GE-PAC 4020/EAI 580

HYBRID COMPUTING SYSTEM

by

Alan Joseph Bomberger

Thesis submitted to the Graduate Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

Chairman, Dr. A. W. Bennett

Dr. J. F. Phelan

Dr. R. H. Miller

Dr. W. A. Blackwell

March 1971

Blacksburg, Virginia

ACKNOWLEDGMENTS

The author would like to express his appreciation to the members of his advisory committee particularly Dr. A. W. Bennett for his guidance and encouragement in writing this thesis. Also, the contribution of whose patience and diligence in carrying this manuscript from rough draft to finished form in the midst of other demanding work must not go unmentioned.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

WHY HYBRID?

Hybrid computation is a discipline designed to reap benefits from the strengths of both digital and analog computation. The analog computer uses operational amplifiers to perform voltage and current analogs of differential equations. Non-linearities are handled by electronic or servo multipliers and function generators. The operational amplifiers are wired to perform the differential equations, and the coefficients are entered as potentiometer settings. In operation, the problem's initial conditions are impressed on the amplifiers before time zero, and then the computer is switched to run. The problem solution then continues as long as desired. The output is in the form of oscilloscope traces or graphs drawn on a plotter. Those can be either with respect to time or with respect to another state variable. This scheme works well for continuous systems with constant coefficients.

The digital computer uses a program controlled arithmetic unit to solve algebraic equations. The arithmetic unit can add, subtract, multiply, and divide. It also can do logical manipulation, bit and sign tests. The computer is programmed to solve the algebraic equations, and the problem data are entered as constants or other equations. This scheme works well for linear systems. Non-linear systems can be accommodated by assuming piece-wise linearity, and using the computer's logic to go from region to region. Another approach is

use of numerical techniques to perform integrations directly. The output is in the form of columns of figures or graphs typed out on a printer.

The hybrid computer is a wedding of the analog and digital computers. No standard configuration exists but, in general, the digital computer furnishes voltages and control signals to the analog computer and receives voltages and status signals in return. A digital program can be designed to call on the analog computer for solutions to previously programmed differential equations. In this system, the digital computer can furnish initial conditions, time-varying references, and various non-linear functions. The outputs can be taken directly from the analog computer or via the digital computer for formatting or data reduction. In this way, the hybrid runs under program control.

Alternatively, the digital computer can be set up to respond to the mode changes of the analog computer and can respond to a fixed way to the analog system. Here, the digital computer can perform as a data-logger and perhaps as a function generator. The computer run is made with the analog computer acting as the master control.

The hybrid system can be extremely flexible. In situations where numerical integration is time consuming, the analog system can perform the required integrations directly. If transport lag enters into a system, the digital computer can pick up a state variable, store it for the appropriate time delay, and return it to the analog computer for continued processing. In the study of stochastic processes, a random

number generator in the digital computer can furnish noise with the required characteristics to the analog computer. If information on the interaction between fast response loops and slow response loops in some system is desired, the slow loops can be programmed in the digital computer and the fast ones in the analog computer.

For functions of more than one state variable or functions derived by experiment, the digital computer can furnish the data to the analog computer during the run. In the realm of information interchange, the digital computer can set different initial conditions on a series of runs and then plot the desired state variables on the same graph. In the same vein, the digital computer can plot the effect of parameter changes on the same graph, ignoring those with too great or too little change. The list of possibilities can be of any length, as the formulation of one configuration leads to several more possibilities. The real use of the hybrid computer is to illuminate a portion of the real or mathematical world. Its amalgamation of continuous analog computation with discrete digital processing, combined with logical power and memory, eminently qualifies the hybrid for this role.

CHAPTER II

ANALOG, DIGITAL, AND HYBRID COMPUTERS

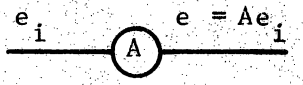
The analog computer solves differential equations by using the electrical properties of resistance and capacitance as mathematical analogs. The computer needs only a few types of computing elements to solve a broad range of problems.[1] The basic functions are listed below:

1. The multiplication of a state variable by a constant coefficient, either positive or negative.
2. The generation of the sum of two or more state variables.
3. The generation of the time integral of a state variable.
4. The generation of the product of two state variables.
5. The generation of functions of a state variable.

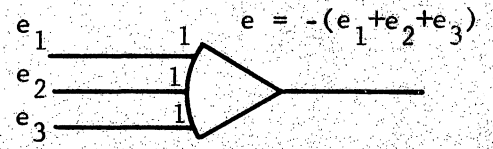
Figure 2-1 shows the basic analog computer elements.

The multiplication by a constant consists of a potentiometer that drops the analog voltage by a specified amount. The sum of two or more state variables is generated by summing them into a d-c amplifier shunted with resistive feedback. The output of the amplifier is then the negative of the sum of the inputs. The time integral of a variable is emulated by a d-c amplifier shunted with capacitive feedback. In machine terms, the output of the amplifier is the negative of the time-integral function. Multiplication is generated by a square law device or a servo-driven potentiometer.

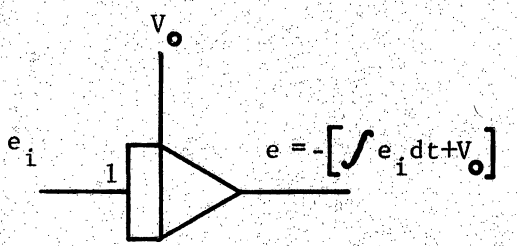
1. Albert S. Jackson, Analog Computation, p. 35.



1. Multiplication by a constant

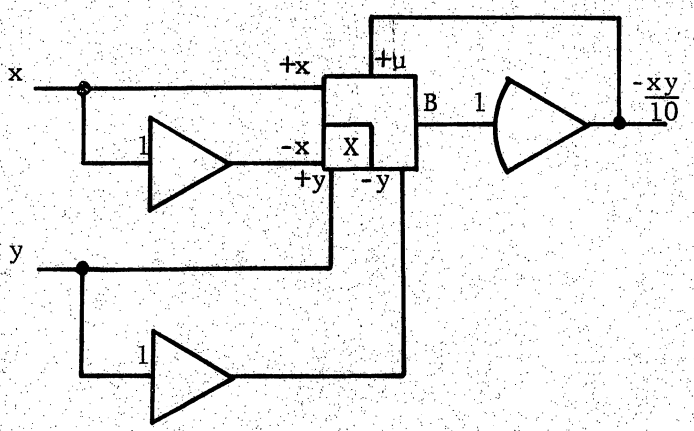


2. Sum of variables



3. Time Integral

4. Inversion



5. Multiplication

FIGURE 2-1. Basic Analog Computer Elements

The basic elements can be wired together to realize a very broad range of differential equations. The basic elements in concert with diodes, resistors, and analog switches or relays can be used to generate a broad class of functions of a state variable. These generators are fairly efficient for functions of a single state variable, but functions of two or more variables are not very straightforward. Unfortunately, complicated functions can take a great deal of hardware to achieve.

The analog computer provides an efficient method of solving ordinary linear differential equations. Consider as an example Legendre's equation.[2]

$$(1 - t^2)\frac{d^2y}{dt^2} - 2t\frac{dy}{dt} + n(n + 1)\, y = 0$$

Let $x = 1 - t^2$

$\dot{x} = -2t$

$\ddot{x} = -2$

An analog computer diagram for Legendre's equation is given in Figure 2-2.

Three control modes are required to run a problem on an analog computer. The first is the initial condition, IC mode. In this mode, the inputs to the integrators are opened up and their outputs are forced to the voltages specified by the initial conditions. In the OPERATE mode, the integrator inputs are hooked up and the problem pro-

2. Ibid., p. 154.

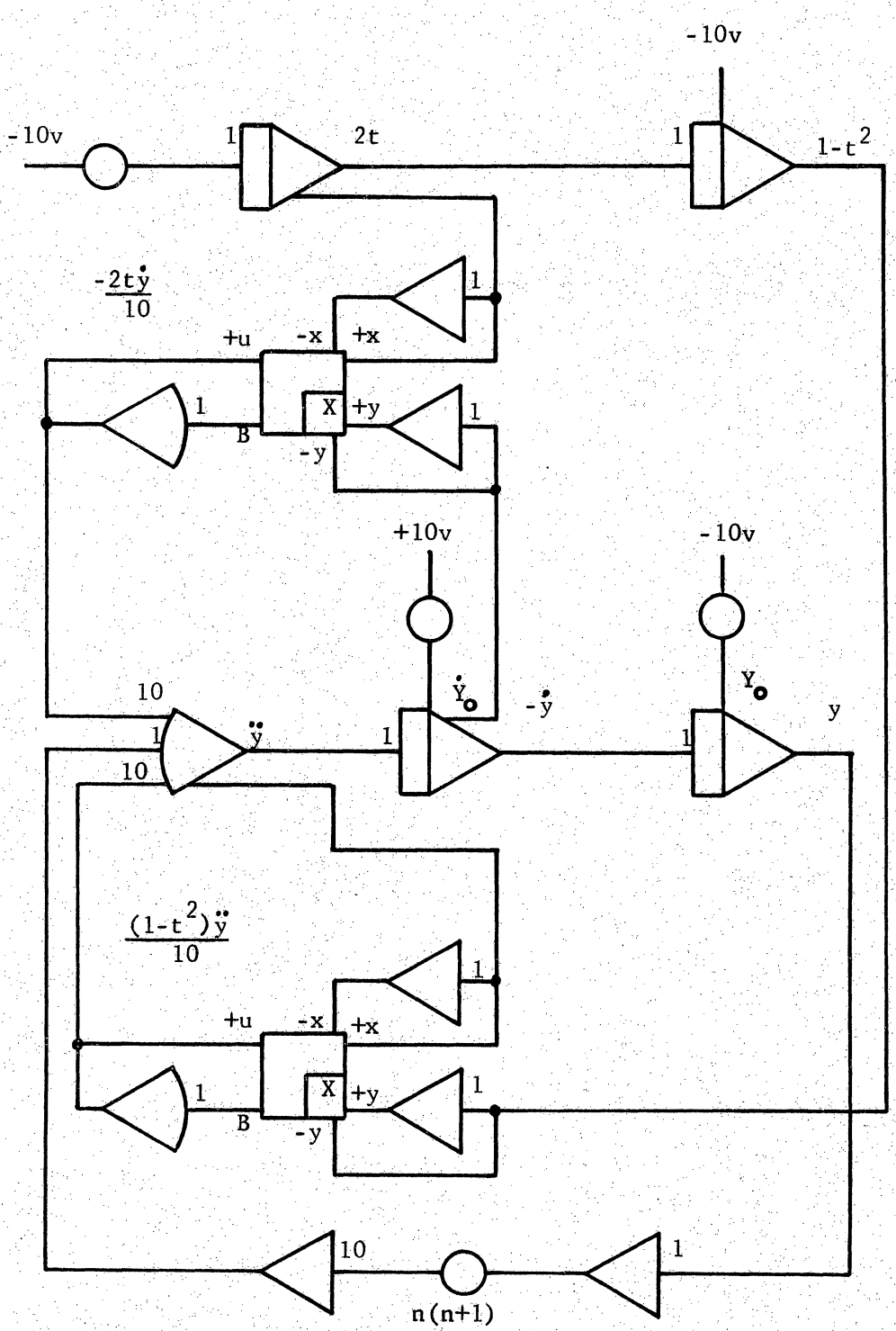


FIGURE 2-2. Analog Computer Diagram of Legendre's Equation

ceeds. Lastly, the HOLD mode halts the computation by opening up the inputs to the integrators. Here, the values of the state variables are held static for checking or recording. If the computer is switched back to OPERATE, the problem continues. Auxiliary modes are also present to provide the following functions. In the PP, program panel mode, an interval timer is provided to switch the computer from OPERATE to IC to OPERATE at a preset rate. This feature provides the capability of running a problem over and over for oscilloscope display. A set pots mode, SP, puts voltage on the addressed potentiometer. This is for the initial specification of the problem, and a static test mode, ST, allows the current into the selected summing junction to be monitored.

The typical time scale for the integrators is one volt per second per input volt. Time scale changes are provided for 10:1, 500:1, and 5000:1 speed-up of the basic rate. Such speed-ups in program running permit many solutions to occur each second for oscilloscope display. The internal timer is proportioned to the speed-up, so that the same portion of the solution is visible in each speed band. Many options are available on various analog computers. The most useful for hybrid work is a set of servo driven potentiometers. This feature allows remote potentiometer setting from an external source.

