

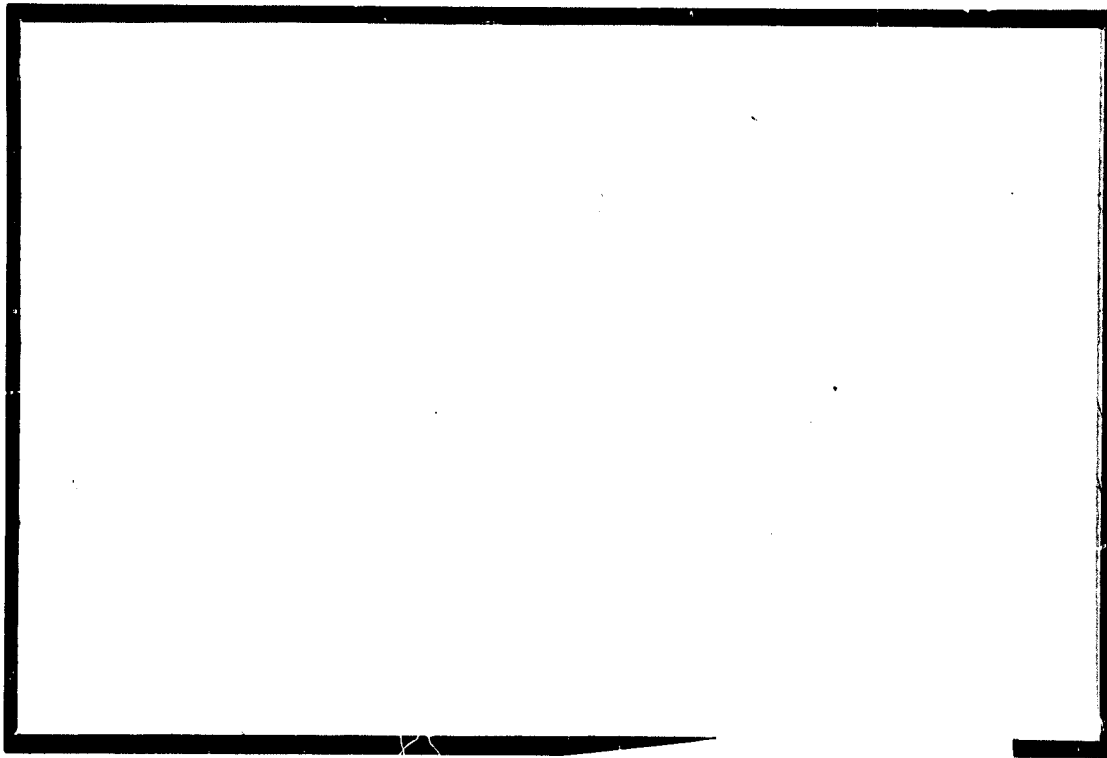
General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

ELECTRICAL

E
N
G
-
N
E
E
R
-
N
G



FACILITY FORM 502

N67 26075

(ACCESSION NUMBER)

94

(PAGES)

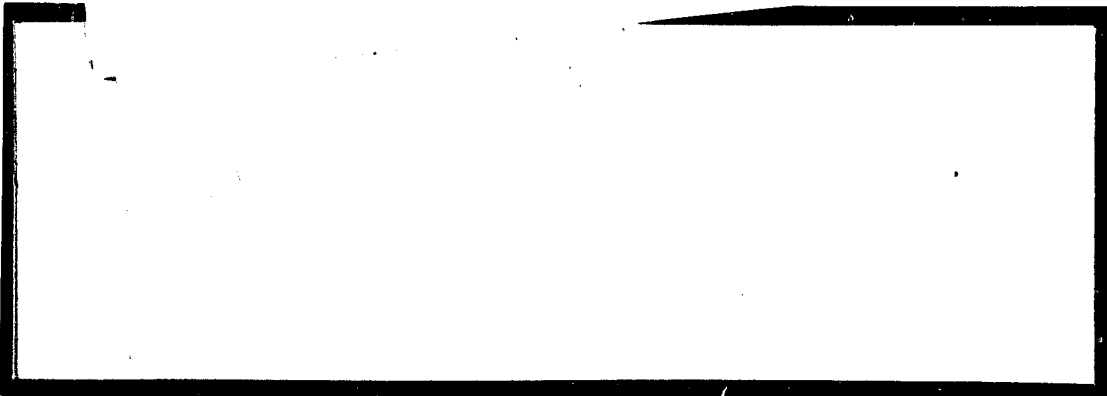
CN 83913

(NAB, CR OR TMX OR AD NUMBER)

(THRU)

(CODE)

(CATEGORY)



AUBURN RESEARCH FOUNDATION

AUBURN UNIVERSITY

AUBURN, ALABAMA

SQT 44506

TECHNICAL REPORT NUMBER 2

DISCRETE METHODS FOR THE IMPLEMENTATION OF CONTROL
NETWORKS USING INTEGRATED CIRCUITRY

PREPARED BY

DIGITAL SYSTEMS LABORATORY

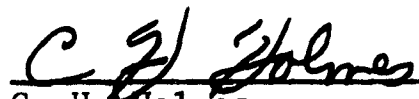
C. C. CARROLL, PROJECT LEADER

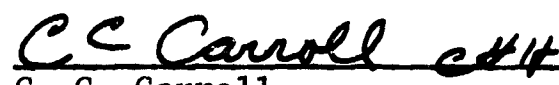
August 1966

CONTRACT NAS8-20163
GEORGE C. MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HUNTSVILLE, ALABAMA

APPROVED BY:

SUBMITTED BY:


C. H. Holmes
Head Professor of
Electrical Engineering


C. C. Carroll
Associate Professor of
Electrical Engineering

FOREWORD

This report is a technical summary reporting the progress of a study conducted by the Digital System Laboratory of the Electrical Engineering Department, Auburn University, under the auspices of the Auburn Research Foundation. The study is focused toward fulfillment of Contract No. NAS8-20163, granted to Auburn Research Foundation, Auburn, Alabama, by the George C. Marshall Space Flight Center, National Aeronautics and Space Administration, Huntsville, Alabama.

ABSTRACT

The digital compensation of a single-axis reference accelerometer loop is investigated through the application of sampled-data control theory and time domain analysis. The resulting digital controller is implemented as a combinational circuit and an arithmetic unit. By other means, another adjustable, digital, lead compensator is developed and implemented. The problem of interfacing the analog and digital networks is examined in detail, and an interfacing scheme is proposed. Finally, a presentation of more sophisticated techniques for future consideration is included.

TABLE OF CONTENTS

LIST OF FIGURES.....	vi
I. INTRODUCTION.....	1
II. COMPENSATION OF THE CONTROL LOOP.....	3
General Consideration of the System.....	3
Sampled Data Description of the Control Loop.....	6
A Generalized Digital Compensator.....	9
Determination of a Unique Digital Controller.....	11
Time Domain Analysis.....	17
III. IMPLEMENTATION OF THE DIGITAL COMPENSATION NETWORK.....	23
Combinational Circuit Approach.....	23
Arithmetic Unit Approach.....	28
IV. A DIGITAL PULSE LEAD COMPENSATOR.....	35
Analog to Digital Network.....	38
The Digital Proportional Section of the Compensator.....	39
Buffer Register Memory.....	42
The Digital Pulse Section of the Compensator.....	42
The Sampling and Pulse Width Control Clock.....	45
V. ANALOG TO DIGITAL CONVERSION.....	47
Open Loop AD Converters.....	47
Closed Loop AD Converters.....	52
Digital to Analog Conversion.....	59
Bipolar Conversion.....	63
Systems Under Consideration for Implementation.....	64
Evaluation of the System to Be Implemented.....	68

VI. TECHNIQUES FOR FUTURE CONSIDERATION.....	70
Quantizing Effects.....	70
Optimal Control Theory.....	71
Linear Sequential Machines.....	74
APPENDICES.....	77
A. Fortran Source Program for Root Locus Plot.....	77
B. Fortran Source Program for Bode Plot.....	82
C. Fortran Source Program for Time Domain Analysis of System.....	83
D. Fortran Source Program for Truth Table of Combinational Network.....	85
REFERENCES.....	87

LIST OF FIGURES

1. Block Diagram of a Digitally Compensated Accelerometer Loop.....	4
2. Block Diagram for Determining the Sampled Closed Loop Response.....	7
3. Block Diagram for a Generalized Digital Controller.....	12
4. Digital Controller.....	14
5. Root Locus Plot for the Compensated System.....	15
6. Open Loop Bode Diagram for the Compensated System.....	18
7. Transition Flow Graph for the Compensated System.....	20
8. System Response Using Time Domain Analysis.....	22
9. Block Diagram of the Digital Compensator.....	24
10. Compensation Network Using Combinational Circuitry.....	25
11. Eight Variable Veitch Diagram for Z_1	27
12. Full Adder Using Two Half Adders.....	29
13. Compensation Network Using an Arithmetic Unit.....	31
14. Complement Gate.....	33
15. Half Adder.....	33
16. Full Adder.....	34
17. A Digital Pulse Lead Compensator.....	36
18. Analog-to-Digital-to-Proportional-Analog Network with Switching Network for Pulse Compensation.....	40

19.	Sequential Circuit to Produce Level-Control for Compensation Pulses.....	43
20.	Time Base Encoder.....	48
21.	Two Bit Direct Threshold Detection Converter.....	50
22.	Three Bit Sequential Approximation Converter.....	51
23.	Basic Closed Loop Analog to Digital Converter.....	53
24.	Subranging Converter.....	55
25.	Subranging with Four Subranges Per Step.....	56
26.	Basic Counter Converter.....	58
27.	Equivalent Circuit of a Four Bit DA Converter Using an Operational Amplifier and a Summing Network.....	60
28.	Equivalent Circuit of a Four Bit DA Converter Using a Bilateral Switching Network.....	61
29.	Counter Converter to be Implemented.....	66
30.	Standard Realization of Machine A.....	75

I. INTRODUCTION

C. C. Carroll, H. T. Nagle, Jr., M. Q. Liner,
H. H. Hull, R. J. Robison and R. White

In recent years, two major objectives in the refinement of compensator designs for the single-axis accelerometer loop have been the miniturization of the packaging techniques, and the improvement of accuracy. In order to fulfill the accuracy requirements, the current designs employ dc analog compensation with ac power amplification.

Although current designs are satisfactory, future applications may require smaller packaging than can be accomplished for a lumped-parameter, dc analog compensator. Therefore, it is desirable to employ micro-miniturized integrated circuitry in a digital compensator for the accelerometer loop.

In this technical report, the compensation of the accelerometer loop is investigated through the application of several digital techniques.

Chapter II is devoted to the use of sampled-data theory in the synthesis of a digital compensator for the stabilization of the accelerometer control loop. In particular, root locus and Bode plot techniques are employed in the specification of the digital compensator. Time domain analysis is used to simulate the compensated system response to an input test signal.

The implementation of the digital compensator, found in Chapter II, is approached in two ways, both of which are discussed in detail in

Chapter III. First, the compensator is implemented with combinational circuitry; and, second, with an arithmetic unit.

An entirely different design procedure for the digital compensator is presented in Chapter IV. A schematic diagram of a pulse lead compensator is shown and its operation is described in detail.

Data conversion techniques are discussed in Chapter V, with a presentation of several open-loop and closed-loop analog-to-digital converters. After evaluating the advantages and disadvantages of each converter, an interfacing scheme is selected to be implemented.

Chapter VI describes several interest areas for future consideration. It is planned to examine the effects of quantizing the signals to be processed by the digital compensator. Optimal control theory, applicable to discrete digital systems, is also under consideration for future study. It is proposed to apply this theory to the accelerometer control loop to determine a time varying sequence of numbers to be used as the optimal input for the system. The implementation of the input sequence might be accomplished using linear sequential machine theory.

II. COMPENSATION OF THE CONTROL LOOP

In this chapter, conventional sampled data control analysis techniques are employed to design theoretically and to test a digital compensation unit for a single-axis reference accelerometer loop of a stabilized platform.

General Consideration of the System

The control loop under consideration may be described adequately by the block diagram of Figure 1. In this block diagram, the loop is compensated by a digital compensator, $D(z)$. Other transfer functions and symbols in Figure 1 are defined:

$G_a(s)$ = transfer function of the accelerometer

$$G_a(s) = \frac{\omega_n^2/H}{s(s^2 + \omega_n^2)}$$

$$\omega_n = 184 \text{ rad./sec.}$$

$$H = 10^5 \text{ gm.-cm.}^2/\text{sec.}$$

$G_p(s)$ = transfer function of the pickoff, preamp, and detector

$$G_p(s) = 562 \text{ volt-dc/rad.}$$

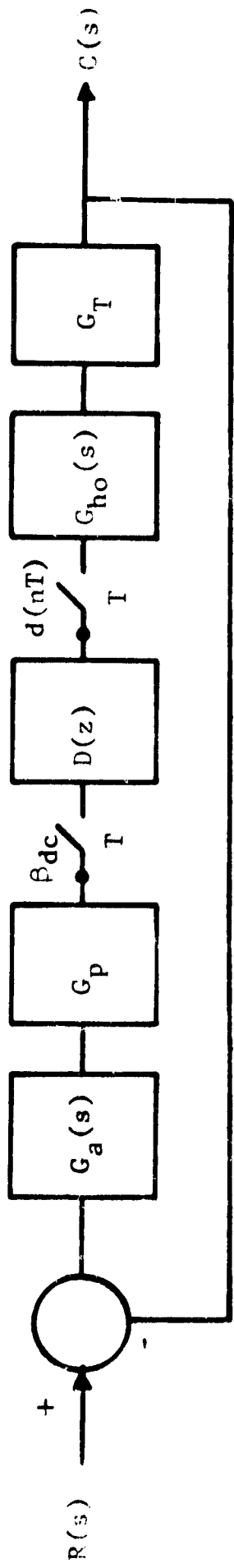


Fig. 1--Block diagram of a digitally compensated accelerometer loop.

$G_{ho}(s)$ = transfer function of a zero-order hold

$$G_{ho}(s) = \frac{1 - e^{-Ts}}{s} \quad T = \text{sampling period}$$

G_T = transfer function of the driver and torque motor

$$G_T = 0.417 \text{ dyne-cm/volt-d.c.}$$

$\frac{\text{—}}{T}$ is a symbol for an impulse sampler

e_{dc} = d.c. voltage output of the detector

$d(nT)$ = output of the digital unit

Although an impulse sampler is unrealizable, an impulse sampler in conjunction with a zero-order hold device is realizable, and it describes adequately the effect of a physical digital compensating unit.

The input, $R(s)$, represents a disturbance torque input, while the output is produced by the torque motor. A design for a stable compensated system with a damping ratio, ζ , of approximately 0.5 is one requirement; and, of more importance, it is required that, for a finite disturbance input, the output of the detector, e_{dc} , be forced to zero

in minimum time (less than 0.5 seconds). The remainder of this chapter will be devoted to a discussion of these considerations.

Sampled Data Description of the Control Loop

For the purpose of sampled data analysis, the block diagram of Figure 1 is transformed into the block diagram of Figure 2. From Figure 2 will be derived the closed loop response of the sampled data system. For this purpose, the following relations are important:

$$E_1(s) = R(s) - C(s)$$

$$E_2(s) = G_1(s)E_1(s)$$

$$E_3(s) = D(s)E_2^*(s)$$

$$C(s) = G_2(s)E_3^*(s).$$

Solving these relations for $C(s)$ in terms of $R(s)$ results in the following:

$$C(s) = \frac{D^*(s)G_2(s)[G_1(s)R(s)]^*}{1 + D^*(s)[G_1(s)G_2(s)]^*}.$$

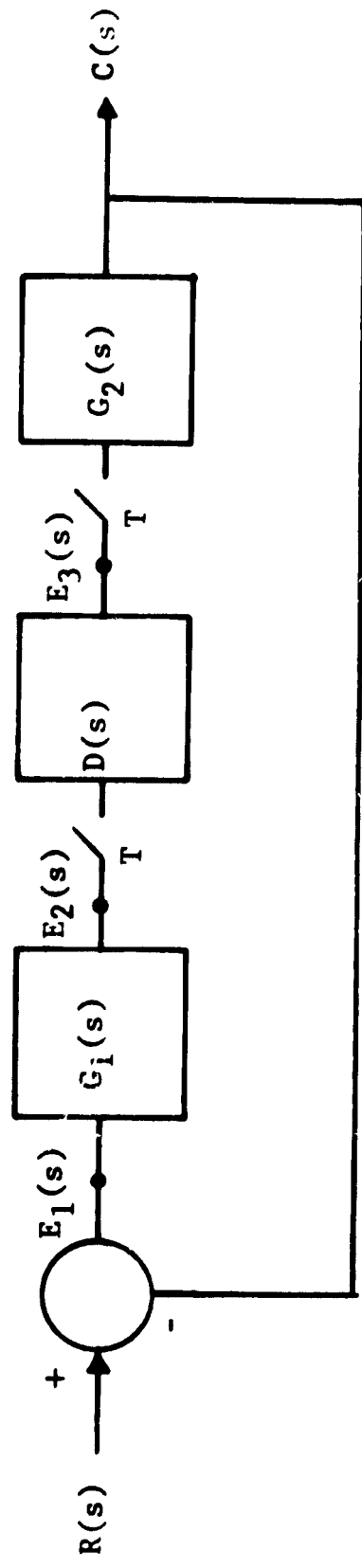


Fig. 2--Block diagram for determining the sampled closed loop response.

Although a control ratio, or closed loop transfer function, cannot be defined, the system stability may still be adjusted by examining the characteristic equation of $C(s)$ as follows¹:

$$1 + D^*(s)[G_1(s)G_2(s)]^* = 0$$

Rewriting the characteristic equation in z transform notation,

$$1 + D(z)G_1G_2(z) = 1 + GH(z) = 0$$

where

$$G_1(s) = G_a(s)G_p = \left[\frac{\omega_n^2/H}{s(s^2 + \omega_n^2)} \right] (G_p) ,$$

$$G_2(s) = G_{ho}(s)G_T = \left[\frac{1 - e^{-Ts}}{s} \right] (G_T) ,$$

but

$$z = e^{Ts}$$

$$G_2(s) = \frac{(1 - z^{-1})G_T}{s} .$$

Now, it is necessary to determine the open loop transfer function, $GH(z)$.¹

$$GH(z) = D(z)G_1G_2(z)$$

$$GH(z) = D(z) \mathfrak{L} \left[\frac{\frac{\omega_n^2 G_p G_T}{H} (1 - z^{-1})}{s^2 (s^2 + \omega_n^2)} \right]$$

$$GH(z) = D(z) \frac{G_p G_T}{H} \left(T - \frac{\sin \omega_n T}{\omega_n} \right) \left[\frac{z^2 + Ez + 1}{(z-1)(z^2 + Fz + 1)} \right]$$

where

$$E = - \frac{2(\omega_n T \cos \omega_n T - \sin \omega_n T)}{\omega_n T - \sin \omega_n T}$$

$$F = - 2 \cos \omega_n T.$$

The problem now becomes one of determining a digital compensator, $D(z)$, that will satisfy the requirements imposed on the system.

