The Microprocessor-Based Synthesizer Controller

H. Donnelly, M. R. Wick, R. W. Weller, G. B. Schaaf, B. Barber, and M. A. Stern
Radio Frequency and Microwave Subsystems Section

The design and implementation plan of a new microprocessor-based controller for the Dana Digiphase Synthesizer is presented. Improvements over the present controller, POCA, are discussed including greater operating capability, increased phase control accuracy and the addition of diagnostics.

I. Introduction

The need to track low level signals with high doppler rates and ranges brought about a demand for Digitally Controlled Oscillators (DCOs) with a programmable frequency capability. Recently, the limited frequency tracking range of one of the Voyager spacecraft receivers underscored the need for this capability on the up-link.

With the advent of microprocessors, significant improvements in control and monitor of the Dana Digiphase Synthesizer were possible with a decrease in the amount of hardware required. The microprocessor provides for additional monitor capability, controller interface processing and functional diagnostics. This new controller incorporates the latest advance in microcomputing technology.

The phase coherent digiphase (Ref. 1) technique of the synthesizer provides a precise frequency source from which the output phase approximates a theoretical ramp when ramping. The synthesizer output is derived from a VCO which is phase locked to a digitally synthesized reference, thus the output phase is predictable and limited only by the inaccuracies of time tagging a ramp start and stop. This Controller performs the measurement and removal of these inaccuracies (or reduces them to a tolerable range of 1 μsec).

II. Description

The Synthesizer Controller, hereafter referred to as the Controller, is designed to control and monitor the Model 7010-S-241 Frequency Synthesizer (Ref. 2) manufactured by Dana Laboratories, Inc. The Controller receives serial data commands from the computer via an industry standard RS-232 command port and outputs parallel BCD frequency commands to the synthesizer. The commands received by the Controller are in ASCII format; they cover a wide variety of Controller functions, including loading a series of frequency ramps, the start time for execution of the ramp sequence editing functions, status reporting and diagnostics.

A sequence of 100 ramps may be stored in the command table. Once loaded, the Controller is ready to execute the ramps in a continuous sequence from any point in the table. This sequence of ramps can be used to approximate a smooth frequency function over time.

A set of specifications for the controller is given in Table I.

III. Controller Development Program

In 1977, development work began under the 310 program on a microprocessor-based controller that would replace the existing Programmed Oscillator Control Assembly (POCA) (Refs. 2 and 3) as controller of the DCO. One engineering model controller was built and successfully tested.

The computing device which controlled this engineering model was a National Semiconductor 16 bit PACE microprocessor. Although the PACE microprocessor was the logical choice at the time, it subsequently was "obsoleted" by the manufacturer.

Though the operational Controller now under development uses much of the hardware and software design of the engineering model developed under the 310 program, the PACE and related circuitry has been replaced with the Intel single-board microcomputer ISBC 86/12.

At the time of this report, the Controller design has been completed, a prototype has been assembled, and functional tests initiated. Test results will be reported in the next DSN Progress Report.

IV. Phase Error Correction Schemes

The Controller development activity focused on the removal of the sources of accumulated phase errors. "Phase error" in the context of this report is defined as a deviation from the equation representing the ideal phase total of a sequence of ramps:

$$\phi(T) = \sum_{K=1}^{N_R} 1/2\, M_K\, T_K^2 + F_K T_K$$

where

$$T \overset{\Delta}{=} \sum_{K=1}^{N_R} T_K \text{, the sum of the series of ramp durations}$$

$N_R \overset{\Delta}{=}$ the total number of ramps in the sequence

$K \overset{\Delta}{=}$ the number designating a single ramp

$M_K \overset{\Delta}{=}$ the slope of the Kth ramp

$F_K \overset{\Delta}{=}$ the initial frequency of the Kth ramp

There are three principal sources of phase error in the DCO.

A. Synthesizer-Delay Induced Phase Error

This source of error is caused by the time delay between the clock pulses in the Dana Digiphase Synthesizer and the Controller. Although both devices operate from the same reference frequency, the Controller is synchronized to real time and the synthesizer is not. To correct this source of error, the 1 PPS signal from the real time reference will be used to synchronize the synthesizer to real time.

B. Phase Error Due to Truncation of the Ramp Rate

This contribution to the phase error is the result of truncation in the ramp rate computation. After the Controller receives a ramp command of a frequency change (Δf) over a time interval (Δt), it computes the ramp rate. This error, due to ramp rate truncation, is reduced by using a sufficient number of digits in performing the ramping summation. The required number of significant digits in the ramp rate computation is discussed in the following.

Let M_I be the ideal ramp rate and M_T be the truncated ramp rate. The phase error due to ramp rate truncation is:

$$\phi_{RT} = 1/2\, M_I \cdot T^2 - 1/2\, M_T \cdot T^2$$

$$= 1/2\,(M_I - M_T) \cdot T^2$$

where

ϕ_{RT} is the cumulative phase error at time T due to ramp rate truncation.

