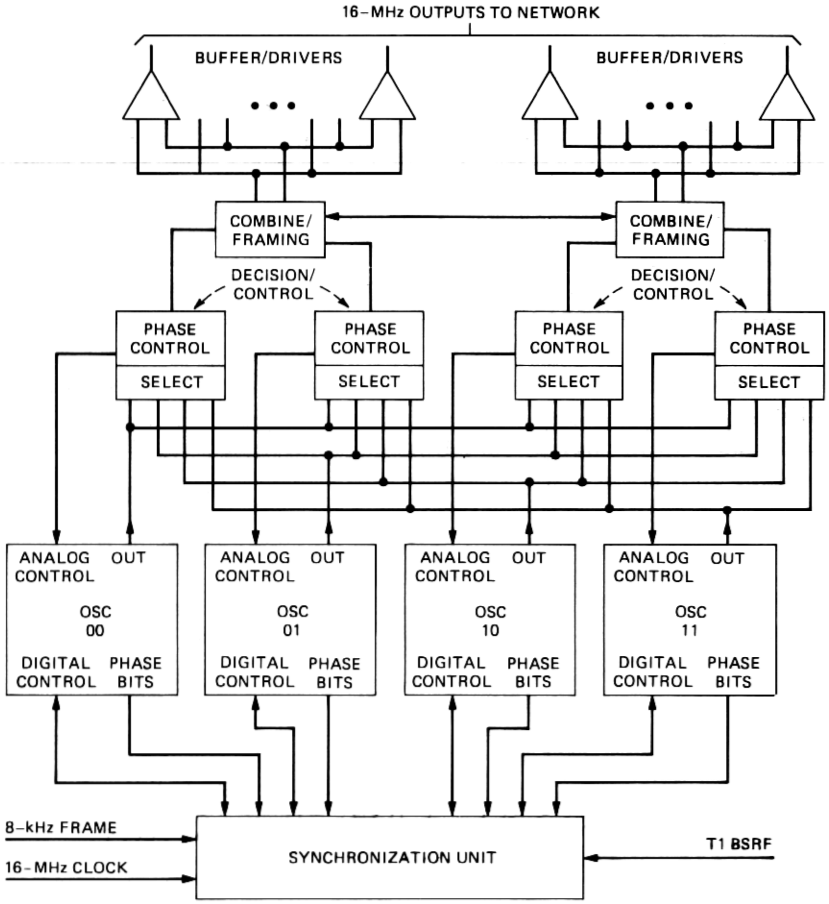

Fig. 1—Network-clock block diagram.

phase-locked to the reference signal, the three slaves are tracked to follow the master with the help of the phase bits. Slave updating algorithms keep the slaves within ±0.5 degree (at 16.384 MHz) of the master.

IV. PHASE-LOCK ALGORITHM AND TIME AND FREQUENCY DOMAIN PERFORMANCE

Since stability in time (or phase) of the 8-kHz frame is what is ultimately important in preventing slips, phase-lock, as opposed to frequency-lock or some other method, was chosen. The phase-lock scheme is similar to what is used in the Digital Data System.[5]

The major components of the loop are the phase comparator, filter section, and the master oscillator. (See Fig. 2a.)

Inputs to the phase comparator are a 4-kHz reference signal, 8-kHz frame pulse, and 4.096-MHz clock, the latter two derived from the local oscillator in No. 4 ESS. A phase comparison consists of starting a counter with the leading edge of the 4-kHz reference signal, and subsequently stopping it with the next 8-kHz frame pulse. The 4.096-MHz clock is counted in between. (See Fig. 2b). Before each count, the counter is preset to −256, thus, yielding a phase comparator output range of −256 to +255 for a given comparison, where 0 corresponds to zero phase error and the exact half-way interspersing of 8-kHz frame pulses and 4-kHz reference pulses. It is evident that a phase comparison is done every 250 μs.

During a typical 8.192-s interval, 2^{15} such phase measurements are made, added, and divided by 2^{15} to yield an average phase error for the 8-s interval. Deviations from this are described later in the operational firmware. This average phase measurement is then multipled by 2^{-15} and added to a running sum called the integral term. Finally, the running sum is added to the average phase measurement, forming a 14-bit frequency control word which is sent to the oscillator.

Since the 8.192-s sampling interval is relatively frequent compared to the time constants of the loop, we can model the system in a continuous form. (See Fig. 2c.) Note that ϕ_R is the phase of the

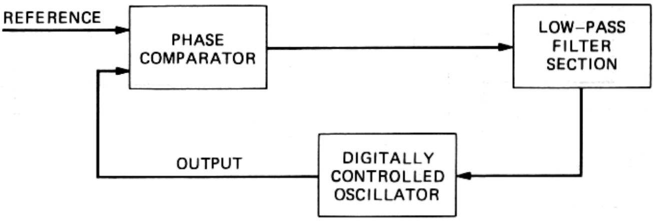

Fig. 2a—Phase-lock loop block diagram.

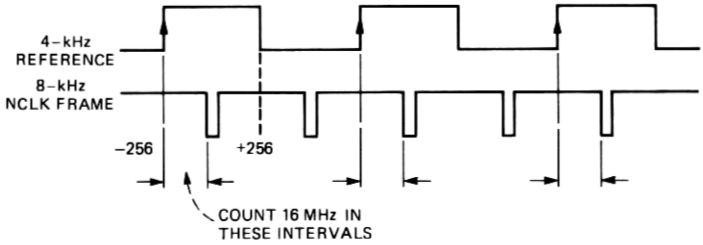

Fig. 2b—Phase-measurement timing.

