IMPLEMENTATION OF A HYBRID SYSTEM INTERFACE

by

Robert Marvin Amundson

Thesis submitted to the Graduate Faculty of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

A. W. Bennett, Chairman

R. A. Thompson

F. G. Gray

June 1975

Blacksburg, Virginia

ACKNOWLEDGMENTS

The author would like to acknowledge the help and support of his committee during his graduate program. He especially thanks Dr. Bennett for his frequent guidance and support of his work in the Computer Engineering Laboratory. He would like to thank many of the people in the Computer Engineering Laboratory for their inputs into the design of the hybrid interface. He would like to thank Barbara Baldwin for her diligent work in the typing of this thesis. Finally, he would like to express appreciation to his wife, Lynn, for her invaluable work in making the drawings.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

Introduction

1.1 Introduction to Hybrid Computation

The application of computing techniques in scientific and engineering fields has recently increased at a phenomenal rate. Computers have been utilized in practically all technical fields. Also, considerable technological advancement has been made in the design of both analog and digital computing components. Likewise, much work has been done on methods for interfacing the computer to other environments. The man-to-machine interface has also received considerable attention, resulting in new software programming techniques. With the recent introduction of the digital microprocessor "computer on a chip", the use of computing elements in real-time systems should grow even faster.

At an early stage of computer development, designers realized that both analog and digital computation have well-defined advantages and disadvantages. In the analog computer, variables are represented by voltages. The analog computer can operate on many variables simultaneously. Also, the application of an operator to a variable is continuous. Thus, the analog program has a close relationship with all program variables. The speed of computation is determined primarily by the bandwidth of the analog computing elements. In addition, certain complicated functions can be computed directly. As an example, integration can be performed directly, giving a true integral rather than an approximation [1,2].

The analog computer has several limitations, however. The accuracy of computation is determined by the accuracy of the analog components. High-accuracy analog computing elements are expensive and complicated. Also, the analog computer can only handle a fixed number of program variables.

In the digital computer, variables are represented by digital values in memory. The number of variables which can be used is practically unlimited. Also, the accuracy of computation is determined primarily by the number of bits of precision which is used to represent a variable. Since system characteristics are determined by software, they are easily changed [1,2].

However, the digital computer is a sequential device, and can only perform one operation at a time. Thus, some functions will take longer to perform on a digital computer than on an analog computer. For instance, integration must be performed by a numerical technique which may consume considerable time. Also, the evaluation of many functions will only be an approximation.

A hybrid computer system is a system which provides an interface between analog and digital computing elements. Thus the programmer can make the best use of the advantages of both the digital and the analog computer. The hybrid system which is described in this thesis was formed by interfacing a General Electric GE/PAC 4020 digital computer to an Electronic Associates, Inc. 580 analog computer. In the remainder of this thesis, these two computers will be referenced as the GE-4020 and the EAI-580.

A hybrid computing system can be used to perform many useful functions for the design engineer. One of the oldest uses for a hybrid system is the simulation of physical systems [1]. The physical system variables can be represented by voltages on the patch panel or values in the digital computer memory. The inclusion of the digital computer in the simulation allows the evaluation of those unusual nonlinear transfer functions which are often found in physical systems. Also, the characteristics of the simulated system can be dynamically changed. This leads to the use of a hybrid computer for optimization of the simulated system. The digital computer can be used to develop the strategy for parameter adjustment. This approach is particularly useful for the optimization of a system for which the strategy changes as the optimum is approached [2].

With the growing emphasis on digital processing of signals, the hybrid computer can be a valuable tool for the evaluation of new processing techniques. Many new signal processing methods are being used in the design of systems. Digital filtering techniques are being used to improve signal filter characteristics, and to provide cheaper filters than are obtainable with analog devices. The fast Fourier transform provides a method for performing quick spectrum analyses. High-speed convolution and correlation techniques open the way for new filtering transfer functions. All of these processing methods involve the conversion of the analog signal to digital samples. The processing is then performed on the samples using digital algorithms such as difference equations. The processed output can then be converted back into an analog signal [3]. Since the hybrid computer

provides access to both the digital and the analog domain, it can serve as a valuable test facility for the verification of algorithms which use these techniques.

The hybrid computer can also be used to evaluate the performance of a sampled-data control system. This type of system occurs whenever a digital processing algorithm is used as part of a control system. Since the digital processor must operate on a variable in a sampled fashion, the control system is operating on discrete data points. The value of a variable is not known during the time interval between two samples. The effects of sampling must be considered in the design of a sampled-data system [4]. A hybrid computer can be used to simulate a sampled-data control system. The digital portion can perform the role of the digital control algorithms, while the analog portion simulates analog components of the control system.

1.2 Organization of Thesis

This thesis describes the GE-4020/EAI-580 hybrid computer system. The hardware implementation is briefly described. Details on specific hardware components are only given where they are needed for proper understanding of the system operation. Emphasis is placed on a description of the interfaces available between the user and the system. The control commands and sequences are discussed in detail. This thesis provides information which is necessary for the user interface at both the language level and the machine level. Also, information is provided which will aid in further hardware development. A discussion of high-level hybrid programming languages is outside the

scope of this thesis.

Chapter I serves as an introduction to hybrid computation. Chapter II gives an introduction to both the digital and analog computer portions of the system. A basic understanding of the operation of both digital and analog computers is assumed. Chapters III through X describe the operation of the various hybrid components in detail. The hardware operation is specified, and the structures of data paths and control commands are listed. Wherever needed for clarification, the operation of analog computing elements is described. Chapter XI goes into the use of an extensive set of digital computer interrupts which are available. These interrupts can be used to augment the speed of a hybrid program. Also, certain interrupts can be used to establish automatic input or output sequencing of some of the hybrid sections which are described in earlier chapters. The interrupts can lessen the burden of the hybrid programmer, and can facilitate the sharing of some hybrid components between two or more programs. Chapter XII summarizes the hybrid computing system, and suggests some improvements for the future.

Chapter II

Hybrid System Design Considerations

2.1 Introduction to the Digital Computer

The GE-4020 digital computer is a 24-bit machine with an accumulator (A) register, a quotient (Q) register, and seven index (X) registers. The accumulator is used as a general purpose data manipulation register. Its contents can be shifted, masked, tested, and gated with boolean functions. In addition, it can be used to operate on positive or negative binary quantities. The Q-register is used in conjunction with the A-register for double word quantity manipulation. The index registers are used for program counters, subroutine linkage, and indexed core addressing.

The GE-4020 has a few special-purpose registers which are not available to the executing program. The P-register holds the core address of the instruction which is currently executing. The I-register holds the current instruction during CPU instruction sequencing. The B-register is a buffer register which is used to communicate both instructions and data between the memory and the arithmetic unit. The arithmetic unit also contains a parallel full adder which is used for arithmetic and logic operations. In addition, the transfer of data between the input/output channels and the arithmetic unit is controlled by the parallel adder. It is a 24-bit data transfer.

A number of instructions are provided by the computer. Among these is a set of I/O instructions. The OPR (Operate) command is used to initialize an I/O channel so that it is ready for an operation. It can also be used to start an I/O operation. The ABT (Abort) com-

mand is used to terminate a channel operation in case of some abnormal condition. The JNR (Jump if Not Ready) command is used to determine the operation status of an I/O channel. The JNE (Jump if No Error) command is used to determine the error status of a channel.

The IN (Input) and OUT (Output) commands perform the actual data transfer. When an IN or OUT command is executed by a program, the data is transferred between the A-register and the I/O channel. The parallel full adder connects the I/O channel multiplexer to the A-register during the data transfer. The channel multiplexer interfaces many channel devices to one parallel adder bus.

The GE-4020 provides an automatic table I/O feature which is driven by interrupts. The interrupt system and table I/O are described in detail in Chapter XI. When activated, the table I/O section of the computer generates an IN or OUT instruction with the desired channel address. However, the instruction execution sequence differs from the sequence for user program I/O instructions. Data is transferred between core memory and the channel multiplexer. The B-register and parallel full adder provide the data communication path for 24 bits.

The 12 bit operand field of an I/O instruction specifies the channel address. In this thesis, the operand address is written according to the GE-4020 assembler language format, and is specified in octal. A "O" followed by two apostrophes indicates that the number between the apostrophes is an octal number.

The channel address and the type of I/O instruction are provided to all devices which are connected to the channel multiplexer. Thus, the exact operation of each instruction is determined by the logic

design in the selected device interface.

2.2 Introduction to the Analog Computer

The EAI-580 analog computer provides a variety of analog computing components which are accessible on the patch panel. Among these are amplifiers and integrators. A set of potentiometers are available for variable scaling. In addition, the EAI-580 provides a number of special computing components, such as the sin-cos and diode function generators.

The EAI-580 mode control section provides a wide variety of methods for controlling the problem solution. A pot set servo amplifier is provided to control the operation of most of the potentiometers. A readout selector system is included to allow voltage readout of various computing components. An overload detection system provides quick detection of analog amplifier overload conditions.

The analog computer also provides a separate patch panel with some digital logic. Gates, registers, differentiators, and counters are accessible on the patch panel. The digital clock and logic mode control section of the EAI-580 is used to control the operation of the logic components. These components can be used to control certain functions within the analog program.

The EAI-580 was designed to allow easy access to all of the signals needed for connection of a hybrid interface. The necessary connections are made by cable connectors in the rear of the computer.

2.3 The Hybrid Interface

The hybrid interface was built using components which were donated by the General Electric plant in Salem, Virginia. It provides an extensive communication path between the GE-4020 digital computer and the EAI-580 analog computer. Figure 2.1 shows a block diagram of the hybrid interface.

The interface provides a method for controlling and sensing the analog computer operating mode from the digital computer. It also enables the digital computer to control the coefficient settings of the servo-controlled potentiometers. The overload status of the analog amplifiers can be detected from the digital computer. The outputs and derivatives of amplifiers can be converted from analog to digital and brought into the digital program. A logic signal communication path is provided between the digital logic patch panel and the digital computer.

In addition, a number of supplemental data paths are provided to expand the flexibility of the hybrid system. An analog output section is available which can provide a set of analog output values to the analog panel trunks under digital computer control. An analog input section is available which can sample a set of analog signals on patch panel trunks and convert them to digital for use by the GE-4020 program. A set of high-speed digitally-controlled attenuators are provided. These attenuators are coefficient devices which are similar to the EAI-580 potentiometers. They can be used for dynamic variation of program parameters. A number of interrupt lines are provided which allow the analog program to interrupt the digital program. Thus, the

hybrid interface provides a complete two-way control and data path between the analog computer and the digital computer.

2.4 Conversion between Analog and Digital

In the design of the hybrid interface, considerable attention was given to the format for communication between analog voltages and digital values. Many formats for conversion between the two domains can be chosen in a hybrid system design. The best known format for digital representation of analog voltages is the natural binary code. In this code, the weight of each bit is related to the weight of other bits by a power of two. The bit weight increases in succession from one end of the binary field to the other. The binary-coded decimal (BCD) code is often used. In this code, each decimal digit is represented by a group of four binary-coded bits. Only the values zero through nine are allowed within each group. The Gray code is a code in which the transition from one value to the next succeeding value involves a change of only one bit position. Many other codes have been used for specific applications [5].

The natural binary code has several advantages over the BCD and Gray codes. It is used as the format for integer arithmetic in most present-day minicomputers. Also, it uses all possible code combinations, providing the maximum number of discrete levels for a certain number of bits of precision. The Gray code also uses all bit combinations, but the bit positions do not correspond to specific bit weights. Thus, software which will handle Gray code values is complicated. The BCD code is a good code to use for human interfacing

with a digital system. A human operator can read a BCD value easily. Also, a BCD value can control a decimal numeric display with a minimum of decoding. However, this code is inefficient in its use of bits.

The natural binary code was chosen for digital representation in the hybrid system. A sign bit was included to cover both positive and negative quantities. When the sign bit is on, the rest of the binary value is coded in two's-complement form. This corresponds to the GE-4020 format for arithmetic values.

Many scale factor schemes can be used for digital value representation [5]. One method sets the weight of the least significant bit at some power of ten. Then each bit to the left has a bit weight of twice the neighboring bit. For a converter with ten data bits and a least significant bit weight of 0.01 volts, the converter can handle a magnitude of up to 10.23 volts. Thus, the converter can handle voltages which are slightly higher than the EAI-580 analog amplifier range. This leeway can be an advantage in hybrid programming.

The other scheme sets the most significant data bit weight at 1/2 times some power of ten. Each bit to the right is 1/2 the value of its neighbor. This method is convenient for the specification of fractional coefficients. However, the maximum magnitude is slightly less than a power of ten. For a converter with 10 bits and a most-significant bit weight of 5.0 volts, the maximum representation is 9.99 volts. Thus, a small portion of the EAI-580 amplifier range is beyond the limits of such a converter.

The digital scaling of analog values in a hybrid system should follow the same format as used for coefficient devices, such as poten-

tiometers. This will simplify the communication of digital values between different devices. For this reason, the second scheme is used in the hybrid interface. The most significant bit is set to 0.5 for coefficient devices and 5.0 volts for voltage devices. Also, the sign bit is in the same bit position for all hybrid devices. Thus, values can be easily transferred from one device to another.