The Controller with its 22 digits of precision obtains:

$$\frac{\phi_{RT}}{8\text{ hrs}} = 1/2 \cdot (10^{-21}\text{ cycles}) \cdot (2.88 \cdot 10^9 \cdot \Delta T)/8\text{ hrs}$$

$$= \frac{1.44 \cdot 10^{-12}\text{ cycles}}{8\text{ hrs}}$$

–a figure significantly better than the controller specification requirement of 10^{-4} cycles/8 hrs.

C. Phase Error Due to the Staircase Construction of Ramps

This source of phase error is caused by the inherent delay due to the stepwise change of frequency commands to the synthesizer. The synthesizer output frequency produces a ramp that follows the staircase control input. This effect is illustrated in Fig. 1.

The cumulative phase error may be considered graphically as the area bounded in the illustration by the ideal ramp slope, the uncorrected ramp pattern, and the line representing the present moment in time.

This error may be expressed as the difference between the cumulative phase of the ideal ramp and the cumulative phase of the staircase ramp:

$$\phi_{SE} = 1/2 \cdot N_T \cdot \Delta F \cdot T - \sum_{N=1}^{N_T-1} \Delta F \cdot \Delta T \cdot N$$

where

$\phi_{SE} \stackrel{\Delta}{=}$ the phase error due to the staircase effect

$N \stackrel{\Delta}{=}$ the number of a single step in a ramp

$N_T \stackrel{\Delta}{=}$ the total number of steps in the ramp

Taking the average value of N over the summation we have:

$$\phi_{SE} = 1/2 \cdot N_T \cdot \Delta F \cdot T - \frac{N_T}{2} \cdot (N_T - 1) \cdot \Delta F \cdot \Delta T$$

Since

$N_T = T/\Delta T$,

$$\phi_{SE} = 1/2 \cdot N_T \cdot \Delta F \cdot T - \frac{N_T - 1}{2} \cdot \Delta F \cdot T$$

$$= 1/2 \cdot N_T \cdot \Delta F \cdot T - \frac{N_T \cdot F \cdot T}{2} + \frac{\Delta F \cdot T}{2}$$

$$\phi_{SE} = + \frac{\Delta F \cdot T}{2}$$

1 Hz/10 μsec is the maximum slope that the Dana Synthesizer can ramp. This results in a staircase ramp phase error of:

$$\phi_{SE} = + \frac{\Delta F \cdot T}{2} = \frac{(1\text{ Hz})(10\ \mu\text{sec})}{2} = +0.5\text{ cycles/sec}$$

The uncorrected staircase ramp has a clearly unacceptable amount of phase error. Two a priori corrections are used to reduce this source of phase error to less than 10^{-4} cycles/8 hrs.

First, a gross correction is accomplished by adding one-half the increment step value to the command frequency prior to the start of a ramp. By this technique, the deviations from the desired theoretical phase value will cancel out and at the completion of a ramp, this half frequency step is subtracted out. Second, the discrete nature of the command frequency prevents the phase deviations from balancing out perfectly; therefore, a fine adjustment is required. This fine adjustment is implemented in hardware and consists of a special register which accumulates the phase error due to the command frequency truncation. When the error reaches a sufficient amount, the least significant digit of the command frequency (10^{-6} Hz) is incremented for one additional cycle (10 μsec). Figure 2 illustrates this phase correction process.

V. Hardware Implementation

Implementation of the Controller logic requires three printed circuit boards: The Intel ISBC 86/12 micro-computer, the "Timing and Interface Logic" (Board No. 1), and the "Frequency Command Logic" (Board No. 2). The functions performed by each board are described in the following sections; a block diagram of the hardware functional elements is shown in Fig. 3.

A. Intel ISBC 86/12 Microcomputer

The ISBC 86/12 Single Board Computer is a complete computer system on a single printed-circuit assembly. The ISBC 86/12 includes a 16-bit central processing unit (CPU), 32K bytes of dynamic RAM, 16K bytes of read only memory, a serial communications interface, three programmable parallel I/O ports, programmable timers, priority interrupt control, Multibus control logic, and bus expansion drivers for interface with other Multibus-compatible expansion boards. For further information on the microcomputer, refer to the Intel ISBC 86/12 reference manuals.

B. Timing and Interface Logic

1. Microcomputer interface. Board No. 1, containing the timing and interface logic, provides the basic interface between the microprocessor, and Boards No. 1 and No. 2. It provides computer bus termination, TTL Gates for address buffering and transceiver elements for bidirectional signal buffering of the microcomputer data lines.

2. Address decode logic. The address decode logic receives the register address from the microcomputer and decodes it into the various enabling signals that force data to be input and output from the internal system registers.