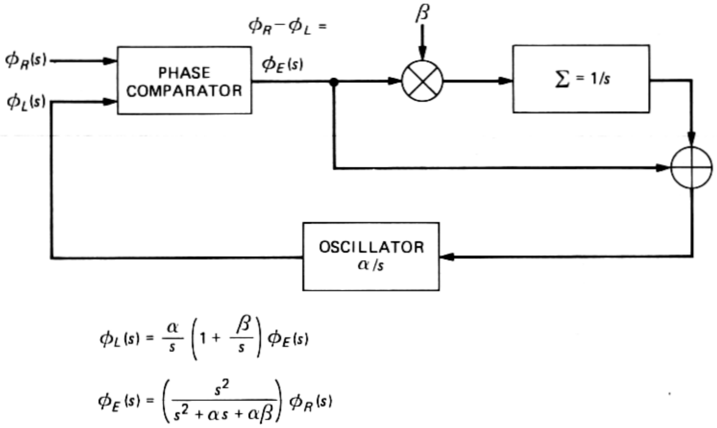
$\phi_R - \phi_L =$
β
$\phi_R(s)$
$\phi_L(s)$
PHASE COMPARATOR
$\phi_E(s)$
$\Sigma = 1/s$
OSCILLATOR α/s
$\phi_L(s) = \frac{\alpha}{s}\left(1 + \frac{\beta}{s}\right)\phi_E(s)$
$\phi_E(s) = \left(\frac{s^2}{s^2 + \alpha s + \alpha\beta}\right)\phi_R(s)$

Fig. 2c—Frequency domain model.

reference signal, ϕ_L the local frame pulse phase, and ϕ_E the difference between them. The multiplier in the integral branch is β, and the oscillator sensitivity and phase comparator gain are combined into a common term α. Therefore, the phase of the local frame pulse can be expressed as

$$\phi_L(s) = \frac{1}{s}\,\alpha\left(1 + \frac{\beta}{s}\right)\phi_E(s), \tag{1}$$

and the phase error is

$$\phi_E(s) = \frac{s^2}{s^2 + \alpha s + \alpha\beta} \cdot \phi_R(s). \tag{2}$$

To determine α, we consider the following: Since the phase comparator has a linear region of 125 μs, corresponding to 512 bits, each bit represents 244 ns. A 1-bit change causes the oscillator to change 5×10^{-11}. Thus

$$\alpha = 5 \times 10^{-11}/244 \text{ ns} = 2.048 \times 10^{-4}/\text{s}. \tag{3}$$

Another way to think of this is in the open-loop sense: Given a 1-bit change in oscillator input, how long will it take for the phase comparator output to change by one bit? It requires 244 ns of phase shift per bit of comparator output, which corresponds to four cycles of 16.384 MHz. Therefore,

$$t = \frac{4}{5 \times 10^{-11} \times 16.384 \times 10^{6}} = 1.36 \text{ h} = \frac{1}{\alpha}. \tag{4}$$

Since the scaling factor at the input of the integral path is 2^{-15}, and updating of the oscillator is done every 8.192 s, β is given by

$$\beta = \frac{2^{-15}}{8.192} \text{ s} = 3.73 \times 10^{-6}/\text{s} \qquad \text{and} \qquad \frac{1}{\beta} = 3.1 \text{ days}, \tag{5}$$

which is the time constant of the integral portion of the phase-locked loop.

A fast-start mode is provided for synchronization under startup conditions. It has the ability to resolve much larger frequency offsets between the clock oscillators and the incoming reference. It is characterized by shortened time constants and a much wider capture range (± 4 parts in 10^7),

$$\alpha' = 32\alpha = 6.55 \times 10^{-3}, \qquad 1/\alpha' = 2.6 \text{ min}, \tag{6}$$

$$\beta' = 512\beta = 1.9 \times 10^{-3}, \qquad 1/\beta' = 8.13 \text{ min}. \tag{7}$$

Figure 3 shows the computation algorithm for both fast-start and normal modes.

V. PHASE AND FREQUENCY RESPONSE

Since it is phase drift in the 8-kHz framing of the No. 4 ESS clock that will cause slips on transmission facilities, it is the phase response of the synchronization unit that we are ultimately interested in. Therefore, let us first determine the response to an input step in frequency. The phase error, in time, is given by

$$\phi_E(t) = \frac{\Delta f}{b} e^{-at} \sin(bt), \tag{8}$$

where

$$a = \frac{\alpha}{2}, \qquad b = \left(\alpha\beta - \frac{\alpha^2}{4}\right)^{1/2}.$$

Figure 4a shows that the phase error builds up with a time constant of $1/\alpha$ to a peak of 4.45 kHz/Hz at 19.92 ks and then dies out with a time constant of $1/\beta$.

Correspondingly, the frequency response to a step in frequency is shown in Fig. 4b. The initial error is approximately ΔF, heading

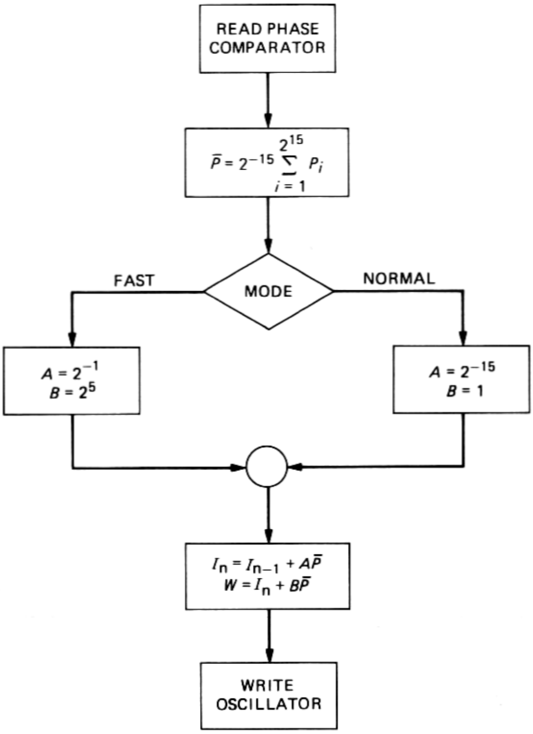

Fig. 3—Synchronization unit—main algorithm.

through zero with time constant $1/\alpha$, and then decaying to zero with time constant $1/\beta$. Note that in the limit, both the frequency and the phase error diminish to zero.