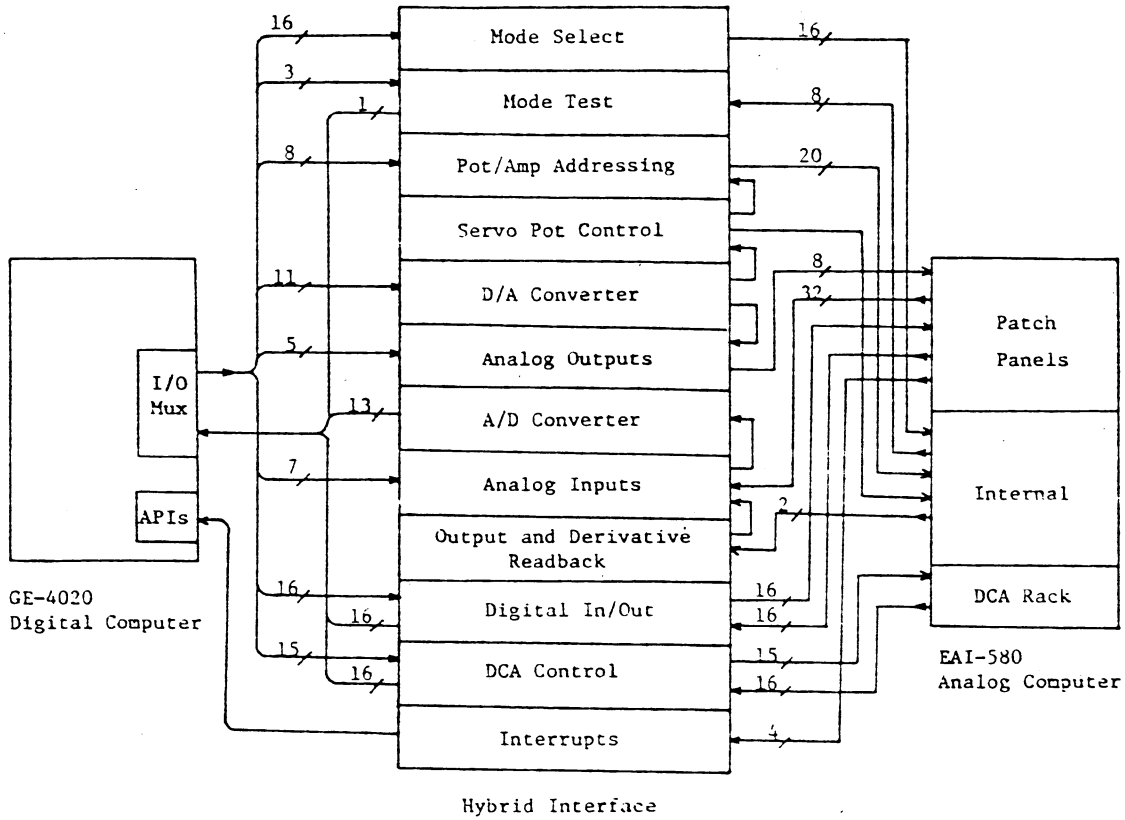


Figure 2.1 Hybrid System Block Diagram

Chapter III

Analog Mode Control and Sense

3.1 Analog Computer Modes

The mode control section of the hybrid interface enables the GE-4020 digital computer to control the operation of the analog components on the EAI-580 analog computer. The mode sense section enables the digital computer to determine which analog modes are selected.

Fig. 3.1 shows a partial view of the analog computer control panel. The mode control pushbuttons are divided into four groups in this figure for the purpose of explanation. The pushbuttons are interlocked within each group, so that the selection of one mode will cancel all others within that group. However, each group operates independently from the other groups.

The Analog Mode group controls the operation of integrators, track/store units, and potentiometers on the analog patch panel. In the IC (Initial Condition) mode, integrators are held to the voltages which are patched into their initial condition input pins. In the HD (Hold) mode, integrators are held to the state which was existant immediately before the HD mode was selected. This allows snapshot sampling of the state of the system. In the OP (Operate) mode, all analog components are operational. This is the compute mode for the analog program. In the SP (Set Pot) mode, the potentiometers are selected for coefficient setup. All other analog components are held inoperative in this mode. The ST (Static Test) mode enables the derivatives of integrators and track/store units to be measured. The derivatives are found by measuring the input summing-junctions. During this mode, the

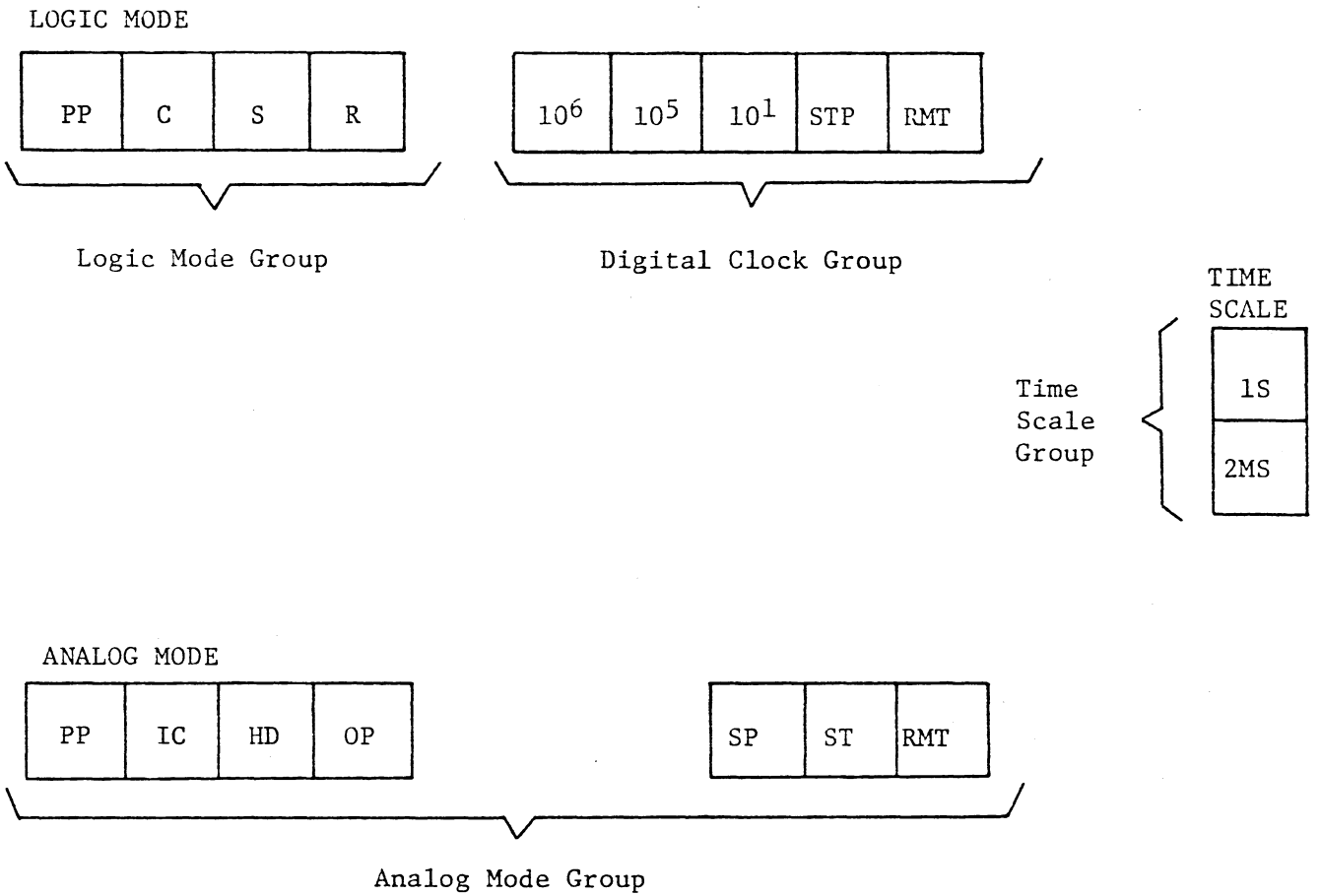


Fig. 3.1 Mode Control Section of EAI-580 Control Panel

integrators and track/store units are held at their initial condition input voltages. This feature is covered in more detail in Chapter IX. The PP (Patch Panel) mode allows a signal from the patch panel to set the mode to IC or OP. If no patch panel signal is present, the mode alternates between the two at a rate determined by an adjustable timer. This mode of operation is used for generating a family of system curves for the analog program, with parameters varying each time the IC mode is entered.

The Time Scale group is used to control the integration time constant of the analog integrators. The two selections are 1S (1 Second) and 2MS (2 Milliseconds). This controls the time frame of the analog system.

The Logic Mode Group is used to control the operation of certain logic functions on the digital patch panel. Among these are the decade counters and general purpose registers.

The Digital Clock group is used to select the rate of the clock which is used by some of the digital panel logic functions. The clock may be selected to one of three rates or to a single-step mode.

3.2 Mode Control Operation

The Analog Mode group and the Logic Mode/Digital Clock group each contain a pushbutton labelled RMT, which means "remote". When depressed, the RMT pushbutton enables mode selection of its group by the digital computer. If the RMT pushbutton is not depressed, mode selection within that group may only be accomplished manually. Mode selection within the Time Scale group may always be accomplished both

manually and by the digital computer. Therefore, the RMT pushbuttons should be depressed when the analog computer is controlled by the digital computer. Otherwise, control commands may be ignored by the analog computer.

Mode control from the digital computer is accomplished by the execution of an OPR 0'1400' command with the appropriate A-register bit configuration. Table 3.1 lists the association between digital computer A-register bits and analog computer modes. The A-register bits may be set to select one mode within each group. If more than one bit in a group is set, an ambiguous situation exists and the result may be any one of the selected modes. The status of a group is not altered if all bits within that group are zero.

The mode control procedure can be summarized as follows:

1. Set the desired bits in the A-register for groups which are to be changed.
2. Execute the command OPR 0'1400'.

The 16 mode control bits are brought out from the GE-4020 on the output data bus. They are gated by the command control logic in the hybrid interface, and then routed to the mode select logic in the EAI-580. The resulting signal in the analog computer is a pulse for each mode which is to be set.

The STP (Step) function in the Digital Clock group performs an additional function. Each time a mode select command is issued to set the STP mode, the digital clock circuit provides a single clock pulse. Thus the digital computer can provide single clock stepping in the same manner as with the STP pushbutton.

Table 3.1 A-register Bits for Mode Selection

	GE-4020 A-register Bit	EAI-580 Mode	
	00	PP	Patch Panel, Analog
Analog	01	IC	Initial Conditions
Mode	02	HD	Hold
Group	03	OP	Operate
	04	SP	Set Pot
	05	ST	Static Test
Time	06	1S	1 Second
Scale	07	2MS	2 Milliseconds
Group			
	08	PP	Patch Panel, Digital
Logic	09	C	Clear
Mode	10	S	Stop
Group	11	R	Run
	12	10^6	1 MHz
Digital	13	10^5	100 KHz
Clock	14	10^1	10 Hz
Group	15	STP	Step

3.3 Mode Sense Operation

Mode sense is an important supplement to mode control. Analog computer modes may be modified by conditions within the analog program as well as by commands from the digital computer. For example, the analog PP (Patch Panel) mode alternately activates the IC (Initial Condition) mode and the OP (Operate) mode. Therefore, it is essential for the GE-4020 computer to be able to sense the analog computer mode at any particular instant.

Sensing the mode of the analog computer from the digital computer is accomplished by the execution of a JNR O'14XO', with bits 03 through 05 (X) of the command specifying the mode to be tested. Thus the user must test for the presence of a certain mode, and will obtain either an affirmative or negative answer about the existence of the tested mode. If the tested mode is currently selected, the JNR command will move on to the following instruction in core. If the tested mode is not selected, the JNR command will skip over the following instruction.

Only the modes of the Analog Mode group can be tested with this mode sense operation. A maximum of five JNR test command executions are needed to completely determine the current analog mode. Table 3.2 lists the command bit values for each mode test.

The mode sense procedure can be summarized as follows:

1. Execute JNR O'14XO' command, with X field determining mode to be tested.
2. The command following the JNR instruction will be executed if the tested mode is active. Otherwise, this command will be skipped over, and the second command

2. after the JNR instruction will be executed.

Table 3.2 JNR O'14XO' Command Bits for Mode Testing

JNR X Field Value	EAI-580 Mode Tested
1	IC
2	OP
3	HD
5	SP
6	ST
7	PP (analog)

Chapter IV

Analog Overload Sense

4.1 Analog Overload Status

All amplifiers and integrators on the EAI-580 analog panel have overload status outputs. The analog overload sense section of the hybrid interface enables the GE-4020 digital computer to sense the occurrence of any overloads. Each amplifier and integrator overload output goes to a corresponding light indicator on the overload panel above the digital voltmeter. A signal is also developed which is the logical "or" of all overload outputs. This overload signal is routed to the hybrid interface for testing by the GE-4020.

Since individual overload indications often flicker on and off, the combined overload signal goes to a holding latch in the hybrid interface, and an overload indication is held until the JNE overload test command is executed. The JNE test then resets the holding latch if it was set. This assures that an overload occurrence will not be missed.

4.2 Overload Sense Procedure

Analog overload sense is accomplished by the execution of a JNE command with the appropriate address. The command will indicate the presence or absence of an overload since the previous overload test. The procedure can be summarized as follows:

Procedure: Execute command JNE O'1400'.

Results : P count goes to P+1 if overload has occurred, thus executing the next instruction.

Results : P count goes to P+2 if overload has not occurred, thus skipping over the next instruction.

Chapter V

Servo-set Potentiometers

5.1 Introduction

The servo potentiometer control section of the hybrid interface provides a digital method for setting the coefficients of potentiometers located in the EAI-580 analog computer. Potentiometers are used in the analog program to multiply a signal voltage by some coefficient between zero and one. They are used for scaling state variables, for providing constants, and for a variety of other functions.