3. Data Port. The Data Port operates in a way similar to the Command Port resident on the microcomputer board. It utilizes the Intel 8251A Programmable Communication Interface IC and receives its clock signals from the programmable counter residing on the Microcomputer Board.

4. Real Time Clock Logic. The Real Time Clock receives a 100 kHz reference frequency from the Dana Synthesizer along with a 1 pulse per second (1 PPS) signal from the external station Frequency and Timing Subsystem (FTS). The 1 PPS signal is synchronized with the 100 kHz reference clock and the reference clock drives the time counters which can be preset to the correct time of day and year.

The Real Time Clock can be preset as follows. One method is to connect the FTS time source and parallel load all digits at once. The second method is to load the time counters in four word transfers from the microcomputer. The internal time clock has a resolution of 10 μsec and a range of 999 days. The source of the actual time used for sequence operations is selectable; it may be either internal or external. If the internal clock is selected, the external FTS will be disregarded and the internal time counter outputs will be presented to the system. If the external FTS signals are lost while the system is in the external time mode, the internal time clock will take over without a significant time delay. The internal clock continuously monitors the external FTS time prior to the changeover, so it is set to station time.

5. Time Tag Logic. A Time Tag has the same resolution and range as the Real Time Clock. It is stored in the Time Tag register along with a frequency control function. The Time Tag is continuously compared to the Real Time Clock and when the time matches, the frequency control ramping function is executed. In this way frequency control operations can be set up at the normal computer transfer rates while being executed with precise time correlation.

The Time Tag logic consists of a storage register for the Time Tag value and a comparator to compare with all the digits of the Real Time Clock. When a match occurs, a signal is generated to interrupt the CPU.

C. Frequency Command Logic

1. Frequency Ramp Control. The frequency ramp control logic receives data from the microcomputer specifying the frequency ramps. It can be instructed to start and stop a ramp, command any fixed frequency, or command positive and negative frequency ramps of any magnitude within the operating range of the Dana Digiphase Synthesizer. The microcomputer writes into three data registers which are accessed by the ramp control logic. One is for the initial frequency value and the other two contain the incremental frequency values. The frequency ramp control logic consists of these data registers, a two-stage serial adder, and storage and control logic. The ramp control sequence is as follows.

The control logic receives a command to load the contents of the initial value register to the output. This operation occurs in the next 10 μsec and the initial value is transferred to the first stage of the output storage. Next, the control logic is commanded to load the incremental frequency into the adder and add this value to the initial value. Simultaneously the initial value is transferred to the record stage of output storage where it is output to the synthesizer. When the add cycle is complete, the new command is residing in the first stage of output storage. Thereafter, every 10 μsec, a new frequency value is produced in the same manner as described above and the values are transferred from stage one of the output storage to the output. This process continues until the microcomputer transmits a stop command.

2. Phase Counter. The phase counter monitors the synthesizer output and reports a cycle count to the microcomputer every 100 msec. The Phase Counter consists of a high-speed synchronized counter capable of running at a maximum rate of 51 MHz.

3. Synthesizer Status. The synthesizer status register consists of three re-triggerable one-shots, each monitoring a specific synthesizer clock signal including 100 kHz, 1 MHz and 5 MHz. If a clock signal is substantially off frequency or stops completely, the respective one-shot will flag the microcomputer. Other signals monitored are "Synthesizer Power On" and a latch that monitors the "Synthesizer No Lock" signal. If phase lock is lost at any time, a "Lock Lost" flag is set in the Controller and is automatically reset when reading the synthesizer status register. Three tell-tale contact closures are also provided. They report "Synthesizer Ready," "Phase In-Lock" and "Synthesizer Power On" available via a connector to an external data monitor.

VI. Firmware Implementation

The firmware in the Controller uses a simple, dedicated, interrupt-driven "Foreground-Background" operating system. Background Tasks are entered into a task queue as a result of messages received from either of two interrupt-driven RS232 ports or from the currently active task. A Task Dispatcher selects the oldest task in the queue for execution. All Background Tasks run to completion and then return control to the Task Dispatcher.

Foreground Tasks are initiated by one of several Background Tasks and interrupt-driven thereafter. Foreground Tasks may be self-terminating or terminated by a Background Task. The Controller firmware is illustrated in Fig. 4 and consists of the following modules:

Control	– contains system initialization logic and the Task Dispatcher
I/O	– performs all I/O with the Controller electronics
RS232	– provides message-oriented I/O with two RS232 ports
Math	– performs 32 digit BCD arithmetic, addition and subtraction in time format (ddd:hh:mm:ss) and conversion from time format to seconds
Report	– formats status reports for output to the RS232 ports
List	– formats listing of the Command List for output to the RS232 ports
Syntax	– processes messages received from the RS232 ports and places calls to the indicated task in the Task Dispatcher
Ramp	– provides two closely related functions – ramp control, which controls the execution of a frequency ramp series, and the Performance Monitor, which detects and reports errors in the operation of the Controller
Diagnostic	– performs diagnostics on the Controller electronics

The Controller includes the firmware necessary to accomplish the following tasks:

(1) Accept data which defines up to 100 frequency ramps, calculate ramp rates, and store the parameters in a "Command List".