Where most analog computers are very much alike, digital computers are very different from one another. Even so, there is a basic organizational similarity between them. The basic digital computer consists of a memory, containing instructions and data, an arithmetic unit

to do calculations and logic, an input-output structure for communication with the outside world, and a control unit for internal scheduling of operations. Figure 2-3 shows a block diagram of a typical computer organization.

The digital computer has an internal set of instructions called machine language. When these words come into the instruction register from memory, they define the computer's operation on that cycle such as add, compare, store, etc. Since this is all the digital computer can do, it is clear that any problem must finally end up in machine language. Higher level programming languages have been developed such as FORTRAN; these are known as user level languages. A compiler program translates the user level program into machine language. The biggest advantage of a user level language is that it can stay relatively standard, and the compiler can tailor the user program to the specific machine.

An experimenter who is trying to use a computer in a scientific endeavor should not have to master the idiosyncrasies of computer hardware. Unfortunately, in hybrid computation, not too much work has been done at the user level.

The hybrid computer consists of an analog and a digital computer wedded by an interface unit. The role of the interface unit is to provide a pathway for the exchange of information between the two computers. The analog computer's information must be digitalized for use in the digital computer; digital information must be converted to analog for presentation to the analog computer. The interface unit

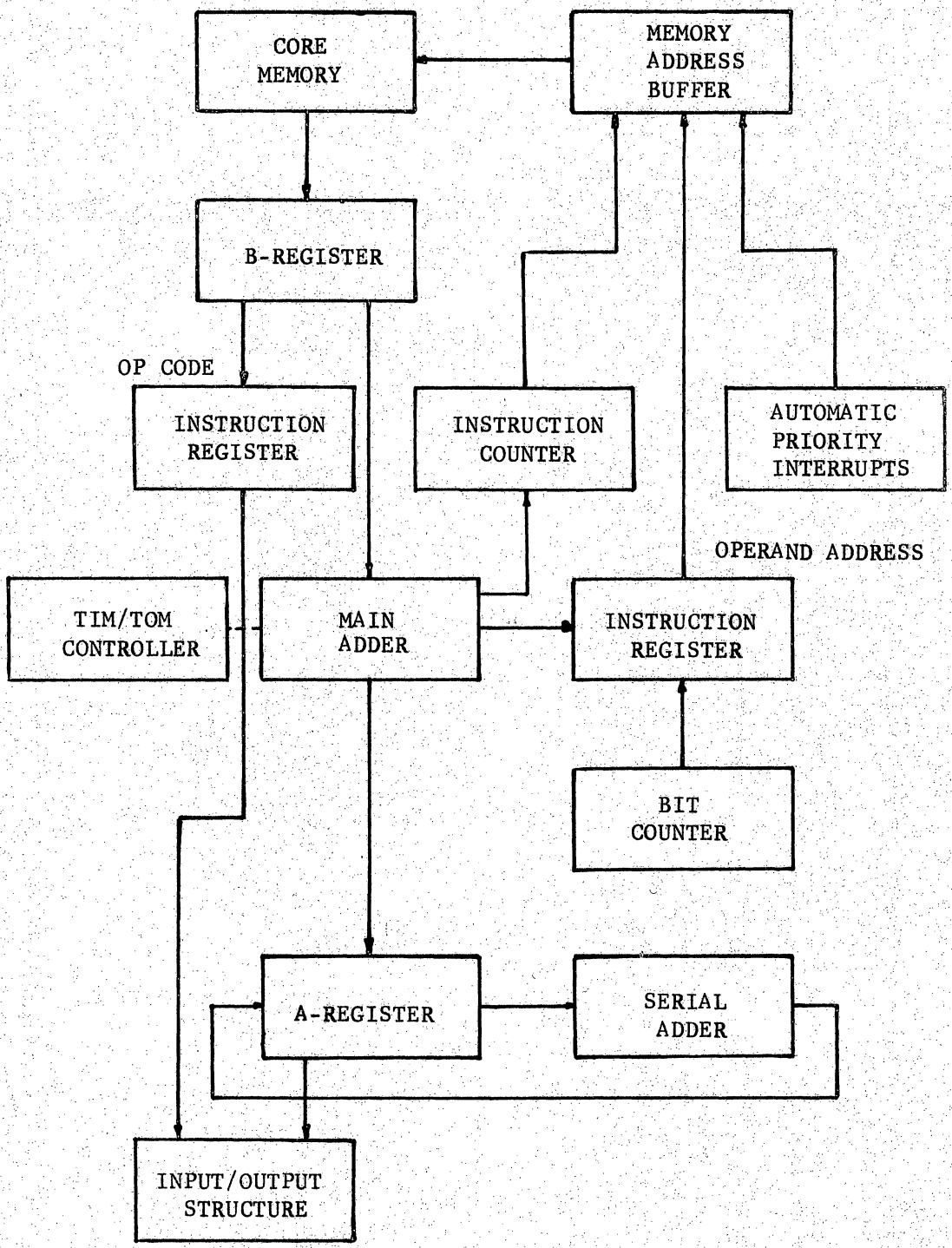


FIGURE 2-3. Typical Digital Computer Block Diagram

provides this basic function. Also important is an exchange of logic information. The digital computer needs to know or alternatively to set the analog computer's mode in order to log a set of problem solutions or to direct the running of a problem. Some analog computers have servo set potentiometers. In this case, the interface unit can provide a means for the digital computer to set them.

The analog computer will also have relays or a digital logic panel. For hybrid work, it is very desirable that the digital computer be able to interact with this equipment. This will allow the digital computer to make program changes in the midst of a problem solution for switching between continuous regions or other parameter changes. Alternatively, the analog computer should be able to call upon the digital computer for some work it needs in the midst of a computation. The role of the interface unit is to allow the analog computer to act as a peripheral to the digital computer, and, if desired, the digital computer can act as a peripheral to the analog computer. Figure 2-4 shows a simplified block diagram of a hybrid system.

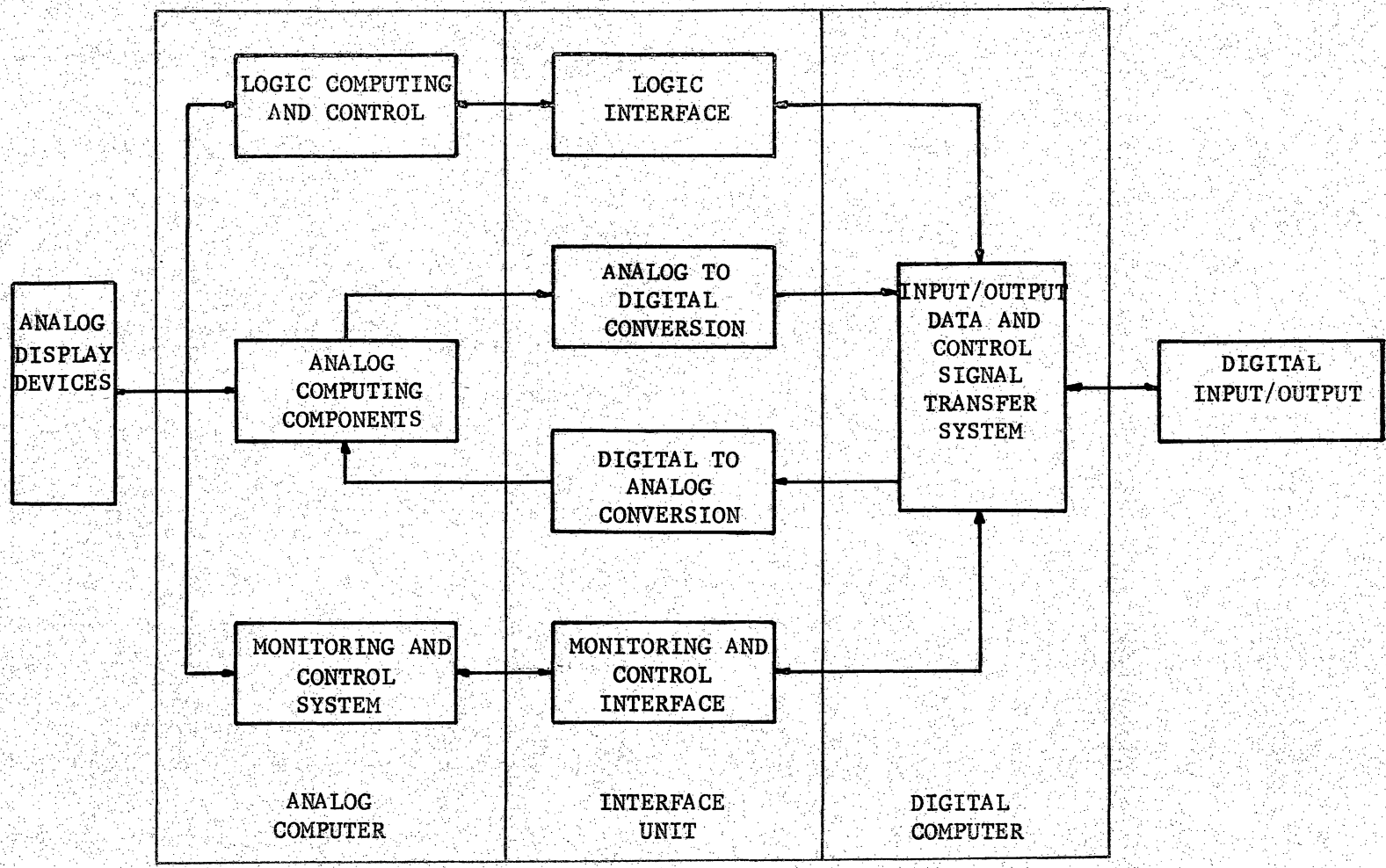


FIGURE 2-4. Simplified Block Diagram of a Hybrid Computer

CHAPTER III

TYPES OF PROBLEMS

Obviously, the hybrid computer can do anything either an analog or a digital computer can. It is also true that most classes of problems can be run on either type of computer. The value of the hybrid rests where, for one reason or another, some portions of the problem can be run better or more easily in one form of computation or the other.

For example, transport lag or time delay.

$$F(s) = e^{-Ts}$$

A Taylor series expansion of this function

$$e^{-Ts} = 1 - Ts + \frac{(Ts)^2}{2!} - \frac{(Ts)^3}{3!} + \cdots + \frac{(-Ts)^n}{n!}$$

could be programmed, but it converges slowly for large Ts. The Padé approximation can be used, but requires a fair number of amplifiers for a good approximation. [1]

$$e^x = \lim_{(u+v)\to\infty} \frac{F_{u,v}(x)}{G_{u,v}(x)}$$

where

$$F_{u,v}(x) = 1 + \frac{vx}{(u+v)1!} + \frac{v(v-1)x^2}{(u+v)(u+v-1)2!} + \cdots + \frac{v(v-1)\cdots 2\cdot x^2}{(u+v)(u+v-1)\cdots(u+1)v!}$$

1. Clarence L. Johnson, Analog Computing Techniques, pp. 128-131.

$$G_{u,v}(x) = 1 + \frac{ux}{(v + x)1!} + \frac{u(u - 1)x^2}{(v + u)(v + u - 1)2!}$$

$$+ \cdots + (-1)^u \frac{u(u - 1) \cdots 2 \cdot x^u}{(v + u)(v + u - 1) \cdots (v + 1)u!}$$

The Padé approximation of e^{-sT} for $u = v = 4$

is

$$e^{-sT} = \frac{(sT)^4 - 20(sT)^3 + 180(sT)^2 - 840sT + 1680}{(sT)^4 + 20(sT)^3 + 180(sT)^2 + 840sT + 1680}$$

The computer wiring diagram for this approximation is shown in Figure 3-1.

A relatively easy to use approximation is [2].

$$e^{-Ts} \cong \exp\left[- 4s \sum_{i=1}^{n} \gamma_i a_i \right] = \prod_{i=1}^{n} \frac{a_i s^2 - 2\gamma_i a_i s + 1}{a_i s^2 + 2\gamma_i a_i s + 1}$$

But this does result in a phase error of

$$\Delta \phi = 2 \sum_{i=1}^{n} \tan^{-1} \frac{2\gamma_i a_i \omega}{1 - a_i^2 \omega^2} - T$$

Even though this formulation uses less amplifiers than the Padé approximation, it is difficult to wire and scale.