Next, if the input takes a step in phase, the phase-error response is given approximately by

$$\phi_E(t) \simeq \left(-e^{-\alpha t} - \frac{\beta}{\alpha} e^{-\beta t}\right) \cdot \Delta\phi_R. \tag{9}$$

As shown in Fig. 5a, the initial offset is of course $\Delta\phi_R$, decaying predominantly with the $1/\alpha$ time constant. The effect on frequency is shown in Fig. 5b.

Finally, we can consider what happens if there is a linear frequency drift in the input signal (or the oscillator output). Although this linear drift is an unlikely situation, it forms a worst-case bound on what the oscillator or the references can do, short of a faulty condition. For a drifting input, moving at ΔF Hz/s^2,

$$\phi_E(t) \simeq \frac{2}{\alpha\beta} + \frac{2}{(\alpha - \beta)} e^{-\alpha t} - \frac{2}{\beta(\alpha - \beta)} e^{-\beta t}. \tag{10}$$

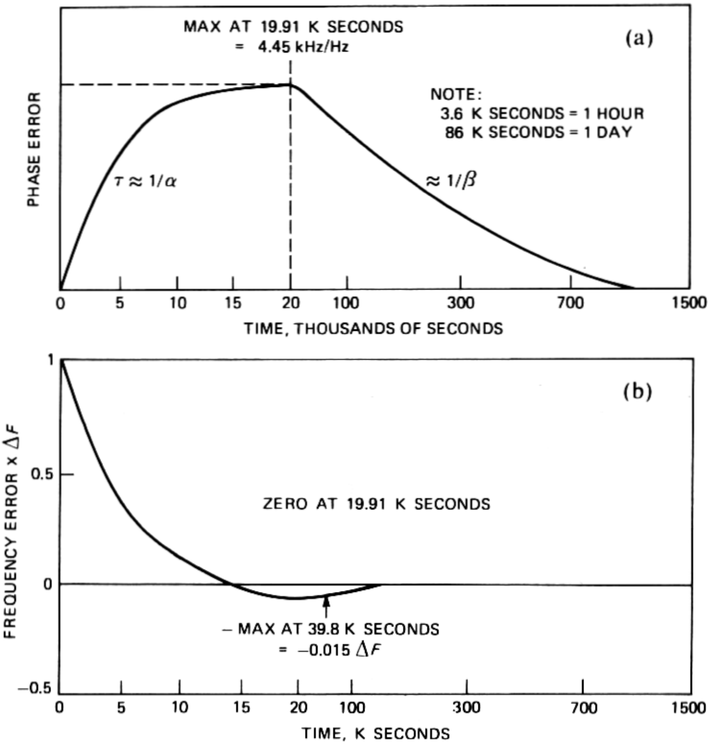

Fig. 4—(a) Phase response to input step in frequency. (b) Frequency response to input step in frequency.

Note that at zero time, the phase error starts out at zero, and in time builds up to a constant offset of $2/\alpha\beta$. This is important, since it determines the maximum phase error for a given drift characteristic of the oscillator. For example, if the worst-case drift is 1×10^{-10} per day, the constant long-term phase error would be about 3 μs, or 12 bits from the phase comparator. This is tolerable, indicating that in the worst-case, the No. 4 ESS frame might shift a constant amount of $\frac{1}{40}$ of a frame, and in no case will phase continuously drift.

VI. SYNCHRONIZATION UNIT HARDWARE DESIGN

Figure 6 is a block diagram of the synchronization unit hardware. The heart of the system is a pair of microprocessors with a fully duplicated Read Only Memory (ROM), Random Access Memory (RAM), I/O, and interrupt complex. A bit-sliced microprogrammed controller has Direct Memory Access (DMA) and interrupt access to the microprocessor community, and performs unit control functions, as well as provides a duplicated interface to the No. 4 ESS Central Control (CC) processor via the peripheral bus. Small microcoded programs in the controller respond to 76 different diagnostic and control orders from the CC by performing microprocessor DMAS and interrupts, data routing, register loading, etc., as required. Three orders are used by the controller for self-diagnosis, one exercising the instruction set, stack, and program counter, the other two testing the program parity, and parity check circuits. In addition to a conventional check of good program parity, one of these two orders increments through a program section where every line has bad parity in a vector form that completely tests the parity tree and check circuits. Parity should fail for every line of this test.

During phase-lock operation, the phase comparator circuit generates an output every 250 μs, representing the relative phase of the incoming reference and local frame. At initial startup, and after a phase "hit," the phase may be arbitrarily adjusted, under program control, by two special phase build-out circuits, one for T1, the other for BSRF. This maximizes the capture range of the loop, and minimizes phase excursions at startup and after reference hits or losses.