(2) Allow modification and display of the Command List.

(3) Output data from the Command List and send commands to the Controller electronics in the sequence required to generate the specified series of ramps.

(4) Monitor operation of the Controller electronics and the synthesizer and report errors and status as required.

(5) Perform diagnostics of the Controller electronics.

The hardware elements used by the firmware are illustrated in Fig. 5:

(1) Synthesizer Output Register – contains a frequency value which is output to the synthesizer every 10 μsec.

(2) Initial Frequency Register – The contents of this register are loaded directly into the Synthesizer Output Register whenever a specific frequency value is required (typically at the beginning of a ramp).

(3) Increment Registers – Two increment registers are provided. While a ramp is executing, the active Increment Register contains the value to be added to the Synthesizer Output Register every 10 μsec in order to generate the desired frequency ramp. The 2nd register is loaded with the increment value for the next ramp and becomes active when a change in ramp rate is required.

(4) Serial Adder – adds the selected Increment Register to the contents of the Synthesizer Output Register and stores the sum in the Synthesizer Output Register.

(5) Controller Clock – maintains time in days, hours, minutes, seconds, down to 10 μsec resolution. The clock is normally driven by an external time source. An interrupt to the processor is generated on every 100 msec clock edge.

(6) Time Tag Register – is loaded in the same format as the Controller clock with the time one of the events listed in item 7 is required to occur.

(7) Time Comparator – generates an output when the values in the Controller clock and the Time Tag Register are identical. This "Time Tag Match" signal generates an interrupt to the processor and also initiates one of the following actions:

   (a) Load the Synthesizer Output Register with the contents of the Initial Frequency Register and stop the Serial Adder.

(b) Load the Synthesizer Output Register with the contents of the Initial Frequency Register, select Increment Register No. 1, and enable the Serial Adder.

(c) Load the Synthesizer Output Register with the contents of the Initial Frequency Register, select Increment Register No. 2, and enable the Serial Adder.

(d) Select Increment Register No. 1 and enable the Serial Adder.

(e) Select Increment Register No. 2 and enable the Serial Adder.

(8) Phase Counter – a 9 digit BCD counter which is incremented once by each cycle generated by the Synthesizer.

VII. Performance Monitor

Two mechanisms used to monitor performance of the system are the Cycle Count Monitor and the Watchdog Timer.

Cycle Count Monitor, a 9 digit BCD counter in the Controller, is incremented once with each cycle output by the synthesizer. The counter is never reset, it "rolls over" to zero. Every 100 msec the contents of the counter are latched into a register which can be read by the processor and the 100 msec interrupt to the processor is generated. When the interrupt is received the processor reads the cycle count value and calculates the absolute difference between the current reading and the reading taken at the previous 100 msec interrupt. This difference is the number of cycles output by the synthesizer in the previous 100 msec interval. The processor adds this value to a 30 digit BCD accumulator to obtain the total number of cycles generated since the counters were reset. Using the Commanded Frequency at the start of the interval and the Commanded Rate, the processor calculates the cycle count expected for the interval. This calculation is carried to 22 fractional digits. This value is then added to an accumulator which contains 30 integer and 22 fractional digits to give the expected cycle count. The integer portion of the expected cycle count is then compared with the generated cycle count. The error bound is +0, −1 since the actual cycle count is derived from a zero crossing detector and gives an uncertainty of one cycle.

The failure of the 100 msec interrupt signal could result in an undetected error condition. Failure of the Time Tag match logic or the Time Tag interrupt could give a similar result since the Performance Monitor operates on the data presented when the Time Tag interrupt occurs. A programmable interval timer supplied on the ISBC 86/12 board is used to monitor the operation of the 100 msec clock interrupt. The device is configured as a programmable one-shot. The one-shot is retriggered by the 100 msec clock interrupt routine. If the 100 msec interrupt fails to appear at the proper time, this Watchdog Timer generates an interrupt to indicate the error condition.

Proper operation of the Time Tag logic is checked by the Performance Monitor, which is driven by the 100 msec clock interrupt. A counter which contains the time interval until the next programmed time tag is maintained in the Performance Monitor. This counter is loaded by the Time Tag interrupt routine and is decremented by the Performance Monitor every 100 msec.

VIII. Diagnostics

A portion of the microcomputer firmware is dedicated to checking the condition of the Controller electronics. Two hardware features are especially useful for troubleshooting purposes.