2. Albert S. Jackson, Op. cit., p. 256.

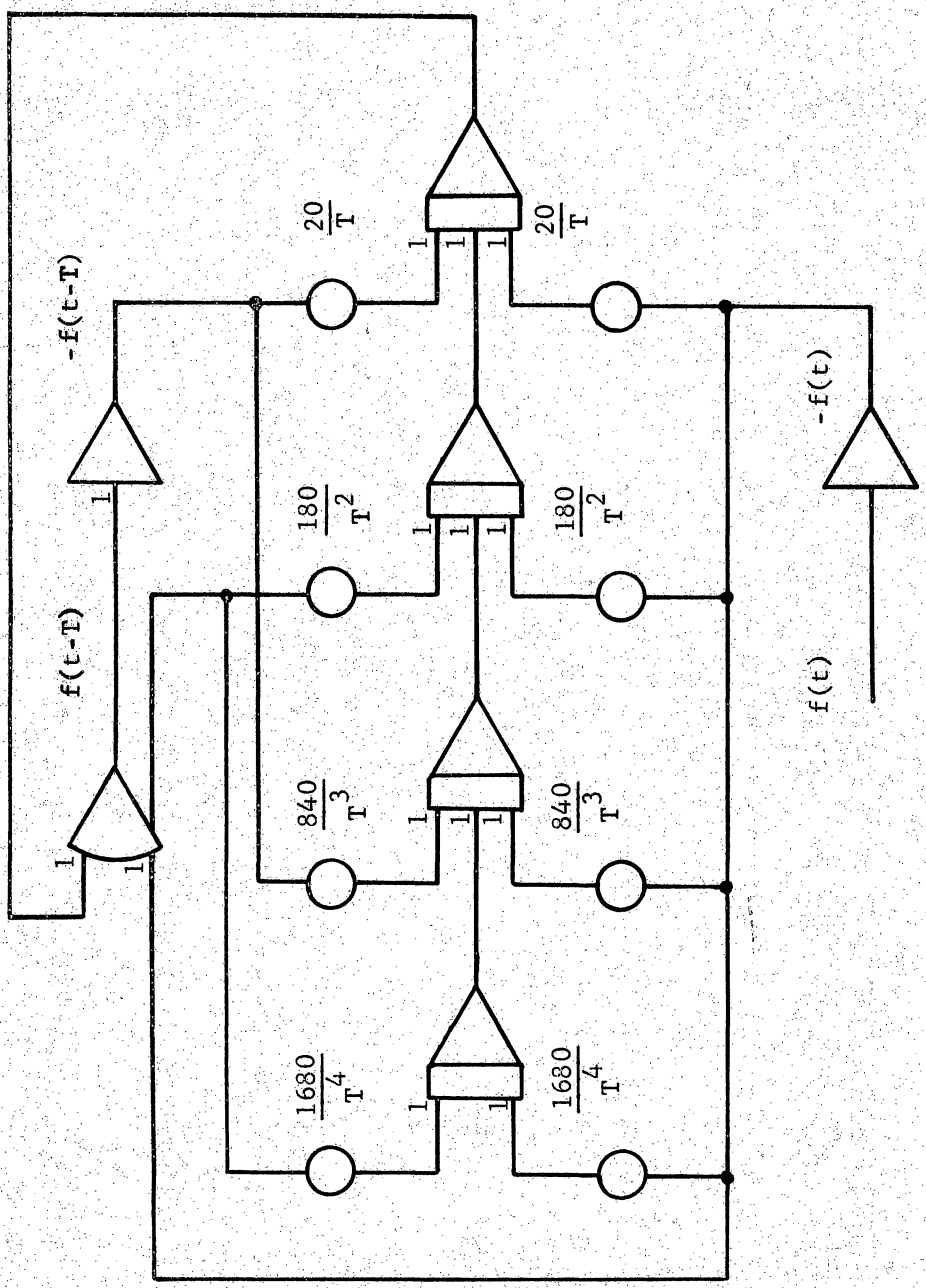


FIGURE 3-1. Generation of The Fourth Order Padé Approximation of a True Time Delay

The hybrid solution to the time lag does not require any amplifiers. To achieve the time delay, the state variable, f(t), is digitalized and stored in the digital computer's memory. After the time T, the value f(t) is converted back to analog and entered appropriately into the problem solution. The mechanization of this function would be a subroutine operating on at least twice the highest frequency of interest, say, every 33 milliseconds for a frequency of 15 radians/sec. The subroutine would call for a conversion of f(t) and look to see if T had elapsed for a previous conversion. If so, that number, f(t - T) would be fed to the analog computer. The minimum T would be $2\Delta T$ where ΔT = 33 milliseconds. The computer memory requirement, n, is then $n = T/\Delta T$. Thirty words of computer memory would suffice for a 1 second delay. Shorter time delays or higher frequencies can be tolerated by driving the subroutine by automatic priority interrupts, if available; in this way, a ΔT of 750 microseconds could be achieved.

Another area in which a digital computer can contribute to a problem solution is in the realm of stochastic systems. In this case, an analog plant can be modeled, and stochastic noise inserted appropriately via the digital computer's D/A channels. This approach can then be extended to the simulation of digital controllers of these types of systems by programming Kalman filter algorithms, and inserting them in the loops. In general, the hybrid computer is a powerful tool in the study of any system with a digital controller. Since the real world is analog, the plant can be modeled on the analog computer; then the loop

can be closed through the digital computer.

In certain types of problems, it is of interest to run a problem with a set of initial conditions. To achieve this, the digital computer is programmed to sequentially set the various initial conditions and then to run the problem. In this way, the analog computer will generate a family of curves which can be superimposed on an oscilloscope or a plotter. Figure 3-2 diagrams this approach. This technique could be very useful in problems where superposition does not hold.

Another useful feature of the hybrid is its ability to easily calculate algebraic functions of the state variables and re-insert them into the running analog problem. For example, take Legendre's equation discussed in Chapter 2. The functions of $\dot{y}$ and y can be calculated digitally and applied at the appropriate points in the system. Also, sets of $\dot{Y}_o$, Y_o, and n(n + 1) can be sequentially programmed to generate a family of solutions. Figure 3-3 is a hybrid block diagram of this problem.

The output of D/A 1 is the variable $\ddot{y}$.

$$\ddot{y} = (2*t*\dot{y} - n*(n + 1)*y) / (1 + t*t)$$

The independent variable t is generated by the digital computer's real time clock. For this formulation, only two analog integrators are needed to solve the problem, and only one digital equation need be solved. Looping to generate a set of $\dot{Y}_o$, Y_o, and n(n + 1) can be used to generate a family of solution curves.