A Generalized Digital Compensator

A physically realizable digital controller (or compensator), $D(z)$, may be expressed in the general form

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}} = \frac{E_2(z)}{M(z)} \frac{M(z)}{E_1(z)}$$

where m and n are positive integers. Also $E_1(z)$, $M(z)$, and $E_2(z)$ are the z transforms of the input, intermediate, and output variables respectively. Define

$$\frac{M(z)}{E_1(z)} = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}};$$

then,

$$E_1(z) = M(z) + b_1 z M(z) + \dots + b_m z^{-m} M(z)$$

$$m(kT) = e_1(kT) - b_1 m[(k-1)T] - \dots - b_m m[(k-m)T].$$

Alternatively,

$$\frac{E_2(z)}{M(z)} = a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n},$$

from which

$$e_2(kT) = a_0 m(kT) + a_1 m[(k-1)T] + \dots + a_n m[(k-n)T].$$

From these results, a generalized block diagram for a digital controller, $D(z)$, can be represented as in Figure 3. In Figure 3, either m or n could be the larger integer.¹

Determination of a Unique Digital Controller

It is desirable to determine a unique digital controller that can be implemented by means of integrated circuit digital logic. Therefore, such a physically constructed digital controller must process discrete, binary coded signals. In other words, in Figure 3, the input $e_1(kT)$, the intermediate variable $m(kT)$, and the output $e_2(kT)$ must be binary coded variables.

To implement the block diagram in Figure 3, the operations of time delaying a signal, multiplying a signal by a constant, and adding two signals must be accomplished. The details of performing these operations by means of integrated circuitry and of interfacing the digital and analog portions of the system with integrated circuitry, are discussed and illustrated in Chapter III.

In the actual choosing of a specific digital controller, $D(z)$, root locus plots and Bode diagrams were employed.

Root locus analysis. Since the characteristic equation in the z domain is an algebraic equation in z , the construction of the z -plane root loci is the same as the construction of the s -plane root loci, and the rules of construction of the conventional root locus diagram are directly applicable.¹

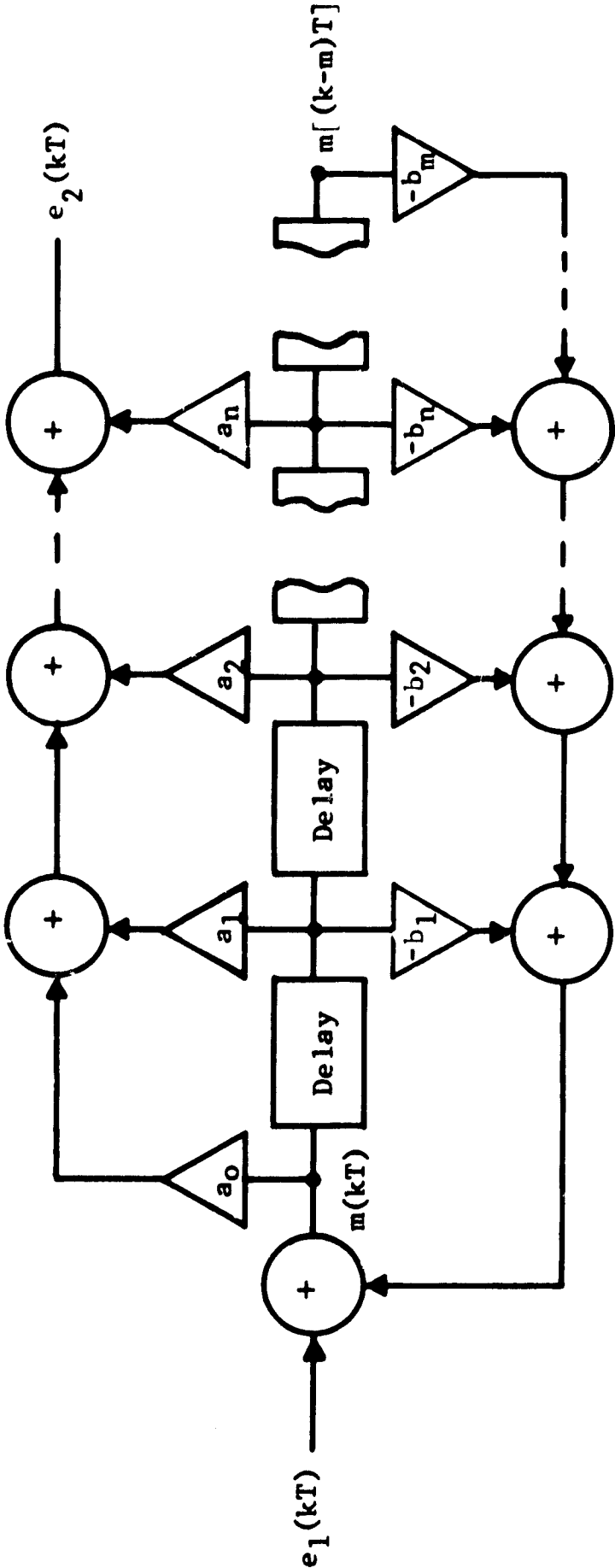


Fig. 3--Block diagram for a generalized digital controller.

In plotting the root locus in the z -plane,

$$|GH(z)| = 1$$

$$\angle GH(z) = 180^\circ + k(360^\circ)$$

where, as shown previously,

$$GH(z) = \frac{(T - \frac{\sin \omega_n T}{n}) G_p G_T}{H} D(z) \left[\frac{z^2 + Ez + 1}{(z-1)(z^2 + Fz + 1)} \right].$$

Choosing a simple form for the digital controller, the following form was selected. (See Figure 4):

$$D(z) = a_0 + 2a_0 z^{-1} = a_0(1 + 2z^{-1}).$$

To demonstrate that this digital controller can adequately compensate the system, a Fortran computer program, shown in Appendix A, was employed in conjunction with the IBM 7040 computer to produce data for the root locus plot shown in Figure 5. The sampling period selected was 0.01 second.

In the z -plane, the region of stability is the interior of the unit circle. The $\zeta = 0.5$ contour is the dotted logarithmic spiral in Figure 5.¹ From the root locus plot, the dominant pair of closed

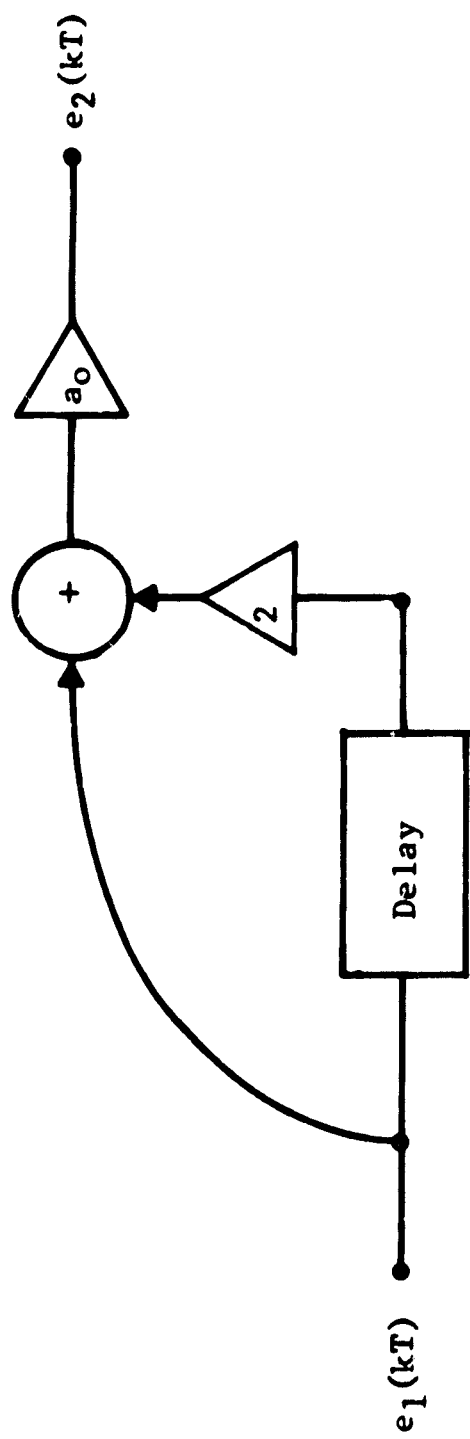


Fig. 4--Digital controller.

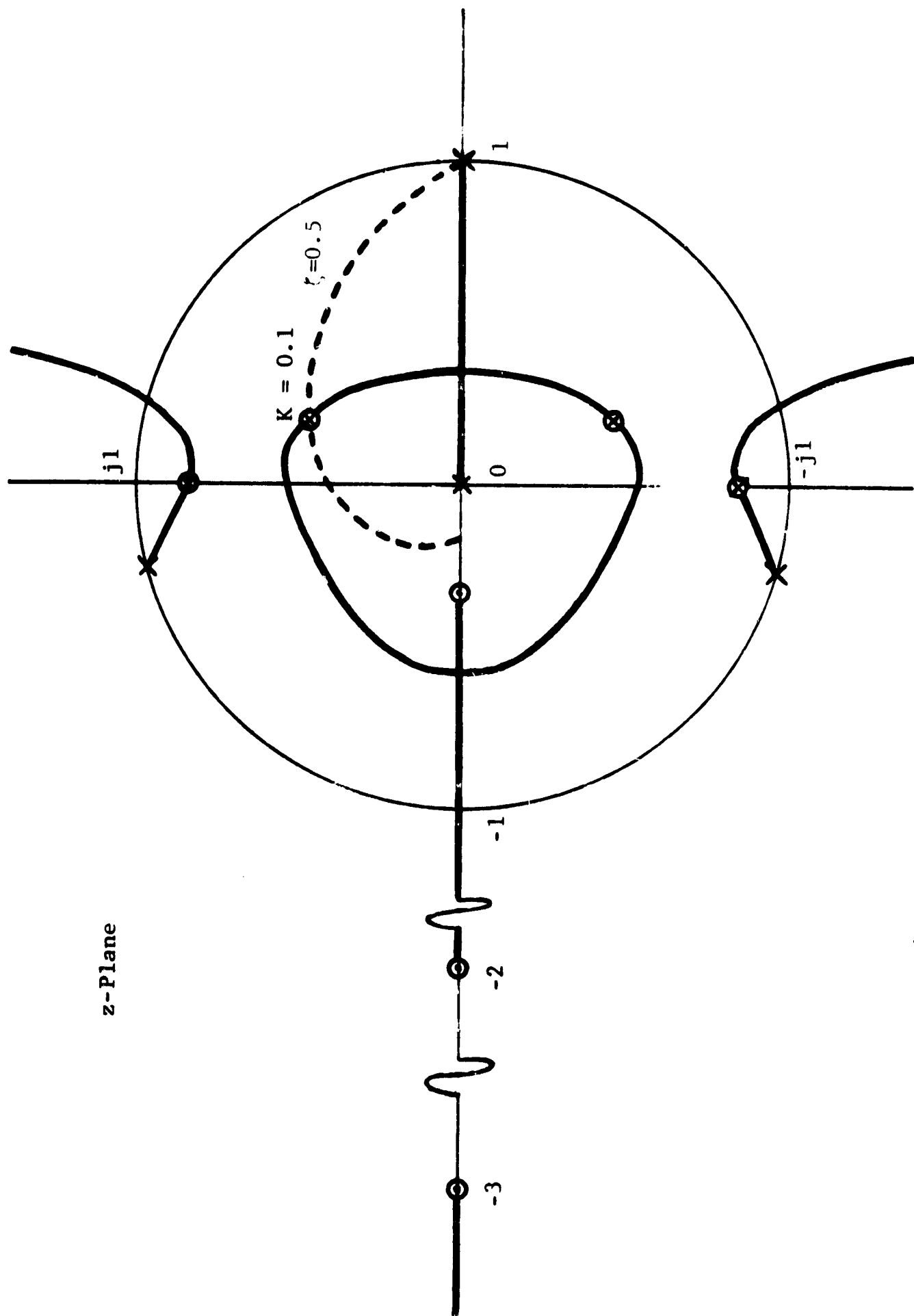


Fig. 5--Root locus plot for the compensated system.

loop roots occur at the $\zeta = 0.5$ contour when the total loop gain is 0.1. Thus, the digital unit gain, a_o , is determined as follows:

$$a_o = \frac{(\text{total loop gain})H}{G_p G_T (T - \frac{\sin \omega_n T}{\omega_n})} = 9,000 .$$

To insure that the phase margin and gain margin are adequate, a Bode plot was produced.

Bode plot analysis. A useful characteristic of sampled data systems is that the frequency response, whether determined in the z , s , or w plane, is unique.

The Bode plot may be constructed using the following relations:¹

$$\text{db} = 20 \log |GH(z)|$$

$$\phi = \angle GH(z)$$

where z is defined as

$$z = e^{j\omega T} \quad 0 \leq \omega T \leq \frac{\pi}{2} .$$

The Bode plot is db and ϕ versus ω_w , frequency in the w plane, with w and ω_w defined by

$$w = \frac{z-1}{z+1}$$

$$\omega_w = \text{Im}(w) = \tan(\omega T/2) .$$

Again, a Fortran program was written (see Appendix B) which produced the data for the Bode plot shown in Figure 6. From the Bode plot, the gain and phase margin indicate stability.

Time Domain Analysis

The final theoretical approach to the problem was to utilize time domain analysis techniques to simulate, on the IBM 7040 computer, the actual compensated system response.

A sample data system is a dynamic system in which one or more variables can change only at discrete instants of time. The sampling operations occur in discrete dynamic elements (the digital controller), as well as in sample and hold elements.²

In order to describe a sampled data system, the state transition matrix method may be employed. The equations which describe the operation of the system are most easily determined from the state transition flow graph.²

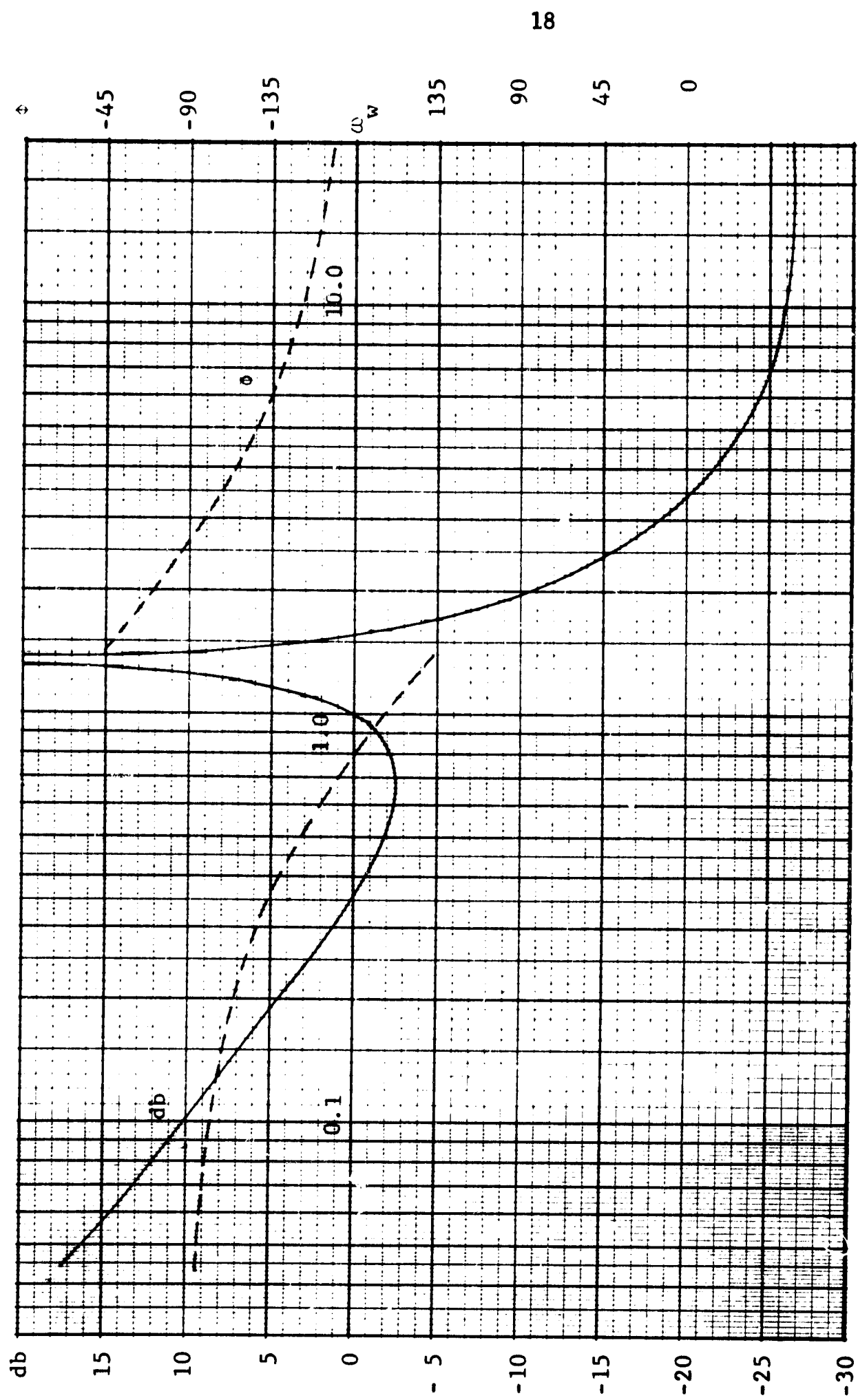


Fig. 6--Open loop Bode diagram for the compensated system.

The state transition flow graph for the system was determined as shown in Figure 7.