(1) A means for the processor to read the contents of the Synthesizer Output Register.

(2) A means to single step the Serial Adder. The diagnostic procedures are intended to fullfill two functions, Confidence Tests and Fault isolation.

Confidence Tests are executed during system power-up and on receipt of the appropriate command from the Command Port. They are intended to verify that the Controller electronics are functioning correctly.

Clock tests – The microcomputer performs several read/write tests on the Real Time clock. A "walking one" pattern series followed by a "walking zero" pattern series is used to test each bit used in the clock register. These tests are followed by a test of each decade. This is accomplished by writing the appropriate pattern, allowing the clock to count once, and then reading the clock value.

Time Tag tests – the Time Tag register, time comparator, and Time Tag interrupt logic is tested by writing the same "walking one" and "walking zero" patterns, first to the clock and then to the Time Tag register. Proper operation of the Time Tag interrupt is checked after each pattern write.

Initial Frequency Register tests – this register is tested by writing a pattern to the register, loading the frequency, setting the clock and Time Tag for a match and then reading the Synthesizer Command Register. "Walking" patterns are used to test each bit.

Increment Register tests – the two increment registers are independently tested. After a pattern is written to the register, the serial adder is stepped through one complete cycle. A correct comparison with the Synthesizer Command Register verifies proper operation. Again, walking bit patterns are used.

Fault Isolation Tests provide a means for maintenance personnel to narrow the fault location to a specific area of the electronics. These tests are identical to the tests performed by the Confidence Tests; however, they provide error data and control functions to a terminal connected to the Command Port. The error data consists of patterns indicating the bit pattern expected and the bit pattern generated by the Controller electronics.

References

1. "Dana Model 7010-S-179 Digiphase Frequency Synthesizer," Dana Publication No. 980428-S-179, July 1971.
2. Wick, M.R., "DSN Programmed Oscillator Development," in *The Deep Space Network Progress Report*, Technical Report 32-1526, Vol. VIII, pp. 111-124, Jet Propulsion Laboratory, Pasadena, California, April 15, 1972.
3. Wick, M. R., "DSN Programmed Oscillator," in *The Deep Space Network Progress Report 42-20*, pp. 167-177, Jet Propulsion Laboratory, Pasadena, California, April 15, 1974.

Table 1. Controller specifications

Frequency Control

Range: 40,000,000.000,000 to 50,999,999.999,999 Hz
Resolution: 1 μHz

External Time Source

Range: 999 days, 23 hours, 59 min, 59.999 sec
Resolution: 1 msec

Internal Time

Resolution: 10 μsec

Command Frequency Rates

Minimum ramp period:	100 msec
Maximum ramp rate:	100 kHz/sec
Ramp resolution:	1 μHz per 100 msec ramp

Instantaneous Frequency Error

At f = 45 MHz and a ramp rate of 100 kHz/sec

$$\frac{f\ \text{error}}{f} = 2.22 \times 10^{-7}$$

At lesser ramp rates, $\frac{f\ \text{error}}{f}$ is reduced proportionately.

A plot of this error is shown in Fig. 6.