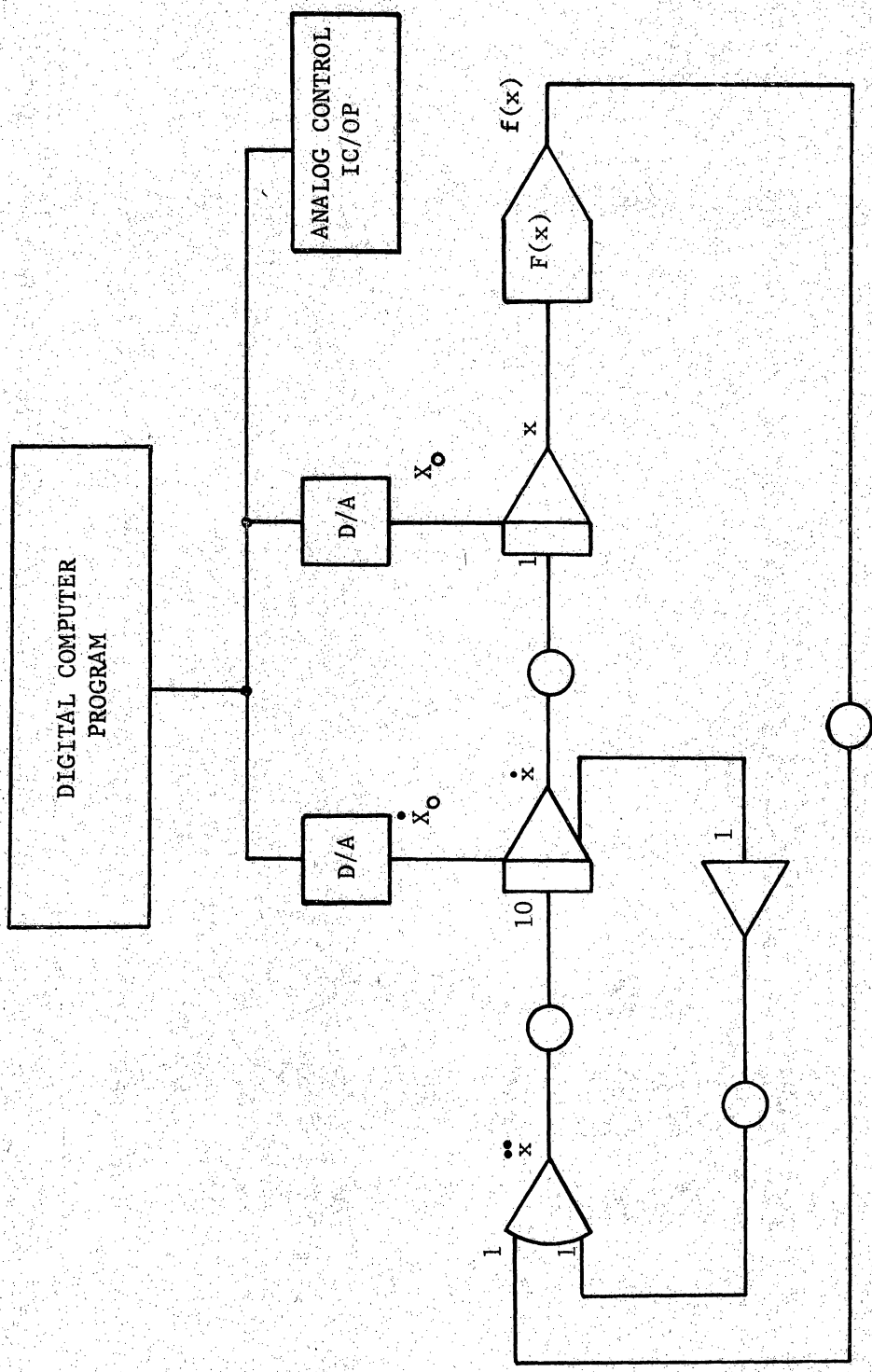


FIGURE 3-2. Digital Computer Control of Initial Conditions

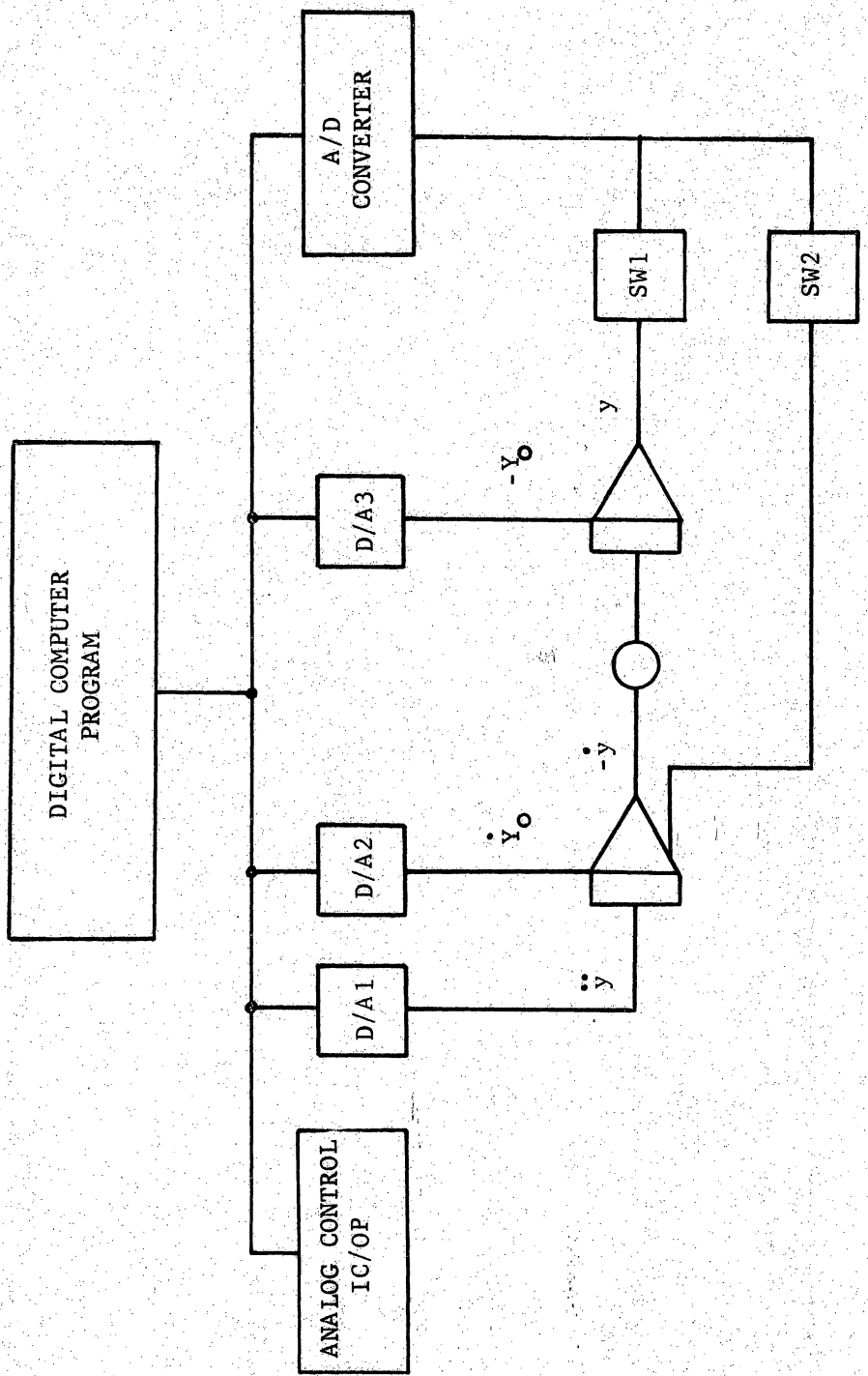


FIGURE 3-3. Hybrid Diagram of Legendre's Equation

CHAPTER IV

ANALOG TO DIGITAL CONVERSION

Analog to digital conversion is the means of quantizing continuous voltage information available in the analog computer into a form suitable for digital computation. The performance of this conversion and its converse, digital to analog conversion, is the main limiting factor in the performance of a hybrid computer system. There are two main considerations for the A to D conversion system in a hybrid computer, speed and accuracy. But since this is a sampled data system, the two are inexorably intertwined.

The conversion speed requirement for hybrid systems can be calculated by considering the largest frequency component of the analog system and the number of state variables to be read in the course of a problem solution. The fastest time constant of an integrator is .2 milliseconds. In a second order solution, this time constant can result in an undamped natural frequency of 5000 radian/second or 796 Hz. To sample this frequency at the Nyquist rate, the A to D converter must operate at 1590 Hz. In other words, the input signal must be sampled once each 629 microseconds. With a converter capable of operating in 50 microseconds, only twelve state variables could be accommodated.

The effect of frequency components higher than half the sampling frequency is that these components are folded into the lower frequency information where they cannot be extracted. Therefore, it is very important that analog problem solutions be scaled such that these high-

er frequencies do not occur in the sampled variables. Therefore, the A to D conversion system in a hybrid computer must be fast to accommodate large numbers of state variables or high frequency components in the problem solution.

The Nyquist sampling theorem assumes that the sample is taken in an infinitesimally short time. This is not the case where the converter operates throughout the sample period. The rate of change of input signal for an error of one least significant digit is equal to the full scale of input times the resolution divided by the conversion time.[1] For a plus/minus 10 volt, 10 bit A to D converter operating in 629 microseconds, the maximum allowable voltage change that does not disturb the least significant digit is

$$\frac{(20)\ (1/2000)}{0.629 \times 10^{-3}} = 15.9 \text{ volts/sec.}$$

For a full scale sinusoidal signal, this corresponds to a frequency.

$$f = \frac{15.9}{2\pi\ (10)} = 0.253 \text{ Hz}$$

This problem occurs because the signal is observed for the full sample time. The situation can be eased by placing a read and hold circuit in front of the A to D converter.

A read and hold circuit very quickly changes up to the input voltage, and then holds that value over a long period determined by its leakage characteristics. This more closely approximates an infinitesimal sample time especially since the read and hold circuit holds the

1. Albert S. Jackson, Op. cit., p. 558.

true value of the input signal at the time that the read and hold gate is disabled. An analysis of the number of samples necessary to reconstruct the continuous function is available in the literature.[2.]

There are three typical types of A to D converters, continuous counting, successive approximation, and voltage to frequency. The first two of these generate an analog signal which is summed with the unknown signal. The operation of the converter then brings the error between the unknown signal and the converter's count to zero. When the error goes to zero, the conversion process is complete.

In the continuous counting converter, the digital number in a counter is converted to analog and summed with the unknown input voltage. Figure 4-1 shows a block diagram of this converter. The counter is then energized in the direction to minimize the error between the counter output and the unknown input voltage. This converter is very appropriate for following a single relatively slowly changing input signal. For multiplexed applications, though, the converter may have to count from full positive or full negative all the way to full negative or positive. For a 10 bit natural binary converter, this is 2048 counts. At a rate of 20 kHz, the conversion will take $\frac{2048 \text{ counts}}{20{,}000 \text{ counts/sec.}} = 0.1$ sec. For hybrid applications, this is an unacceptably long time.

The successive approximation converter searches for the unknown input voltage by testing the analog value of binary weighted bits

2. Beckey and Karplus, Hybrid Computation, pp. 129-135.

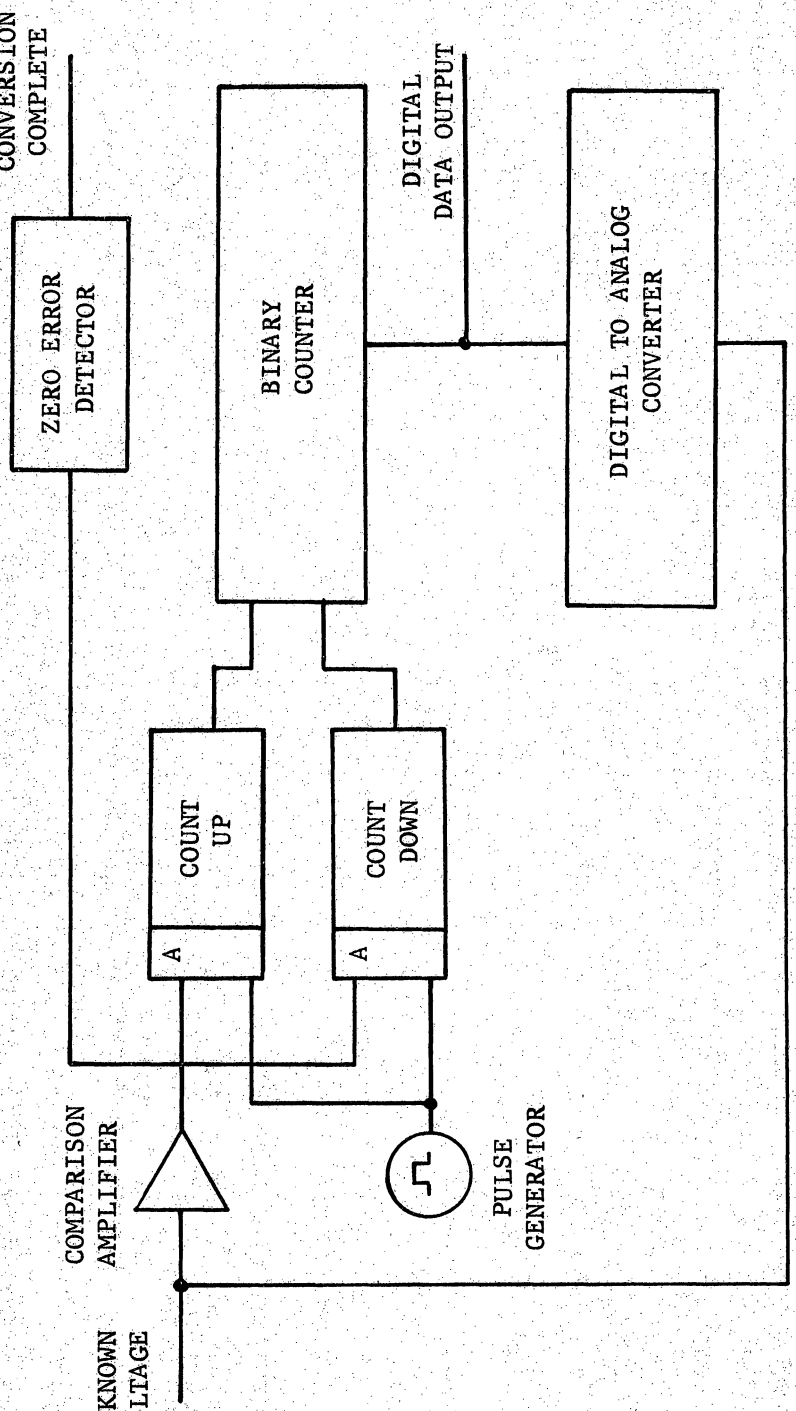


FIGURE 4-1. Continuous Counting Analog to Digital Converter

against the unknown voltage, proceeding from the most significant to the least significant. Figure 4-2 diagrams this approach. First, the positive full scale value is summed with the input. If the error is negative, this amount of signal is maintained in the summing junction, and a negative sign bit is inserted in the answer register. Next, the most significant binary bit is tested; if the error is negative, the unknown voltage is smaller than this, and that bit is not inserted into the answer register. The unknown voltage is then tested against all subsequent bits. If the unknown voltage is greater than the developing digital number, that bit is inserted into the answer register, and its signal is maintained in the summing junction. When all bits have been tested, the conversion is complete. Since the full magnitude is tested first, negative inputs are referenced to that point and are rendered as 2's complement numbers.

For a 10 bit natural binary converter operating at 20 kHz, the conversion will take $\frac{10 \text{ counts}}{20{,}000 \text{ counts/sec.}}$ = 500 microseconds. A converter of this type is acceptably fast for hybrid applications. Integrated circuit hardware exists which can improve these speeds by an order of magnitude. Using these speeds, a 10 bit continuous counting converter can operate in 10 milliseconds, and a successive approximation converter of the same precision can operate in 50 microseconds.