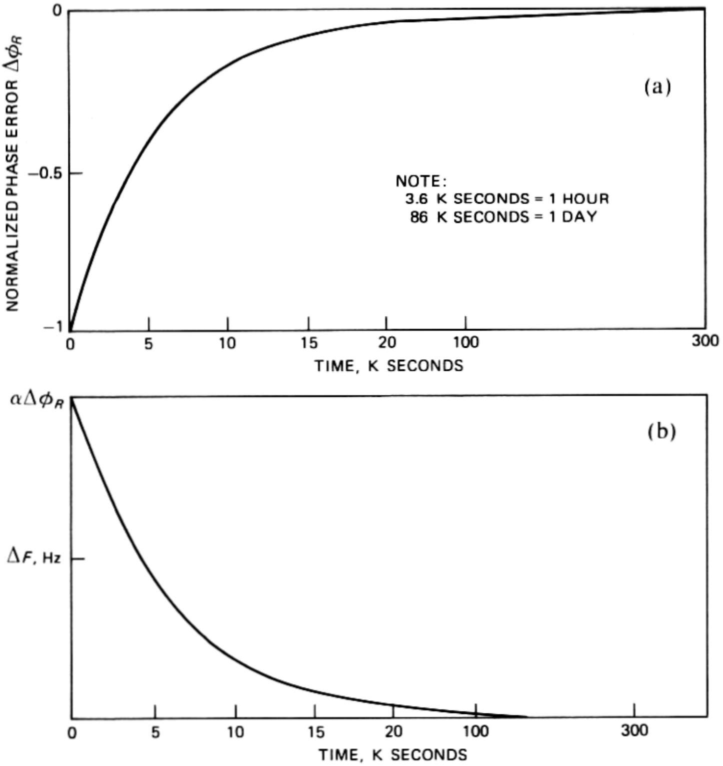

Fig. 5—(a) Phase-error response to step in phase. (b) Frequency response to step in phase.

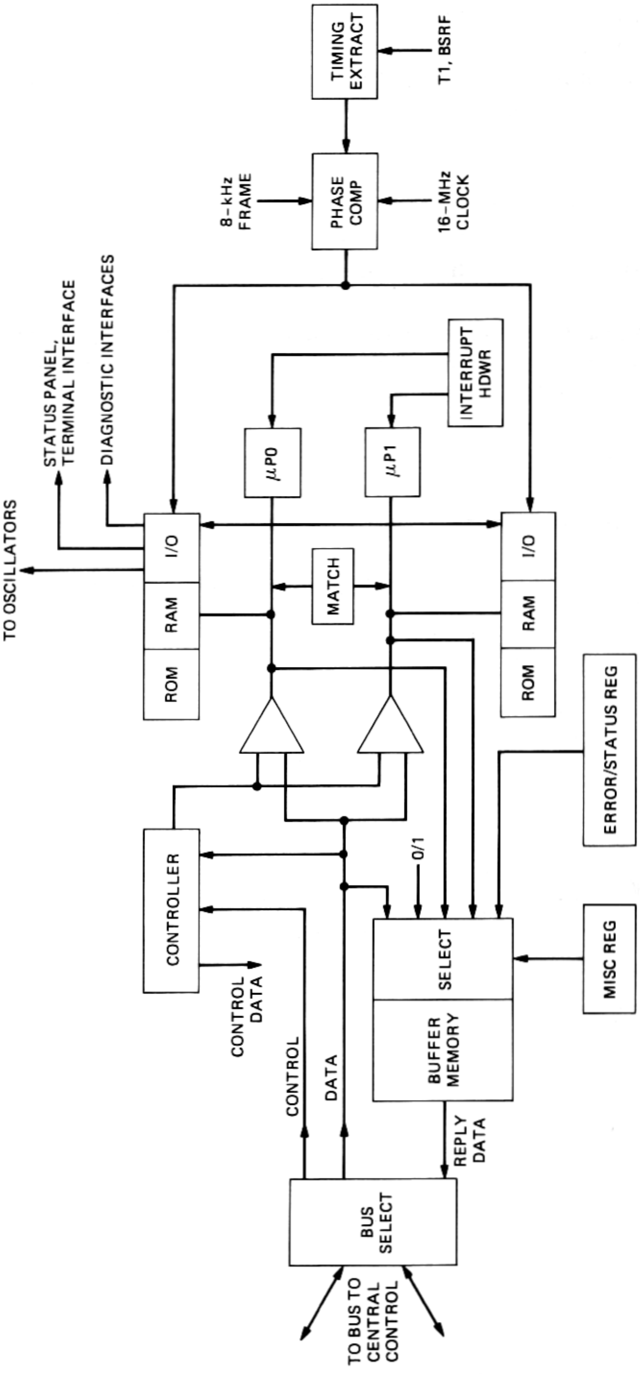

Fig. 6—Synchronization-unit hardware block diagram.

The operational program then reads the 250-μs comparison samples, and performs the algorithm described before. Oscillators are updated every 8.192 s nominally, plus computational and administrative overhead.

It is extremely important that bad data should never be written to an oscillator. Several levels of hardware and firmware protection are employed to ensure the quality of oscillator writes. The first and most important is the duplication of the microprocessors, including 32 K of program ROM, 4 K RAM, 51 I/O ports, and the interrupt hardware. The processors run synchronously and parity over address and data is matched between the two. A mismatch inhibits writing to the oscillators, interrupts and halts the processors, and sets an error in the error register. To further protect the data sent to the oscillators, I/O ports containing oscillator data are cross-coupled: That is, when a processor writes a port and then reads it back, it reads back that port from the other processor. Thus, each processor can match its data against the other, and a write is performed only if the data is the same. Even then, a write to an oscillator will succeed only if it is performed exactly simultaneously by both processors. After the write, the data is read back from the oscillator and is once again verified against what was calculated.

The cross-coupled I/O ports are also used by the diagnostics to verify programs and data stored in ROM. Each processor can read out its own ROM and exchange it with the other via these ports and, thus, check the two for equality.

Another hardware feature that aids in diagnosing the processors is a pair of address cross-coupled read/write I/O ports. When microprocessor 0 writes ports A and B, and then reads them back, it reads A and B, respectively. Processor 1, however reads them back as B and A, respectively, effectively doing an address interchange of data. A section of diagnostic code uses these ports to create differential code in RAM, which is subsequently executed to exercise and check the bus parity match circuits.

To monitor performance of the unit, a status panel provides a readout of the current phase error and integral term, as well as indications of the operating mode and reference in use. An EIA compatible interface on the status panel also allows monitoring of the synchronization process by a terminal. This terminal interface, in conjunction with about 5 K bytes of the program, provides various monitoring modes, as well as firmware utilities, described later.

VII. FIRMWARE

The synchronization unit microprocessor-based firmware is comprised of three parts: Operational programs, diagnostics, and utilities.