From the transition flow graph, the following relations were derived:

$$\begin{aligned} x_1(t) = & x_1(t_0) + \frac{1}{\omega_n} \sin[\omega_n(t-t_0)]x_2(t_0) + \frac{1}{\omega_n^2} [1-\cos[\omega_n(t-t_0)]] \\ & \cdot x_3(t_0) + \frac{1}{H}[(t-t_0) - \frac{1}{\omega_n} \sin[\omega_n(t-t_0)]] [R-G_T d(kT)] \end{aligned}$$

$$\begin{aligned} x_2(t) = & \cos[\omega_n(t-t_0)]x_2(t_0) + \frac{1}{\omega_n} \sin[\omega_n(t-t_0)]x_3(t_0) \\ & + \frac{1}{H} [1-\cos[\omega_n(t-t_0)]] [R-G_T d(kT)] \end{aligned}$$

$$\begin{aligned} x_3(t) = & \cos[\omega_n(t-t_0)]x_3(t_0) - \omega_n \sin[\omega_n(t-t_0)]x_2(t_0) \\ & + \frac{1}{H} [\omega_n \sin[\omega_n(t-t_0)]] [R-G_T d(kT)] . \end{aligned}$$

The state transition equations are obtained by substituting

$$t = (k+1)T$$

$$t_0 = kT .$$

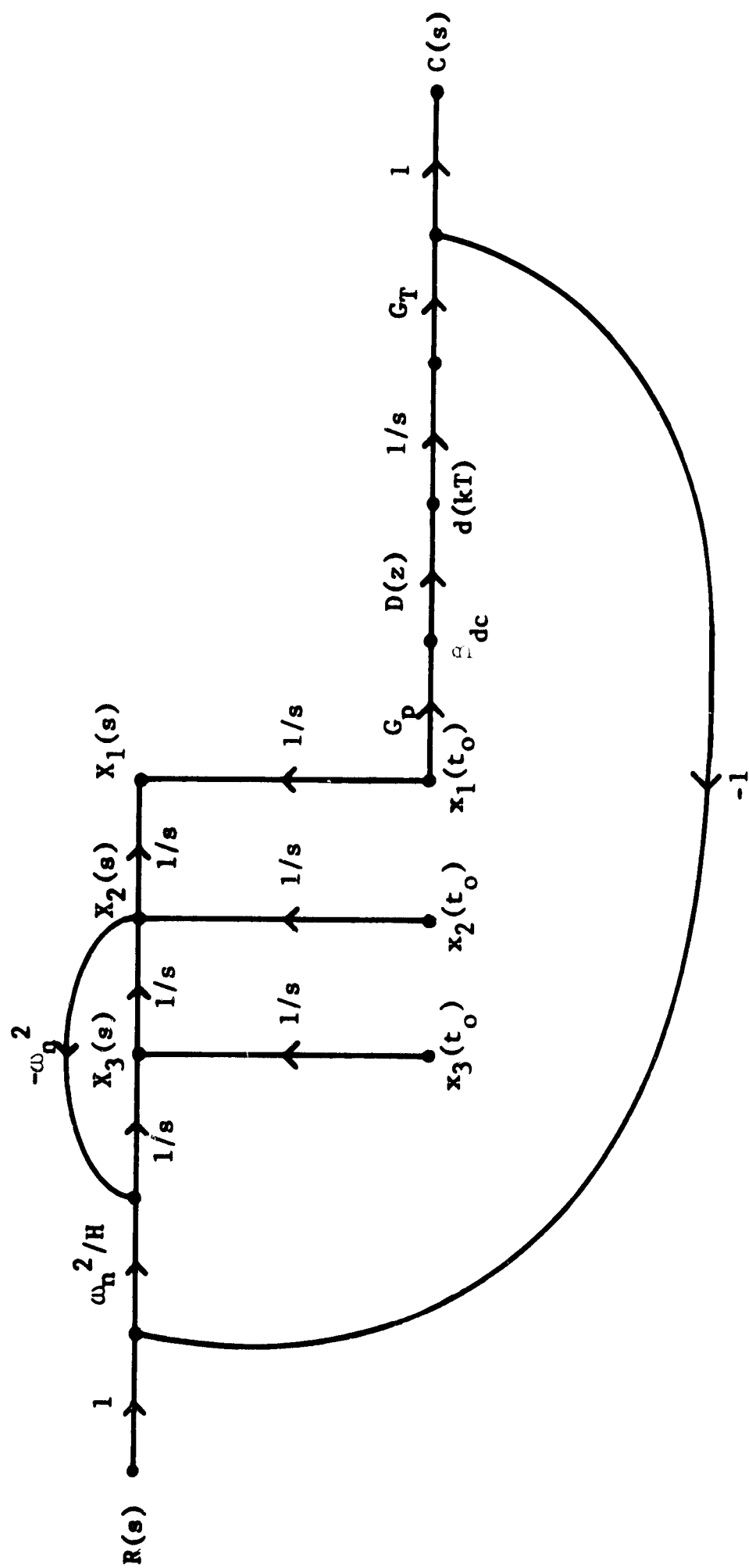


Fig. 7--Transition flow graph for the compensated system.

For the digital controller of Figure 4, the describing equation is

$$\frac{d(z)}{\beta_{dc}(z)} = a_o(1 + 2z^{-1})$$

$$d(kT) = a_o[\beta_{dc}(kT) + 2\beta_{dc}[(k-1)T]]$$

where the input β_{dc} must be quantized according to some interfacing scheme. The scheme chosen for the simulation quantized β_{dc} into fifteen levels between +2.5 volts and -2.5 volts.

The above state transition equations were incorporated in the Fortran program of Appendix C.

For testing purposes, the input signal is:

$$r(t) = R[\mu(t) - \mu(t - .10)]$$

$$R = 5.0 \times 10^4 \text{ gm-cm}$$

The system response to this input signal is shown in Figure 8. Note that the system is indeed stable, and that the settling time (0.45 sec.) is adequate.

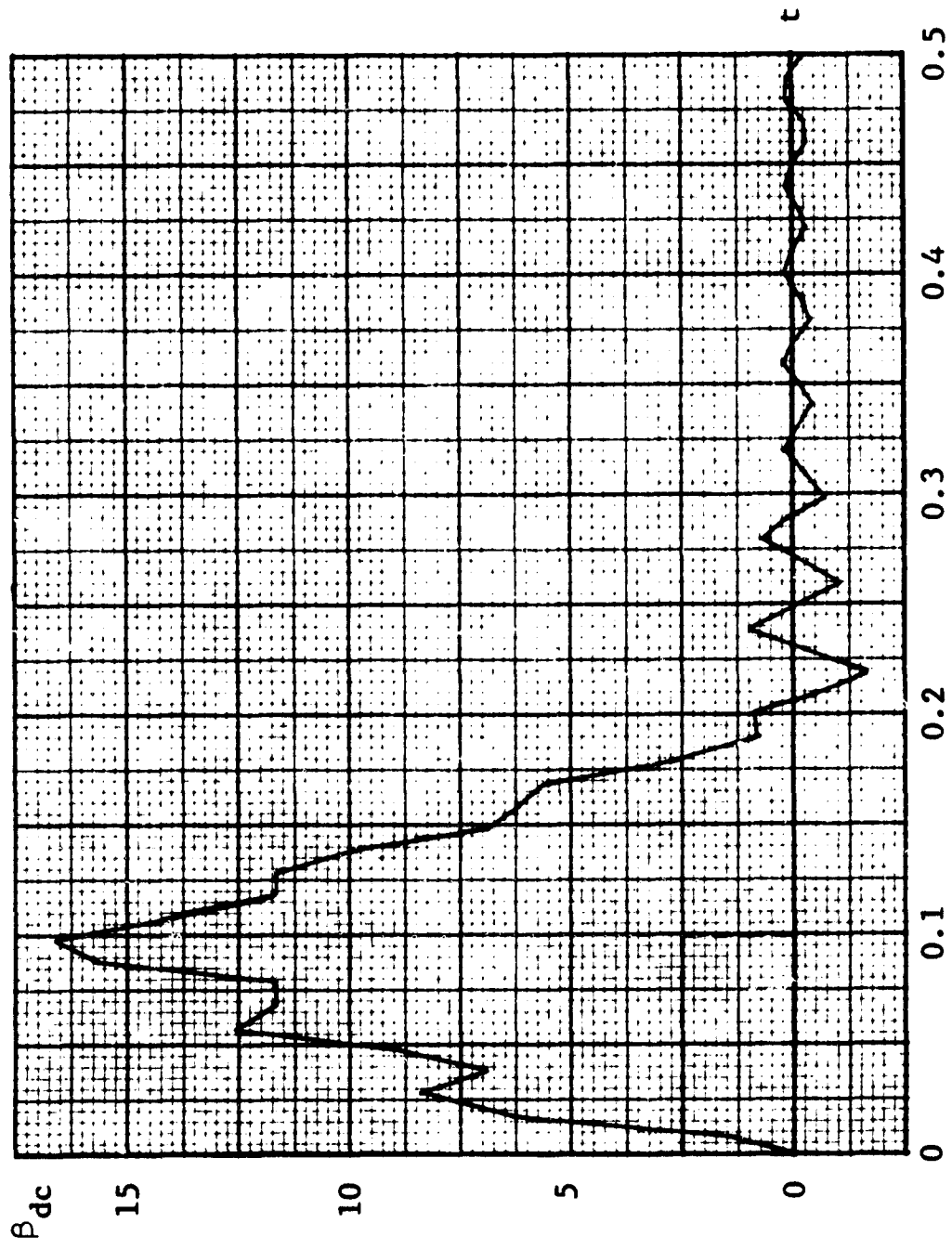


Fig. 8--System response using time domain analysis.

III. IMPLEMENTATION OF THE DIGITAL COMPENSATION NETWORK

The implementation of a digital compensation network of the form $D(z) = (z + A)/z$ (see Figure 9) was approached by two different methods. The classical approach, implementing a combinational network using a truth table, was used, in addition to a more sophisticated arithmetic unit approach. Compensation networks were designed using both approaches, as discussed in the following paragraphs.

Combinational Circuit Approach

Combinational circuitry may be used to realize the block diagram of Figure 9. The function of the combinational circuit is to multiply the previous value of the input by a constant, and to add this product to the present value of the input. This can be accomplished with the block diagram illustrated in Figure 10.

A four bit (one sign bit and three magnitude bits) representation was chosen for each input level. Since the immediate past input as well as the present input are of interest, the output is a function of eight variables.

A computer program, Appendix D, was written to obtain the truth table, using the four bit representation for the output. The output can be represented by four functions, Z_1 , Z_2 , Z_3 , and Z_4 , each

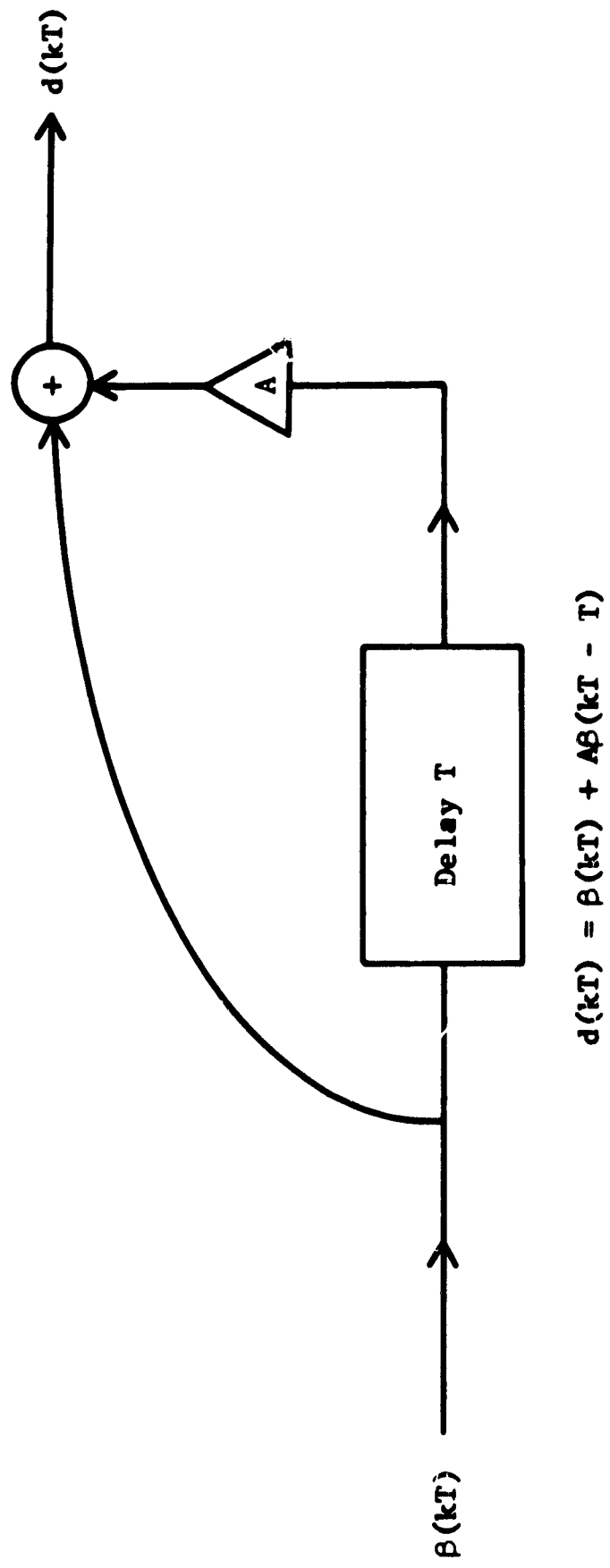


Fig. 9--Block diagram of the digital compensator.

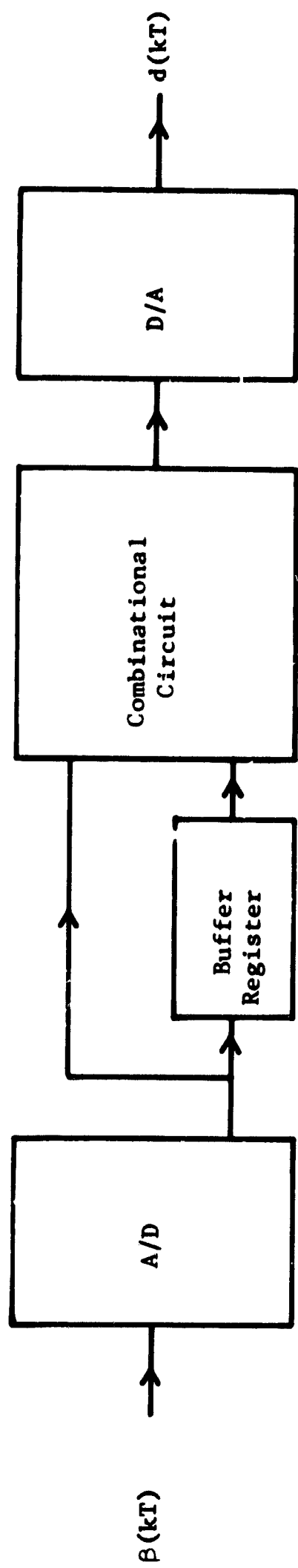


Fig. 10--Compensation network using combinational circuitry.

representing one bit of the output. From the truth table, these four functions are mapped on an eight-variable Veitch diagram, as in Figure 11. After all the appropriate input states (which are defined by squares on the diagram) have been designated, they can be grouped to minimize the circuitry required to implement the output function.^{3,4}

The computer program was used to obtain the truth table for two cases, $A = 2$ and $A = 3$. The output functions obtained were mapped, and, since the functions for the case $A = 3$ appeared simpler, they were chosen to be implemented.

The output functions obtained for the case $A = 3$ are as follows:

$$\begin{aligned} Z_1 = & AE + EF + EG + BCE + BDE + BEH + CEH \\ & + ABFG + AFGH + ABCFH + ABDFH + ACDFG \end{aligned}$$

$$\begin{aligned} Z_2 = & A'E'F + E'G'H' + ABEF + ABE'F' + ACEF + ACE'F' \\ & + ADEF + ADE'F' + AEFG + AE'F'G + AEFH + BEFG \\ & + B'E'FG' + EFGH + AB'C'E'H' + A'B'F'G'H' + ABF'GH \\ & + BCEFH + BCE'F'G + BE'F'GH + B'D'E'FH' + C'D'E'FG' \\ & + CDEFG + A'C'D'F'G'H' + ABCDF'G + A'B'C'EF'G' \\ & + CDE'F'GH \end{aligned}$$

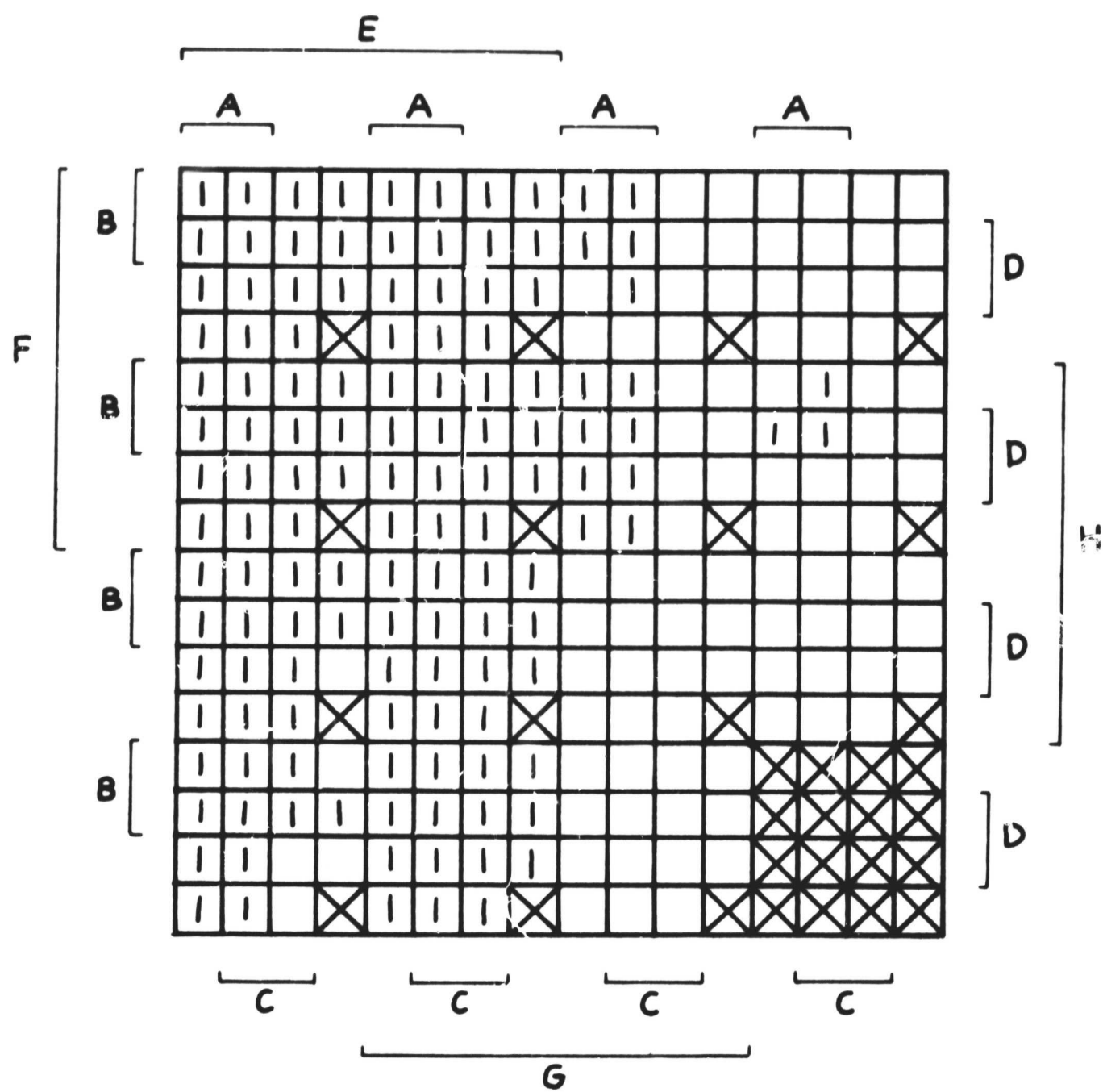


Fig. 11--Eight variable Veitch diagram for Z_1 .

$$Z_3 = A'f_3 + Af'_3$$

$$\begin{aligned} f_3 = & BE'FG + BE'F'G' + B'F'G'H' + C'E'F'H' + CE'F'G' \\ & + E'FGH + BCE'G'H + BEFG'H' + B'EFG'H + BDE'G'H \\ & + B'C'DE'F' + B'D'E'F'G + B'C'EFG + BEF'GH + CEFG'H' \\ & + C'EFG'H + C'D'F'G'H' + B'C'DEFH' + B'D'EFGH' + B'CDE'GH' \\ & + B'C'D'E'G'H' + BCDEF'G \end{aligned}$$

$$Z_4 = B'f_4 + Bf'_4$$

$$\begin{aligned} f_4 = & CDF'H' + CE'F'H' + CF'G'H' + CFG'H + C'D'FH' + C'D'F'H \\ & + C'FGH' + C'EF'H + C'F'GH + DF'G'H' + D'F'GH + D'FGH' \\ & + EF'GH + DE'F'G'H. \end{aligned}$$

These functions require a large combinational network; therefore, another solution to the problem was sought.

Arithmetic Unit Approach

The elements needed for an arithmetic compensation network are a delay element, a multiplier, and an adder. A buffer register can be used as the delay element. Also, if the case $A = 2$ is chosen to be implemented, the buffer register can be used as a multiplier, since multiplying by two in binary corresponds to a shift in the binary point.