The third type of A to D converter develops a frequency which is proportional to the unknown voltage. Figure 4-3 shows a block diagram of this converter. A counter then samples the frequency for a known period. After the sample period, the count in the counter is the

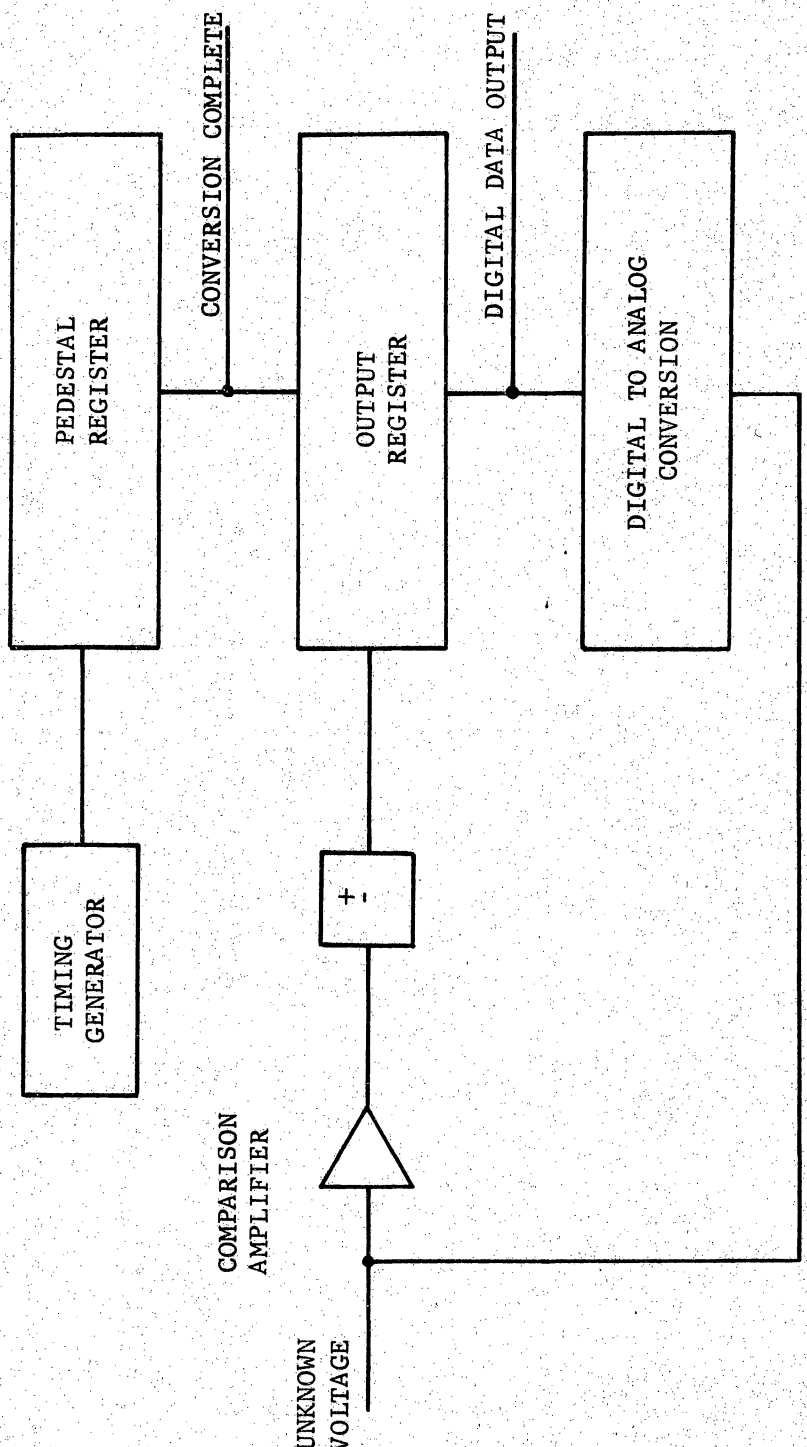


FIGURE 4-2. Successive Approximation Analog to Digital Converter

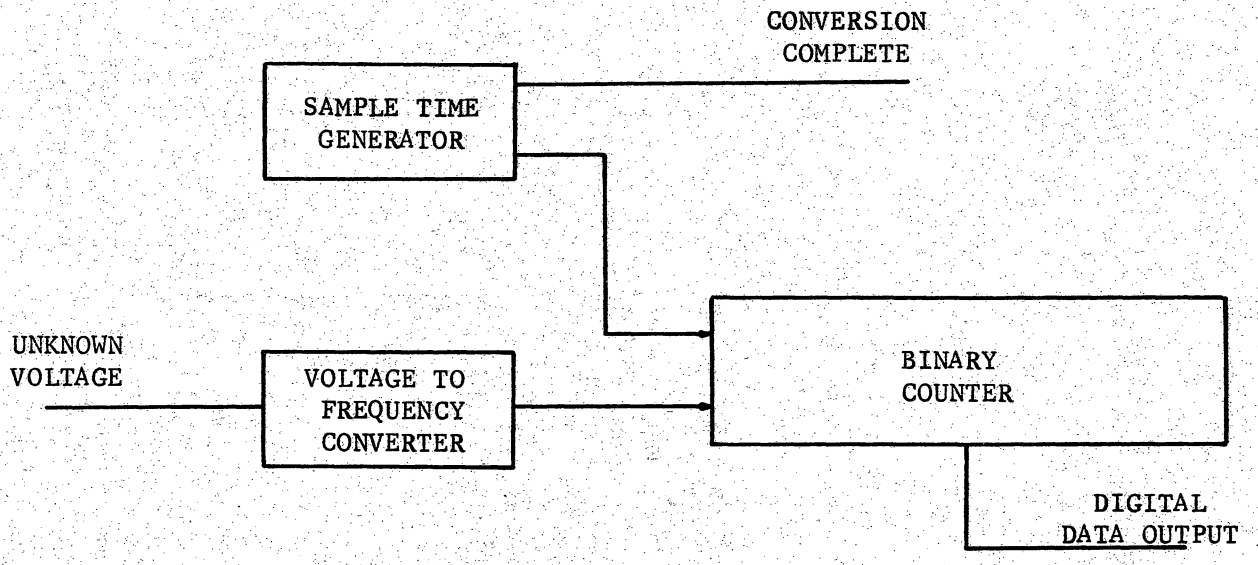


FIGURE 4-3. Analog to Pulse Rate Analog to Digital Converter

digital equivalent of the unknown voltage. A major strength of this procedure is its independence of common mode errors. If the sampling period is an integral multiple of the common mode frequency, the effects of the common mode cancel out over each successive cycle. This cancellation is particularly useful if the major contaminant of the unknown signal is the power line frequency, since the sampling period can easily be synchronized with that frequency.

Another strength of this conversion system is that it can be used to integrate any shape signal with respect to time.[3] As the input signal changes, the converter outputs a frequency proportional to the signal's height. By counting over the period of interest, the total number of counts is proportional to the area under the curve. Unfortunately, the conversion times for high precision readings can be rather long. For a converter whose output is 100 kHz at a full scale 10 volt input signal, a one part in a thousand voltage determination will take $\frac{1000 \text{ counts}}{100{,}000 \text{ counts/sec.}}$ = 10 milliseconds. As with the continuous counting converter, this time is unacceptably long for hybrid applications.

3. "Vidar 260 Voltage-to-Frequency Converter", p. 3-2.

CHAPTER V

IMPLEMENTATION

The VPI hybrid computer system is composed of an EAI® 580 Analog Hybrid Computing System, a GE/PAC® 4020 Process Computer, and an Interface Unit specially designed to marry the two computers. Figure 5-1 shows a block diagram of the hybrid system. The hybrid interface consists of five modules. First, there is the hybrid control module which mediates status and control information between the two computers, and the D to A converter module which provides analog voltages from digital information for use in the analog computer. Then, there is the analog scanner/controller which switches voltages from the analog computer to the A to D converter, and the A to D converter itself. Finally, there is a logic interface which exchanges digital signals between the analog computer's logic module and the digital computer.

The GE/PAC 4020 has a group of I/O commands called GEN II instructions. The format of these consists of eight octal fields.

2 5 X S K3 K2 K1 K0

The first two fields identify the instruction as a GEN II. The X field identifies an auxiliary register, the contents of which are added to the instruction upon execution. During execution of a GEN II command, the 4020 generates two timing signals, Phase A and Phase B. These signals identify the output bit configurations as I/O commands to the peripherals. The S field identifies the I/O action desired.

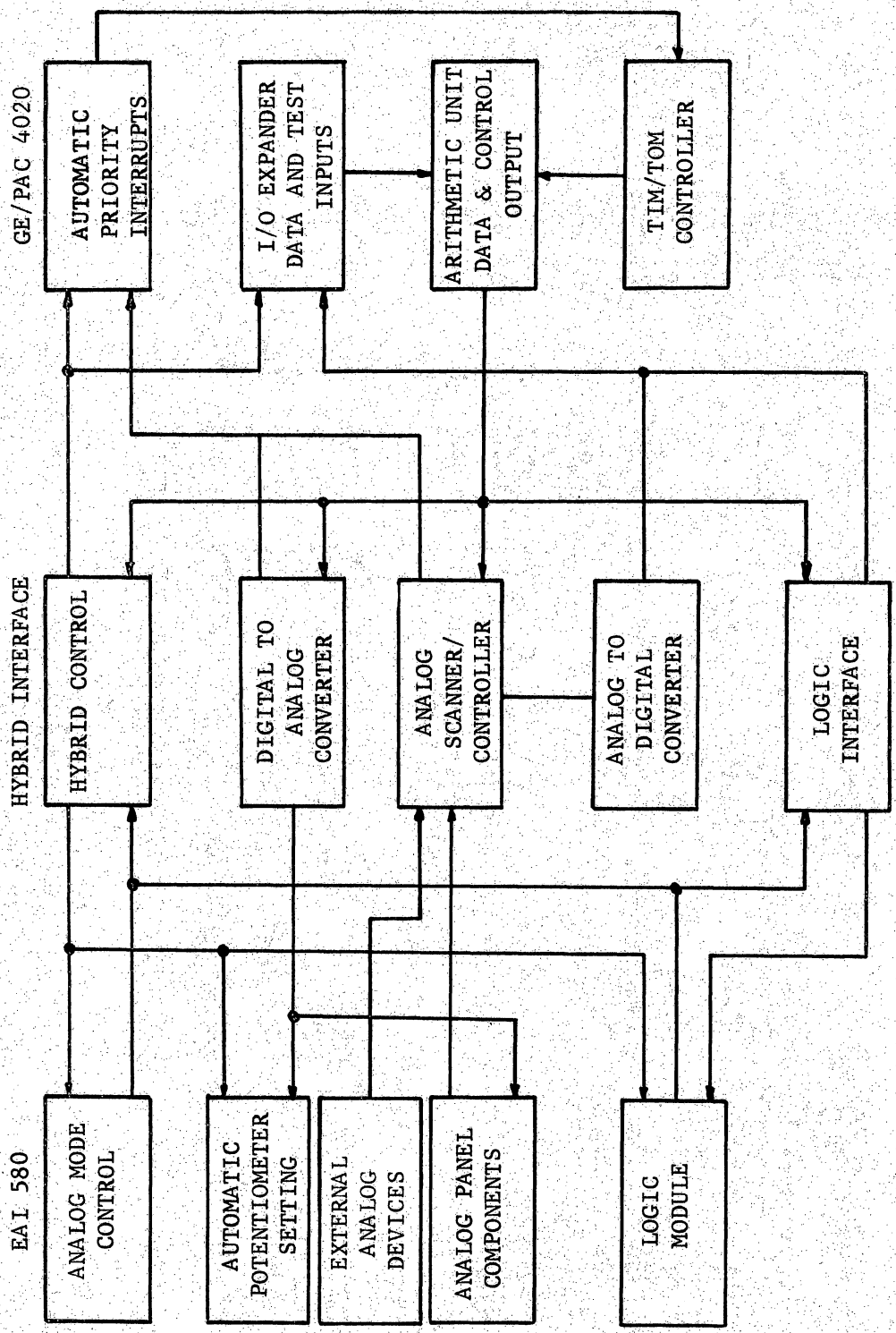


FIGURE 5-1. Hybrid Interface

S = 0	Select device	SEL
S = 1	Activate device	ACT
S = 2	Operate device	OPR
S = 3	Abort device	ABT
S = 4	Output to device	OUT
S = 5	Input from device	IN
S = 6	Test Ready	JNR
S = 7	Test Error	JNE

The in and out commands can be emulated by the TIM, table input to memory, and TOM, table output from memory functions. In response to an interrupt, the TIM/TOM controller will generate an in or an out command to the implemented peripheral, except that the data transferred in or out goes through the B-register directly to or from memory. This transfer does not disturb the running program, and appears to the peripheral as a typical GEN II interchange. The A to D and D to A converters can exchange data with the GE/PAC 4020 using these instructions.

In the GEN II instruction, the K3 and K2 fields define the major device address and K0 the minor.

K3, K2 - - K0 = 1100 D to A Converter

K3, K2 - - K0 = 1101 Logic Interface

K3, K2 - - K0 = 1200 A to D Converter

K3, K2 - - K0 = 1400 Hybrid Control

The K1 field defines the test to be performed by the Test Ready instruction. The peripheral's reaction to this field is specific to the device. The hybrid control module makes the greatest use of this

function. Figure 5-2 shows a block diagram of the hybrid control module.

The EAI 580, in slave mode, accepts external inputs to control its internal functions. With an operate command, S = 2, bits in the GE/PAC 4020 A-register will select the EAI 580 mode.

Operate Command S = 2

A-register bit	EAI 580 function selected
15	STEP
14	10^1
13	10^5
12	10^6
11	RUN
10	STOP
09	CLEAR
08	PROGRAM PANEL, DIGITAL
07	2 MSEC.
06	1 SEC.
05	STATIC TEST
04	SET POTS
03	OPERATE
02	HOLD
01	INITIAL CONDITIONS
00	PROGRAM PANEL, ANALOG

This control information allows the GE/PAC 4020 access to the complete repertory of the EAI 580.