The purpose of the operational program is to implement the phase-lock algorithm. The diagnostic portion performs tests on the hardware. The utilities provide the tools necessary for program development. In the following sections, these three areas will be described in detail. A hierarchy chart appears in Fig. 7, and a state diagram in Fig. 8.

7.1 Operational firmware

As shown in Fig. 7, the operational program consists of two parts, the "main loop" and the interrupt handlers. The main loop is where the unit usually resides, and is where the phase-lock algorithm is performed. The interrupts are the means by which the main loop is entered, or they change the operating environment, or they perform various other tasks related to synchronization.

A state diagram is shown in Fig. 8. The "phase-lock mode" state is where the main loop is executed. The processors are in a "halted" state after reset and trap events. In the "free run" state the oscillators are not written. Certain events (like interrupts) cause a transition from one state to another.

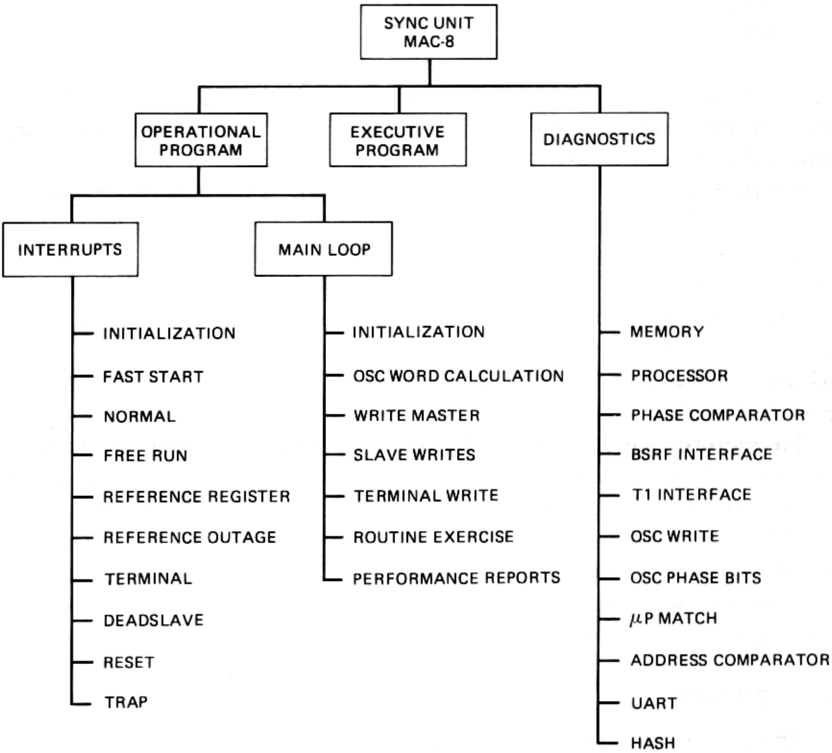

Fig. 7—Firmware structure.

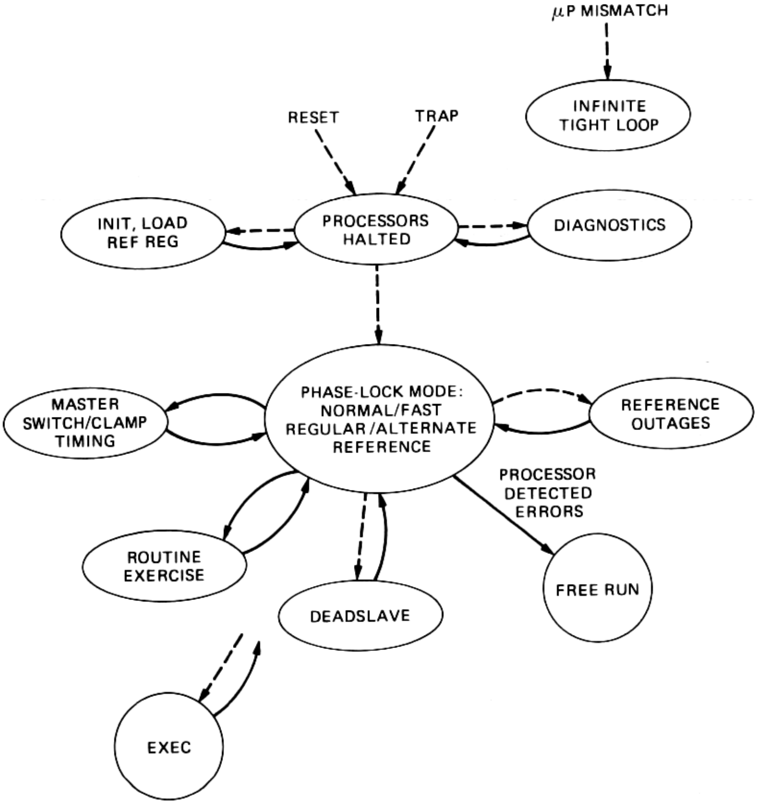

Fig. 8—Processor states.

7.1.1 Main loop

The main loop is an 8-s cycle that performs the synchronization algorithm in Fig. 3. In this loop, a phase measurement is made, the master written, and the slaves written. When the synchronization unit is in fast start or normal mode, this loop is executed repeatedly. There are seven main parts to the cycle (Fig. 7).

The first part of the main loop consists of the miscellaneous tasks. One such task is the maintenance of event counters such as:

(*i*) The number of 8-s cycles executed since the beginning of fast start.

(*ii*) Time elapsed since the last power up.

(*iii*) The duration of reference outages.

Also, the I/O is read for the master oscillator number. The RAM is read in a special location called the reference register to see whether BSRF or T1 is the reference signal to use, and bits in the I/O are written to select that signal. An I/O port is read to find out which references have outages and another I/O port is written to screen irrelevant or unwanted interrupts. The integral terms and oscillator words are copied into a special area of RAM which is read by the CC and put into the "critical registers" for the synchronization unit.