A parallel adder capable of adding five information bits was required. The adder network chosen comprises five full adders capable of adding two information bits together with a carry bit. The full adder used in this network comprises two half adders and an "or" gate.^{4,5}

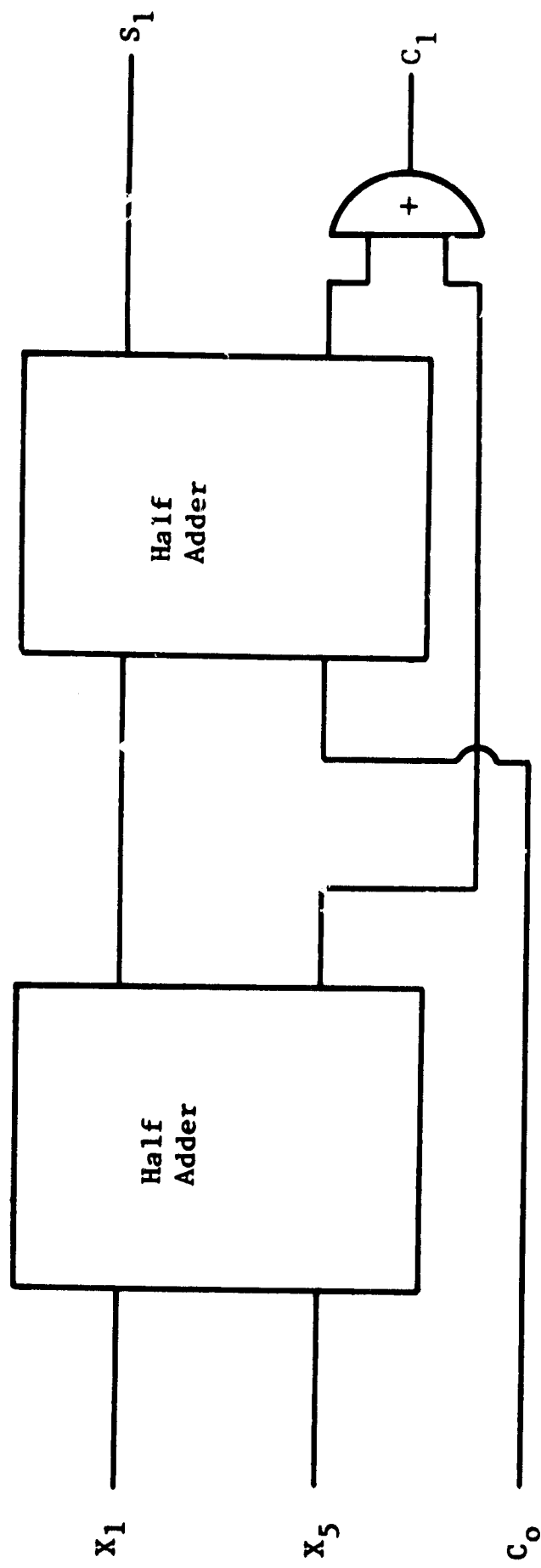


Fig. 12--Full adder using two half adders.

The fallacy of choosing this adder is that it adds magnitudes. Since the most significant bit is a sign bit, some provision had to be made for the sign. The complement of a negative number can be added readily to a positive number, or vice versa; therefore, a complement network was necessary in some cases.⁶ The one's complement is used in this system, since it is easier to implement.

The circuitry required for the complement network can be determined by the construction of a truth table. Since the sign bit determines whether or not the complement operation should be performed, it and the input magnitude bit must be considered as the inputs in the truth table.

In order to prevent erroneous outputs during the time the adder is performing its operation, the output of the adder must be gated with a clock pulse to insure that only correct information is transmitted.

The arithmetic compensation network is shown in Figure 13.

Construction of the arithmetic compensation network involves the implementation of a multiplier, three identical complement networks, five full adder units, and a gate network. (The five full adders comprise the adder shown in Figure 12.)

Since the case $D(z) = \frac{z+2}{z}$ was chosen for implementation, a buffer register is used as a multiplier, and as the one time interval delay needed for the network. The least significant element in the register is set to provide a "zero" true output continuously, thus performing the multiplication by two function.

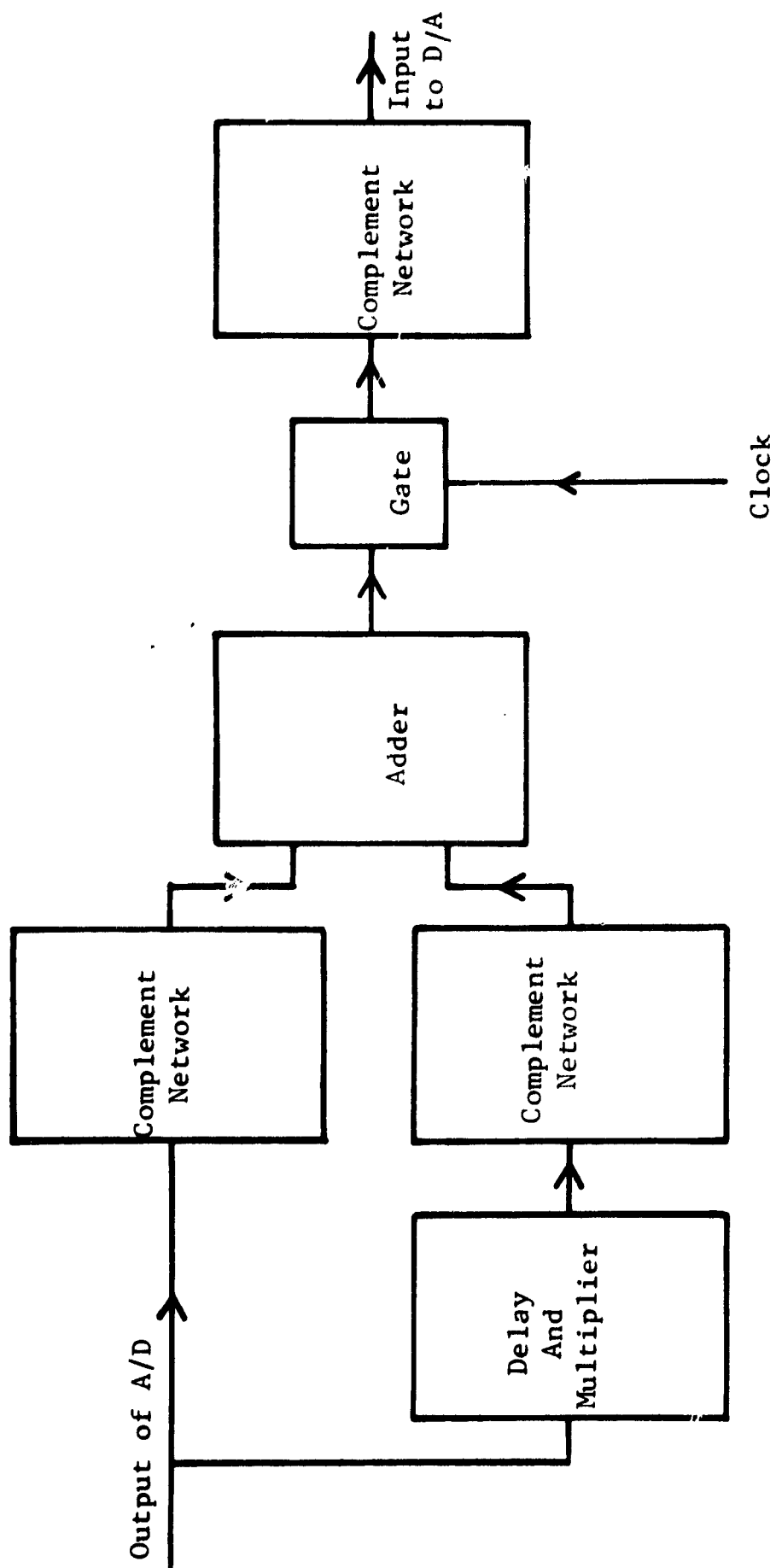


Fig. 13--Compensation network using an arithmetic unit.

Since the truth table for the complement network indicates that the "exclusive or" function will provide the proper output, "exclusive or" gates are required for each of the bits representing magnitude. The combination of NAND gates shown in Figure 14 performs the necessary operation.

The full adder constructed for the network is a combination of two half adders and the "or" function. The half adder can be built using NAND logic as shown in Figure 15.

The full adder is as shown in Figure 16. Five such full adders comprise the adder unit used in the compensation network.

The gating network required to prevent erroneous output is provided simply by "ANDing" the adder output with a clock. The clock frequency is half the input frequency, and allows the adder to complete its operation before any output is emitted.

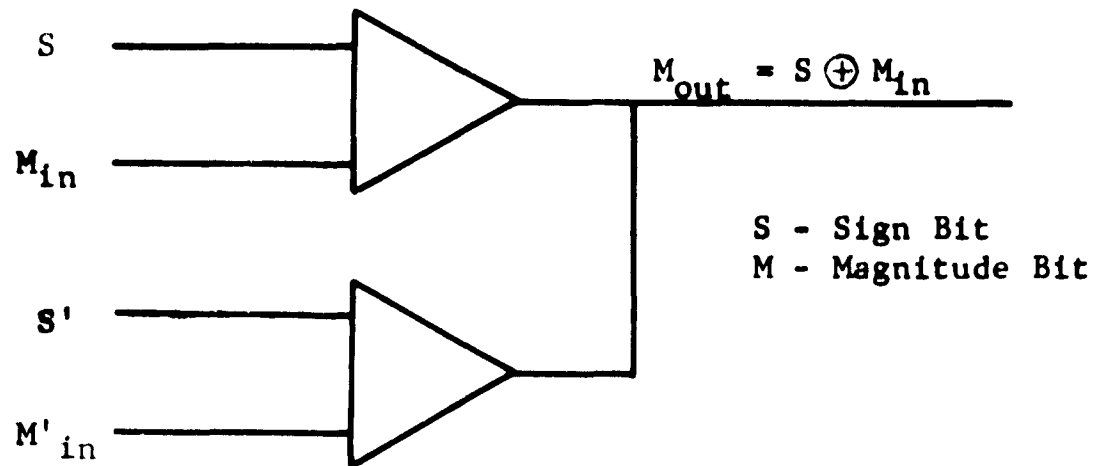


Fig. 14--Complement gate.

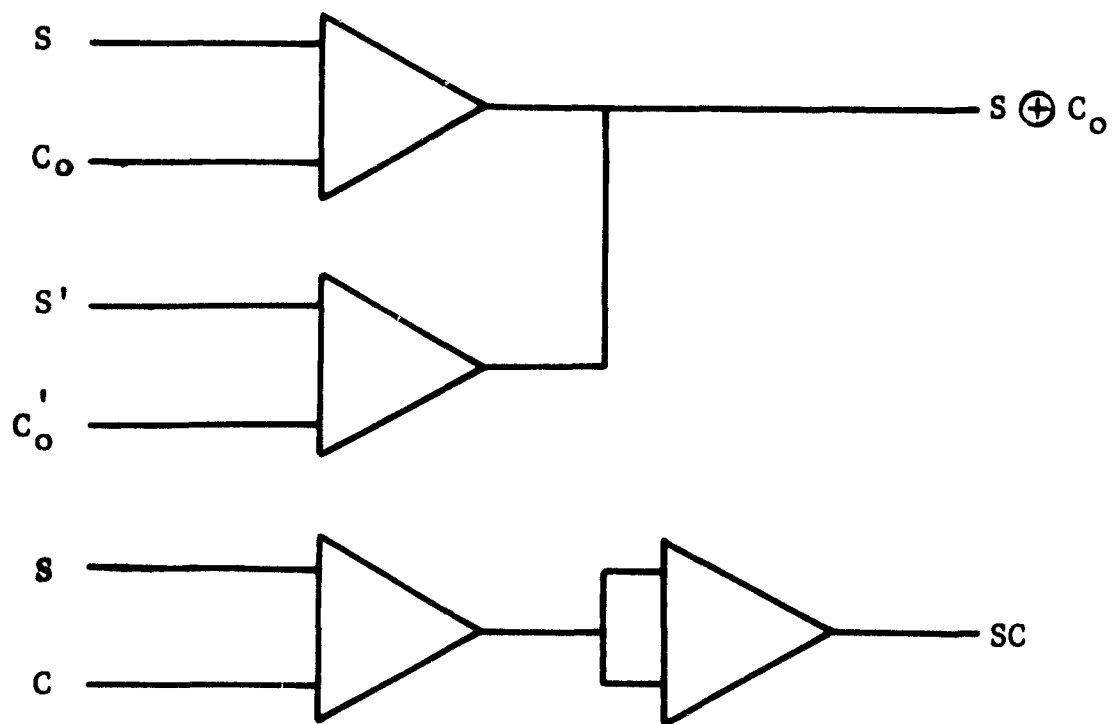


Fig. 15--Half adder.

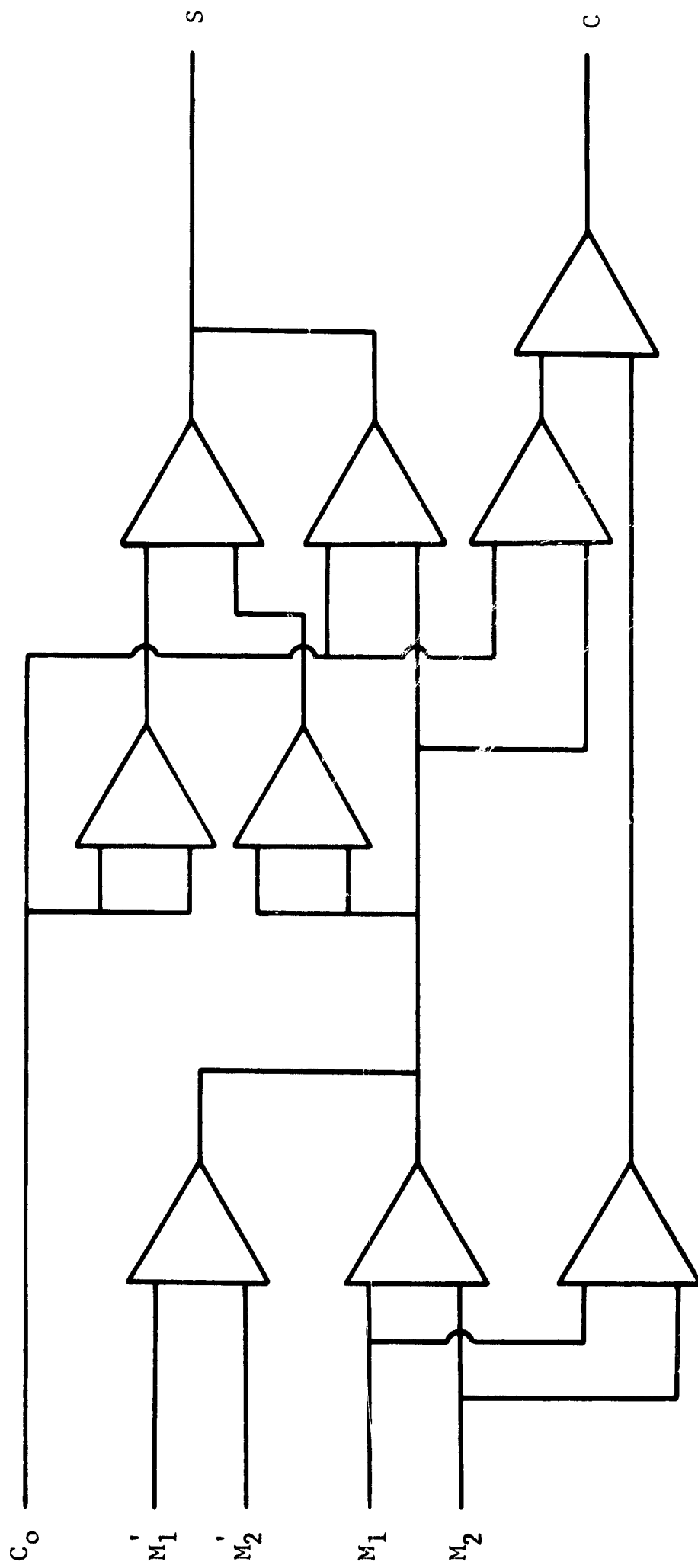


Fig. 16--Full adder.

IV. A DIGITAL PULSE LEAD COMPENSATOR

The system described in this section is based upon a concept developed in a paper titled "Compensating Pulse-Data Servos" by Gawlowicz, Taft, and Wijnschenk.⁷ This system comprises five sections with their functional relationships indicated in Figure 17. The five sections are the AD (analog to digital) network, the digital proportional network, the buffer register memory, the pulse compensator section, and the clock.

As shown in Figure 17, the AD network furnishes a converted error signal to both the proportional network section and to the buffer register. The function of the proportional section is to produce an output that is equal to its input multiplied by a constant. The value of this constant may be made adjustable. The second output of the AD network, which goes to the buffer register,⁸ is shifted through the register by one of the clock signals. This shifting process makes available the digital value of the previous error signal together with its present value. This sequential circuit working in conjunction with the associated combinational circuitry in the digital pulse compensator, determines whether the sense of the error signal is positive or negative, and whether the error is increasing or decreasing. This information allows a pulse to be switched into the output where the output may already be proportional to the input error signal. Pulses having

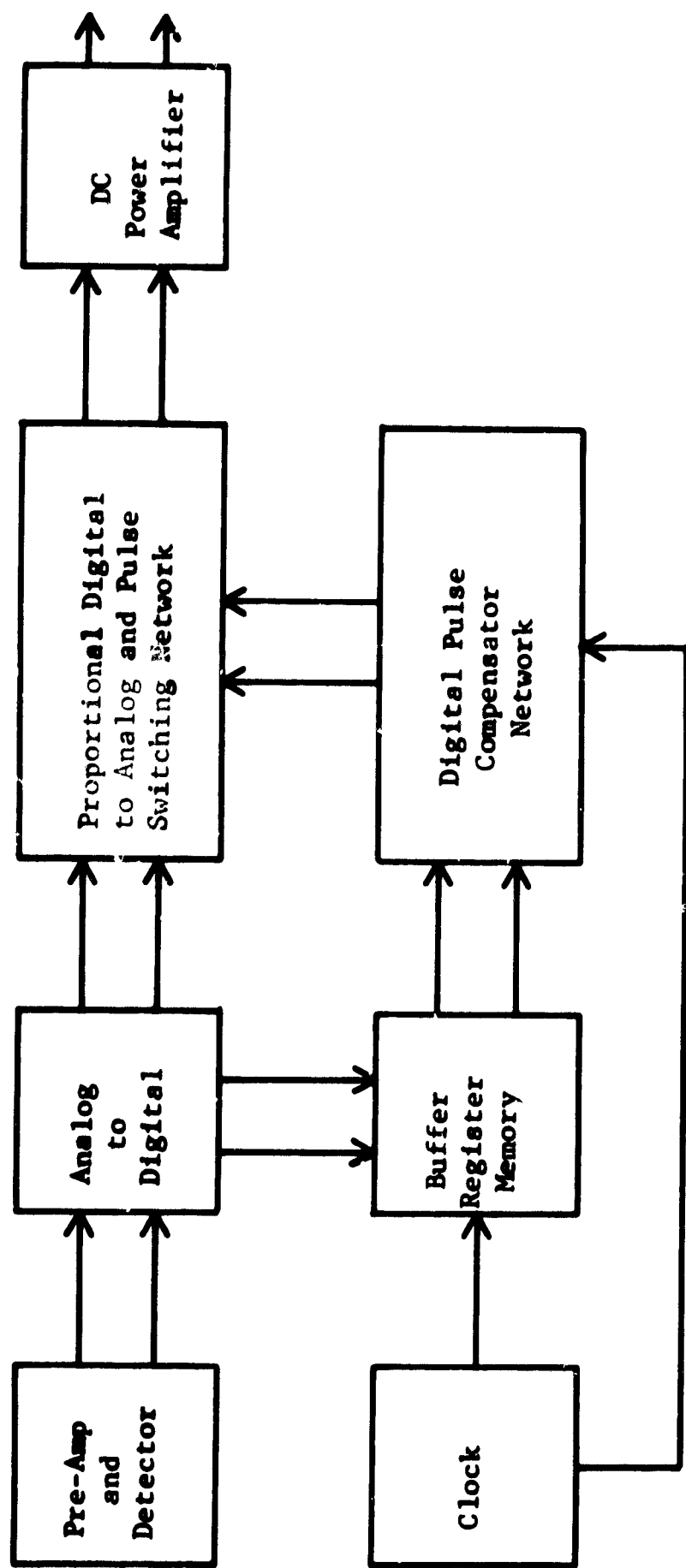


Fig. 17--A digital pulse lead compensator.

the same polarity as the error signal can be switched into the output lines when the error is positive and increasing. On the other hand, when the error is positive and decreasing, pulses having opposite polarity to the error signal can be switched into the output.