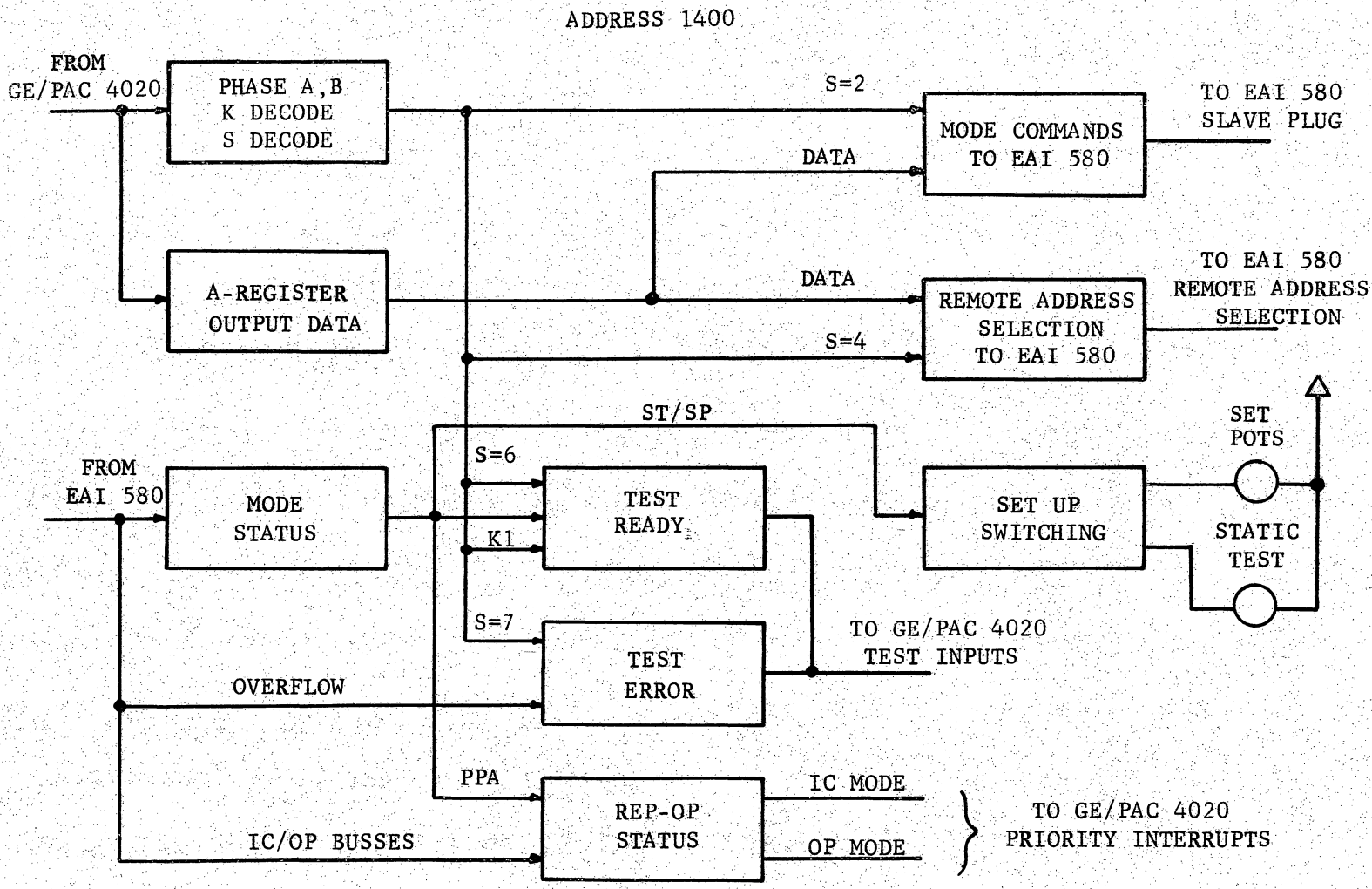


FIGURE 5-2. Block Diagram of Hybrid Control Module

In the other direction, the GE/PAC 4020 can test the status of the EAI 580. This is accomplished by decoding the K1 field of the test ready command.

Test Ready Command S = 6

K1 decode	580 mode tested
0	REMOTE I/O
1	INITIAL CONDITIONS
2	OPERATE
3	HOLD
4	SLEW
5	SET POTS
6	STATIC TEST
7	PROGRAM PANEL, ANALOG

This information allows the GE/PAC 4020 to test the current status of the EAI 580.

In the program panel, analog mode, the analog computer is switched from initial conditions to operate and back again by the REP-OP timer card in the EAI 580. These transitions fire automatic priority interrupts in the GE/PAC 4020 allowing it to synchronize with the analog processing.

In the course of a solution, an operational amplifier may saturate, creating an overflow condition. This intelligence is stored in the hybrid controller until the GE/PAC 4020 issues a test error command, S = 7. Phase B of the test error command clears this storage element. In this way, an analog overflow condition is remembered until

the GE/PAC 4020 tests to see if it has occurred.

The hybrid controller can select any addressable element on the EAI 580 by operating its remote address selection relays. This function is used to set potentiometers in the set pots mode and to monitor the summing junction currents in the static test mode.

An out command, S = 4, to the hybrid controller will load the 4020's A-register into flip-flop storage to pick up the address relays. The least significant A-register bits must be two BCD fields containing the number of the address required. If the out command is issued to the hybrid controller when the EAI 580 is in the set pots mode, its servo amplifier is triggered to set the addressed potentiometer to the voltage held by the D to A output module. If, for some reason, the servo loop fails to adjust the pot, an abort command, S = 3, issued to the hybrid controller will clear the servo loop. In the static test mode, the addressed summing junction is connected to the A/D converter for analysis by the GE/PAC 4020.

Also contained in the hybrid controller module are four generalized priority interrupts. These can be connected to alert the GE/PAC 4020 to any desired logic transition in the EAI 580.

The hybrid control module therefore acts as an intermediary for exchange of control and status information between the digital and analog portions of the hybrid system.

The D to A converter module provides analog voltages from digital information. Figure 5-3 shows a block diagram of the D to A converter module. With an OUT command, S = 4, the 4020 outputs its A-register

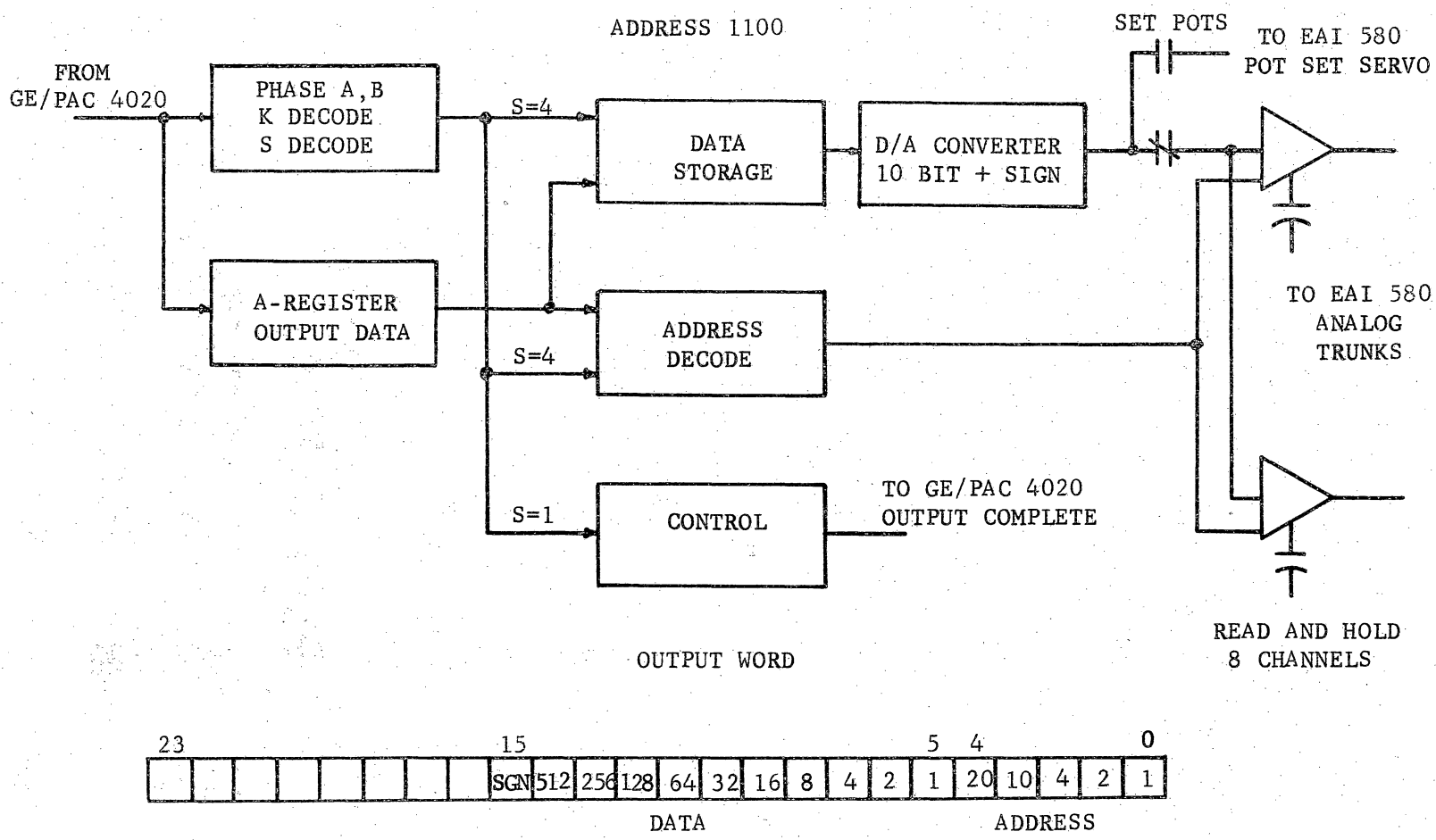


FIGURE 5-3. Block Diagram of D to A Converter

into the data and address storage registers in the module. The format of the A-register information consists of two fields, five bits of address, and eleven bits of data. The address field can accommodate 32 output channels of which eight are implemented. The data field can define a 10 bit binary number plus sign. In this scheme, negative outputs are coded as 2's complement numbers. The D to A converter output is then gated to the read and hold circuit defined by the address field. The basic precision of the D to A converter is one part in a thousand. The leakage of the read and hold cards is defined at 0.1%/sec. in order that the GE/PAC 4020 need update each channel only once per second. For this accuracy, the analog voltage must be held on the read and hold card for 150 microseconds. To guarantee this timing, another 50 microseconds are needed between outputs; therefore, the digital computer can update a given channel each 200 microseconds. A TOM channel can be used to transfer a list of digital values from memory to this module. When a channel has been updated, the output complete line can fire a TOM interrupt signaling the GE/PAC 4020 to send data to the next channel. This gives a top frequency of 50 kHz or, for all eight channels, 6.25 kHz. Since high frequency components are present due to the sample and hold nature of the device, a filter should be used in the analog solution if rapidly changing signals are to be applied to a running problem in the EAI 580.

In the set pots mode, the output of the D to A converter is switched to the EAI 580's potentiometer setting servo. In the automatic pot set mode, then, the potentiometer addressed by the hybrid control

module is set to the value of the voltage found here in the D to A converter module. The D to A converter module is the path that allows digital information to be transferred to the analog computer, both for set-up as in initial conditions or potentiometer settings and dynamically in a running problem.