The next part in the process is the calculation of the master's new 14-bit control word. This program first decides whether to call a routine to perform a phase build out. (The shifting of the reference signal to produce a desired phase error.) For the beginning of fast start, it is necessary to build out to a phase error of zero, whereas after a phase hit we build out to the previous phase measurement. T1 and BSRF build out are performed by different hardware. From the firmware's point of view, the build out algorithm is as follows:

(*i*) Write the build-out register (T1 or BSRF) to 0 and take a phase measurement, A.

(*ii*) If B is the desired phase error, then write the build-out register to C, where

$$C = \frac{A - B}{16} + 32, \qquad A < B,$$

$$C = \frac{A - B}{16}, \qquad A > B.$$

The accuracy of this procedure is ±9 bits of phase error. For the BSRF, the accuracy is enhanced to ±3 bits by using a diagnostic access port to cut off the BSRF signal for 13 μs, which causes a shift in the reference signal as it appears at the phase comparator. This is done repeatedly until the desired accuracy is achieved. The process never requires more than seven iterations.

The next step in the process is taking a phase measurement. The phase comparator comes up with a phase measurement every 250 μs. The phase error takes on the values between -255 and $+255$, and appears in a 16-bit I/O port. A firmware routine forms the sum of 2^{12} consecutive phase measurements (takes 1 s) and then computes the average phase error. During the process, adjacent 250-μs phase measurements are checked for jitter by requiring that they differ by no more than two bits. If they do differ by more than two, then the event is recorded and the summing routine starts over. Starting over too often results in an error reported to CC.

The time spent updating slaves can consume a considerable portion of the 8-s cycles, therefore, the phase measurement time is varied to compensate by subtracting the time spent updating slaves in the last cycle from 8 s. The resulting time dictates the number of consecutive 1-s phase measurements which are made, which varies from 1 to 7. The average phase error is then computed from the 1-s averages. The difference between adjacent 1-s phase measurements is monitored, and, if it exceeds three, then a phase hit has occurred, and a build out to the last good phase measurement is performed.

The 8-s phase measurements are monitored from cycle to cylce. The decision to transfer from fast start to normal mode is made based on the magnitude of the phase error (must be ≤ 1) and the slope of the phase error (must be $\leq 1/80$ bit/s). In normal mode, if the slope of the phase error curve exceeds 164/40 or 164/400 or 164/4000 bits/s, then an error is reported to CC and the unit enters a free run state. If the magnitude of the phase error exceeds 230 at any time, then an error is reported and the free run state entered.

Having gone through the checks in phase error, the next step is to update the integral term (see Fig. 3). The new integral term is equal to the old one plus a constant, A, multiplied by the phase error. The master's word calculation follows as the integral term, plus another constant, B, multiplied by the phase error. The integral term, a 29-bit quantity, can be thought of as the 3-day average of the oscillator word, a 14-bit quantity.

Before writing the master with the just-calculated word, a basic check is made for a reasonable value. If the word presently in the master differs from the new word by more than 256 bits (4096 for fast start) then processor-detected error is set (a bit in the I/O which results in an interrupt to the CC) and free-run mode is entered. If the difference is greater than 64, then the master is walked by steps of 64 to the desired value, with the slaves updated at each stage. This prevents NCLK frame detected phase errors from occurring.

Anytime an oscillator is written, a common routine is called which performs some defensive checks and then writes the oscillator. The first check made is that the word to write cannot differ by more than 256 from the word in the oscillator. The second is that both processors must agree on the word that is to be written. This is accomplished by using a crossed I/O port where each processor can read what the other wrote.

The next line of defense is in the hardware. The program writes an I/O port, which will strobe the data in the crossed ports into the oscillators if (1) the oscillator clamps (CC-controlled gates which, when set, inhibit oscillator writing) are not set, (2) the processors are synchronous, and (3) there are no errors in the hardware error source register.

The last line of defense is that the program always executes a 100-ms pause between oscillator writes. This allows time for the NCLK frame to detect a phase error and interrupt the CC, which, in turn, will set the oscillator clamps so that no more writes can be made. Since the clock frame can run on three oscillators, no degradation of service occurs—only a loss of redundancy.

The next step in the process is to deal with the slaves. There is an I/O port which reflects the state of the oscillator clamps. Those slaves which are unclamped are updated. As a first approximation, whatever frequency control word offset was last written to the master is written to each slave. Next the oscillator phase bits are used to more precisely align the slaves to the master.

There is a 2-bit phase indication (slave relative to master) in the I/O for each slave. These take on the values 0, 1, or 2 depending on whether the slave is within ±.5 degrees of the master, greater than +0.5 degrees, or less than −0.5 degrees, respectively. The zero condition window, usually about 50 oscillator word-bits wide, is measured periodically. If a slave is found with nonzero phase bits (after writing the master's change to it), then one-half of the window width is added or subtracted to the word in the slave. If the phase bits are still nonzero for the slave, then it is walked towards its window until the phase bits become zero.

The next step in the main loop is the writing of periodic data to the terminal. The following information is written: phase error; the four oscillator words; the integral term; which oscillators are enabled; the master; the mode (fast start or normal mode); the reference (BSRF, or T1); and the cycle counter.

As part of the main operation loop, routine exercise is performed every hour when the unit is in normal mode. The following diagnostic routines are called: ROM test, I/O test, processor self-test, phase comparator test, BSRF interface test, T1 interface test, oscillator buffer test, and the oscillator write tests. Any failures result in a processor-detected error and free run.

A set of performance reports is maintained. These reports include: number and duration of free runs, reference outages, phase hits, frequency offsets, and fast starts.

7.1.2 Operational interrupts

The operational interrupt handlers perform various functions while the unit is in a phase-locked mode or they may be used to place the unit into a phase-lock mode. The interrupts come from the CC or directly from hardware in the unit. The interrupt handlers are listed in Fig. 7.