Figure 5.1 is a block diagram of the hybrid interface components which are used to set the potentiometers. For simplicity, most of the control logic is not shown. The D/A (Digital to Analog) converter is used to provide a reference voltage to the servo amplifier. This voltage is equal to the desired coefficient multiplied by ten. A set of address select relays is used to select the address of the pot to be set. When the analog computer is in SP mode, the relays are connected to a pot addressing network in the EAI-580. The servo amplifier drives a motor in the selected potentiometer, and will "slew" the pot until the voltage at the pot wiper is the same as that provided by the D/A converter. Since the high side of the selected pot is switched to +10 volts, the pot servo will stop when the pot is set to the desired coefficient. As the pot action is mechanical, the operation may take a number of seconds to complete.

A digital computer command is provided to load the D/A converter register with the coefficient. Another command is provided to load

the address select register and start the pot servo. The address select bits are brought out of the GE-4020 in a two-digit BCD format. The address select decode logic converts each digit to a one-of-ten output format, and drives the appropriate "tens" relay and "units" relay. A command which will abort the pot servo operation is available in case of an error. A test command is supplied to test for completion of the pot set operation.

5.2 Use of the Potentiometers

Potentiometer patch fields on the EAI-580 computer may contain one of three different types of coefficient devices:

1. Handset potentiometers, which are controllable only at the analog computer.
2. Servo-set potentiometers, which can be controlled by the digital computer or at the analog console.
3. Digitally-controlled attenuators (DCA's), which are high-speed coefficient devices controllable only by the digital computer.

The servo-set potentiometers in use at present are located in pot fields 10 through 39 and 60 through 69. The pot set servo amplifier and control circuitry does not, however, check the validity of the selected pot address. If an attempt is made to set either the handset pots or the DCA's, the servo amplifier will search unsuccessfully for a servo null until it is reset. This may be performed manually by depressing the CLR pushbutton at the lower right corner of the analog control console. A circuit has been added to the EAI-580 to turn off

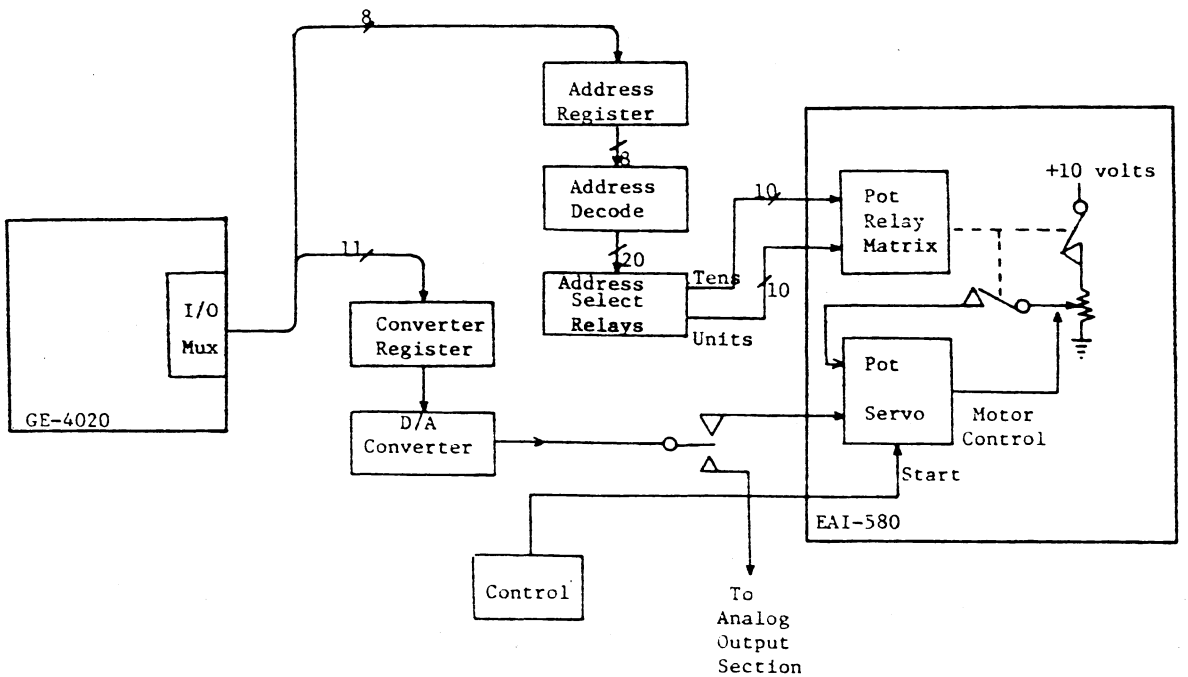


Figure 5.1 Servo Set Potentiometer Control

the servo amplifier automatically if a servo null is not attained within 10 seconds after the start of a pot set operation. Nevertheless, a valid address should always be chosen for a servo-set operation.

All potentiometers with an address ending with the digit "4" provide the programmer access to both the low (LO) and high (HI) sides of the pot. Such a pot has no internal connection between the low side and ground. This feature is useful to provide a variable resistor for some special circuit configurations. One example is a variable-gain amplifier, using the variable resistor as a feedback element. However, such a pot cannot be set by the pot set servo unless its low (LO) side is grounded via a patch cord. An attempt to set an ungrounded pot will result in the same condition as with the invalid servo pot address.

The pot servo control circuitry in the EAI-580 must be placed under digital control before any pots can be set from the digital computer. This is accomplished by depressing the DIG pushbutton on the analog computer control panel. When depressed, this pushbutton disables analog computer control of the servo set potentiometers, placing them under complete control of the digital computer. Also, the analog computer must be set to the SP mode before the servo can operate. The mode may be selected manually or as described in Chapter III.

5.3 Coefficient Selection

A digital value must be provided to the D/A converter before the pot can be set. The converter then provides the necessary analog voltage to the input of the servo amplifier. If the same coefficient is to be set into more than one pot consecutively, then the D/A converter

register only needs to be loaded once with the value. The register will hold a selected coefficient until it is changed by another command from the digital computer.

The servo pot control section of the hybrid interface shares the D/A converter with the analog output section. Once a pot set operation is started, the converter output must remain constant until it is completed. Otherwise, the servo will stop with the wrong coefficient. Therefore, the analog outputs are not usable during pot set operations.

The coefficient selection is accomplished by setting the proper bits in the A-register and executing an OUT O'1100' command. There is no need to wait for the D/A converter to settle before going on to select the pot address. This is because the pot address select logic has a built-in wait interval before starting the servo.

Table 5.1 shows the bit weight values for the specification of the pot coefficient. Note that a coefficient of 1.0 cannot be obtained. The selection procedure can be summarized as follows:

Procedure:
1. Load the A-register with coefficient.
2. Execute OUT O'1100' command.
3. Go on to select pot address.

5.4 Pot Address Selection and Servo Start

The pot address is selected by setting the proper bits in the A-register and executing an OUT O'1400' command. In addition, the execution of this command signals the servo to start the pot set operation. The pot address is coded in BCD form in the A-register.

The command causes the contents of the A-register to be loaded into

the address select register. Also, a timer is started in the hybrid interface to allow time for the relay contacts to settle to the new address. After this timer finishes, the hybrid interface sends a start pulse to the servo amplifier.

Table 5.2 shows the bit weight value for the specification of the pot address. The pot addressing procedure can be summarized as follows:

Procedure:
1. Load the A-register with address.
2. Execute OUT O'1400' command.
3. Wait for servo completion.

5.5 Test for Servo Ready

A test can be made by the GE-4020 to ascertain whether the hybrid is "ready" to set a pot. This is accomplished with the JNR O'1400' command. The command will indicate "not ready" if any of the following conditions exist:

1. The DIG pushbutton is not depressed.
2. The address select relay contacts are settling to a new address.
3. The pot servo is setting a pot.

If none of these conditions exist, then the test command will indicate "ready", and the hybrid interface can start a pot set operation. The command can be used at the beginning of a series of pot setting actions to assure that the servo is under digital control. It can also be used to wait for the completion of each pot operation.

The action of the ready test command can be summarized as follows:

Procedure: Execute JNR 0'1400' command.

Result : The command following the JNR instruction will be executed if the pot section is ready to set a pot. Otherwise, this command will be skipped over, and the second command after the JNR instruction will be executed.

To avoid tying up the processor with a wait loop whenever a pot is slewing, an interrupt is provided to signal the completion of the operation. This interrupt is explained in Chapter XI. The programming environment in the processor will determine whether the interrupt or the JNR command should be used.

5.6 Pot Set Abort

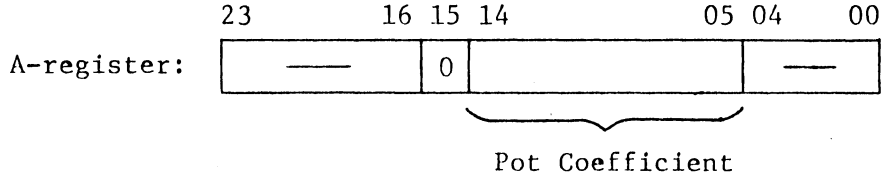
The command ABT 0'1400' is used to abort a pot set operation. This command will turn off the servo amplifier.

Although an automatic circuit exists in the EAI-580 to abort an unsuccessful pot operation, the hybrid interface does not send any abort status to the digital computer. Thus, when the servo shuts down, the JNR ready test will indicate "ready", giving the misleading implication that the operation was successful.

This problem may be avoided by counting the elapsed time of a pot operation in the GE-4020 program. Since the pots normally take less than 5 seconds to set, the program can abort the pot servo after about 8 seconds. In this way, the automatic timer is never allowed to time out. The purpose of the automatic timer is to reset the servo ampli-

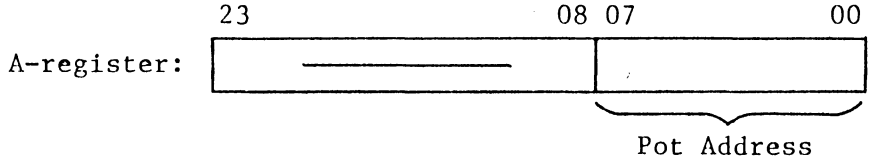
fier when it is started by a transient, since the servo start logic is sensitive to noise.

Table 5.1 A-register Bit Weights for Pot Coefficient



A-register bit	Pot Coefficient	Fractional Expression
14	0.5	1/2
13	0.25	1/4
12	0.125	1/8
11	0.0625	1/16
10	0.03125	1/32
09	0.015625	1/64
08	0.007812	1/128
07	0.003906	1/256
06	0.001953	1/512
05	0.000976	1/1024

Table 5.2 A-register Bit Weights for Pot Address



A-register bit	Pot Address Weight
07	80
06	40
05	20
04	10
03	8
02	4
01	2
00	1

Chapter VI

Digitally Controlled Attenuators

6.1 Introduction

The hybrid interface provides digital communication with a set of digitally controlled attenuators, or "DCA's". A DCA is a coefficient multiplying type of D/A converter which can be used in the same manner as a servo-set potentiometer. Since the servo potentiometers take several seconds for a coefficient change, they are suited only for long term setup applications. The DCA provides the added features of high speed and high resolution. They can be used for dynamic variation of problem variables. The DCA's are located in potentiometer patch fields 40 through 59.

Figure 6.1 is a block diagram of the DCA's and the hybrid interface logic. For simplicity, most of the control logic is not shown. Each DCA has its own coefficient register and D/A converter. The coefficient register performs the signal reconstruction function by holding the selected value until a new selection is performed. This method constitutes a zero-order hold reconstruction [4]. In addition, this hold method guarantees that there will be no leakage drift in the output voltage over a long hold period. The D/A converter is a current summing-junction operational amplifier type, with the reference voltage supplied from the input HI pin on the analog patch panel [1].

The connection between the data lines to the DCA rack and the desired DCA is selected by the address register. The address register is loaded by a command from the digital computer. The selected DCA may be loaded with a coefficient by another digital computer command.

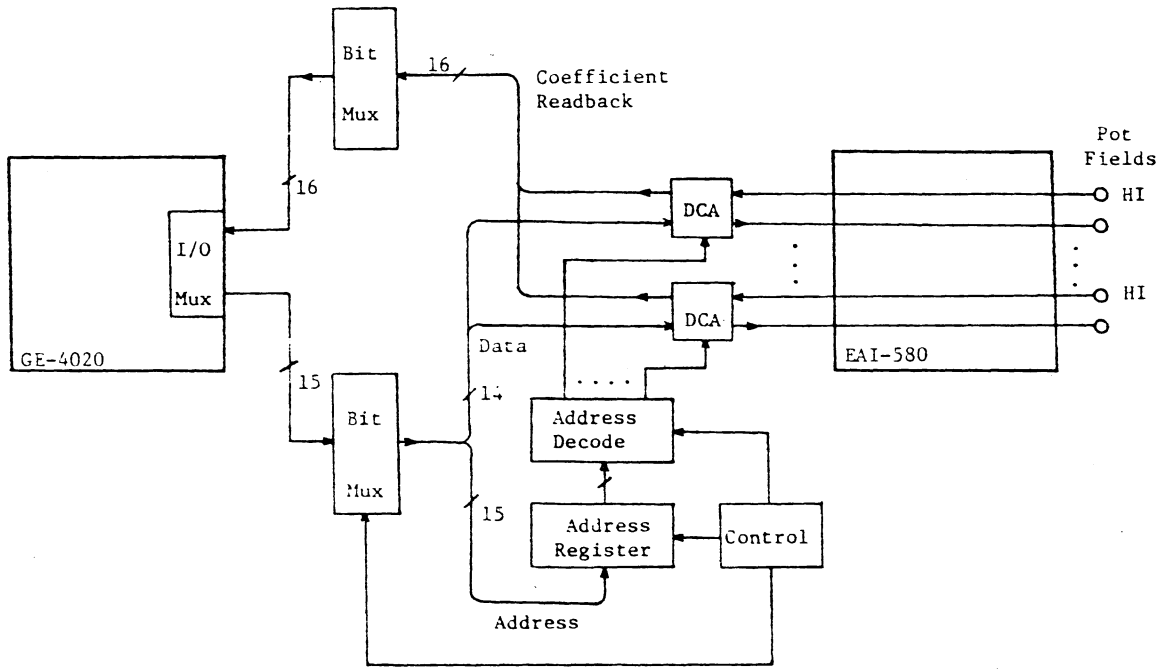


Figure 6.1 DCA Control Interface

The contents of the coefficient register can also be read back into the GE-4020 by an input command. The address register is only controlled by the digital program. Therefore, multiple operations can be performed on a particular DCA without reloading the address register each time.