Phase Error

Maximum accumulated phase error: 10^{-4} cycles/8 hour ramp

Data Storage

Maximum number of ramps: 100 ramps

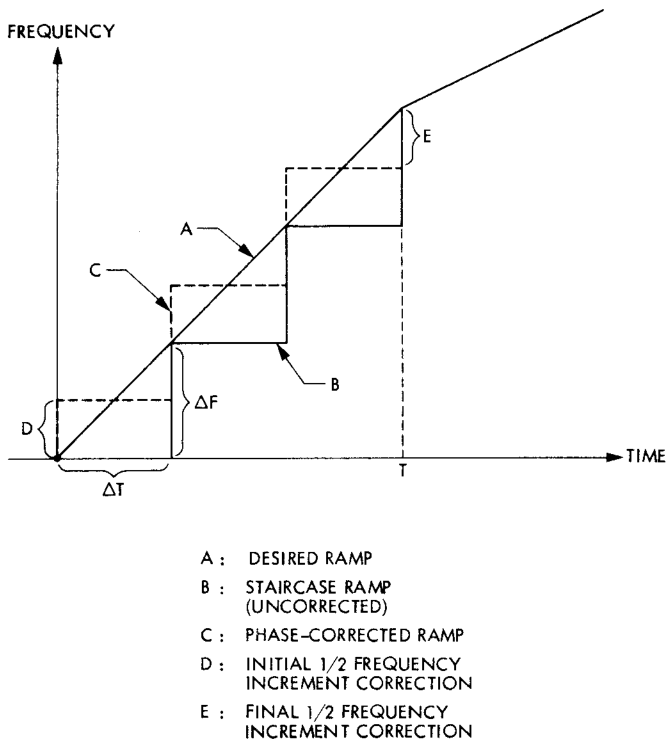

Fig. 1. Frequency ramp construction