Init is an order from CC to initialize RAM and the I/O ports. Fast start, normal, and free run are all orders that cause the unit to enter the respective states. During free run, the program is in a loop which increments a free-run duration counter. The reference change order is used by the CC to load the reference register.

The terminal handler processes requests from the terminal. A set of commands exist which allows one to change the terminal printing format, have the machine performance report printed, run routine exercise, or print information about the last processor-detected error. In addition, the status panel may be placed in a trace mode, where the present program state and last interrupt processed are displayed. With another command one may exit to the executive program.

The dead-slave interrupt is used to bring on line a new oscillator, or one whose frequency is very different from the master's. In this region, the slave is out of phase-lock with the master and the 2-bit phase indication in the I/O is meaningless. The phase comparator is used to measure the frequency difference between the slave and the master. The output of the slave is used for the reference input to the phase comparator and then the rate of change of phase error measured. Based on the frequency difference calculated, a new slave word is calculated and written to the slave. The process continues until the clock frame is able to phase lock the slave and then the phase bits are used to bring the slave in.

Reset is the highest level of interrupt. Upon receiving this interrupt, some code is executed which initializes variables and then the processors halt. Interrupts from the halted state result in synchronous operation of the processors.

Executing an illegal OP code results in a trap. The trap handler records the program counter in RAM, prints it on the terminal, and then halts.

7.2 Diagnostics

The diagnostic routines are listed in Fig. 7 and are briefly described here.

The memory tests are divided into ROM check and RAM check. For ROM check a crossed I/O port is used where each processor reads what the other one wrote. By reading each ROM location, writing the data to the crossed port, and reading the crossed port, the processors compare the two ROMS.

The RAM test uses a conventional walking 1's and 0's algorithm. In order to discover a wider class of faults, the RAM test routine uses I/O ports for scratch pad memory, rather than RAM. The read/write I/O ports have been previously tested by CC-based diagnostics.

A processor self-check is implemented in the firmware. Here the instruction set is verified for various addressing modes by performing sample problems and comparing results against constants.

The phase comparator, BSRF interface circuitry, and T1 interface circuitry are tested by firmware routines. The build-out circuits are also tested by these routines. Test vectors are applied through diagnostic access points in the I/O.

The ability to read and write oscillators is verified by a firmware routine. Also, the performance of the oscillator phase bits is checked. The clamps between the synchronization unit and the oscillators are also tested here.

Other diagnostic routines test the processor match circuitry, the address comparator, and the terminal interface. A routine can be called by CC, which performs a hash sum over the memory to verify the version of the program.

7.3 Utility program

A utility (executive) program is also included in the synchronization unit firmware. This program deals with commands typed in at the terminal. The commands enable one to read and write memory, load program in ROM-emulation, set break points, add patch code, and initiate execution of program. The utility program is a useful tool for program development as well as hardware troubleshooting.

VIII. PHYSICAL DESIGN

The NCSU is a J-coded 8-inch-high by 36.25-inch-wide unit, using 1A Technology hardware. The unit contains 18 "FG" type circuit packs, each being uniquely coded apparatus with two codes having additional MC coding for the documentation of firmware. These packs are arranged in three side-by-side apparatus housings occupying the majority of the unit. (See Fig. 9.) The circuit packs are connected by backplane wiring. In the case of the microprocessor bus leads and all of the +5 volt power and ground distribution, the wiring is via the multilayer printed wiring board. The bus leads being in the multilayer board provide more noise immunity than possible with wired connections. The natural flow of these leads would form long parallel horizontal runs. The danger in so much parallel exposure is crosstalk. To counteract the natural flow, a system of routing the leads as shown in Figure 10b was adopted. Contrast this with the natural flow as shown in Figure 10a; note that while the length has been slightly increased, the amount of parallel exposure has been cut in half. Figure 11 is a photograph of the A section of one the layers of the multilayer printed wiring board, showing the actual routing.

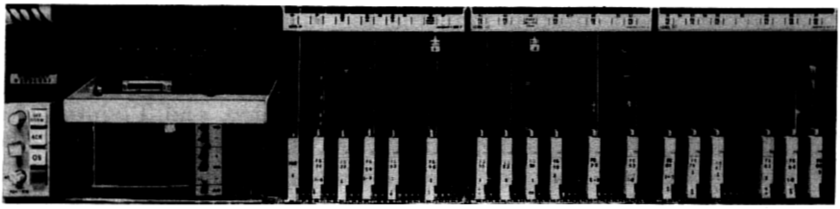

Fig. 9—Network-clock synchronization unit (front view).

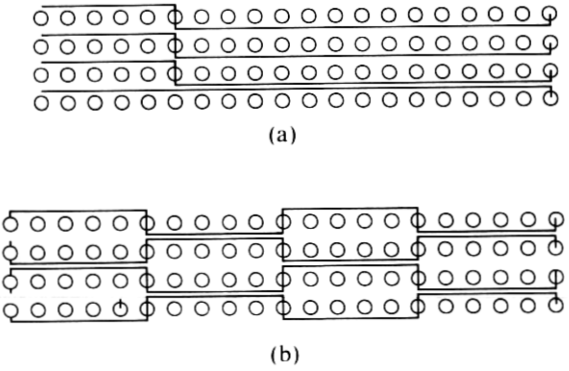

Fig. 10—Patterns. (a) Natural flow. (b) Anticrosstalk.

The unit power consists of a 140-V to 5-V power converter, power control relays, power switch, and two circuit packs.

The unit also contains a status panel (see Section IV), a terminal strip, and the jack for the BSRF.

IX. TESTING AND PERFORMANCE EVALUATION

To verify the phase-lock algorithm and its implementation, an experiment is performed. We start with an oscillator whose frequency is within 5×10^{-11} of a stable reference. Then the oscillator's word is changed by 3814 bits or a frequency offset of 1.83×10^{-7}. The unit is placed in the fast-start mode and the phase error versus time history is recorded.