The digital multiplexers in the interface provide the mapping between GE-4020 bit positions and the DCA data lines. Both data and address information are conveyed by the output multiplexer. The coefficient readback data is conveyed by the input multiplexer.

The GE-4020 "TIM/TOM" feature, which emulates input or output commands for table I/O, is described in Chapter XI. This feature is not provided for control of the DCA's. The operation of a DCA unit is faster than the command sequence which is used to control it. Therefore, no wait intervals are necessary. The table I/O feature would only lengthen the operation time.

A DCA has a resolution of 14 bits. There is no provision for a sign bit, so the coefficient is always expressed in positive magnitude form. However, the DCA is an inverting device. Therefore, the output Y of a DCA can be expressed as a function of its input X and coefficient K according to the following formula:

$$Y = -K\,X$$

The coefficient may range from 0.0 to 0.99994. A 1.0 coefficient would only be possible with the addition of a fifteenth coefficient bit, due to the bit weight scheme utilized. The tradeoffs of this scheme were discussed in Chapter II. A unity coefficient can be approximated with adequate accuracy by using a coefficient with all bits on.

6.2 Specifications

Specifications for the DCA circuit are as follows:

Settling time to within 5%	:	10 microseconds
Settling time to within 1%	:	20 microseconds
Output voltage range	:	+10 to -10 volts
Output impedance	:	100 ohms

6.3 Use of the DCA's

The DCA coefficients cannot be selected with the EAI-580 pot servo amplifier. They are under complete control of the GE-4020 digital computer. However, a DCA may be selected for analog readout when the "DIG" pushbutton is not depressed. Under this condition, the "P" pushbutton should be depressed for pot readout to the digital voltmeter. The voltmeter will read the output signal of the DCA which is selected by the address pushbuttons. Even when the analog computer is in SP mode, this output will be the coefficient multiplied by the existing input after inversion. The SP mode does not place a reference voltage on the high side (input) of the DCA. Therefore, the user must patch a 10 volt reference into the DCA to read its coefficient directly from the analog computer.

A DCA with an address which ends in the digit "4" does not need to have the LO pin grounded on the patch panel, and operates identically to the other DCA's. In fact, the patch panel LO pin is not connected to anything internally. Therefore, the DCA cannot be used as a variable resistive element in the manner of a servo-set potentiometer.

6.4 DCA Control

The DCA control section of the hybrid interface provides two GE-4020 commands. The DCA address selection is performed by setting the appropriate bits in the A-register and executing an OPR 0'1401' command. This command loads the address into the address register in the DCA controller. This address will not change until another such command is executed. The DCA coefficient is then selected by setting the appropriate bits in the A-register and executing an OUT 0'1401' command. The value will be loaded into the coefficient register of the addressed DCA. The OPR command for address selection does not need to be executed again until it is desired to address a different DCA.

Since the settling time of a DCA is comparable to the instruction execution time of the command sequence used to control it, there is no need for a ready test command. By the time a program is ready to update a DCA value, the DCA will have already settled from the previous output command.

The DCA address is formatted in bits 07 through 00 of the A-register. The address field is a natural binary form of the DCA patch panel address, with an offset to make the address of zero correspond to the first DCA. Thus, the A-register address field can be calculated by subtracting 40 from the DCA panel address, and forming the binary representation of the resulting decimal number. Table 6.1 summarizes the address specification. The following procedure is used to load the address register:

Procedure: 1. Load A-register with the number corresponding to the desired DCA address.

2. Execute OPR O'1401' command.

The DCA coefficient is formatted in bits 14 through 01 of the A-register. This alignment was chosen to dedicate the most significant bit to the same bit position as used for the servo pot and analog output specifications. Table 6.2 shows the coefficient format. The following procedure is used to load the coefficient register of the selected DCA:

Procedure: 1. Load A-register with coefficient.

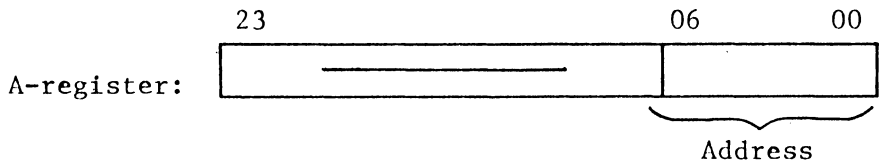
2. Execute OUT O'1401' command.

6.5 DCA Coefficient Readback

The hybrid interface provides a GE-4020 command which can be used to read the coefficient of a DCA. The DCA selected by the address register is read by an IN O'1401' command, and the coefficient is placed into the A-register. Table 6.3 shows the format for the result.

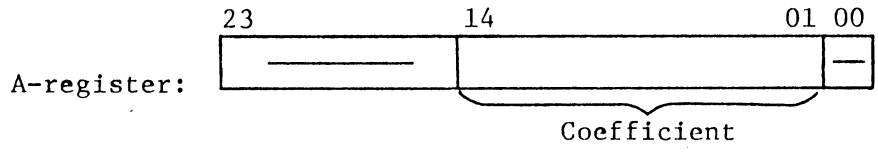
The coefficient readback feature is useful when a DCA is used to provide an output which is a function of its previous output value. Instead of referencing a memory cell containing the previous value, the program simply reads back the value from the DCA. However, this procedure will only save program time if the addressing scheme for retrieving the value from core is based on indirect addressing. The execution time for an input command to the DCA unit is longer than for a direct memory reference. Also, if more than one previous coefficient is needed to calculate the new coefficient, the readback procedure has little benefit.

Table 6.1 DCA Address Specification



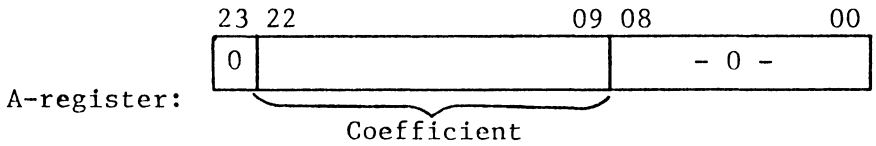
Decimal Value of Bits 06 through 00	DCA Address
0	40
1	41
2	42
3	43
:	:
9	49
10	50
:	:
19	59

Table 6.2 A-register Bit Weights for DCA Coefficient



A-register bit	DCA coefficient (fraction)	DCA coefficient (decimal)
14	1/2	0.5
13	1/4	0.25
12	1/8	0.125
11	1/16	0.0625
10	1/32	0.03125
09	1/64	0.015625
08	1/128	0.007812
07	1/256	0.003906
06	1/512	0.001953
05	1/1024	0.000976
04	1/2048	0.000488
03	1/4096	0.000244
02	1/8192	0.000122
01	1/16384	0.000061

Table 6.3 DCA Coefficient Readback Format



A-register bit	DCA coefficient (fraction)	DCA coefficient (decimal)
22	1/2	0.5
21	1/4	0.25
20	1/8	0.125
19	1/16	0.0625
18	1/32	0.03125
17	1/64	0.015625
16	1/128	0.007812
15	1/256	0.003906
14	1/512	0.001953
13	1/1024	0.000976
12	1/2048	0.000488
11	1/4096	0.000244
10	1/8192	0.000122
09	1/16384	0.000061

Chapter VII

Analog Outputs

7.1 Introduction

The analog output section of the hybrid interface provides a set of analog output signals from the GE-4020 digital computer to the EAI-580 analog patch panel. These outputs are medium speed operation outputs with a resolution of 10 bits plus a sign bit. Thus, they have a lower resolution than DCA's, but can provide both positive and negative outputs under digital control. They are slower than DCA's, but still much faster than servo-set potentiometers for providing analog signals. The analog outputs do not provide an input for coefficient multiplication, and are strictly voltage outputs. Thus they can only be used to provide constants or forcing functions to an analog program. Either DCA's or servo-set potentiometers must be used for state variable parameter variation.

Figure 7.1 is a block diagram of the analog output section of the hybrid interface. It utilizes a single D/A converter, an address register to select the analog output circuit for connection to the converter, and an analog sample/hold circuit to each output for signal reconstruction. Since the outputs are produced on sample/hold circuits, the analog signal is typical of zero-order hold reconstruction [4]. However, the outputs will exhibit a leakage drift during long hold periods, as the sample/hold circuit uses a capacitor for storage of the sampled voltage. The outputs do not have filters for elimination of high frequency components caused by the sampling process. If filtering of an output is needed before application to the analog program,

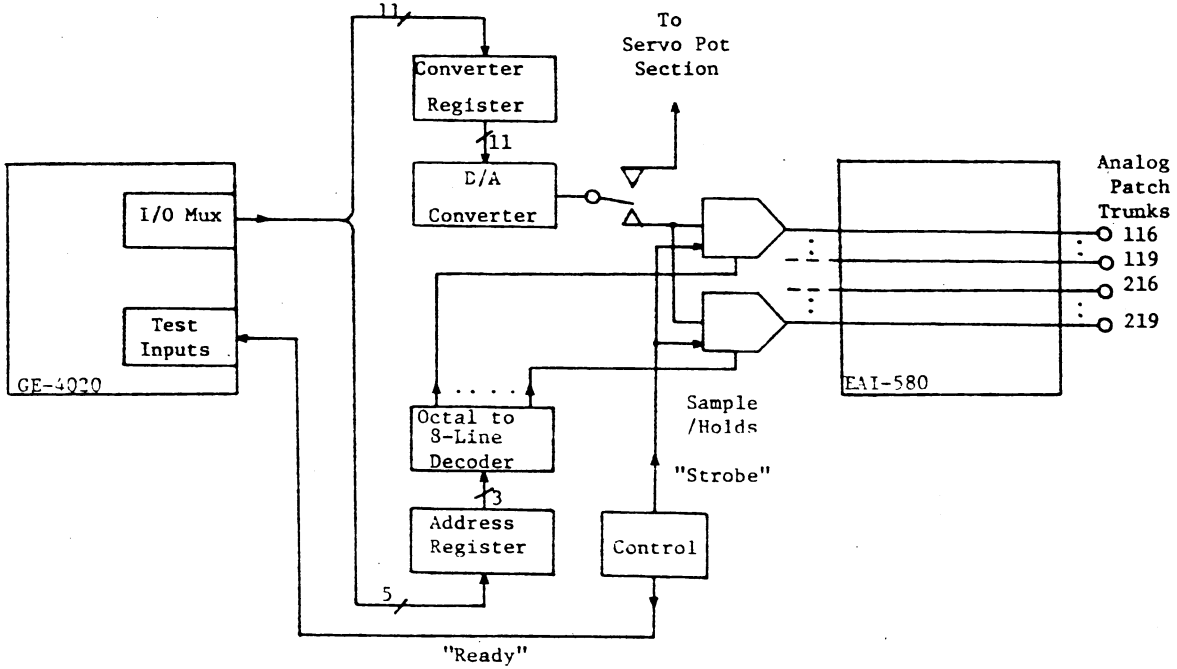


Figure 7.1 Analog Output Section

the filtering should be performed on the patch panel.

A digital computer command is used to transfer the A-register contents to the output address and D/A converter registers. This command also starts a timer to allow time for the converter to settle to the selected value. At the end of the timeout, the decoded output of the address register provides a strobe (sample) pulse to the selected sample/hold circuit. The input to each sample/hold circuit is fed from the converter output. Therefore, the selected output circuit will be updated to the D/A conversion value. Other analog outputs are not affected.

A ready test command is provided to test the status of the ready line in the analog output logic. This signal is derived from the converter timer. When the timer is inactive and no sample/hold circuits are sampling, the converter is ready. Otherwise, it is busy, and the digital computer must wait.

Chapter XI describes the "TIM/TOM" table I/O feature of the GE-4020. The hybrid interface provides the capability of using this automatic I/O feature for analog outputs. A set of value/address specification words can be placed into a table in memory. A digital computer command is then used to start automatic output. An interrupt is provided to allow the programmer to wait for completion without repetatively executing a ready test command. This feature is described in detail in Chapter XI.

The D/A converter uses the current summing-junction operational amplifier method of conversion [1]. The D/A converter has a resolution of 10 bits plus a sign bit. For a positive output, the sign bit

is off (0), and the remaining bits constitute a natural binary representation of the magnitude. For a negative output, the sign bit is on (1), and the remaining bits are a negative number in two's-complement form. Only the converter value bit field is complemented, however.

As mentioned in Chapter V, the D/A converter is also used for servo pot set operations. Therefore, pot set and analog output operations may not occur at the same time. For instance, after a pot set operation is initiated, it must be completed before the analog outputs can be updated. Also, if the analog computer is in the SP mode, the D/A converter does not connect to the analog output multiplexer. Therefore, a time interval for relay settling should be allowed when switching from SP mode to any other mode. This time should be approximately 5 milliseconds. This also allows time for certain switching relays in the EAI-580 to settle. Analog outputs are not affected by any other hybrid operations.

The analog outputs are available at trunks 116 through 119 and trunks 216 through 219 on the analog patch panel. The multiplexer can be expanded to provide more analog outputs in the future.