The clock section furnishes two outputs. The negation output goes to the buffer register and shifts the register contents on each negative transition. The assertion output goes to the enabling input of a gate that controls the width of the compensating pulses. The clock repetition rate determines how often a compensating pulse may occur, and the clock pulse width determines the width of each compensating pulse. Both the pulse repetition rate and the pulse width of the clock pulses are adjustable.

These component sections, working together, produce a digital output consisting of two parts. The first part of the output signal is proportional to the converted analog error signal. The second part consists of fixed height pulses whose width and polarity can be controlled. When a pulse control signal occurs, the output level is switched to the proper value until the pulse control signal ends. At the end of the pulse control signal, the output level is switched back to the value at the terminals of the proportional section. The composite output signal described above is transmitted to the torque motor through a dc power amplifier.

This complete system, except for several resistors and capacitors, can be constructed using Computer Control Company's μ -pac cards.⁸

The following cards are needed:

- 4 ST-335 Schmitt Trigger Pac
- 1 LD-330 Lamp Driver Pac
- 2 BR-335 Buffer Register Pac
- 1 BP-330 Blank Pac
- 1 MV-335 Multivibrator Clock Pac
- 1 DG-335 Selection Gate Pac
- 2 DI-335 Nand Gate Pac

The next section of this chapter presents in greater detail the operation of the analog to digital network.

Analog to Digital Network

The analog to digital converter section is a simple open loop type. Because the analog error signal is limited in size, the Schmitt trigger circuits on Computer Control Company's ST-335 μ -pac⁸ can be used as level detectors. In fact, four Schmitt trigger circuits are used to detect positive error levels, and four additional ones are used to detect negative error levels. The resultant digital output appears on each of eight lines. Since each level detector is set to trigger at a specific level, only certain combinations of outputs can possibly occur. This situation produces many redundancies in the Veitch Diagram³ used to simplify the combinational circuits. This combinational circuitry is found in the pulse compensation section and will be discussed in that section of this chapter.

A given analog error signal will be either positive or negative, unless it is too small to be detected. Suppose the analog error is less negative than the smallest negative level, and less positive than the smallest positive level; then, each of the four negative level detectors will be activated, and each of the positive level detectors will not be activated. Under the conditions just described, the output of the proportional section is zero. In view of the fact that the Schmitt trigger pac output is (+6) volts (logic one) when it is not activated, the output of each positive level detector needs to be inverted. The analog to digital section is shown as the leftmost part of Figure 18.

The Digital Proportional Section of the Compensator

One half of the digital proportional section includes four lamp driver transistor switches. These transistor switches are used to connect resistors either across an output resistor R or to the positive supply line. The size of these resistors (R_1, R_2, R_3, R_4) and the resistor R may be calculated to give a proportional output voltage across resistor R . In effect, the proportional section is a simple DA (digital to analog) converter. Its resistors are chosen to give a proportional output rather than the usual direct output. The μ -pac line⁸ includes the LD-330 lamp driver pac, which has enough switches on one card to construct the digital proportional section. By referring to Figure 18, it can be seen how an input level signal can switch the

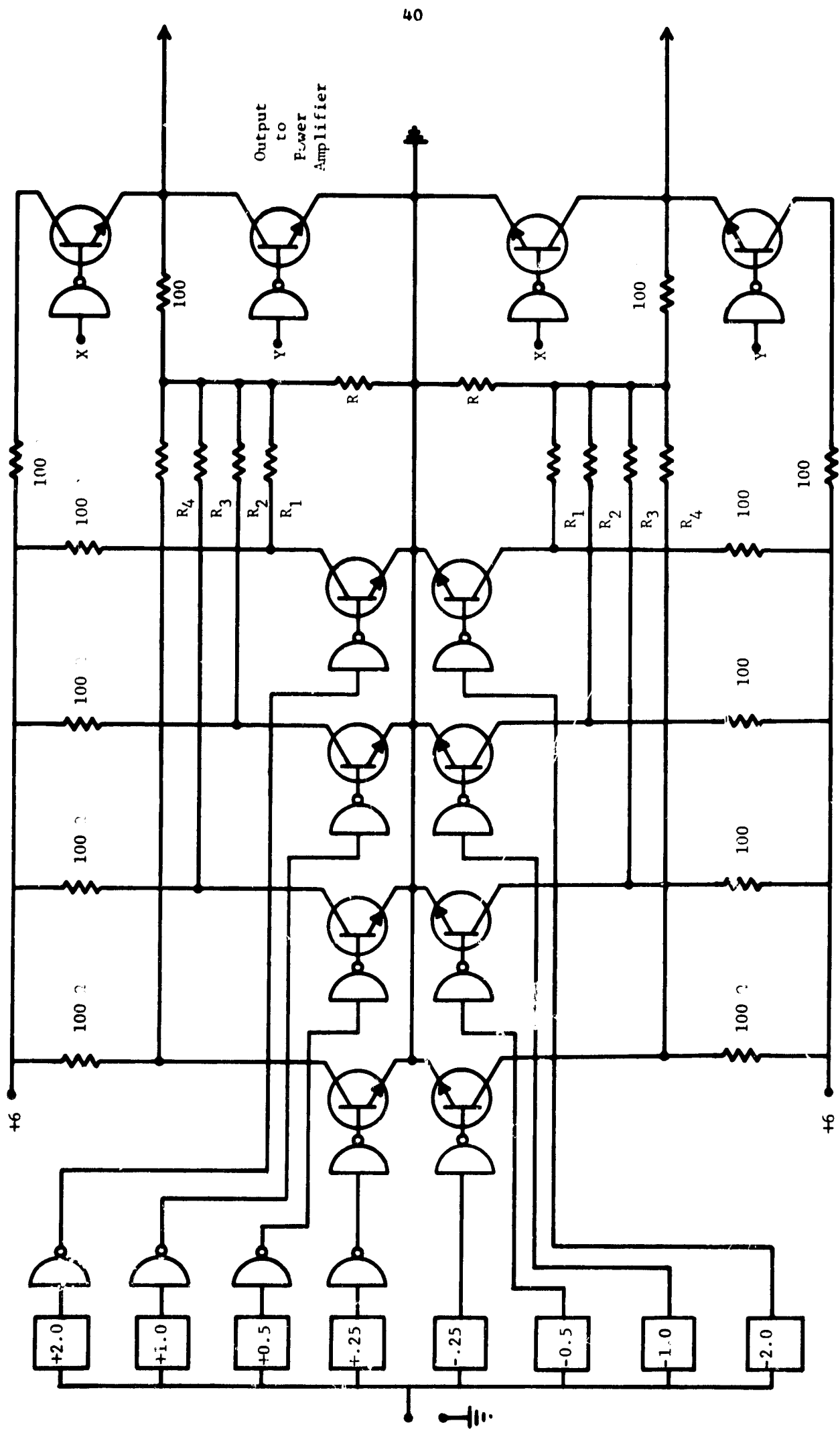


Fig. 18--Analog-to-digital-to-proportional-analog network with switching network for pulse compensation.

proper resistor from ground to the positive supply terminal. The 100 Ω resistors shown in Figure 18 are used to limit transistor current when its collector is switched to ground potential.

The input requirement of the power amplifier is such that neither of its two input terminals can be permanently grounded. Furthermore, the direction of the current through the torque motor depends upon the polarity of the input signal to the power amplifier. These requirements are responsible for the duplication of elements in the digital proportional section. For a given polarity of analog error input to the AD converter section, one particular half of the digital proportional section will be in operation. The two halves of the digital proportional section have a common ground; and, since the high side of each half connects to only one of the two output lines, the polarity of the output signal depends upon the half that is in operation.

To summarize, the digital proportional section produces an output that has the same polarity as, and is proportional to, the converted analog error input signal. The value for the proportionality constant was selected on the basis of experimental tests. The gain of the analog compensator was found to be very close to 2.0, which is the value used in the design of the proportional section. The switches with inputs labeled X and Y (shown in Figure 18) are not part of the digital proportional section. They are used for switching the compensating pulses into the output, and will be discussed in the pulse section of this chapter.

Buffer Register Memory

The buffer register memory is simply a four element shift register.⁸ The four signal level outputs from the AD converter are connected to the shift register. These four level signals are shifted to the output terminals of the register on the negative going transition of each clock pulse. If the AD converter output changes during the interval between clock pulses, the voltage levels on the two sides of the shift register will be different. The signal levels on the two sides of the shift register, when considered at the same instant of time, furnish information to the combinational circuitry in the pulse section of the compensator. There are two identical buffer register memories: the one being used is determined by the polarity of the analog error input to the AD converter. Figure 19 shows a buffer register memory, and, also, the combinational circuitry that is in the pulse control part of the digital pulse compensation section. The next portion of this chapter deals with the digital pulse compensation section.

The Digital Pulse Section of the Compensator

The digital pulse compensation section includes combinational circuitry, gating circuitry, and transistor switches labeled X and Y. The combinational circuitry develops a level control signal used to generate a pulse. For example, when the analog input error signal is

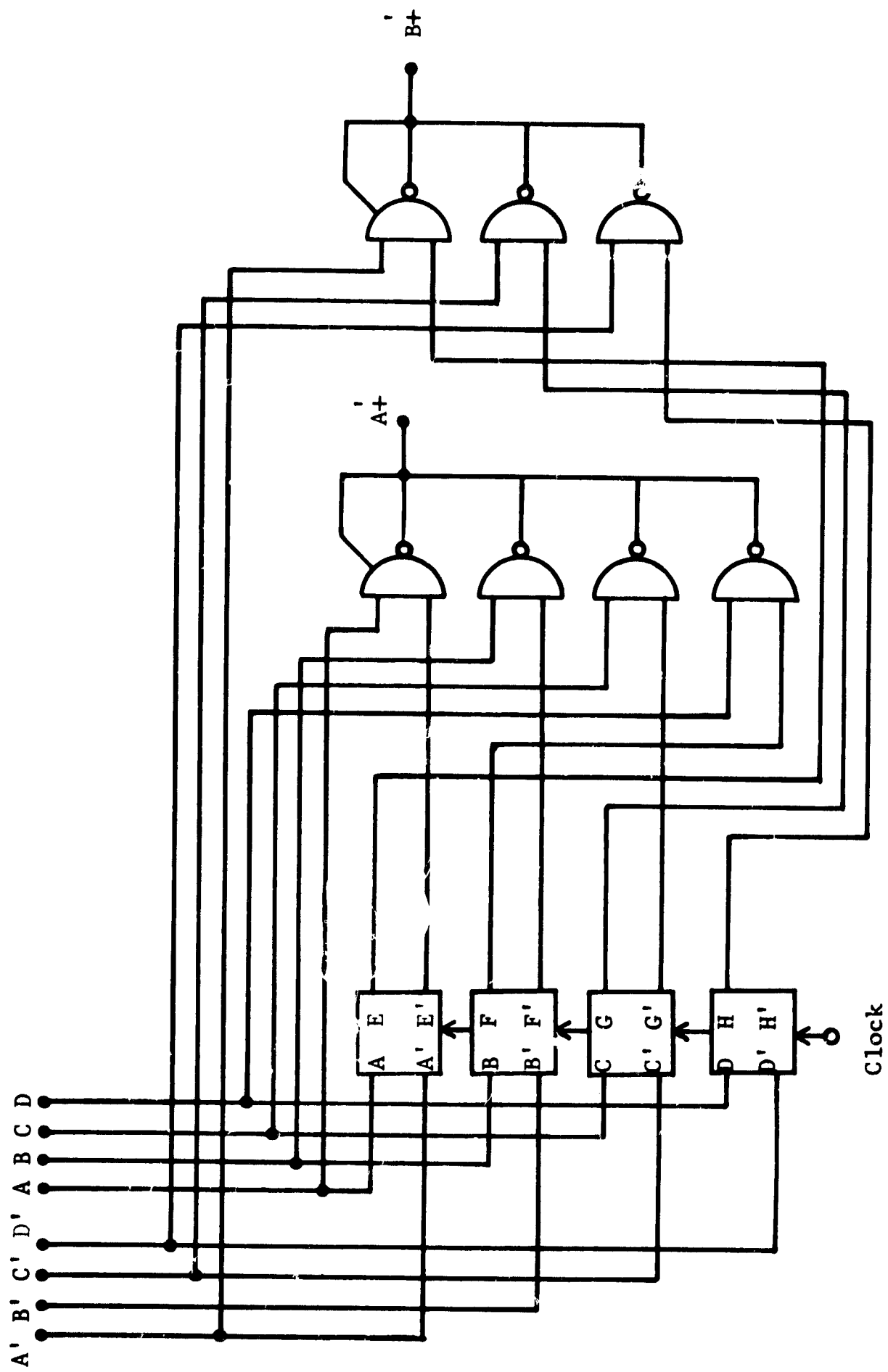


Fig. 19--Sequential circuit to produce level-control for compensation pulses.

positive and increasing, lines marked (A, B, C, D, E', F, F', G') will have signal levels such that a positive pulse level control will be produced. This level control signal is allowed to pass through a gate circuit to transistor switches labeled X in Figure 13. The gate circuit is enabled by clock pulses so that its output becomes a pulse control signal. The pulse control signal closes transistor switches X resulting in a positive pulse of (+6) volts being applied to the output line connected to the power amplifier. In effect, the output line is simply transferred from the voltage level present at the proportional section output to the (+6) volt supply. When the pulse control signal to transistor switches X is removed, the output voltage level returns to the value present at the output of the proportional section.

Output pulses of opposite polarity to the error signal can be obtained by closing transistor switches labeled Y in Figure 18. A pulse control signal to close switches labeled Y is developed by a method similar to the one used to develop the X control signal. The level control signal for this purpose comes from the combinational circuit whose inputs are from lines (A', C', D', E, G, H). In this case, the level control signal is also gated by the clock pulse before being connected to transistor switches Y in Figure 18.

In summary, it may be said that the digital pulse section will deliver a positive pulse when the error is positive and increasing, but will deliver a negative pulse when the error is positive and

decreasing. There are two other possibilities. When the error is negative, and is becoming more negative, a negative pulse is delivered. Finally, when the error is negative, and is becoming less negative, a positive pulse is delivered. In all cases, the output pulse width is controlled by clock pulses that can be adjusted. The remaining section to be discussed is the clock section, and its discussion follows.

The Sampling and Pulse Width Control Clock

The clock used in this system is composed of a self-starting, free-running multivibrator, a pulse shaper, and a pulse amplifier. It is a Computer Control Company's MV-335 μ -pac.⁸ The operating frequency is adjustable in steps by selecting appropriate capacitors, and has an adjustable resistor on the card for further variation of frequency. The pulse width is also adjustable from 45 nsec. to the maximum width of 50 percent of the pulse period.

There are two outputs from this clock. The assertion output delivers pulses that rise to (+6) volts (logic one), and fall to (0) volts (logic zero). This particular output enables a gate through which pulse control level signals are transmitted. The duration of the clock assertion pulse thereby determines the duration of the digital compensation pulses. The negation output, however, delivers pulses that fall from (+6) volts to (0) volts and return to (+6) volts. This output is connected to the buffer register and causes it to shift its contents on each negative transition.

Based on the above considerations, one may observe that the occurrence and the duration of compensation pulses are controlled by the clock.

V. ANALOG TO DIGITAL CONVERSION

Analog to digital conversion techniques are quite numerous with respect to methods of implementation. Basically, such digitizing procedures may be divided into two very general types, those involving open loop systems and those involving closed loop systems. The fundamental principles of operation of several AD methods that will illustrate the aforementioned two general types of systems will be discussed.

Open Loop AD Converters

Open loop AD converters do not involve feedback. This fact necessarily limits the "self error checking" capability of an open loop system.

Assuming that the analog signal is electrical, this signal could be used to regulate the frequency of an oscillator, an astable multivibrator for example. The output of this oscillator would then be counted over a fixed time interval to produce the digital result.⁹ This scheme is illustrated by the block diagram of Figure 20.

Instead of varying the frequency being counted, the time interval of the counting could be varied as some function of the analog signal.

Both methods require gating circuits, a counter, and two oscillators or clocks, one of a fixed rate and the other of a variable rate. In

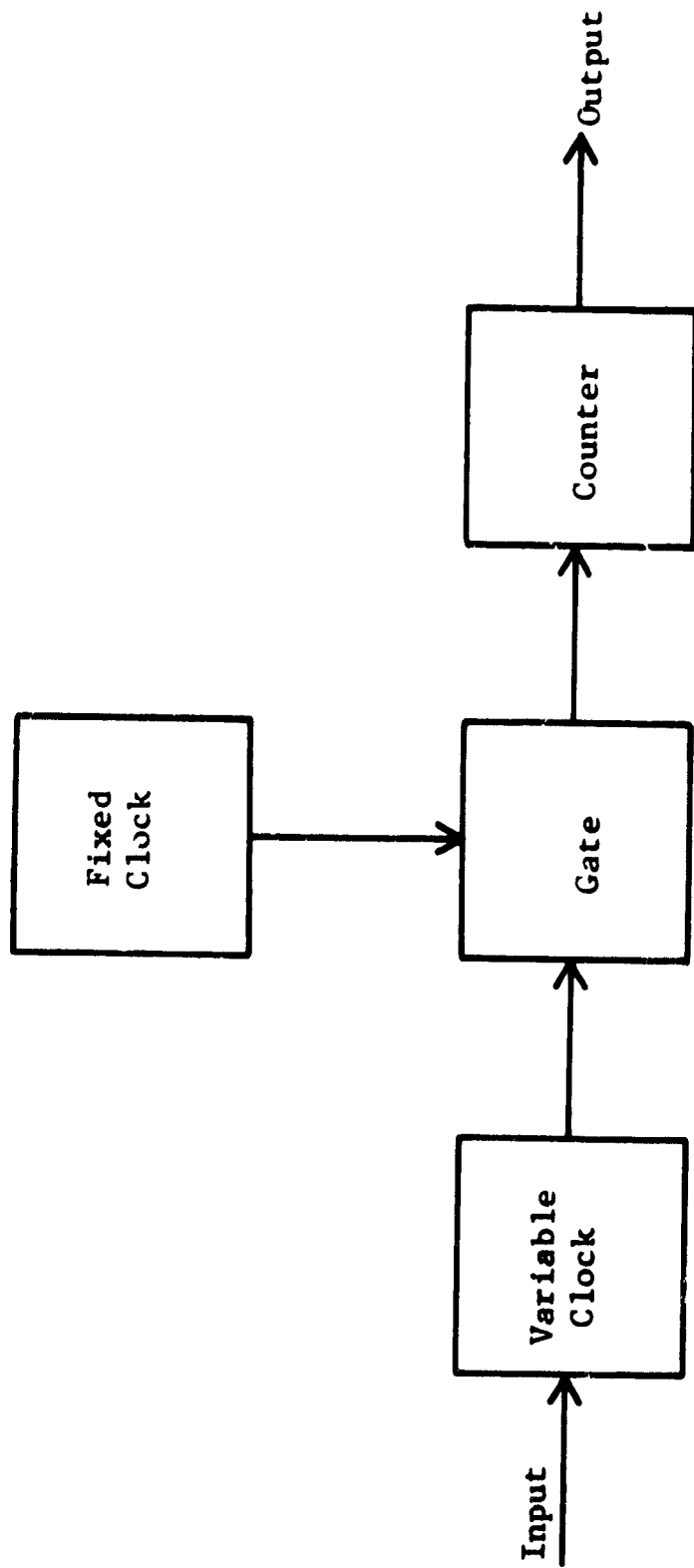


Fig. 20--Time base encoder.

the first case, the variable clock drives the counter through a gate that is operated by the fixed clock. In the second case the functions of the two clocks are transposed, the counter being driven by the fixed clock, and the gate being operated by the variable clock. These two schemes, in effect, utilize an intermediate analog. The magnitude of the analog signal is not sensed directly.⁹

An open loop method which does involve direct level detection is a direct threshold detection converter shown functionally in Figure 21. The system illustrated is capable of two bit binary resolution of an analog signal within a range from 0 to E volts. The accuracy of such a system is largely dependent upon the quality of the comparators or level detectors employed in the circuit. An ideal comparator would be enabled as soon as the voltage exceeded its threshold level, and the device would be disabled as soon as the input level dropped below the threshold level. In the actual case, the enabling and disabling levels differ by a finite amount. The higher the quality of the comparator, the smaller the difference between enabling and disabling levels becomes. No information is gained if the levels of two comparators overlap.