Using eq. (8) with $\alpha = 6.29 \times 10^{-3}$/s and $\beta = 1.96 \times 10^{-3}$/s (for a cycle time of 8.0 s and an oscillator sensitivity of 4.8×10^{-11}/bit), we

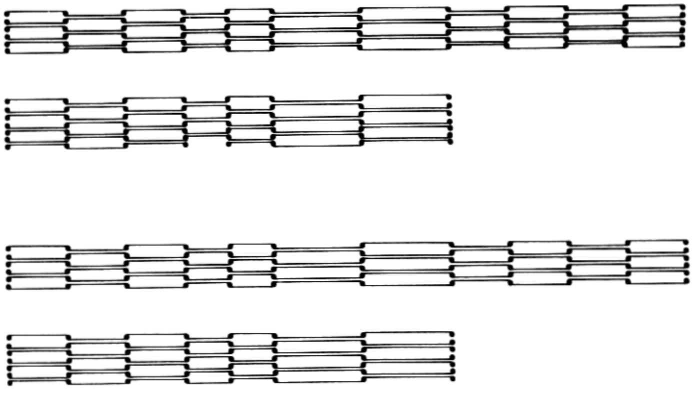

Fig. 11—Actual bus routing.

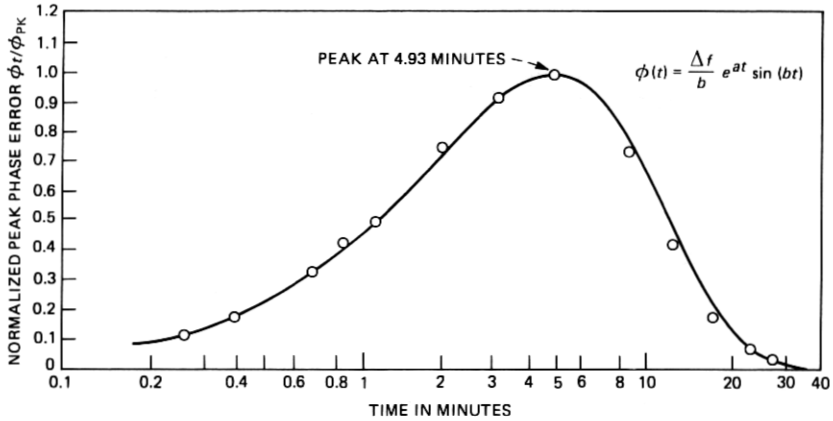

Fig. 12—Fast start.

can calculate that the peak phase error should be 85 bits of phase comparison and should occur at $t = 296$ s. The experimental result is a peak of 87 at $t = 284$ s.

The above experiment is repeated for normal mode with a frequency offset of 200 oscillator control word bits, which is equal to 0.962×10^{-8}. The experimental peak phase error is 188 and the predicted is 188.2. The peak time for the experiment is 346 min versus a predicted time of 344.

The theoretical phase error expression is normalized with respect to the theoretical peak value, and the resulting curve plotted in Fig. 12 for fast start and Fig. 13 for normal mode. The normalized data is plotted as x's in the figures. The agreement between theory and reality is quite good.

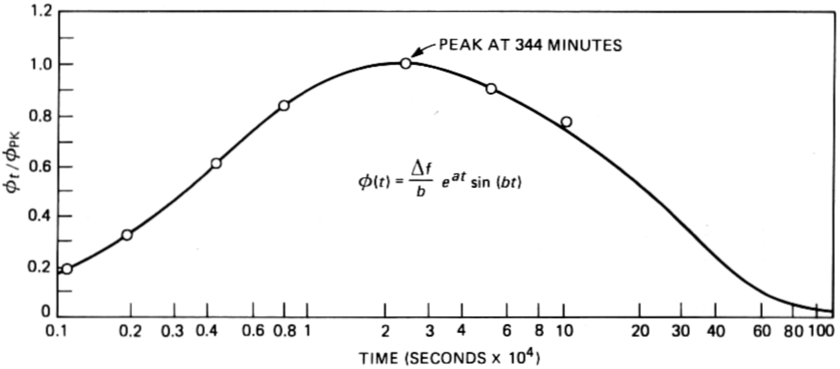

Fig. 13—Normal mode.

X. SUMMARY

Synchronization of the No. 4 ESS is performed by phase-locking the oscillators in the NCLK to the Bell System Reference Frequency or to the framing on a T1 line from another No. 4 ESS. Thus, the No. 4 ESS is in the upper layers of the treelike timing structure of the Switched Digital Network. A digital second-order phase-lock is used, exhibiting excellent convergence and stability characteristics. It is implemented with duplicated and matched microprocessors as part of the loop. This, along with extensive operational and hardware checks ensures a high degree of confidence in data written to the oscillators. Most diagnostics for the synchronization hardware are contained in the microprocessors. Finally, experimental testing of the unit has shown actual performance to conform very well to the predicted time domain response.

REFERENCES

1. A. E. Ritchie and L. S. Tuomenoksa, "No. 4 ESS: System Objectives and Organization," B.S.T.J., *56,* No. 7 (September 1977), pp. 1023–4.
2. J. E. Abate et al., "The Switched Digital Network Plan," B.S.T.J., *56,* No. 7 (September 1977), pp. 1312–3.
3. J. E. Abate et al., "Synchronization Considerations for the Bell System Switched Digital Network," ICC 1979 Proceedings, Boston, Mass., June 10, 1979.
4. J. H. Huttenhoff et al., "No. 4 ESS: Peripheral Systems," B.S.T.J., *56,* No. 7 (September 1977), pp. 1034–5.
5. B. R. Saltzberg and H. M. Zydney, "Digital Data System: Network Synchronization," B.S.T.J., *54,* No. 5 (May–June 1975), pp. 881–6.