7.2 Specifications

Specifications for the analog output section are as follows:

Approximate absolute accuracy	:	±0.5%
Time required to complete update of an analog output	:	250 microseconds
Drift rate of an analog output in the "hold" state	:	0.1% per second

(Note: Drift is toward 0 volts.)

Output slope while changing from one value to another	:	0.15 volts per microsecond
Output voltage range	:	+10 volts to -10 volts
Output impedance	:	2 ohms
Maximum output noise	:	5 millivolts R.M.S.

7.3 Analog Output Operation

Unless automatic table I/O is used, an analog output is specified by loading the A-register in the GE-4020 with the proper bits to specify both the analog output address and the desired output value. Table 7.1 shows the format for this specification word. The specified analog output is updated by executing an OUT O'1100' command. Refer to Chapter XI for the procedure when using automatic table output.

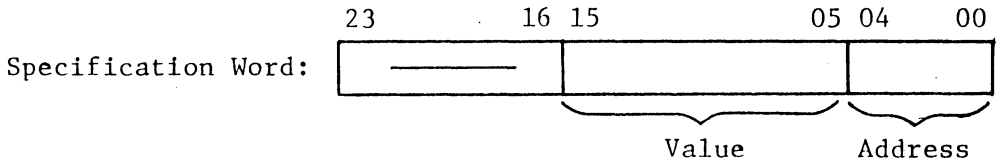
Since the update operation takes 250 microseconds for completion, a command is provided which will test for the completion of the analog output update. This command is the JNR O'1100' instruction. This instruction will either execute or skip over the next instruction, depending on the status of the D/A converter control logic.

The operation of analog outputs can be summarized by the following procedure:

Procedure:
1. Set the desired address and value fields into the A-register.
2. Execute OUT O'1100' command. Converter is now started.
3. Execute JNR O'1100' command.

Result : If conversion is not complete, the second instruction after the JNR command will be executed.
Before starting another conversion, the program should go to step 3 again.
If conversion is complete, the first instruction after the JNR command will be executed.

Table 7.1 Analog Output Value and Address Format



Bit	Analog Output Voltage	Analog Output EAI-580 Units
15	SIGN	SIGN
14	5.0	1/2
13	2.5	1/4
12	1.25	1/8
11	0.625	1/16
10	0.3125	1/32
09	0.15625	1/64
08	0.078125	1/128
07	0.0390625	1/256
06	0.01953125	1/512
05	0.009765625	1/1024

SIGN Bit: 1 = Negative, bits 14 through 05 are in 2's complement form.

0 = Positive, bits 14 through 05 are in positive binary form as shown.

Value of Bits 04 through 00	Analog Panel TRK
0	116
1	117
2	118
3	119
4	216
5	217
6	218
7	219

Chapter VIII

Analog Inputs

8.1 Introduction

The analog input section of the hybrid interface allows the digital computer to sample and digitize analog voltages present on analog input trunks on the EAI-580 analog patch panel. It operates independently from the other hybrid sections. It utilizes a single A/D converter, a sample/hold circuit to hold the signal for conversion, and a digitally-addressable input selection analog multiplexer.

A block diagram of the analog input section is shown in Figure 8.1. The address of the input trunk to be sampled is placed into the address register by the proper GE-4020 command. The contents of this register are decoded in the interface to control the input selection multiplexer. Thus, one input "switch" is always closed. However, the common side of the multiplexer is not connected to anything until the A/D conversion is started. When a conversion process is started, the sample/hold circuit samples the common side of the multiplexer, thus obtaining the signal presently selected by the address register. The A/D converter is also started. The sample/hold circuit is a zero-order hold which is used to keep the sampled value constant for the time duration of A/D conversion [5]. When the conversion is complete, the resulting digital value in the output register may be brought into the digital computer, and another input sample may begin.

The A/D converter has a resolution of 11 bits plus a sign bit. For a positive input, the sign bit is off (0), and the remaining bits constitute a natural binary representation of the magnitude. For a

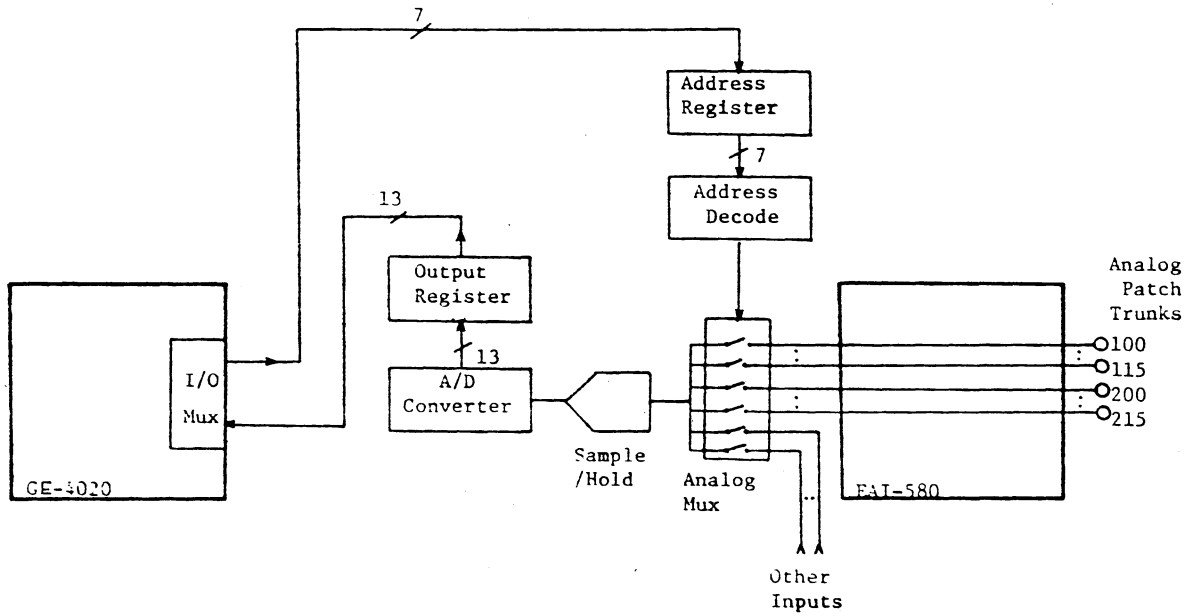


Figure 8.1 Analog Input Section

negative input, the sign bit is on (1), and the remaining bits are a negative number in two's-complement form. The A/D converter uses a successive-approximation technique which starts with the most significant bit (sign) and works down to the least significant bit. The conversion time is a constant, and is independent of the input voltage [1].

The analog input multiplexer will handle 128 analog inputs. Out of these, 32 inputs go to the trunk field of the analog patch panel. Two additional inputs are used for amplifier output and derivative readout. Also, twenty inputs are available in remote cable RT-1. The remaining analog inputs are implemented, but are not connected to anything outside the hybrid interface.

The analog multiplexer is addressed by the decoded output from the address register. The address for the multiplex switches is coded as a binary value from 0 to octal 177 (decimal 127).

Shannon's sampling theorem shows that the sampling frequency of an analog input sampler must always be greater than twice the frequency of the highest Fourier transform component present in the input signal [6]. Since the A/D conversion time is fixed, the theorem is turned around to show that the input signal must contain no spectral components above half of the analog sampling frequency. The analog sampling frequency in this case is the rate of conversion applied to a particular input, rather than to the A/D converter. If more than one analog input is continuously sampled, the sampling rate for a particular input will be the A/D conversion rate divided by the number of sampled inputs.

If the sampling criterion is violated, the result will be an effective "folding" over of the frequency axis. Another term for this effect is "aliasing". Frequency components will be present in the digital samples which were not present in the original signal [3]. If a signal in the analog problem has any frequency components which violate this criterion, the signal should be pre-filtered before application to an analog input trunk. No input filtering is implemented, so any necessary pre-filtering will have to be performed on the patch panel.

Another sampling consideration is related to the fact that the A/D converter can only handle one signal at a time. If the hybrid programmer wants to acquire the "state" of the analog program by sampling a number of state variables, a "skew" effect will be exhibited between individual variables [1]. Only one state variable can be sampled at one instant, so each variable will be separated in time from the previously sampled variable. Therefore, the acquired state variables will not represent the exact state of the problem at a particular instant. If a sufficiently large number of state variables are involved, the skew may cause significant error.

This problem could have been eliminated by placing a separate sample/hold circuit at each analog input. However, accurate analog sample/hold circuits are too bulky and expensive to be used for a large number of inputs. Also, some analog inputs may not be related to a state variable group, and the programmer must be able to handle their sampling separately. If analog sample skew represents a significant error, the sample/hold circuits on the analog panel may be used

to latch the state variables at the desired time. Then the A/D converter can leisurely sample each latched state variable.

Two commands are available to start the conversion process. The same command which loads the address register also initiates a conversion. In addition, a separate command is available which will initiate a conversion without modifying the address. This allows the programmer to take successive samples of the same input trunk without specifying the address each time. A ready test command is provided to test for completion of the converter. An input command is available to bring the converted value into the GE-4020.

8.2 Specifications

The specifications for the analog input section are as follows:

Approximate absolute accuracy	:	±0.25%
Total input conversion time	:	500 microseconds
Input voltage range	:	+10 volts to -10 volts (+12 volts to -12 volts maximum)
Input impedance	:	100 K ohms

8.3 Analog Input Operation

An analog input address is specified by loading the A-register in the GE-4020 with the proper bits to specify the trunk, and executing an OUT 0'1200' command. The execution of this command loads the address register with the specified address, and starts the A/D

converter. If it is desired to use the contents of the address register without change, the command OPR O'1200' is used.

Table 8.1 shows the bit assignments for trunk address specification. Bits 09 through 03 are used as addressing bits, and the remainder of the word is ignored. For trunks 100 through 115, the value of the addressing field (09 through 03) can be calculated by subtracting decimal 100 from the trunk number. For trunks 200 through 215, the value of the addressing field can be calculated by subtracting decimal 72 from the trunk number. For the remote cable inputs, the address field values range from octal 50 to octal 73.

The JNR O'1200' ready test command can now be used to determine the status of the A/D conversion. If a conversion is in progress, the JNR command will cause a jump to the current location plus 2. If the converter has finished, execution of the JNR command simply advances the program count to the current location plus 1. The command may constitute a "wait" loop. However, it is often desirable to allow the digital computer to do something else during the conversion. At any rate, it is necessary to test for a "ready" converter status before transferring the output of the A/D converter into the A-register and starting another sample.

Once conversion is complete, the resulting digital value must be brought into the digital computer by executing an IN O'1200' command. The digital representation of the signal value will then be transferred into the A-register and can be processed by the program. Table 8.2 shows the bit assignment for the resulting input value.

In addition to indicating the value of the analog input, one bit of the A/D output is placed into the resulting value word to indicate that the converter has produced an overflow. Converter overflow occurs when the input signal sample is a value which is too large (either negative or positive) to be represented by 11 bits plus a sign bit. If an overflow has occurred, bit 00 will be off (0).

A command is also provided to test for converter overflow. Execution of the JNE O'1200' error test will cause the program count to skip the next instruction if no overflow occurred. If an overflow has occurred, the command causes the program count to execute the next instruction. Whether the overflow status is determined by a bit test of the converter result word or by execution of this JNE command, the JNE command must be executed once to clear the error condition. For this reason, the overflow bit test is not recommended for overflow determination. However, the overflow bit becomes important when the analog input section is operated by program interrupts. It enables the program to determine which trunk address out of a table of trunks caused the overflow.

The "TIM/TOM" table I/O feature is described in detail in Chapter XI. Several interrupts are provided in the analog input section to facilitate automatic table operation of analog sampling. Refer to Chapter XI for details on automatic analog input operation.

When the table I/O feature is not used, the following procedure summarizes command control of the analog inputs:

Procedure:

1a. If address update is desired, set proper address bits in the A-register and execute OUT O'1200'.

1b. If address update is not desired, execute OPR O'1200'.

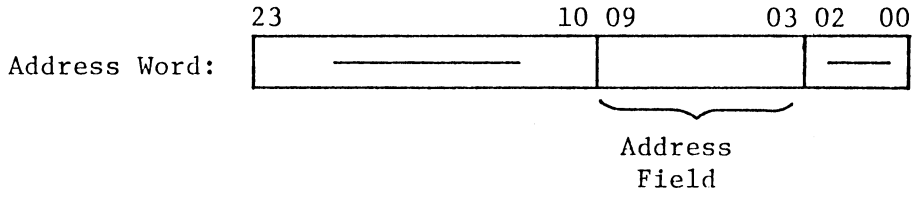
2. Either allow GE-4020 to work on some program or calculation, or go immediately to a wait loop using JNR O'1200'.

3. Execute JNR O'1200'. If the command causes a jump to P+2 (conversion not finished), go back to the JNR command. If the command causes the P count to go to P+1 (conversion complete), branch out of the loop to the next step.

4. Conversion has been finished. Transfer the converted value into the A-register with IN O'1200' command. (This step may be interchanged with step 5.)

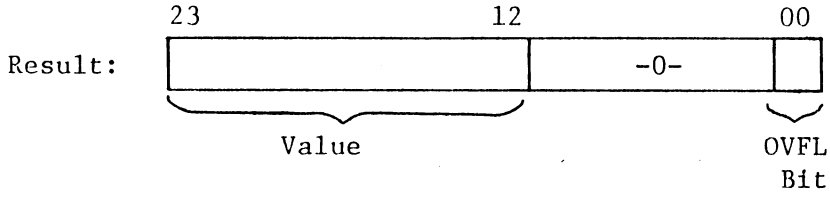
5. Test for converter overflow with JNE O'1200' command. If the command causes a jump to P+2, process the input and start another input sample (1a or 1b). If the command causes the P count to go to P+1, branch to an error handling routine to take care of the overflow. (This step may be interchanged with step 4.)