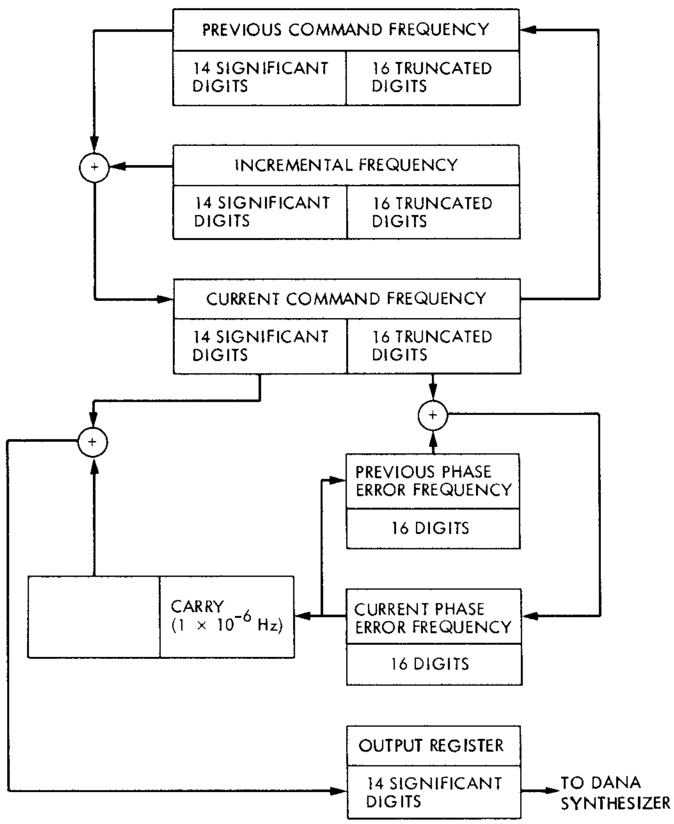

Fig. 2. Phase error correction mechanism

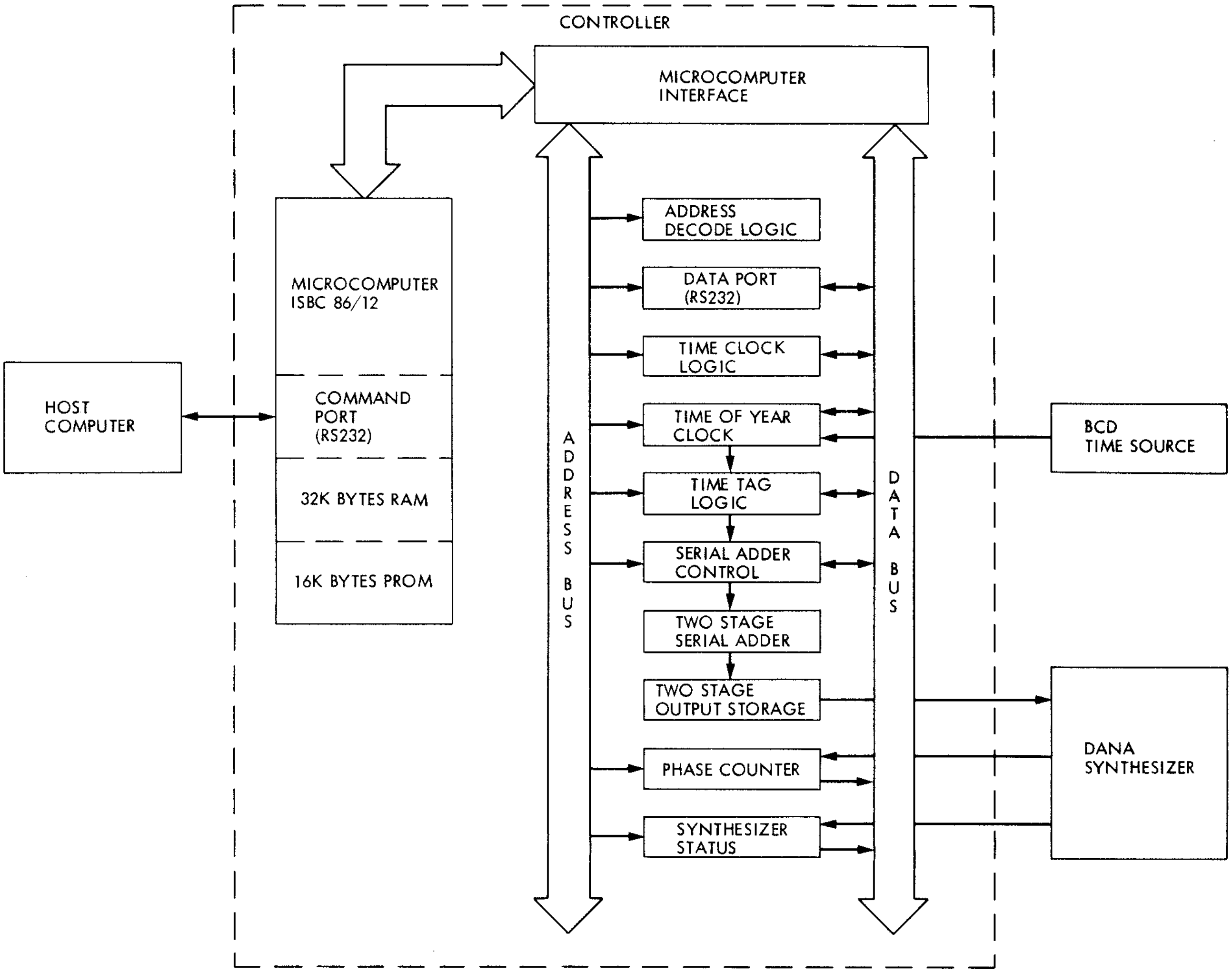

Fig. 3. Hardware block diagram

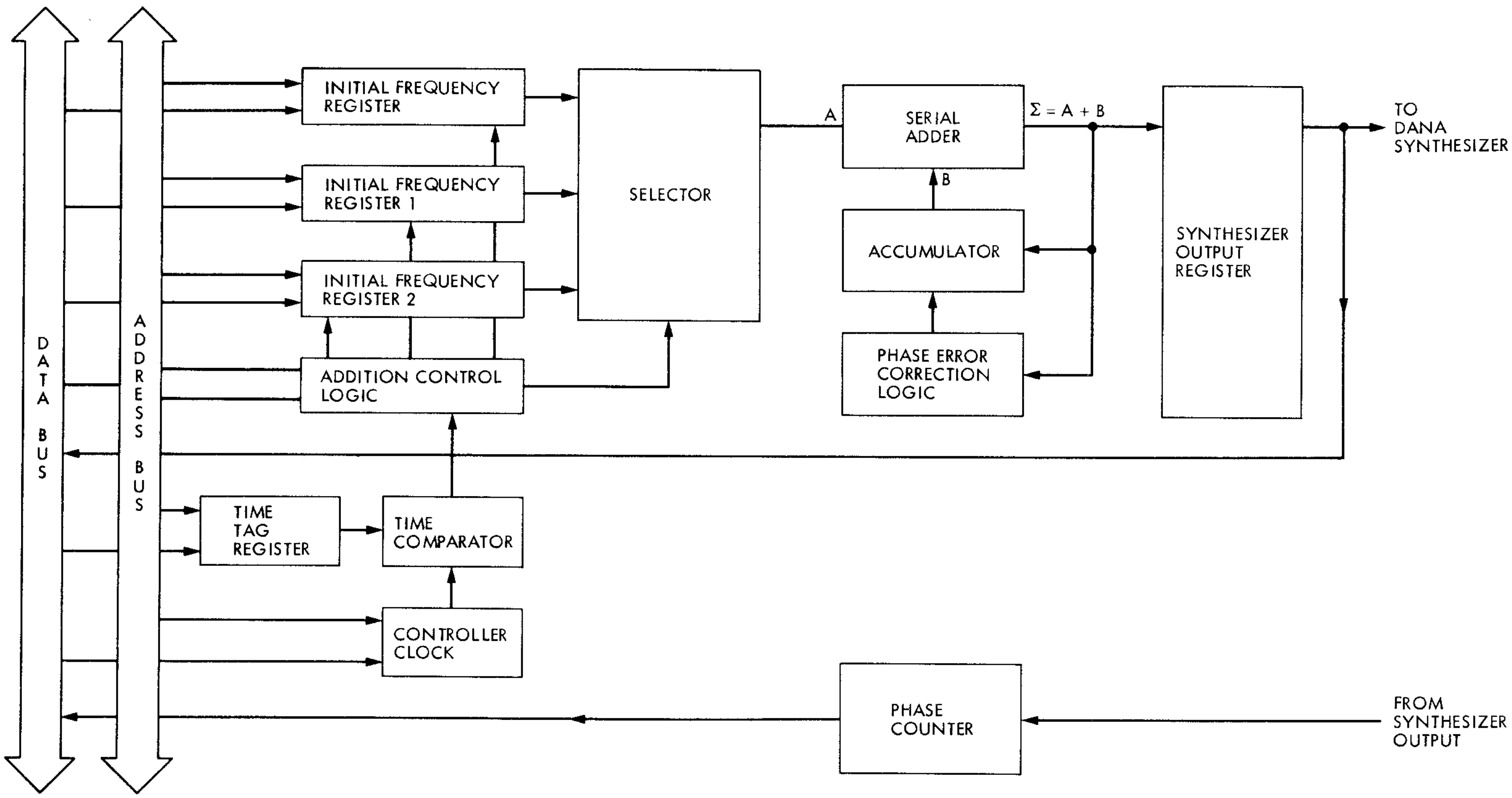