As might be inferred from Figure 21 a resolution of n bits requires $2^n - 1$ comparators. Such a system then, although capable of good speed, is limited in applicability because of the complexity required for high resolution and the comparator sensitivity limitations.¹⁰

Sequential approximation converters require only n comparators for a resolution of n binary bits. Figure 22 is a block diagram of

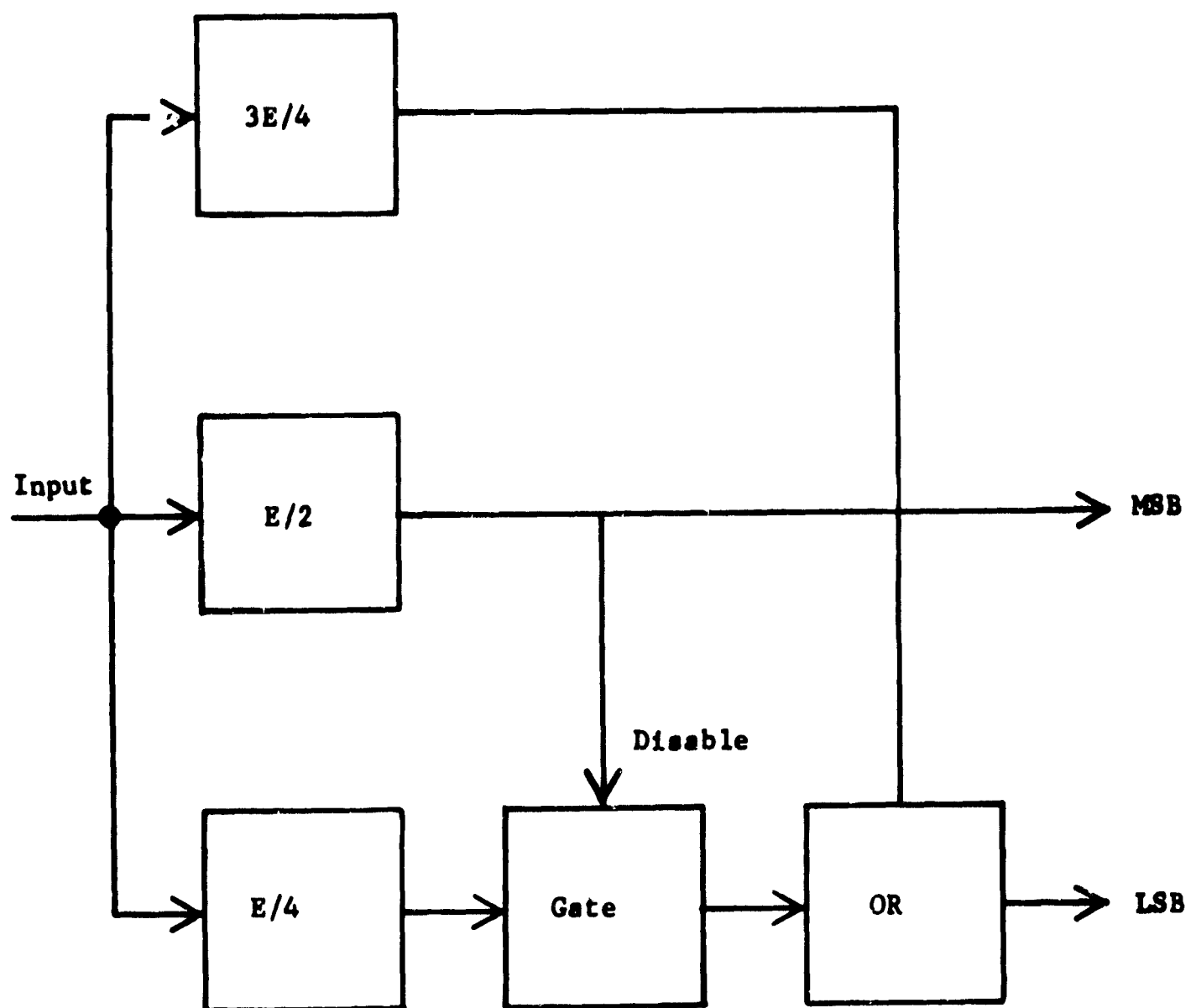


Fig. 21--Two bit direct threshold detection converter.

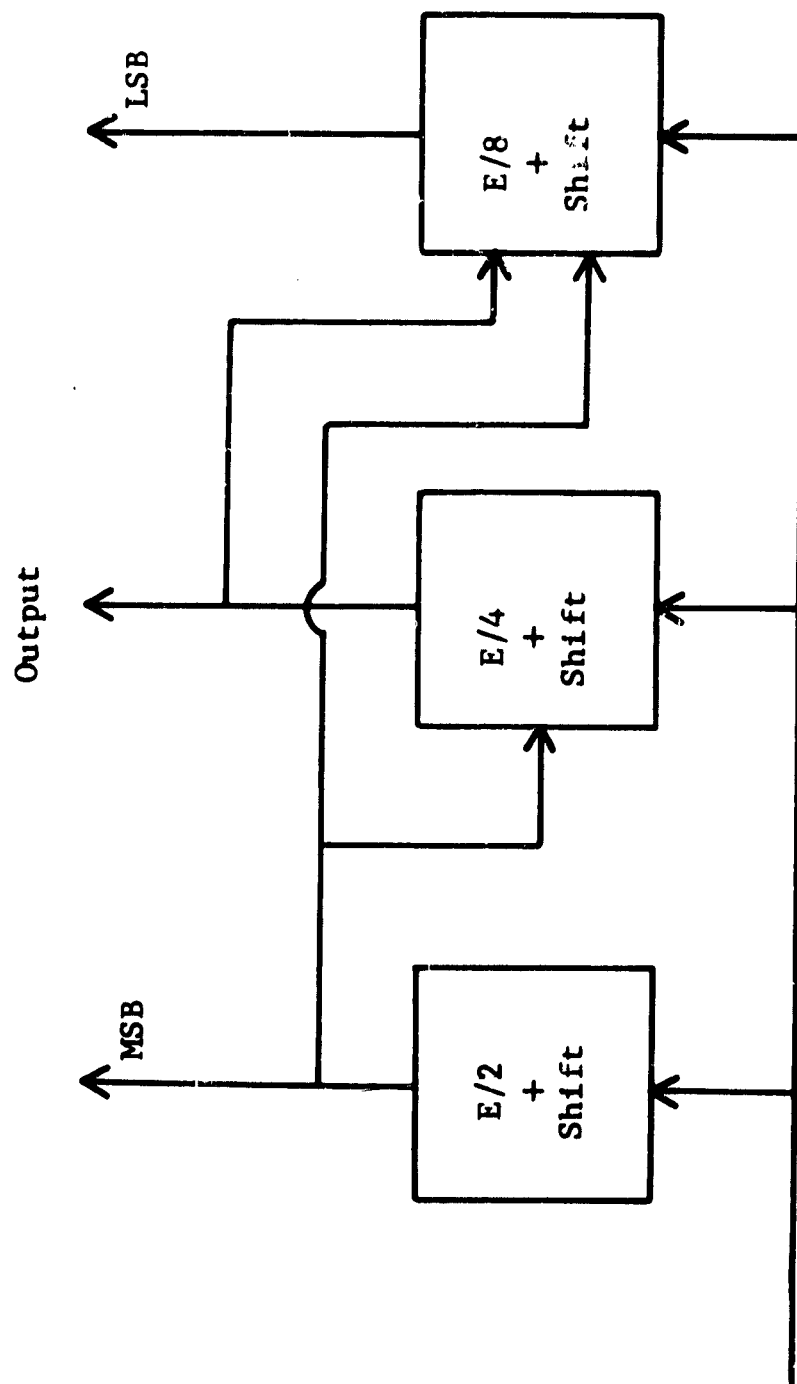


Fig. 22--Three bit sequential approximation converter.

a three bit sequential approximation converter. The encoding range is assumed to be from 0 to E volts. The enabling level of each comparator is determined by its initial setting and a shift. The initial setting of a comparator is the amount of voltage represented by an assertion of the bit which the comparator determines. The indicated shift is the sum of the initial settings of enabled comparators that determine higher order bits. In practice, the reference voltage applied to all comparators, other than the comparator determining the most significant bit, would be supplied by a digital to analog conversion device. The existence of a DA converter will be assumed for the moment.⁹

The preceding examples represent some of the basic principles employed in open loop converters. As previously mentioned such systems do lack an ability to sense or correct any error in digitization. Open loop systems are capable of relatively high speeds and are applicable as low resolution systems.

Closed Loop AD Converters

Closed loop converter operation is illustrated by Figure 23. The closed loop AD converter consists of a difference detector or comparator, an open loop AD converter, and a DA converter in the feedback path. The conversion process is implemented until the feedback analog is determined to be equal to the analog input; then the conversion process is stopped and the digital output is read or stored.

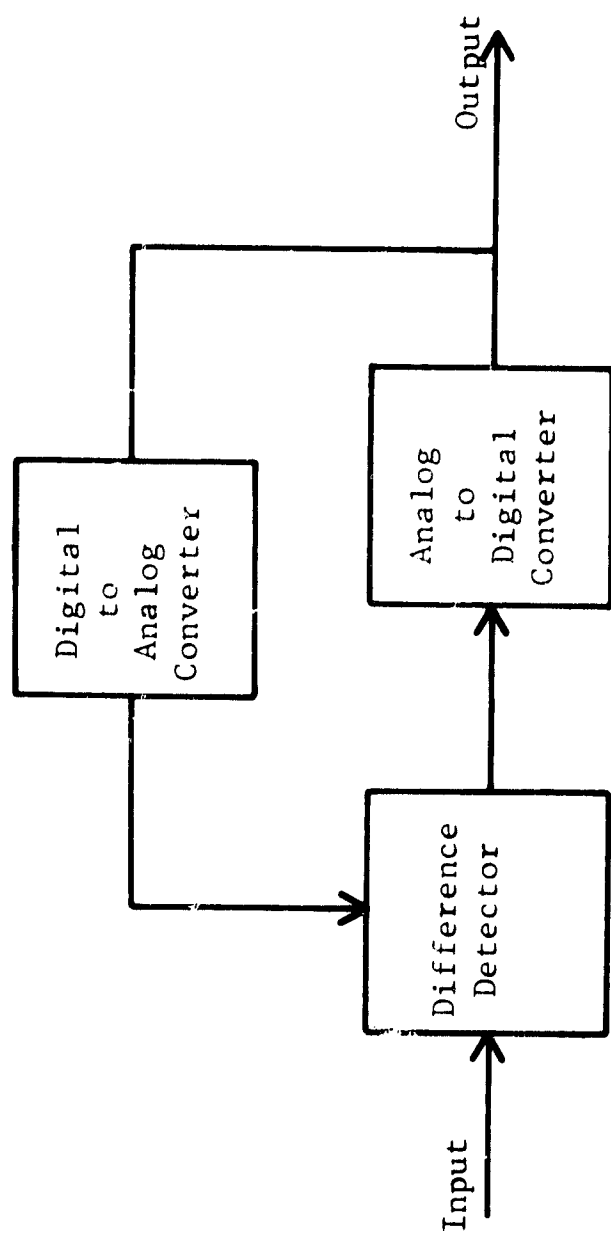


Fig. 23--Basic closed loop analog to digital converter.

For the most part, closed loop converters operate either on a subranging principle or on a counting principle. Considering subranging first, this principle first detects which one of several equal increments of the input range contains the analog input. Once this increment is determined, the increment is then "divided" into the same number of equal parts as was the input range in the initial step. The process would then be continued until the desired resolution was reached. If such a process has S subranges per trial, then after T trials a resolution of one part in S^T will be obtained.¹⁰

Figure 24 is a block diagram of a subranging converter. Such a system must be synchronous, and, therefore, requires a clock. The system must either contain as many comparators as there are subranges per step, or else one comparator must successively have the proper reference voltages applied to implement this action. The control register and programmer contain logic circuits and appropriate DA networks to implement the necessary comparisons.¹⁰

Figure 25 shows the process of subranging. The first conversion step divides the full scale voltage into subranges, in this instance four. The subrange containing the analog is determined. This subrange is then divided into four parts and the process repeated. A subranging converter, operating in the mode that is defined by Figure 25, would be capable of a binary resolution of four bits.

If a subranging converter has two subranges per step, it is known as a successive approximation converter. In the event that the desired

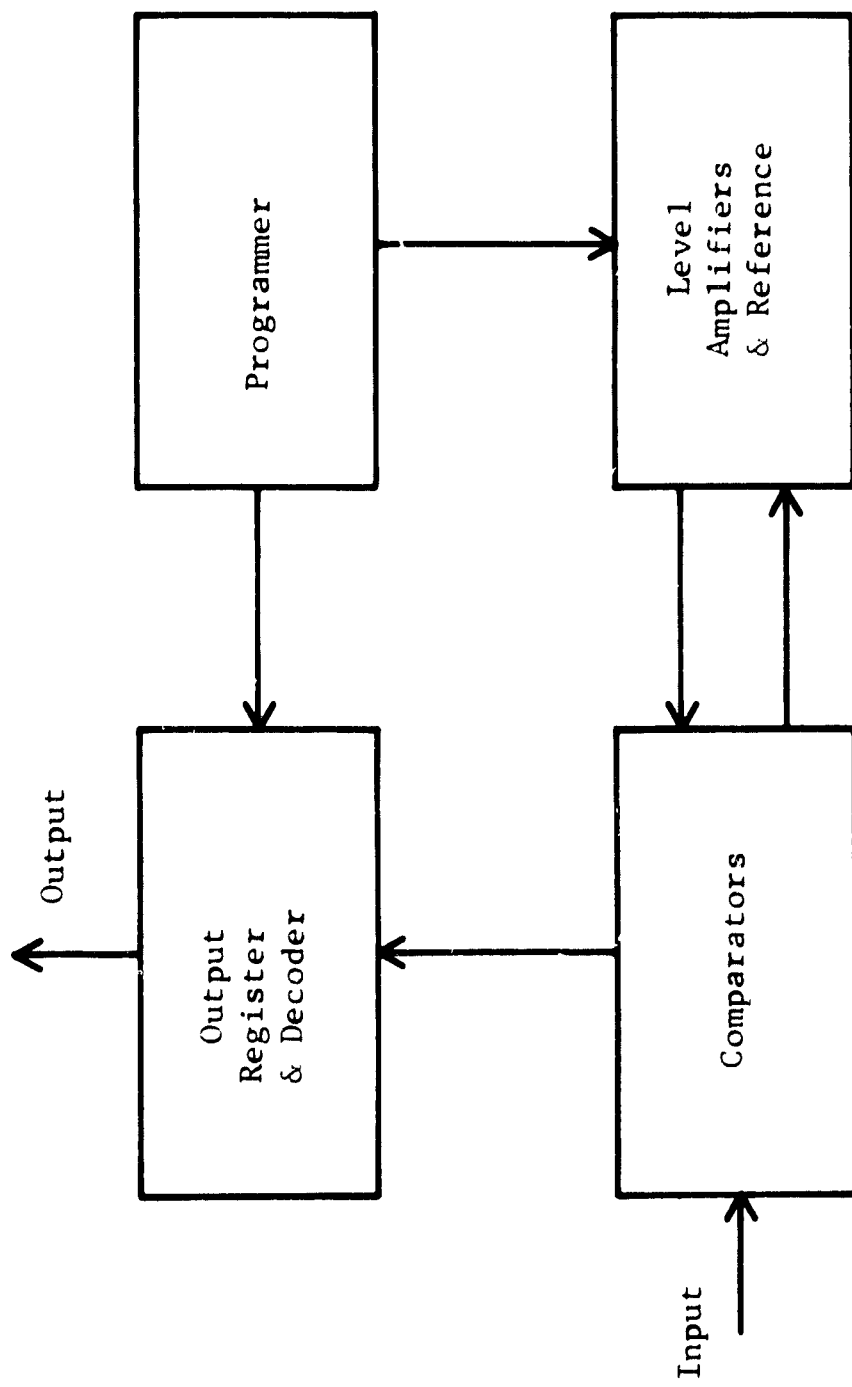


Fig. 24---Subranging converter.