Table 8.1 Analog Input Address Specification



Address Field (octal)	Analog Patch Panel TRK ---
0	100
1	101
2	102
3	103
4	104
5	105
6	106
7	107
10	108
11	109
12	110
13	111
14	112
15	113
16	114
17	115
20	200
21	201
22	202
23	203
24	204
25	205
26	206
27	207
30	208
31	209
32	210
33	211
34	212
35	213
36	214
37	215

Table 8.2 Analog Input Result Format



Bit	Analog Input Voltage	Analog Input EAI-580 Units
23	SIGN	SIGN
22	5.0	1/2
21	2.5	1/4
20	1.25	1/8
19	0.625	1/16
18	0.3125	1/32
17	0.15625	1/64
16	0.07812	1/128
15	0.03906	1/256
14	0.01953	1/512
13	0.00976	1/1024
12	0.00488	1/2048

SIGN Bit: 1 = Negative, two's-complement form for bits 22 through 12.

0 = Positive, natural binary

OVFL Bit: 1 = Conversion overflow

0 = No overflow

Chapter IX

Amplifier Output and Derivative Sampling

9.1 Introduction

The hybrid interface provides a method for selecting and reading amplifier outputs or derivatives. The addressing which is used to select potentiometers as described in Chapter V can also be used to select certain analog patch panel components. Figure 9.1 shows a block diagram of the hybrid interface components which are used for amplifier output and derivative sampling.

When the analog computer is not under digital control, the digital voltmeter can be connected to a number of analog components. This is always the case unless the "DIG" pushbutton is depressed. The DVM control pushbuttons determine the connection. When the "P" pushbutton is depressed, the output of the potentiometer which is selected by the address pushbuttons is displayed. When the "A" pushbutton is depressed, the output of the selected amplifier is displayed. When the "D" pushbutton is depressed, the "derivative" of the selected amplifier is displayed.

The derivative is the sum of the input signals, with scale factors introduced for the gain of each input. The derivative of an amplifier is found by the following formula:

$$D = \sum_i k_i v_i \qquad \text{for all inputs } i,$$

where k_i is the gain of the input and v_i is the input voltage.

If the amplifier is used only as a summing amplifier, the derivative will simply be the inverse of the amplifier output. In this case

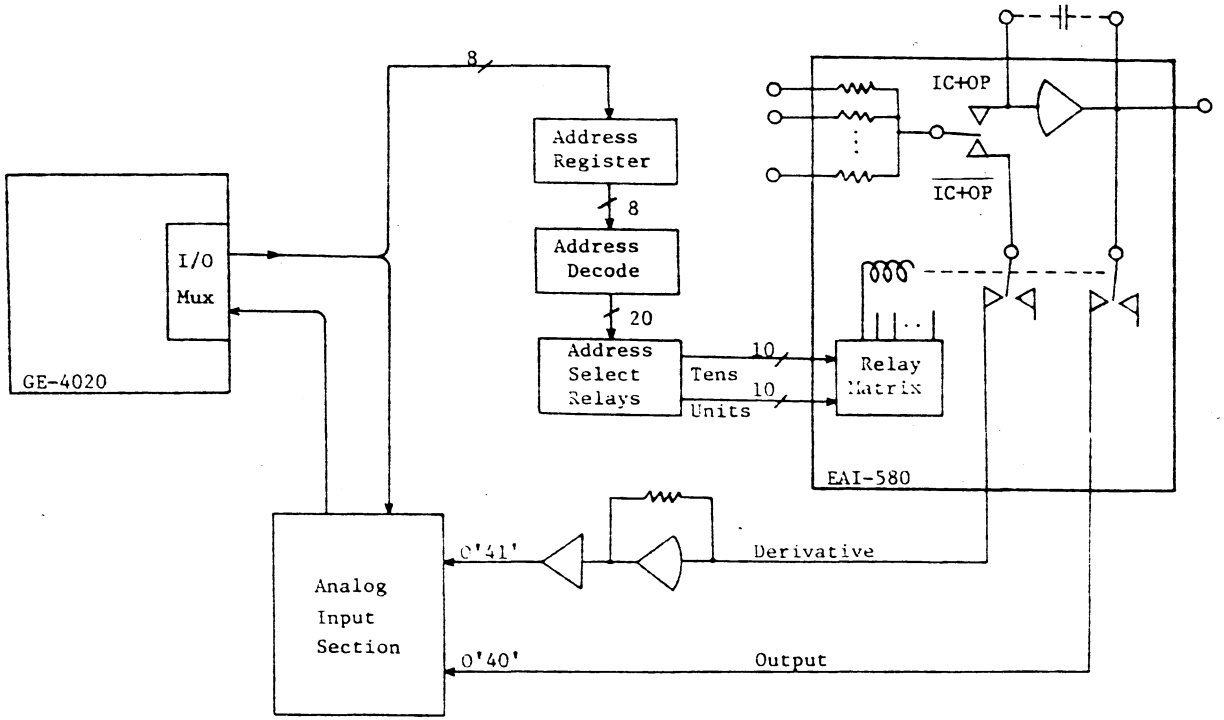


Figure 9.1 Amplifier Output and Derivative Sampling

the derivative has no significance, and therefore is not displayed by the EAI-580. However, if the amplifier is used as the active element of an integrator or track/store unit, then the derivative has significance. In the integrator case, the derivative readout will be the actual time derivative of the integrator output.

In the track/store case, the readout bears no direct relationship to a time derivative. However, the readout will still have significance in certain problems. The track/store unit can be used as a component of a sampled-data system. It performs the zero-order hold function. In the z-domain, the zero-order hold has a transfer function of unity [4]. Also, with the proper timing of the sample/hold control signal, it can be set up to perform the unit delay function. This function has a transfer operator of 1/z. When used as a sampled-data function block, the 1/z operator plays a similar role to that of the 1/s (integration) operator in a continuous data system. One use for the unit delay function is in the implementation of a discrete difference equation for digital filtering [3]. In this case, access to both inputs and outputs of the delay block can be valuable. Thus, a readout of the input sum to a track/store unit can provide important data in certain programs.

Derivative readout is also a valuable tool for program patching verification. When the analog computer is in the IC or ST modes, all integrators are held to their initial condition voltages. The initial condition of an integrator is equal to the voltage patched into its initial condition input. All other analog components operate as normal. Thus, each amplifier input sum is calculable when the patched configuration, the initial condition voltages, and the potentiometer settings

are known. By comparing the calculated derivatives with the actual derivative readouts, a mismatch will suggest a patching error.

Since derivatives of simple summing amplifiers cannot be selected for readout, the amplifier outputs should be sampled instead. The outputs will be the negative of the desired derivatives. These outputs can be used as additional comparative data during program verification.

The ST (Static Test) mode is almost identical to the IC mode. One difference is in the operation of the track/store units. During the ST mode, the track/store units are set to voltages present at their initial condition inputs. During the IC mode, track/store units operate as normal. Therefore, static test performs a complete initialization of the analog program, including any components which constitute a sampled-data system.

The other difference between the static test and initial conditions modes is that the IC mode does not permit derivative readout. The IC mode is intended only for integrator initialization, whereas the ST mode is used for complete program initialization and derivative readout.

During operation of the analog program, the HD (Hold) mode can be entered to freeze the system at its current state. The HD mode allows derivative readout of integrators, as well as output readout of all components. In this manner, instantaneous state variable values and their derivatives can be acquired.

When the analog computer is under digital control with the DIG pushbutton depressed, the analog DVM readout and addressing pushbuttons are disabled. The address select logic is then controlled by the

hybrid interface. The address select relays determine the patch panel address for readout.

When the analog computer is in the SP mode, the address relays will select both the addressed pot and the addressed amplifier. This is because the pot addresses and amplifier addresses on the patch panel are not mutually exclusive. The addressed pot is selected for operation with the servo amplifier. The addressed amplifier is selected for derivative readout. However, the amplifier output line is held to zero. If an integrator or track/store derivative is selected for readout in the set pot mode, it must be remembered that the addressed potentiometer is taken out of the analog circuit for servo operation. Therefore, the analog network may be effectively altered during this mode.

Two signal lines are brought into the hybrid interface from the EAI-580 readout section. One line is connected to the output of the selected amplifier. The other line is connected to the derivative select line. If the analog mode is not IC or OP, then the selected amplifier input is disconnected, and its summing-junction resistors are switched to the derivative select line. The derivative readout operational amplifier takes the place of the disconnected amplifier and provides an output voltage which is the negative sum of the inputs. An additional amplifier then converts this signal to a positive sum. The resulting amplifier output and derivative readout signals are brought into the digital computer by way of two dedicated analog inputs.

The operation of the amplifier output and derivative readout lines is summarized for each analog mode in Table 9.1.

Table 9.1 Amplifier Output and Derivative Operation

Amplifier Readout

Mode	Amplifier	Integrator	Track/Store
IC	$-\sum$	IC	$f(\sum)$
HD	$-\sum$	$f(t_H)$	$f(\sum)$
OP	$-\sum$	$f(\sum, t)$	$f(\sum)$
SP	0	0	0
ST	$-\sum$	IC	IC

Derivative Readout

Mode	Amplifier	Integrator	Track/Store
IC	0	0	0
HD	0	$\sum$	0
OP	0	0	0
SP	0	$\sum$	$\sum$
ST	0	$\sum$	$\sum$+IC

Explanation:

$\sum$ is the notation for $\sum_i k_i v_i$

where k_i = gain of input i, v_i = input voltage to input i.

f represents the operation function of the analog component.

IC is the initial condition input voltage.

t_H is the time instant at which the HD mode was activated.

9.2 Output and Derivative Sampling

The address relays are set by loading the A-register with the appropriate address specification and executing an OUT 0'1400' command. Table 9.2 shows the bit weights for address selection. If the analog computer is in the SP mode, this command will also start a pot set operation. This problem can be avoided by changing the analog mode, setting the address relays, and then going back to the SP mode. The digital program should check that the analog mode is not SP before changing the address relays for amplifier readout.

Since the address relays need around 5 milliseconds to settle when they are changed, a timer is started by the OUT command. A ready test command is provided which tests the status of this timer. It is the same command which is used for the servo potentiometer ready test. If the analog computer mode is not SP, the test will only indicate the status of the relay settling timer.

Once an amplifier is addressed and the relays have settled, the output and derivative readout signals can be sampled by the analog input section. They may be sampled more than once, and at any desired rate. The analog computer mode can also be changed between samples.

The EAI-580 has only one timing restriction on operations which change the analog computer mode during a running program. The "hold" function in the integrators and track/store units is performed with relays. Therefore, when the analog mode is switched to HD, approximately 5 milliseconds must be provided before attempting amplifier readout. This time interval will have to be set up by the hybrid

programmer, as a timer is not provided. The IC and OP modes are switched electronically, and pose no timing problems.

The amplifier output readout line is connected to the analog input which corresponds to an address field of octal 40. The amplifier derivative readout line is connected to the analog input which corresponds to an address field of octal 41. Refer to Chapter VIII for operation of the analog input section of the hybrid interface.

The procedure for output and derivative readback can be summarized as follows:

Procedure: 1. If address is to be changed:

a. Make sure analog mode is not SP.

b. Load A-register with address specification as per Table 9.2.

c. Execute OUT 0'1400' command.

d. Execute JNR 0'1400' command.

If DIG pushbutton is not depressed or address relays are settling, the command will skip over the following instruction. Otherwise, the next instruction will be executed.

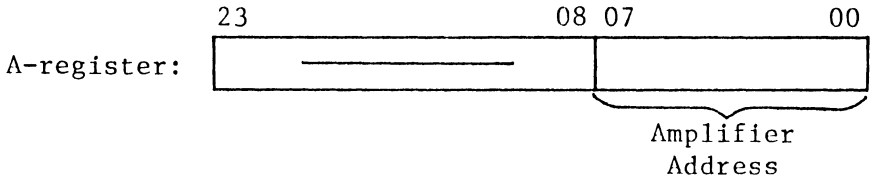
e. If JNR indicates "not ready", go to step 1d.

2. Address select is now finished.

a. For amplifier output readout, set address field for analog input (bits 09 through 03) to octal 40. Bring in analog value as described in Chapter VIII.

Procedure: 2. b. For derivative readout, set address field for analog input (bits 09 through 03) to octal 41. Bring in analog value as described in Chapter VIII.

Table 9.2 A-register Bit Weights for Amplifier Address



A-register Bit	Amplifier Address Weight
07	80
06	40
05	20
04	10
03	8
02	4
01	2
00	1

Chapter X

Digital Output and Input

10.1 Digital Output

The digital output section of the hybrid interface enables the digital computer to transfer digital signals to the digital patch panel of the EAI-580. Sixteen digital trunks can be addressed by the execution of an OUT O'1101' command, with A-register bits corresponding to logic trunks. All bits which are on (1) will place a positive (1) pulse of 16 microseconds duration on the corresponding output trunk. If the EAI-580 program is to store this condition, the trunk will have to be connected to a latching circuit. All trunks corresponding to A-register bits which are off (0) will remain low (0). One output command can output to all sixteen trunks simultaneously. Refer to Table 10.1 for the correspondence between A-register bits and digital panel trunks.

The digital output procedure can be summarized as follows:

Procedure:
1. Set those bits in A-register which correspond to desired "1" logic signals on digital panel.
2. Execute OUT O'1101' command.