Fig. 4. Hardware elements used by the firmware

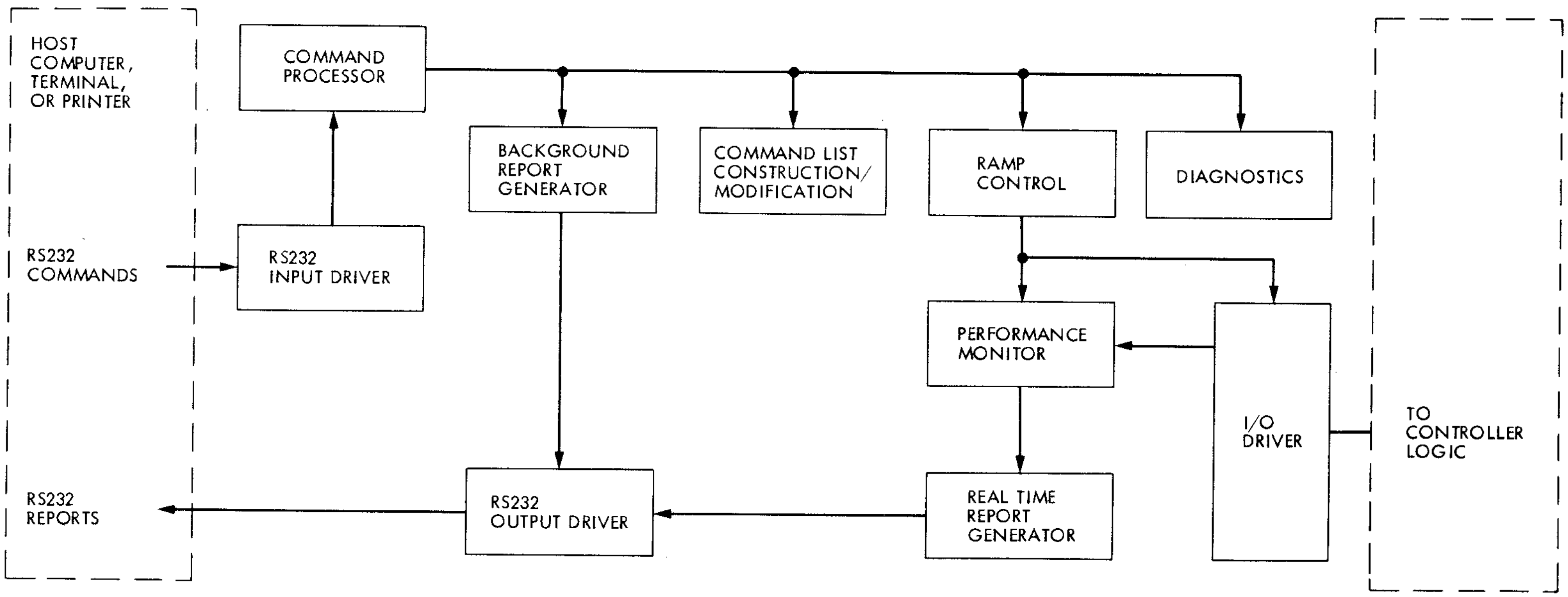

Fig. 5. Firmware block diagram

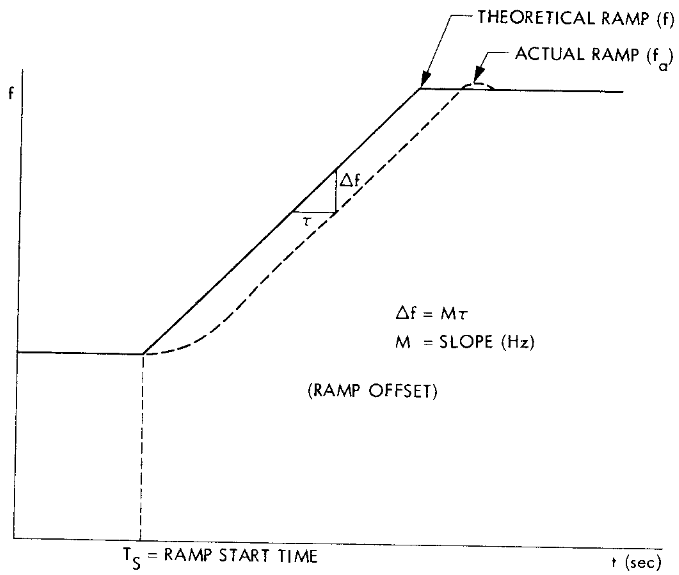

Fig. 6. Instantaneous frequency error