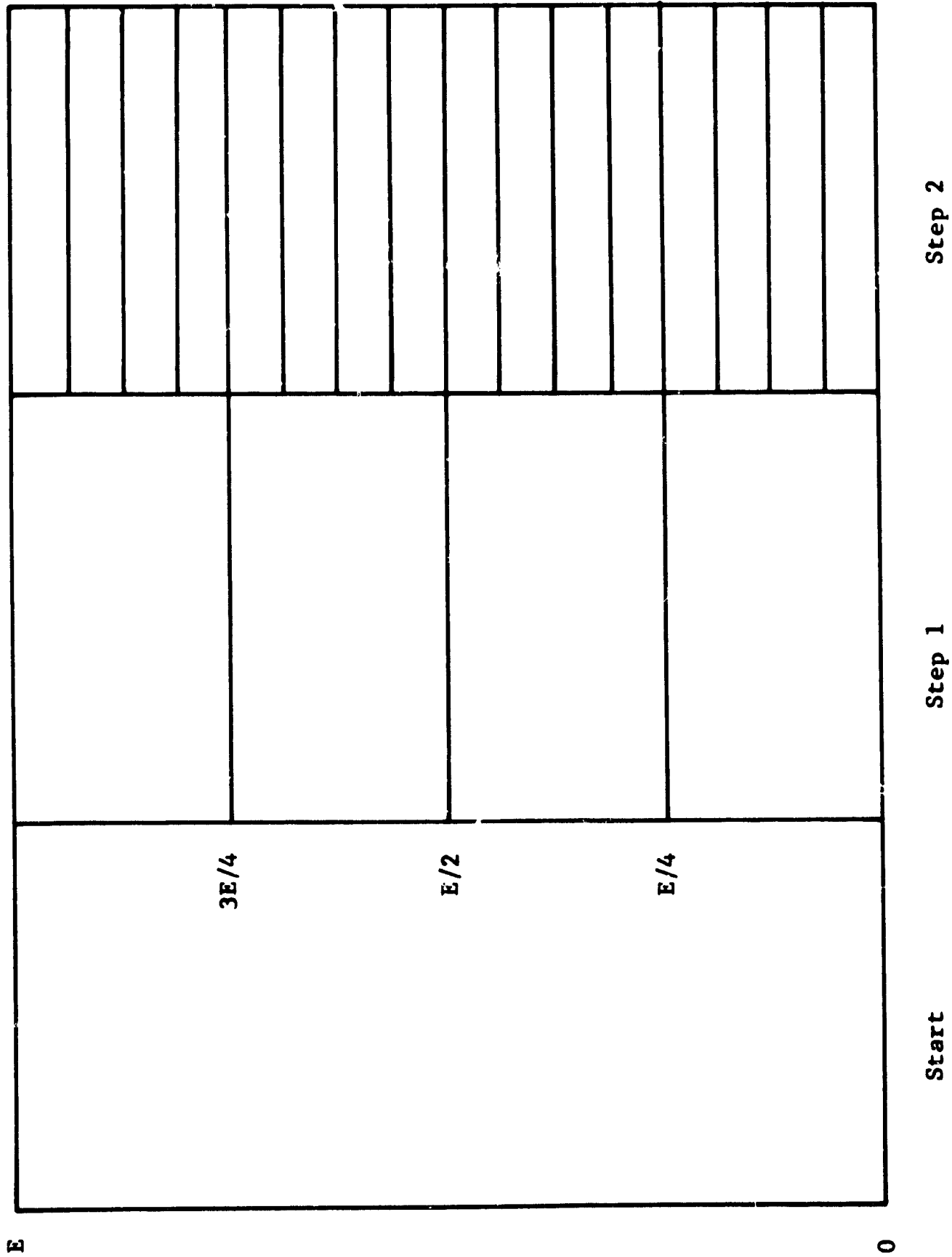


Fig. 25--Subbranging with four subranges per step.

resolution is achieved in one step, the subranging process is identical to the direct threshold converter of Figure 21. Subranging converters are necessarily complex because of the amount of logic circuitry required to implement the subranging.¹⁰

Converters operating on a counting principle are the other basic closed loop converter type. Figure 26 shows such a converter. The clock is used to run the counter whose output is fed into a buffer register. The counter also supplies the input to a DA converter. The counter is allowed to count until the analog feedback is equal to the analog input. When the comparator detects this null state, clock pulses to the counter are inhibited. The programmer implements readout of the digital result and resets the counter to the initial state.¹⁰

The basic counter converter can provide a readout only at intervals of time that are equal to the time required for the counter to count the maximum number of counts. Also, the signal presented to the converter must be continuous during the conversion interval or sample and hold circuitry is required.¹⁰

The basic counter converter may be modified to give a continuous readout if the input signal does not change at a rate faster than that represented by one count. If the signal meets this requirement, the counter section may be modified to function as an up-down counter. The programming and control register directs an up count when the comparator detects that the analog input exceeds the feedback analog. A down count is directed when the feedback analog exceeds the input analog.¹⁰

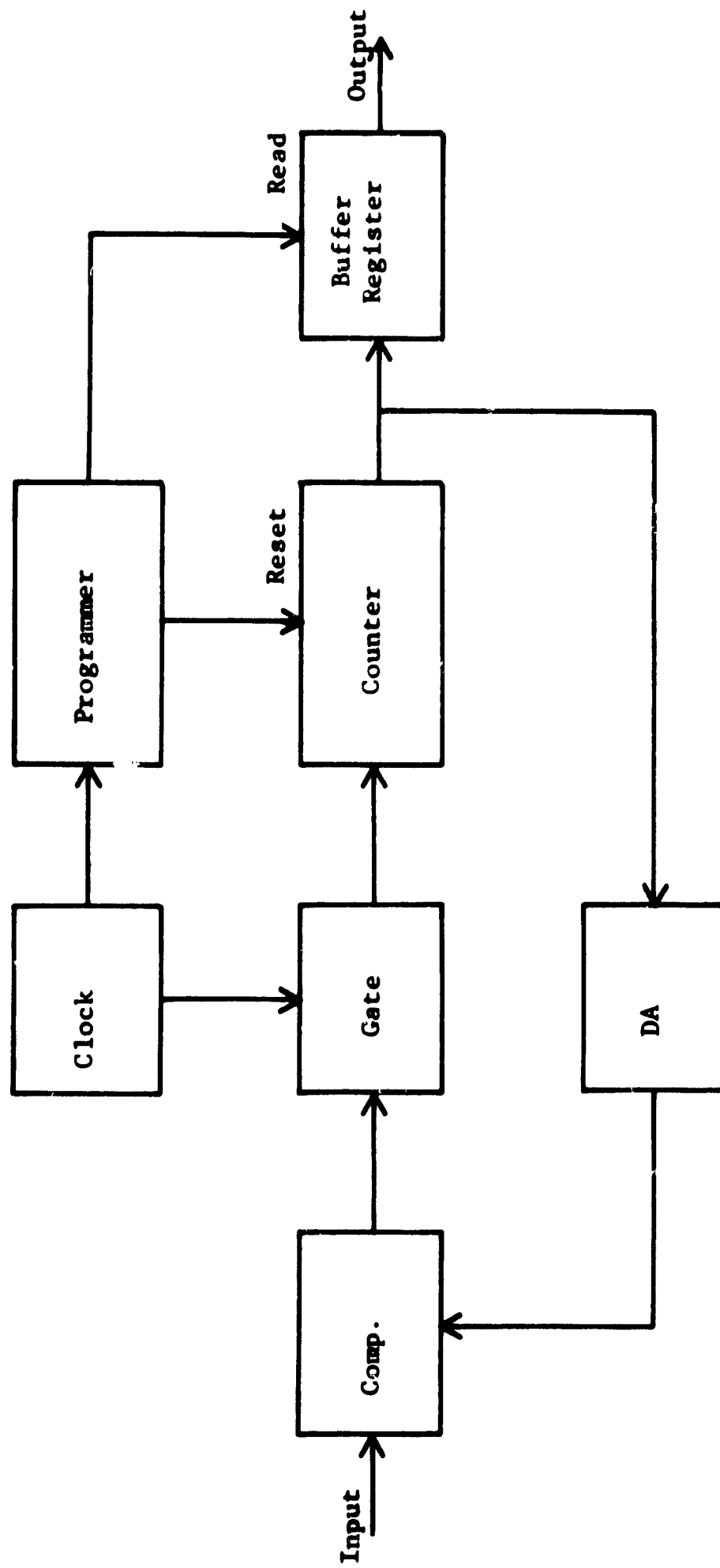


Fig. 26--Basic counter converter.

The AD converters presented by no means encompass all of the known system types. It is true that all AD systems encompass some of the principles put forth in this presentation.

Digital to Analog Conversion

A digital to analog converter transforms a digital input into a representative analog output. For the purposes of this consideration of DA methods, an analog conversion of a binary coded digital input will be examined.

To convert a digital signal into an analog signal, one of two principles must be employed. Figures 27 and 28 illustrate the equivalent networks of the two principles.

A DA converter implementing its operation in the manner of Figure 27 requires an operational amplifier and a network of appropriately weighted resistors. The gain of an operational amplifier operating in the feedback mode as an amplifier may be expressed by the following equation:

$$G = - \frac{R_f}{R_i}$$

where G is the gain, R_f is the value in ohms of the feedback resistor, and R_i is the value in ohms of the input resistance. The negative sign results from the inherent inversion of an operational amplifier.

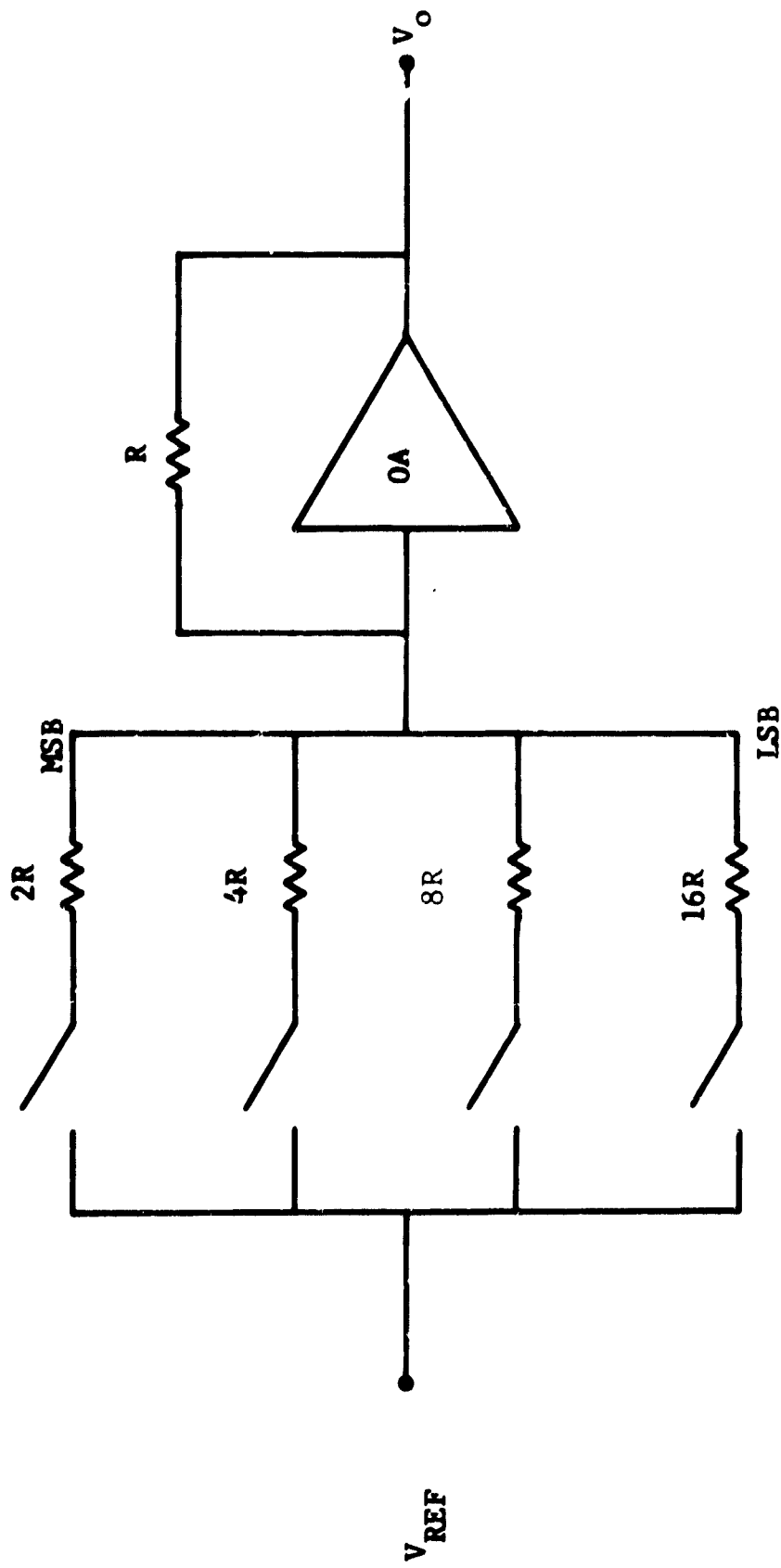


Fig. 27--Equivalent circuit of a four bit DA converter using an operational amplifier and summing network.

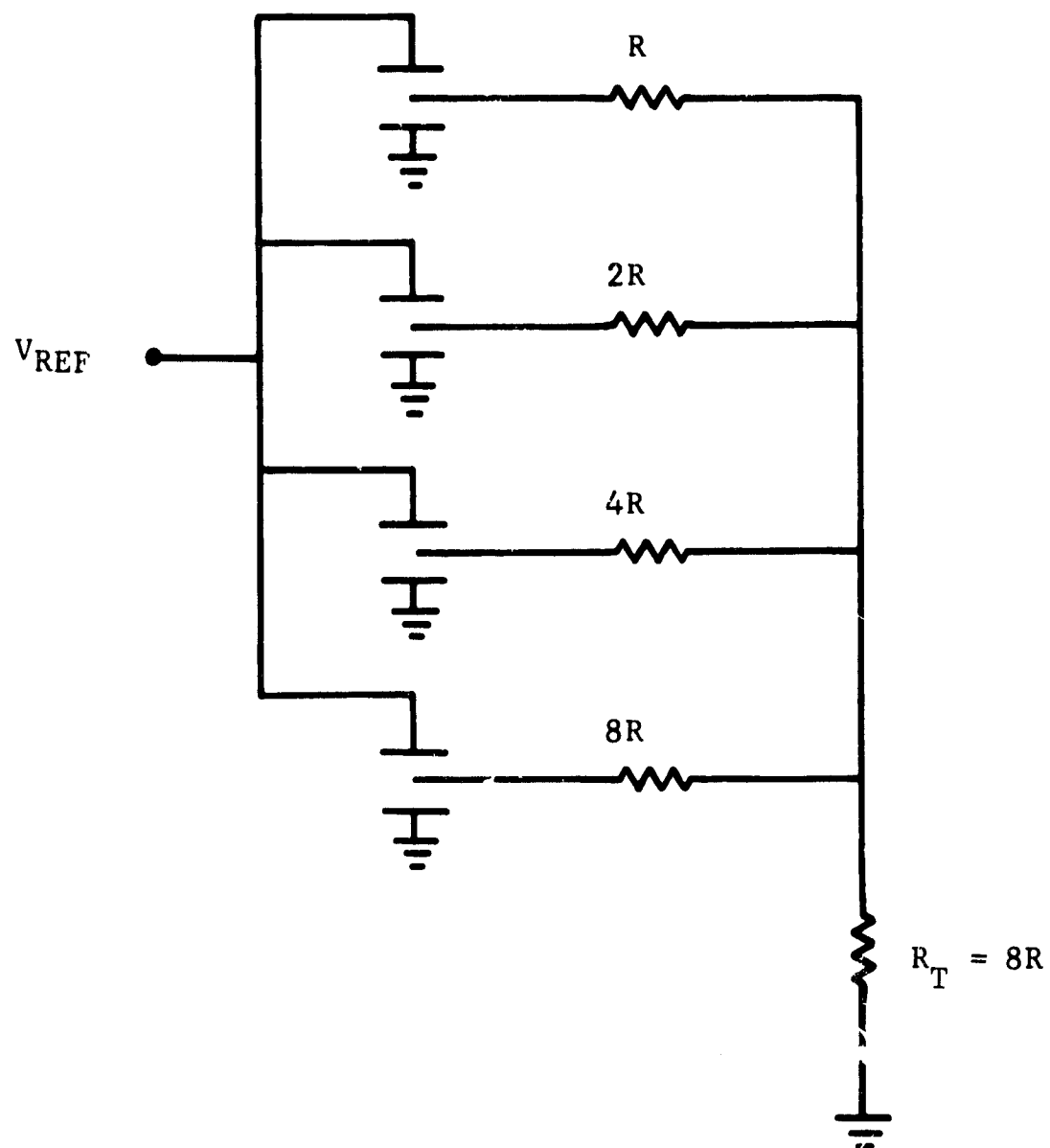


Fig. 28--Equivalent circuit of a four bit DA converter using a bilateral switching network.

In the case of the conversion of a four bit digital input to an analog output, as shown in Figure 27, R_i will consist of four binary weighted resistors. The assertion of a bit connects the resistor representing that bit to the reference source. The most significant bit would be represented by resistor $2R$. The value of the feedback resistor would be one half that of the resistor representing the most significant bit. V_o , the output voltage, may be expressed as:

$$V_o = - \frac{R}{R_{eq}} V_{ref} .$$

R_{eq} is the parallel combination of the resistors representing asserted bits. The voltage inversion requires that the reference voltage have a polarity that is opposite in sign to the desired output polarity.¹¹

The analog of a digital input may be achieved by a converter whose equivalent circuit is represented by Figure 28. The output of this network is taken across resistor R_T which is equal in ohms to the resistor representing the least significant bit. If a bit is not asserted, the representative resistor is grounded and is in parallel with the terminating resistance. The advantage of such a network is that it presents a constant impedance to the load. The disadvantage is that care must be taken to insure that the output is not affected by the load resistance.⁹

Bipolar Conversion

To this point no consideration has been given either to the digitizing of a bipolar signal or to the conversion of a digital signal to a bipolar analog. Needless to say, bipolar signals do exist and must be taken into account.

With regard to AD conversion of a bipolar input signal, the simplest procedure, in principle, is to offset the input signal so that the converter always is presented with a unipolar signal. Another alternative would involve the utilization of two unipolar converters of opposite polarity conversion capability. Some polarity sensing device could be used to gate the input to the appropriate network. Or, instead of two converters, one unipolar converter and a polarity sensor could be used either to gate the input directly to the converter or, if the input were of incorrect polarity, the sensor would cause the analog to be passed through an inverter before being presented to the AD converter. Still a fourth method, which would be applicable only to closed loop systems employing a DA converter in the feedback loop, would be to generate a bipolar feedback analog. This would require that the comparator be capable of accepting both positive and negative input signals and reference voltages in addition to being capable of discriminating absolute voltage magnitudes.

In the case of any bipolar AD conversion of n bits, one bit must necessarily be interpreted as a sign bit; the remaining $n-1$ bits will be available for magnitude representation.⁹ If the AD

output is to be an input to a standard logical arithmetic unit, some coding of the AD output may be required so that, for example, 1100 and 0100 may be interpreted as quantities of opposite sign but equal in absolute magnitude.

Bipolar DA conversion can be achieved by using the sign bit of the digital input to select the reference voltage of proper polarity that is to be applied to the network.⁹ In another application the sign bit could be used either to produce the analog directly or to cause the analog to be directed through an inverter before appearing at the output terminals.⁹ At first glance, it would appear that the latter method of DA conversion might be the most desirable, since only a single polarity reference is required. However, the inversion process, would probably be implemented by an operational amplifier having unity gain. An operational amplifier would require both positive and negative supply voltages; hence, no real simplification would result.

Systems Under Consideration for Implementation

For the digital system under study, it has been decided to construct an AD converter which will provide a digital output of three binary coded bits plus a sign bit. It is desired that this converter be constructed of integrated logic components.

Since the number of bits of resolution to be used currently is relatively low, ordinarily complex AD systems have not been eliminated from consideration. The determining factor in the choice of a system,

at present, is the availability of appropriate components. Every attempt is being made to employ equipment that is either on hand or is readily available from manufacturers of integrated circuit elements.

Two methods of AD conversion are currently being investigated. One system will operate as a direct threshold detection converter along the principles of the converter of Figure 21. In order to realize a conversion of three binary magnitude bits plus a sign bit, the system requires fifteen level detectors and appropriate combinational circuitry to give an output coding that would be compatible with the proposed system's arithmetic unit. In order to increase the magnitude resolution of the AD converter by one bit, an additional sixteen level detectors would be required; and, as a consequence, the coding circuitry would be made more complex. For these reasons, an expansion of this system is not planned. Such a converter is being examined because the necessary basic components are available. Furthermore, it is felt that satisfactory operation may be achieved with a minimum of experimentation, and that design and fabrication of auxiliary components may be held within feasible limits.