The analog scanner/controller interfaces with the GE/PAC 4020 to switch input voltages to the A to D converter. It also provides timing pulses to synchronize the GE/PAC with the conversion process. Figure 5-4 shows a block diagram of the scanner/controller. An out command, S = 4, issued to address 1200 loads the T-register with the address of the point to be scanned and generates a pulse to start the A to D converter. The address in the T-register is decoded and applied to the high speed switches. The decoded T-register information then turns on the selected analog switch which connects its input to the A to D converter. The switches are FET's, so there is no appreciable delay between selection and connection. When the A to D converter goes not busy, that transition signals the GE/PAC 4020 that the converted data is ready. When the GE/PAC does an in command, S = 5, to take in the converted data, the controller signals the GE/PAC 4020 to output the next point to be scanned. These signals can fire TIM and TOM interrupts in the GE/PAC. In this way, the current value of a set of state variables can be maintained in the GE/PAC 4020 memory for processing in real-time without program intervention. If the A to D converter overflows due to too great an input voltage, this information is stored in the scanner/controller and also entered as bit 0 of the

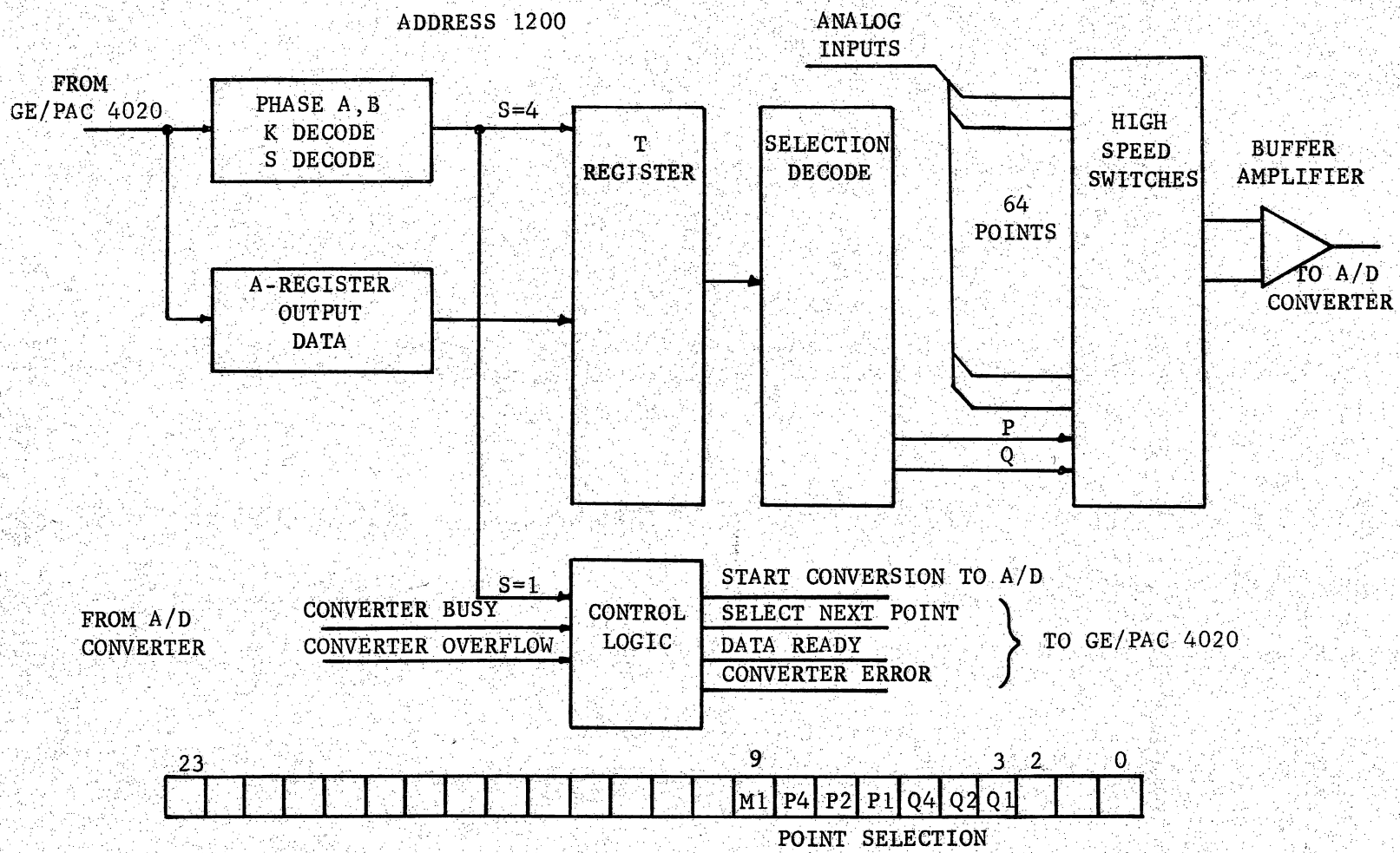


FIGURE 5-4. Block Diagram of Analog Scanner/Controller

A to D converted output word. When the GE/PAC 4020 tests for error with a test error command, S = 7, the error flip-flop is cleared. In this way, an overflow in the A to D converter is remembered until the GE/PAC 4020 takes a look at it. By issuing an operate command, S = 2, the GE/PAC 4020 makes the scanner/controller start an A to D conversion cycle without selecting a new point. In this way, the computer can follow a single state variable without having to re-address for each conversion.

The A to D converter is of the successive approximation type operating at a basic rate of 62 microseconds per bit. The precision of the converter is 11 bits, so a point can be converted in 682 microseconds. The start pulse takes 10 microseconds, and the GE/PAC 4020 in command takes 26 microseconds; therefore, the total conversion time is 718 microseconds corresponding to a scan frequency of 13.9 kHz. A read and hold circuit with a sample time of 10 microseconds is implemented ahead of the A to D converter. Figure 5-5 shows a block diagram of the A to D converter system.

This module responds to a start pulse from the scanner/controller. The analog input is up to plus/minus 10.23 volts from the read and hold card. Starting with the sign bit, a current equivalent to that bit position's bit power is applied to the summing junction of the comparison amplifier. If the incoming signal is more positive than the bit power, the bit is accepted. The single-shot then strobes that bit position from the shift register into the C-register which maintains the bit's current as the conversion proceeds to the next bit position. If

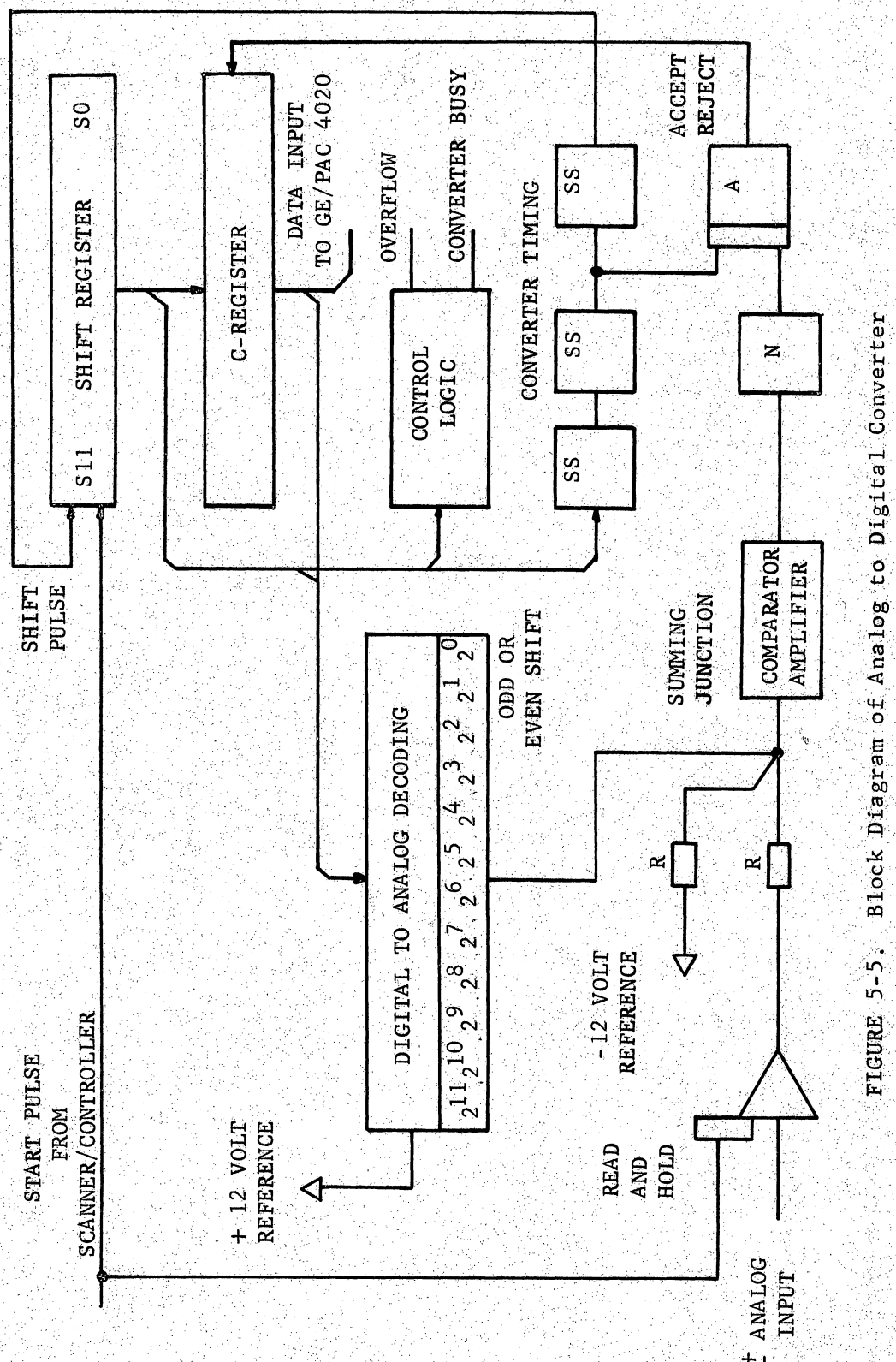


FIGURE 5-5. Block Diagram of Analog to Digital Converter

the unknown signal is more negative than the bit power, the bit is rejected, and the single-shot does not strobe that bit position into the C-register. If the input signal is negative, the most significant bit of the C-register is not set, and the current from its D to A converter is removed from the summing junction. Thus the sign bit is set, and the conversion proceeds referenced full negative. The conversion algorithm then generates the 2's complement of the input signal. This occurs since all subsequent bit tests are made with respect to the maximum negative input. When the last shift register bit is set, the conversion is complete. If the entire C-register is set and the input signal is still greater than the C-register number, a positive overflow has occurred. Alternatively, if at this time the entire C-register is cleared and the input signal is less than the C-register number, a negative overflow has occurred. This overflow signal is stored in the scanner/controller module.

The converter busy signal goes true when the start pulse comes in and stays true while the timing single-shots are firing. When the conversion is complete, the timing pulses stop, and the busy signal goes false. This transition signals the GE/PAC 4020 that the converted data is available in the C-register. The GE/PAC then issues an in command, S = 5, and takes the data into its A-register. The converted data is formatted with the sign bit in bit position 24, that is left justified. The overflow flag appears in bit position 0. When the data appears in the GE/PAC, it can be tested for validity, bit 0 false, and shifted to right justify the data and maintain the sign. In this manner, the

converted data is an eleven bit natural binary number with negative values expressed in 2's complement form.

The logic I/O module exchanges digital information between the GE/PAC 4020 and the EAI 580 Logic Expansion Group. Figure 5-6 shows a block diagram of this module. In response to an in command, S = 5, the module transfers the status of 16 bits of data to the GE/PAC 4020's A-register. The true state of an input line is +5 volts or open. True input data appears as a 1 in the digital computer. In response to an out command, S = 4, the data contained in the GE/PAC 4020's A-register is transferred to the module's output drivers. The output is a 13 microsecond pulse. The true state of the outputs is +5 volts resulting from a 1 in the digital computer.

The hybrid interface system is seen to provide a generalized communications path between the GE/PAC 4020 digital computer and the EAI 580 analog computer. This interface scheme allows either computer to act as the master control, depending upon the class of problems to be studied, and it provides for full interchange of a problem's parameters between the digital and analog regimes as a problem solution unfolds. The system is fast enough to study many large problems in real-time with good accuracy. Hopefully, this system will impose no major impediment to the imagination and competence of the experimenter.