10.2 Digital Output Specifications

Specifications for the digital output are as follows:

Output polarity	:	Non-inverted (1 bit gives 1 on trunk)
Output pulse width	:	16 microseconds
Output type	:	TTL (+5 and 0 volts)

Table 10.1 Digital Output/Input Correlation to A-register Bits

A-register Bit	Digital Panel Output TRK
15	200
14	201
13	202
12	203
11	204
10	205
09	206
08	207
07	208
06	209
05	210
04	211
03	212
02	213
01	214
00	215

A-register Bit	Digital Panel Input TRK
23	100
22	101
21	102
20	103
19	104
18	105
17	106
16	107
15	108
14	109
13	110
12	111
11	112
10	113
09	114
08	115

10.3 Digital Input

The digital input section of the hybrid interface enables the digital computer to acquire digital signals from the digital patch panel of the EAI-580. Sixteen digital trunks can be sampled by the execution of an IN O'1101' command. The states of the logic inputs are then represented by bits in the A-register. All trunks which are at a logic level of 1 will place a 1 bit at the corresponding bit position in the A-register, and all 0 trunks will be represented by a 0 in the A-register. One input command will sample all sixteen trunks simultaneously. Refer to Table 10.1 for the correspondence between A-register bits and digital panel trunks.

The digital input gates in the hybrid interface are TTL logic gates, and have an internal pullup to logic 1. Therefore, all unpatched digital input trunks will display a logic 1 state.

The digital input procedure can be summarized as follows:

Procedure:
1. Execute IN O'1101' command.
2. Bits in A-register represent logic states of the digital input trunks.

10.4 Digital Input Specifications

Specifications for the digital inputs are as follows:

Input polarity	:	Non-inverted (1 bit = logic 1 on TRK)
Input sampling time	:	27 microseconds
Input type	:	TTL (+5 and 0 volts)
Input load	:	1 TTL load

Chapter XI

Hybrid Interrupts

11.1 Introduction to GE-4020 Interrupts

Digital computers which are tailored to interaction with a real-time analog environment utilize a hardware concept known as the program interrupt. A program interrupt is a hardware function implemented within the CPU which allows a signal from the external environment to interrupt the program currently executing and branch the CPU to another task or routine. This function is important to the quick handling of high-speed input and output operations. The active program does not need to periodically check for I/O completion, since any change in status will automatically interrupt the processor. Thus, the processor can utilize more of its time performing useful work.

The GE-4020 computer provides 64 separate interrupts, each with an individual dedicated core address for a response instruction. The interrupt response locations are fixed at core addresses 200 through 277 octal. Separate interrupts allow the ability to perform many simultaneous I/O operations with minimum interaction. Each interrupt has a comparative hardware priority which is based on its corresponding core address, with the lowest core address having the highest priority. The hardware prioritization of interrupts is necessary to resolve a simultaneous occurrence of more than one interrupt. When this condition occurs, only the highest priority one is allowed to signal the processor. The others are stored in the interrupt hardware until they have priority. However, the hardware priority assignment has very little effect on the time it takes for a program to receive an inter-

rupt once it is triggered. The organization of the software in the computer is the primary consideration for system response time.

The GE-4020 has two kinds of interrupts, referred to as the TIM/TOM interrupt and the branching interrupt. The kind of interrupt assigned to a response address is determined by hardware wiring options. The TIM/TOM ("Table Input to Memory", "Table Output from Memory") interrupt is a special kind of interrupt which is used to set up automatic input or output sequencing. The contents of the response address are decoded by the CPU to determine the number and type of characters to be transferred and the core address for the transfer. The format for the TIM/TOM control word is explained in detail in the GE-4020 instruction reference manual.

The channel address assignment and the direction of transfer for a particular TIM/TOM interrupt are determined by a hard-wired diode matrix board in the GE-4020. For a TOM interrupt, the 12-bit channel address and an OUT command skeleton are brought together with the logical "or" function, and the resulting hardware generated instruction is injected into the CPU. This command causes the transfer of data from memory to a peripheral. Likewise, a TIM interrupt causes the formation and injection of an IN command, transferring data from a peripheral into memory.

Each character transfer causes an automatic update of the TIM/TOM control word, and triggers the interrupt again if more characters are to be transferred. The characters are passed by "stealing" cycles from the normal CPU sequencing, and are transferred into or out of memory using the B-register as an intermediary. The communication of the

characters with the hybrid interface is accomplished with two 24-bit data busses which connect to the parallel adder register, one for input and one for output. The parallel adder links to the B-register for TIM/TOM I/O. Since the B-register is not available to programs, the TIM/TOM operation is completely transparent to the current running program.

The branching interrupt is the kind of interrupt which is used to signal the processor to branch to a program which has been waiting for the interrupt. The core cell corresponding to a branching interrupt normally contains an SPB (Save Place and Branch) instruction, with the instruction operand specifying the address of the routine which is expecting the interrupt. When it occurs, the SPB instruction is executed, and the current P-register contents (current place) and CPU status are stored into index register X1 for later reference.

A branching interrupt is used to signal the completion of the I/O transfer of a group of characters for an I/O device. Thus, each TIM/TOM interrupt is assigned a corresponding branching "echo" interrupt. The echo interrupt is triggered when the TIM/TOM control word indicates that all characters have been transferred. This condition is referred to as TIM/TOM control word "depletion". In addition, a branching interrupt can be used to indicate a change in the status of a device or external process. Certain interrupts are used to signal error conditions, and some interrupt inputs are wired for user-defined purposes.

The GE-4020 interrupts are also divided into two classes. Those with response addresses from 200 through 237 octal are called "primary"

interrupts, while those from 240 through 277 octal are called "secondary" interrupts. The secondary interrupts are inhibited from interrupting the processor whenever the PAI (Permit Automatic Interrupts) flip-flop is reset. A program may reset the PAI flip-flop by executing the IAI1 command. Interrupts which are triggered during the inhibited state are stored in the interrupt hardware until released. The PAI flip-flop can be set back on by the PAI command, or by an LPR (Load Place and Restore) command with the appropriate operand contents.

The ability to inhibit branching interrupts is essential to the proper operation of interrupt servicing software. For this reason, branching interrupts are normally assigned to secondary response locations. When a branching interrupt transfers control from one program to another, the activated program must save any registers it will use which were also in use by the interrupted program. This is necessary so that they can be restored before control is transferred back. The registers should be saved in core locations which are dedicated for use only by the activated program. To avoid confusion, this register save operation must be completed before any further interruptions can be allowed. Therefore, it should be performed with secondary interrupts inhibited. Further, since index register X1 is dedicated for use by the SPB command to save machine status and place, its contents are automatically changed by a branching interrupt occurrence. Therefore, an interrupt-activated program must either include X1 in the register save operation, or inhibit secondary interrupts until it returns control back to the interrupted program.

In contrast, primary interrupts are not controlled by the state of the PAI flip-flop. If necessary, the execution of an IAI2 command will place all interrupts in an inhibited state, and a PAI command is used to release the interrupt system again. This kind of interrupt control is only needed for the special case of an SPB instruction placed in a branching-type primary interrupt location. This case will be covered in more detail in Section 11.2. Normally, the primary interrupt response will consist of a DMT (Decrement Memory and Test) instruction or a TIM/TOM control word. Since these response instructions do not affect program status, they do not need to be inhibited and are always placed in the primary interrupt class.

All interrupts are automatically inhibited for one instruction sequence following the execution of certain commands, such as branch and jump instructions. This is necessary to assure that the right place value is always saved by a branching interrupt occurrence. It also assures that subroutines which are called by the SPB command will have at least one inhibited instruction cycle in which to save the return address and status contained in register X1. A full list of these inhibiting commands is shown in Table 11.1.

When using interrupts, caution should be observed to avoid the occurrence of a long chain of uninterruptable instructions. For example, when a JNR ready test instruction is used to wait for a ready status on a device, the not-ready sequence of instructions should contain at least one interruptable instruction. The NOP (No Operation) instruction is often used for this purpose. If a long chain of inhibiting instructions is executed, the interrupts may be kept inhibited

for long enough to miss an interrupt event. For example, if a second interrupting signal occurs at the input of an interrupt before the first interrupting event has been answered by the CPU, then the second event will be lost. For this reason, all inhibiting operations should be kept as short in time as possible.

A DMT instruction in a branching-type interrupt location performs a special function. The contents of the operand address is decremented by one, and if the contents become negative an "echo" signal is generated in the interrupt system. When wired appropriately, this echo signal can be used to trigger another interrupt. This feature is used to perform counting functions, and is utilized in particular for the system 60 Hz clock timer. Interrupt location octal 223 is triggered by a 60 Hz signal, and the DMT response instruction is used to divide this rate down. Thus the corresponding branching-type interrupt at location octal 240 is triggered by the echo signal at some multiple of 1/60 second. To keep the clock running, the echo interrupt must place the dividing constant back into the location which is referenced by the DMT command. The value of this constant is one less than the desired divide count. Normally, the system clock timer uses a constant of 59, thus providing a timer echo interrupt at 1 second intervals.

11.2 Enhancements to the Interrupt System

Two CPU modifications have been made at V.P.I. & S.U. to enhance the operation of the GE-4020 in our hybrid and operating system environments. They can both be switched on for modified operation or off for unmodified operation under software command control. The control com-

mands and operating modes are summarized in Table 11.2.

The first modification deals with the save location for machine status and place when used by an SPB branch command in an interrupt location. When operational, the modified mode causes this status to be saved in memory cell 0 instead of index register X1 whenever the save operation is the result of a primary class interrupt. This modification is necessary when the hybrid system is used under the operating system environment. The operating system software uses index register X1 whenever it has control of the CPU, and so it must be able to inhibit interrupts from possibly destroying this register contents. Since primary class interrupts cannot be controlled by the PAI flip-flop, it was decided to dedicate a special core location for primary interrupt save operations.

The second modification deals with the conditions which turn the PAI flip-flop on and off. When operational, the modified mode causes all SPB commands to automatically set the PAI flip-flop, if it was previously reset. In the normal mode, SPB commands will reset the PAI flip-flop. The modified mode should be operational when using the operating system, since this will make it unnecessary for users to place a PAI command at the beginning of subroutines. The operating system environment thus enables user programs to be concerned with interrupt control commands only under special situations.

11.3 Analog Mode Change Interrupts

The hybrid system provides a set of interrupts to detect certain changes in the mode setting of the EAI-580 analog computer. When the

Table 11.1 List of Non-interruptable Instructions

BRU	JNR	CLZ	PAI
BTR	JCB	CMO	IAI1
BTS	JDR	CMZ	IAI2
SPB	JNO	CNO	XEC
LPR	TXH	CNZ	LDX
LDP	CLO	All Quasis	

Table 11.2 Interrupt Mode Control Commands

Command (Octal)	Operation
25000100	Set mode for primary interrupt save in memory cell 0.
25000102	Reset save mode to normal
25000104	Set mode for automatic PAI after SPB command
25000106	Reset permit mode to normal

analog mode group is set to the PP (Patch Panel) mode, the IC (Initial Condition) and OP (Operate) modes are also alternately activated. Since this operation is performed in the EAI-580, a set of interrupts is needed to allow the user program to sense this mode change without continually testing the analog computer status.

A branching-type primary interrupt, dedicated to response address octal 211, is triggered when the analog mode group changes from PP+OP to PP+IC. A corresponding branching-type secondary interrupt is dedicated to response address octal 276. It is fired from an echo signal caused by a DMT instruction at the primary interrupt location. Thus, the DMT instruction can be used to generate an interrupt to the user program when the analog mode has cycled a certain number of times. The constant placed at the location referenced by the operand of the DMT instruction determines the number of mode change cycles per program interrupt. This constant should be one less than the desired number of mode change cycles.

The mode cycling feature can be used to generate a family of system response curves. Each time the mode changes to PP+IC, the EAI-580 digital logic panel can be used to change certain system parameters in the analog program. Then when the mode changes to PP+OP, a new curve is added to the family of curves. The secondary interrupt can then be used to signal the completion of the family of curves. Of course, the user program may want to retain more of the control in the digital computer. If the user program is to be interrupted at every mode change event, then the location referenced by the DMT command should contain zero. This constant must be reestablished at every

occurrence of the secondary interrupt.

A branching-type primary interrupt is dedicated to location octal 212, and is triggered when the analog mode group changes from PP+IC to PP+OP. A corresponding branching-type secondary interrupt is dedicated to location octal 277. It is fired from an echo signal caused by a DMT instruction at location octal 212. These two interrupts are used in the same manner as the two interrupts which detect a mode change to PP+IC.

11.4 Servo-set Potentiometer Interrupt

The hybrid interface provides a branching-type secondary interrupt to indicate pot set completion. The interrupt response address is octal 250. It is triggered when the servo amplifier stops searching for a null. This means either that the pot was successfully set to the proper value or that the null was not obtainable. There is no status returned to permit the user to distinguish between these two modes of completion. The user should make sure that a valid pot set operation is specified.

The pot set completion interrupt makes it possible for the hybrid user program to release the CPU for use by other programs after starting a pot set operation. This is important in a multi-program operating system environment, since the pot set operation takes several seconds for completion.

11.5 Analog Output Interrupts

Two interrupts are provided by the hybrid system to operate the

analog output section. A TOM output interrupt is assigned to response address octal 217, and generates an OUT O'1100' command. The contents of the response address should be a TOM control word specifying the following information:

1. Number of words of analog output selection in an address/value table.
2. Address in core of the address/value selection table.
3. One character per word packing, so that the hybrid interface can use all 24 bits of the transfer.

Each word in the address/value selection table is formed in the same format as shown in Table 7.1.

The TOM interrupt enables the user to set up a table-driven update of many analog outputs with a single program operation. Once the table is established and the TOM control word is stored into the response location, the process may be started with an ACT O'1100' command. The ACT (Activate) command triggers the TOM interrupt, and the operation becomes automatic. At this point, the user can wait for a completion interrupt.