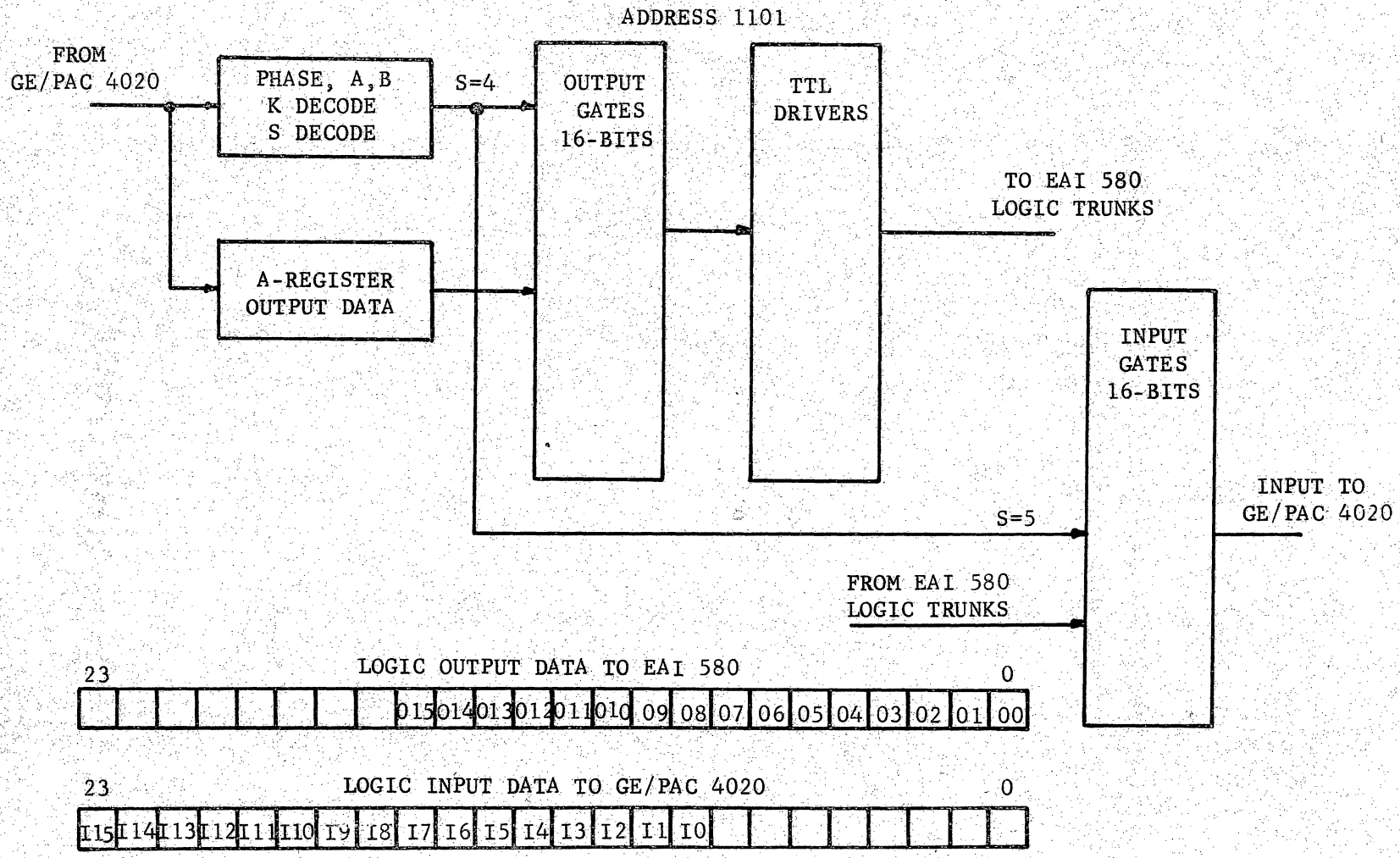


FIGURE 5-6. Block Diagram of Logic I/O Module

CHAPTER VI

SUMMARY

The Hybrid Computing System is a generalized tool for studying natural phenomenon. For this reason, a comprehensive analysis of its capabilities is out of the question. Even so, it might be illuminating to sketch out the operation of the system as it generates a set of solution curves for Legendre's Equation mentioned in Chapters Two and Three.

As a first step, the experimenter would wire the hybrid diagram shown in Figure 3-3. The digitally computed value y would be connected to the first read and hold channel shown in Figure 5-3. The initial conditions $\dot{Y}_o$ and Y_o would be connected to channels two and three. The analog computed values $\dot{y}$ and y would be connected to two of the high speed switches in the analog scanner/controller shown in Figure 5-4. These interconnections are available on the analog program panel.

Analogous to the wiring of the analog computer is the programming of the digital computer. A good way to show the steps the digital computer must take is to develop a flow chart which can then be reduced to coding for operating the machine. Figure 6-1 shows such a flow chart. In this simple example, the experimenter defines $\dot{Y}_o$, Y_o, $\Delta\dot{Y}_o$, ΔY_o, n, $\dot{Y}_{of}$, and t_f. The flow chart assumes that the operating system is making the current values of $\dot{y}$ and y available in the computer's memory and that D/A 1, 2, and 3 are being output continuously. This is possible due to automatic I/O hardware in the GE/PAC 4020 which allows

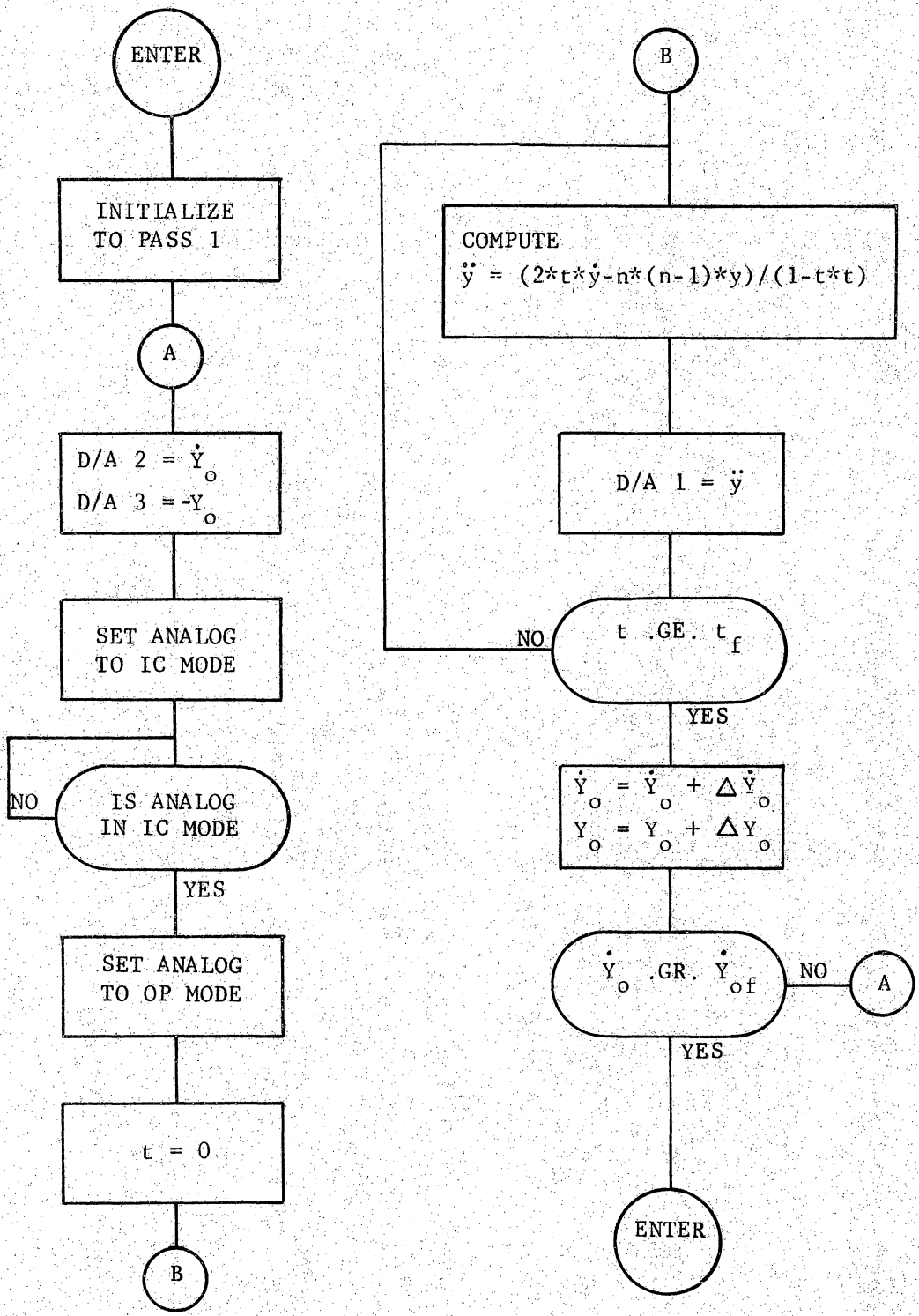


FIGURE 6-1. Flowchart for Solution of Legendre's Equation

data to be sent and received without disturbing the running program. Also, the real time clock must be running.

When the program begins to run, the initial values of $\dot{Y}_o$ and Y_o are placed on the D/A output channels, and the analog computer is switched to the IC mode. When the analog computer gets into the IC mode, the integrators used will immediately attain the values $\dot{Y}_o$ and $-Y_o$. At this point, the program switches the analog computer to the RUN mode and sets the clock to zero. The next step is to compute the value of $\ddot{y}$ and output it on D/A 1. Since the values of $\dot{y}$ and y are being updated as fast as the A/D converter can bring them in, and $\ddot{y}$ is output at a similar rate, this calculation will stay in step with the analog computation of the integrals. When t_f is reached, the run is completed, and new values of $\dot{Y}_o$ and Y_o are calculated. If $\dot{Y}_o$ has not reached its final value, the new initial conditions are impressed on the integrators, and the next pass begins. The output of this problem might well be a picture of $\dot{y}$ versus y on the analog computer's oscilloscope. It would then be a family of curves whose initial conditions are changed from run to run by an amount $\Delta\dot{Y}_o$ and ΔY_o.

This example certainly minimizes both the computing power and some of the problems associated with the Hybrid Computing System, but it is intended to clarify, perhaps, the intent of the system and to show the interaction between the digital and the analog computing regimes.

BIBLIOGRAPHY

1. "EAI 580 Analog/Hybrid Computing System, Maintenance Series, Console Components." Publication No. 00.800.2056-0. Electronic Associates, Inc., June, 1969.

2. "EAI 580 Analog/Hybrid Computing System, Maintenance Series, Logic Expansion Group." Publication No. 00.2058-0. Electronic Associates, Inc., October, 1968.

3. "GE/PAC 4020 System Manual." General Electric Company, Publication GET-3460.

4. Jackson, Albert S. Analog Computation. New York: McGraw-Hill Book Company, Inc., 1960.

5. Johnson, Clarence L. Analog Computer Techniques. New York: McGraw-Hill Book Company, Inc., 1956.

6. Katell, E. "Hybrid Computation Comes of Age." EAI Applications Reference Library, Bulletin No. 852541, January, 1969.

7. Ledley, Robert Steven. Digital Computer and Control Engineering. New York: McGraw-Hill Book Company, Inc., 1960.

8. Vichnevetslsy, R. "Error Analysis in The Computer Simulation of Dynamic Systems: Variational Aspects of The Problem." IEEE Transactions on Electronic Computers, Vol. EC-16, No. 4, August, 1967.

9. "Vidar 260 Voltage-to-Frequency Converter." Vidar Corporation Technical Manual 114-1, Mountain View, California, February, 1966.

10. Bekey, George A. and Karplus, Walter A. Hybrid Computation. New York: John Wiley & Sons, Inc., 1968.

The vita has been removed from the scanned document

GE/PAC 4020/EAI 580
HYBRID COMPUTING SYSTEM

by

Alan J. Bomberger

ABSTRACT

Hybrid computation is a discipline designed to reap benefits from the strengths of both digital and analog computation. This system welds the capabilities of a GE/PAC® 4020 Process Computer and an EAI® 580 Analog/Hybrid Computing System by means of a special Hybrid Interface.

The Hybrid Interface consists of five modules. First, there is the hybrid control module which mediates status and control information between the two computers, and the D to A converter module which provides analog voltages from digital information. Then, there is the analog scanner/controller which switches voltages from the analog computer to the A to D converter, and the A to D converter itself. Finally, there is a logic interface which exchanges digital signals between the analog computer's logic module and the digital computer.

The interface scheme allows either computer to act as the master control, and it provides for full interchange of a problem's parameters between the digital and analog regimes as a problem solution unfolds. The system is fast enough to study many large problems in real-time with good accuracy.