Even though the output operation is automatic, the D/A converter will still only handle one output point at a time. Therefore, there will be a time skew between the update of individual output trunks. Since the TOM interrupt may be inhibited by certain commands within the currently executing program, the time skew may vary. This factor should be considered in the determination of system response characteristics.

A completion interrupt is provided with response address octal 247. It is a branching-type secondary interrupt. This interrupt is triggered from the echo completion signal of the TOM interrupt, indicating that all of the analog output operations specified by the address/value table have been performed. The time interval from the start of a table-driven analog output to the generation of the echo interrupt is roughly the D/A conversion time of 250 microseconds multiplied by the number of words in the selection table. The JNR ready test should not be used to determine the completion of the analog output update, since the test will indicate "ready" between each D/A converter cycle.

Since the D/A converter is shared between the analog output and servo set pot sections of the hybrid interface, the two analog output interrupts are inhibited from triggering when the analog computer is in SP mode. This is necessary to avoid accidental operation of these interrupts. Also, the automatic output operation can be accidently started by the execution of the OUT O'1100' command. If the interrupts are locked out by the lockout switch on the GE-4020 console, then the analog output interrupts will not operate. Also, if the analog output TOM interrupt response is a depleted control word, then the interrupt will not start the automatic output. However, if the control word is not depleted and interrupts are active, then the A-register contents during the OUT command will be used for the first output specification, and the address/value selection table will be used for further output specifications. To avoid accidental operation, a dummy depleted control word of octal 77777777 should be stored into the TOM location if the

OUT command is to be used with interrupts active.

11.6 Analog Input Interrupts

Four interrupts are provided by the hybrid system to operate the analog input section. A TIM interrupt is provided to bring the A/D converter value into the digital computer, a TOM interrupt is provided to output address selection words to the converter, and two branching interrupts are provided to signal analog input completion. These interrupts are summarized in Table 11.3.

A primary TIM interrupt is assigned to response address octal 214, and generates the IN 0'1200' command. The contents of the response address should be a TIM control word specifying the following information:

1. Number of analog values to be sampled and brought into memory by the TIM operation at one value per word.
2. Address in core of the table to receive the input values.
3. One character per word packing, so that the hybrid interface can use all 24 bits of the transfer.

Each value which is brought into memory will follow the same format as shown in Table 8.2. The TIM interrupt is triggered automatically by the hybrid interface when the A/D converter has finished the conversion of a sample.

A primary TOM interrupt is assigned to response address octal 215, and generates the OUT 0'1200' command. The contents of the response address should be a TIM control word specifying the following information:

1. Number of words of analog input selection in an address table.
2. Address in core of the address selection table.
3. One character per word packing, so that the hybrid interface can use all 24 bits of the transfer.

Each word in the address table is formed in the same manner as in Table 8.1.

The TIM and TOM interrupts enable the user to set up a table-driven sampling of analog inputs. Once the address selection table is established and the two TIM/TOM control words are stored into the response locations, the process may be started with an ACT O'1200' command. This triggers the TOM address select interrupt, and the operation becomes automatic. The converter fires the TIM input interrupt when a value is ready, and the TIM input command triggers the TOM address select interrupt to start another conversion. While the operation is proceeding automatically, the user can wait for a completion interrupt. It must be remembered, however, that the A/D converter will still only handle one analog input at a time. Therefore, there will be a time skew between individual input trunk samples. Since the TIM and TOM interrupts may be inhibited by certain commands within the currently executing program, the time skew may vary.

Two completion interrupts are provided, one of which is a primary class interrupt, and the other of which is a secondary class interrupt. Two interrupts are provided to allow considerable flexibility in the method of performing analog sampling. The primary interrupt is dedicated to response address octal 216. This interrupt is triggered from

the echo completion signal of the TIM value input interrupt, indicating that all of the desired analog samples have been brought into the value table in core memory. It is a branching-type interrupt which may contain either an SPB or a DMT command.

If the response instruction is an SPB command, then the subroutine which handles the interrupt must inhibit all interrupts, perform its function, and then return to the place of interruption. The subroutine could start another set of samples on the same analog trunks. This procedure would allow the user to simply obtain the latest sample of the desired trunk from the input table, while the sampling process is automatic and continuous. However, it is difficult for more than one program to share the A/D converter by this method. When using the primary interrupt with an SPB response instruction, the machine modification to save status in core cell zero should be operational.

The primary completion interrupt is also triggered when A/D converter overflow occurs. This condition means that the input voltage is too large in magnitude to be represented by the converter digital output. When an overflow occurs, the interrupt is prohibited from being fired again until the overflow condition is cleared. The condition can only be cleared by executing the JNE O'1200' error test. Therefore, the routine which answers the analog input completion interrupt should always test for converter overflow.

The secondary interrupt for completion status is dedicated to response address octal 246. It is a branching interrupt which is only fired by an echo from a DMT instruction at the primary completion

interrupt address. When the secondary interrupt is to be used, the primary interrupt at octal 216 must contain a DMT instruction, with the operand pointing to a core location containing the value zero. Each time the interrupt occurs, the zero must be re-established in this core location. This scheme provides interrupt branching at either a primary or a secondary level.

11.7 External User Interrupts

The hybrid interface provides four "external" interrupt inputs which are accessible on the EAI-580 digital patch panel and are available for user-defined purposes. They are wired to digital panel trunks 116 through 119, and may be patched to any digital function on the EAI-580. A transition from logic 0 (Ground) to logic 1 (+5 volts) on an external interrupt input will trigger the interrupt. Each interrupt input is dedicated to both a primary and a secondary branch-type response location. The interrupt response locations are summarized in Tables 11.3 and 11.4. The primary interrupts are assigned to locations octal 205 through 210, and the secondary interrupts are assigned to locations octal 272 through 275.

The primary class response location which is associated with an external interrupt input may contain an SPB or a DMT command. If the response instruction is an SPB branch command, then the interrupt service routine should be executed with all interrupts inhibited, and should perform a quick function. Also, the machine modification to save primary interruption status into core cell zero should be operational. An example of this type of external interrupt response would

be a routine which sends a certain clock sequence to the digital program on the EAI-580 whenever the interrupt occurs.

If the response instruction in a primary response location is a DMT command, then the operand of the instruction points to a core location which is used as an event counter. Initially the contents of this location are set to a value which is one less than the desired number of interrupt input events per user program interrupt. When the specified number of events has occurred, the corresponding secondary echo interrupt is triggered, and the hybrid user program is signalled. This feature is useful to the user program in the GE-4020 for determining when a certain number of clock cycles have transpired in the digital logic area. The constant which is placed into the event counter location should be zero if the secondary interrupt is to be activated for every interrupt input event. Also, the constant must be re-established in this location whenever the secondary interrupt occurs.

11.8 Summary of Hybrid Interrupts

A wide variety of events in the EAI-580 analog and digital sections and within the hybrid interface can be detected by using the hybrid interrupts. This provides a valuable method for the synchronization of the digital program in the GE-4020 and the analog program on the EAI-580 patch panels.

The user of a program in the GE-4020 digital computer must be aware of the total interrupt environment in which the program is operating, since any enabled interrupt may be triggered. In particular, the system 60 Hz clock interrupt is active whenever interrupts are

enabled. Also, many I/O devices which are not related to the hybrid system use certain interrupts.

Tables 11.3 and 11.4 give a summary of the interrupts which are used by the hybrid system. The addresses are shown in octal in these tables.

Table 11.3 Hybrid Interrupt Dedications

Response Address	Response Type	Class	Echo to Address	Hybrid Section
205	SPB/DMT	PRI	272	External
206	SPB/DMT	PRI	273	External
207	SPB/DMT	PRI	274	External
210	SPB/DMT	PRI	275	External
211	SPB/DMT	PRI	276	Mode Change
212	SPB/DMT	PRI	277	Mode Change
214	TIM	PRI	216	Analog Input
215	TOM	PRI	---	Analog Input
216	SPB/DMT	PRI	246	Analog Input
217	TOM	PRI	247	Analog Output
246	SPB	SEC	---	Analog Input
247	SPB	SEC	---	Analog Output
250	SPB	SEC	---	Servo Pot
272	SPB	SEC	---	External
273	SPB	SEC	---	External
274	SPB	SEC	---	External
275	SPB	SEC	---	External
276	SPB	SEC	---	Mode Change
277	SPB	SEC	---	Mode Change

Table 11.4 Hybrid Interrupt Operation

Response Address	Activation Event	Function Performed
205	0 to 1 change on Dig TRK 116	Event count, user defined
206	0 to 1 change on Dig TRK 117	Event count, user defined
207	0 to 1 change on Dig TRK 118	Event count, user defined
210	0 to 1 change on Dig TRK 119	Event count, user defined
211	Analog mode change to PP+IC	Mode change count or signal
212	Analog mode change to PP+OP	Mode change count or signal
214	A/D converter completion	A/D converter result input into core
215	ACT O'1200' command or A/D converter ready for new conversion	Output of new address word to analog input section
216	TIM echo from 214 or A/D converter overflow error	Analog input operation complete or error signal
217	ACT O'1100' command or D/A converter ready for new conversion	Output of new address/value word to analog output section
246	DMT echo from 216	Analog input operation complete or error signal
247	TOM echo from 217	Analog output operation complete signal
250	Servo-set pot operation completion	Pot servo ready signal
272	DMT echo from 205	Event, user defined
273	DMT echo from 206	Event, user defined
274	DMT echo from 207	Event, user defined
275	DMT echo from 210	Event, user defined
276	DMT echo from 211	Mode change count or signal
277	DMT echo from 212	Mode change count or signal

Chapter XII

Conclusion

12.1 Review

The GE-4020/EAI-580 hybrid computer interface provides an extensive communication path between the digital and analog computers. The digital computer controls the analog computer by the use of a wide variety of commands. Also, a large set of commands is available for the control of analog and digital sampling. The analog computer can interrupt the digital computer. Also, many interrupts are utilized to provide programming flexibility for sampling and control software.

12.2 Suggestions for Future Work

One of the problems which is often encountered in hybrid programming has to do with the lack of exact time information. The characteristics of any sampled-data system are dependent on the accuracy and stability of the sampling interval. The only sampling reference in the hybrid interface is the A/D conversion time. The GE-4020 does not contain a timer which is fast enough to be used as a sampling reference in hybrid programming. Therefore, the programmer cannot easily control the sample rate of the system.

To solve this problem, a digitally-controlled time interval timer should be added to the hybrid interface. The digital computer should be able to load a time register with a time interval value. The register should then count down at a fixed rate, and interrupt the GE-4020 when the count of zero is reached. The digital computer should also be able to interrogate the present time register value.

This will provide a means for the time "stamping" of samples. Thus the time interval timer would allow considerable flexibility in the programming of sampled-data systems.

The analog-digital conversion technology has advanced considerably since the construction of the hybrid interface. An A/D converter is now available which has a conversion time of less than the GE-4020 command sequence time. This can be used to eliminate the need for wait command sequences in the digital program. It can also virtually eliminate time skew between samples in a multiple variable ensemble. Therefore, the present A/D converter should be replaced or supplemented by such a high-speed converter.

BIBLIOGRAPHY

1. A. W. Bennett: Introduction to Computer Simulation, West Publishing Co., Inc., St. Paul, 1974.

2. G. A. Bekey and W. J. Karplus: Hybrid Computation, John Wiley & Sons, Inc., New York, 1968.

3. B. Gold and C. M. Rader: Digital Processing of Signals, McGraw-Hill, Inc., New York, 1969.

4. S. C. Gupta and L. Hasdorff: Fundamentals of Automatic Control, John Wiley & Sons, Inc., New York, 1970.

5. Engineering Staff of Analog Devices, Inc., Analog-Digital Conversion Handbook, Analog Devices, Inc., Norwood, Massachusetts, 1972.

6. R. M. Oliver, J. R. Pierce, C. E. Shannon, "The Philosophy of Pulse Code Modulation," Proc. I.R.E., 36, No. 11, November 1948, pp. 1324-1331.

7. "The GE-PAC 4000 Process Computer System," Publication No. 019-E-69620, Vol. 1, General Electric Communication Products Department, Phoenix, Arizona, 1966.

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

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

The vita has been removed from the scanned document

IMPLEMENTATION OF A HYBRID SYSTEM INTERFACE

by

Robert Marvin Amundson

(ABSTRACT)

Both analog and digital computation have certain comparative advantages and disadvantages. A hybrid computer system provides an interface between analog and digital computing elements. With the use of a hybrid computer, the programmer can utilize the best features of both analog and digital computation.

A hybrid interface was constructed to couple the GE-4020 digital computer and the EAI-580 analog computer. The interface provides a method for controlling and sensing the analog computer operating mode from the digital computer. It also enables the digital computer to control the coefficient settings of servo-controlled potentiometers. The overload status of the analog amplifiers can be detected from the digital computer. The outputs and derivatives of amplifiers can be converted from analog to digital and brought into the digital program. A logic signal communication path is provided between a digital logic patch panel and the digital computer.

In addition, a number of supplemental data paths are provided to expand the flexibility of the hybrid system. An analog output section is available which can provide a set of analog output values to analog patch trunks under digital computer control. An analog input section is available which can sample a set of analog patch trunk signals and

convert them to digital for use by the digital program. A set of high-speed digitally-controlled electronic attenuators is provided. A number of interrupts are available to allow the analog program to interrupt the digital program. Thus, the hybrid interface provides a complete two-way control and data path between the analog and digital computers.

Documentation is provided at both the language level and the machine hardware level. Emphasis is placed on a description of the user-system interface. The digital control commands and analog system characteristics are described in detail. Information is provided which will aid in further hardware development.