The second AD system being considered is a basic counter converter similar to that of Figure 26. A more complete block diagram of the system under investigation is shown in Figure 29. As in the case of Figure 26, the counter is clock driven and is permitted to run until the comparator detects that the feedback analog exceeds the input analog, at which time, the counter will be inhibited. The counter

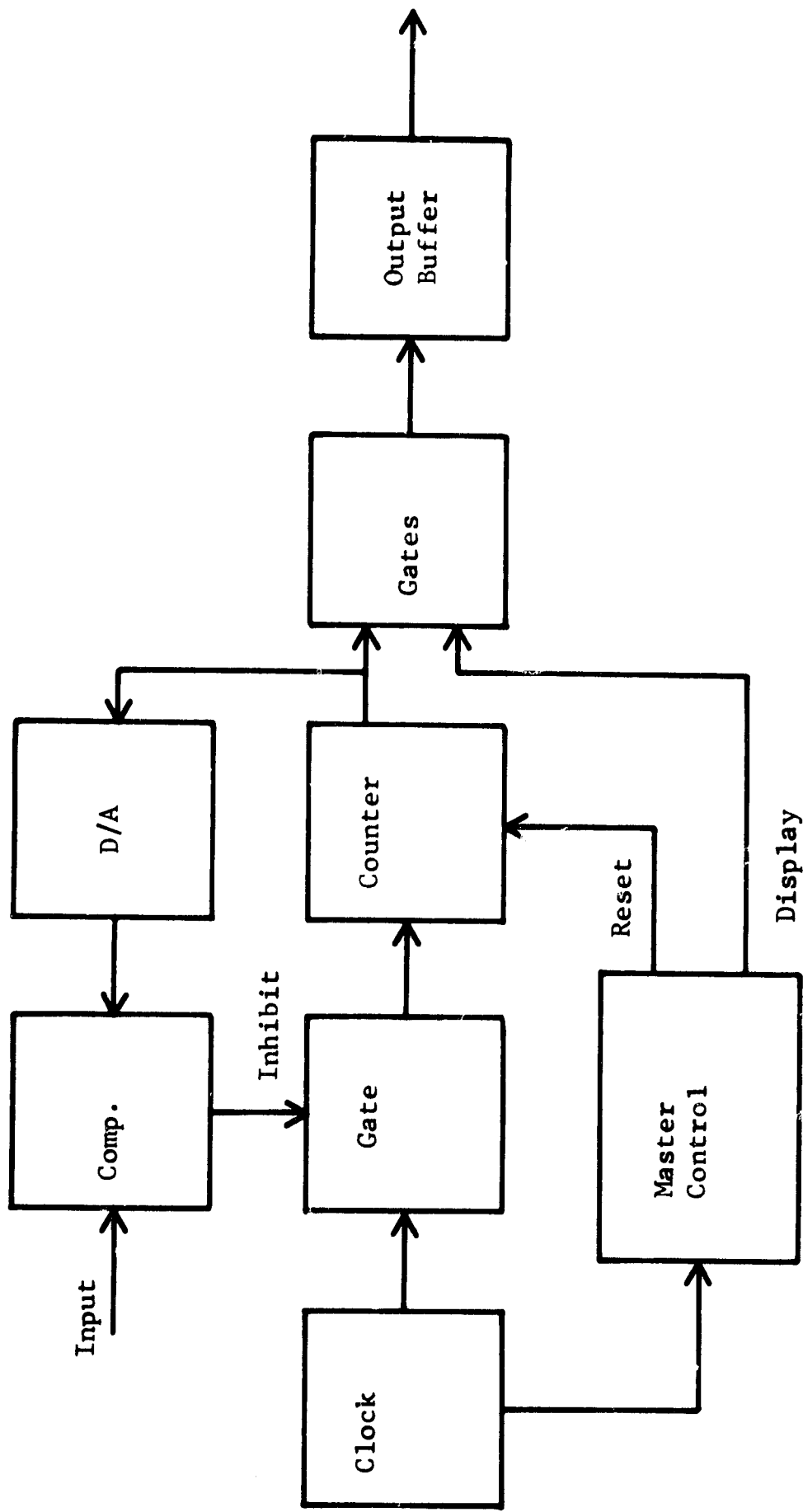


Fig. 29--Counter converter to be implemented.

will function in the binary mode, with the 0000 state being eliminated by presetting a 1 in the least significant bit, in conjunction with the reset command from the master control. The master control will supply the reset to the counter section, and will gate the counter output to the buffer. The input to the output buffer will be gated so that the buffer will hold the conversion result until a new conversion has occurred. The DA feedback analog will be generated by an array whose principle of operation was defined by Figure 28 and the related text.

The level detector or comparator being considered for use in this AD process is currently a μ A710 high-speed, differential comparator manufactured by Fairchild Semiconductor. Investigation of the properties of this device, in particular the input impedance matching, is in progress. The impedance presented by the error signal source and the DA network to the comparator must be as nearly equal in magnitude as possible, and of an order less than 200 ohms.

The counter stage can be realized by the application of one of several types of integrated circuit counter blocks on hand. The master control unit will actually consist of counter stages whose outputs will be logically combined to yield the required timing and control pulses.

At present, an integrated circuit DA converter is not available commercially. This has required the design, fabrication, and investigation of several configurations. Initially, an attempt was made to implement the DA conversion by means of an operational amplifier and summing resistor network similar in principle to Figure 27. However,

it was found that the changing input impedance presented by the summing network did not permit proper functioning of the operational amplifier.

DA systems currently under investigation can be represented by the equivalent circuit of Figure 28. As previously noted, such a configuration presents a constant impedance to the load. However, the magnitude of the resistors that must be employed so as not to exceed the current rating of the switching elements is such that the output impedance is considerably greater than 200 ohms, the maximum input impedance to the comparator.¹² This condition requires the use of an impedance matching network between the DA output and the comparator input.

A network that functions in the desired DA mode has been devised using conventional elements. It is being refined to improve both precision and switching time. The maximum switching speed of the DA governs the maximum conversion speed of the AD system.

The ability of this system to digitize a bipolar input signal is to be realized by offsetting the analog input signal. A precision reference will be employed as the offset voltage.

Evaluation of the Systems to be Implemented

The open loop system will, in theory, lend itself to fabrication from available elements. An offsetting of the input signal would not be necessary, since level detectors capable of sensing both positive

and negative levels are at hand. The fabrication of such a system will not yield a system which can be readily modified to give higher resolution, but the resultant system could be made compatible with the logic circuitry, and it should entail fewer problems than the closed loop device.

Since the number of bits of resolution to be employed at present is not necessarily fixed in number, it is desirable to have a system under study that will lend itself to modification for increased resolution. The counter converter fulfills this qualification of expandability. Although the problems to be surmounted before a functional model is available are more numerous than in the case of the open loop system, it is felt that a satisfactory device can be implemented.

VI. TECHNIQUES FOR FUTURE CONSIDERATION

The following paragraphs deal with three areas of interest encountered in the synthesis of the accelerometer loop compensation network. Each interest area is briefly outlined and is then examined as a possible method to be selected for synthesizing the digital compensator.

Quantizing Effects

Using sampled-data control techniques, a sampled-data system may be compensated theoretically. The digital controller may be described by the block diagram of Figure 3. The only restrictions placed on the digital compensator (by sampled-data theory) is that the multiplier constants be finite.

In practice, however, the actual implementation of the digital controller using integrated circuitry places further restrictions on the multiplier constants. The field of allowable values for the constants is determined by the circuit configuration to be used in implementing the block diagram of Figure 3, and by the encoding method to be used in quantizing the input signal of the digital controller. Mathematical formulation of these allowable fields must be accomplished before these effects may be employed in designing a digital compensator.

Optimal Control Theory

One branch of optimal control theory, discrete dynamic programming, is especially suited for use in the analysis of digital systems.

Discrete dynamic programming may be used, in general, to determine the controller for an m^{th} order system. The m^{th} order system may be described by the following set of first order linear differential equations:

$$\dot{x}_i = f_i(x_1, \dots, x_m, u_1, \dots, u_s, t) \quad i = 1, 2, \dots, m$$

where $u_1, \dots, u_s$ are the system inputs; $x_1, \dots, x_m$, the state variables.

It is proposed that the integral, termed the cost function,

$$V = \int_0^T (\underline{x}' A \underline{x} + \underline{u}' H \underline{u}) dt$$

be minimized by choosing the system inputs as specific functions of time. $\underline{x}$ and $\underline{u}$ are vectors of the m state variables and s forcing functions. A and H are symmetric, constant matrices.

The discrete version of the above system is given by the following relations:

$$\dot{\underline{x}} = B\underline{x} + C\underline{u}$$

$$V = \sum_{n=0}^{N-1} (\underline{x}'_n A \underline{x}_n + \underline{u}'_n H \underline{u}_n) h .$$

The time interval T is subdivided into N equal intervals h , and the forcing function $\underline{u}$ are to be held constant during each subinterval.

Solving the set of m first-order differential equations, the following transition equation is found:

$$\underline{x}_{n+1} = F \underline{x}_n + E \underline{u}_n .$$

When the series V is minimized by the choice of $\underline{u}_0, \underline{u}_1, \dots, \underline{u}_{N-1}$, the result is a function only of the initial state $\underline{x}_0$, because $\underline{x}_n$ is obtained from $\underline{x}_{n-1}$ by the transition equation.

It may be shown that

$$V_n(\underline{x}_0) = \underline{x}_0' G_n \underline{x}_0$$

$$\underline{u}_n = - D_n \underline{x}_0$$

where G_n and D_n are matrices defined below:

$$D_n = (hH + E'G_{n-1}E)^{-1} E'G_{n-1}F$$

$$G_n = (A + D_n'HD_n)h + (F - ED_n)'G_{n-1}(F - ED_n).$$

These equations form an iterative process to be carried out from $n = 1$ to $n = N$. At each subinterval of time, the forcing functions which will minimize the cost V are produced.

In assessing the practicability of the above design procedure, it should be noted that the iterative equations are well suited for use on a digital computer. They can be programmed in matrix form, which is convenient for high-order systems.

A disadvantage of discrete dynamic programming is that each forcing function $\underline{u}_n$ is formed as a linear combination of all the state variables. However, in practice, all the state variables are seldom accessible. Fortunately, techniques are available to eliminate some of the state variables from the feedback matrix.

For the accelerometer control loop, it is desirable to eliminate two state variables from the feedback matrix. Then one could determine the forcing function u (at each subinterval) that would drive a given disturbance value of β to zero in minimum settling time. The forcing function u could be produced as a time varying sequence of numbers whose magnitudes depend upon the size of the disturbance of β .

Linear Sequential Machines

The basic building blocks from which any linear network can be realized are linear machines, whose transfer functions are described by a rational form in the delay operator $P(D)/Q(D)$. It can be shown that linear (binary) sequential machines have the characteristics of these network building blocks.¹³ The linear sequential machines discussed here will be binary machines.

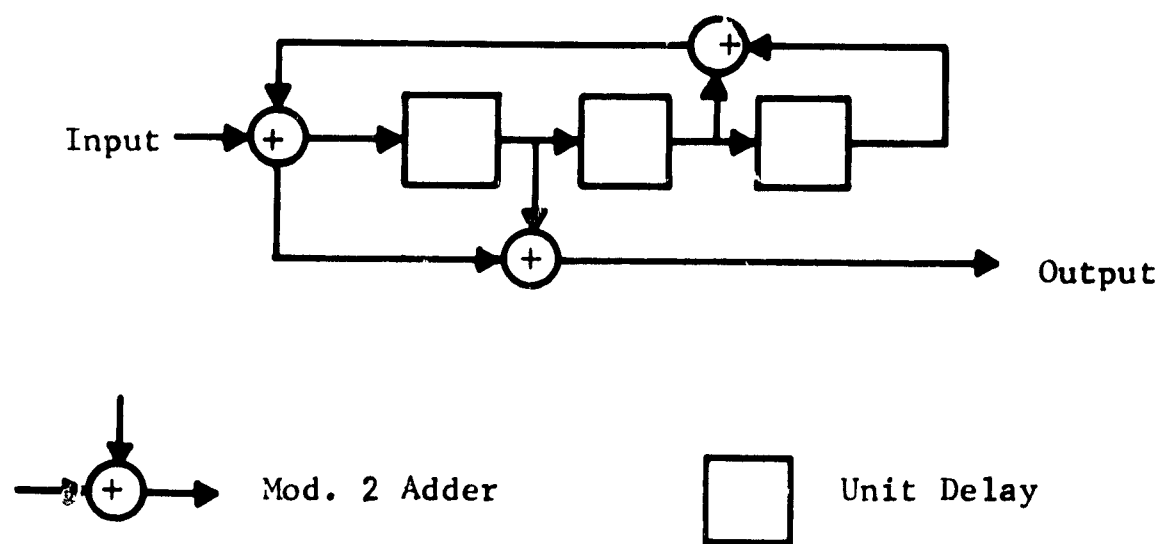
Linear sequential machines are, by definition, sequential machines whose next state and output are linear functions of the previous state and the present input. Linear switching networks, as they are often called, are particularly useful as code generators or code translators.¹³

A given transfer function in the delay operator, D , may be realized with a linear sequential machine comprising a finite number of modulo-two adders (exclusive-or elements) and storage devices (shift register elements).¹⁴ The simplicity of these circuit elements makes the linear sequential machine an attractive approach in realizing transfer functions.

As an example of transfer function realization, suppose machine A has the transfer function:

$$T(D) = \frac{1 + D}{1 + D^2 + D^3}$$

Machine A is implemented as shown in Figure 30.



$$T(D) = \frac{1 + D}{1 + D^2 + D^3}$$

Fig. 30--Standard realization of machine A.

Optimal control theory, as discussed in the previous section, proposes a technique for determining the forcing function u which will drive a given error disturbance β to zero in a minimum amount of time. The forcing function can be produced as a time varying sequence of numbers whose magnitudes depend upon the magnitude of the disturbance.

This time varying sequence may be generated with a linear sequential machine, and this function of the machine could be termed "code generation" or "code translation". Suppose β is encoded using n bits, and that a unique sequence is to be generated for all possible values of β . For a serial input there will be a serial output: one output bit upon acceptance of each input bit with k additional bits of output, where k is the number of unit delays in the machine. Therefore, the length of the output sequence is not restricted to the length of the input sequence n , but is equal to $n + k$. Since there is a unique output for each input, there is a transfer function which describes the function of the sequence generator; this transfer function may be realized in the same manner as shown in Figure 30.

APPENDIX A

FORTRAN SOURCE PROGRAM FOR ROOT LOCUS

THIS PROGRAM FINDS ROOTS OF THE COMPENSATED CHARACTERISTIC EQUATION USING MUELLER'S METHOD

SOURCE DECK

```

C      ROOT LOCUS FOR ACCELEROMETER LOOP
      DIMENSION AR(6),AI(6),SAVE(10)
      1  FORMAT(1H0,1PF14.4)
      2  FORMAT(1H1,3HK=,F12.4)
      10 FORMAT(4F11.4)
      11 FORMAT(1H1)
      WT=1.84
      F=-2.0*(WT*COS(WT)-SIN(WT))/(WT-SIN(WT))
      E=-2.0*COS(WT)
      DO 5 I=1,5
      READ(5,10) A,B,C,D
      G=0.001/1.1
      DO 2 KX1=1,101
      G=G*1.1
      IM=I
      N=4
      AR(1) = 1.0
      AR(2) = G+C+E-1.0
      AR(3) = 1.0 - F+C*E-C+D+E*G+A*G
      AR(4) = D*E-D+C-C*E-1.0+C+G*E*A+G*B
      AR(5) = D-D*E-C+G*A+E*B*C
      CALL SOLVER(N,IV,AR,AI,SAVE)
      WRITE(6,3) G
      WRITE(6,1) (SAVE(J), J=1,8)
      2  CONTINUE
      WRITE(6,11)
      * CONTINUE
      STOP
      END
$IBFTC SOLVER DECK
      SUBROUTINE SOLVER(N,IV,AR,AI,SAVE)
      DIMENSION AR(9),AI(9),SAVE(16)
      C      BEGIN DATA INPUT
      M=N+1
      IF (IM)24,23,24
      23 DO 30 J=1,M
      AI(J)=0.0

```

```

30    CONTINUE
24    INDEX=2
C     END DATA INPUT
C     BEGIN MULLERS METHOD
5     KOUNT = 0
      TOL=1.E-10
      ITPR=50
      P1R=AP(N+1)+AP(N)+AP(N-1)
      P1I=AI(N+1)+AI(N)+AI(N-1)
      P2R=AP(N+1)
      P2I=AI(N+1)
      U=AR(N)**2-AI(N)**2-4.0*(AR(N-1)*AR(N+1)
C-AI(N-1)*AI(N+1))
      V=2.0*AR(N)*AI(N)-4.0*(AR(N-1)*AI(N+1)
C+AI(N-1)*AR(N+1))
      Q=AR(N)
      R=AI(N)
      RAD=1
      GO TO 500
50    TEMP=RADR**2+RADI**2
      IF(TEMP) 510,510,511
510   XR=.5
      XI=C.0
      GO TO 512
511   XR=-2.0*(AR(N+1)*RADR+AI(N+1)*RADI)/TEMP
      XI=-2.0*(AI(N+1)*RADR-AR(N+1)*RADI)/TEMP
512   HR=XR
      HI=XI
      WR=-XR
      WI=-XI
      KDOLY=1
      GO TO 41
504   TEMP=(ABS(PR)+ABS(P1))/(ABS(P2R)+ABS(P2I))
      IF(TEMP-1.0) 505,505,506
506   WR=.5*WR
      WI=.5*WI
      HR=-WR
      HI=-WI
      XR=HR
      XI=HI
      GO TO 51
505   P3R=PR
      P3I=PI
      DEN=ABS(P3R)+ABS(P3I)
      IF(DEN) 7,7,5211
5211  XPO=XR
      XIO=XI
      IF(ABS(P1R-P2R)-1.E-25) 521,521,525
521   IF(ABS(P1I-P2I)-1.E-25) 522,522,525
522   IF(ABS(P1R-P3R)-1.E-25) 523,523,525
523   IF(ABS(P1I-P3I)-1.E-25) 524,524,525
524   WR=1.0
      WI=C.0

```

```

GO TO 53
525 TEMP=WR+1.0
DR=TEMP*P3R-WI*P3I
DI=TEMP*P3I+WI*P3R
OR=TEMP*P2R-WI*P2I
RI=TEMP*P2I+WI*P2R
TEMP=WR*P1R-WI*P1I-Q+P3R
FI=WR*P1I+WI*P1R-R+P3I
FR=WR*TEMP-WI*FI
FI=WR*FI+WI*TEMP
Q=FR-Q+DR
R=FI-R+DI
U=Q*Q-R*R-4.0*(DR*FR-DI*FI)
V=2.0*Q*R-4.0*(DR*FI+DI*FR)
KRAN=2
GO TO 500
526 TEMP=RADR**2+RADI**2
WR=-2.0*(DR*RADR+DI*RADI)/TEMP
WI=-2.0*(DI*RADR-DR*RADI)/TEMP
53 HRO=HR
HIO=HI
HP=WR*HRO-WI*HIO
HI=WR*HIC+WI*HRO
XR=HR+XRO
XI=HI+XIC
KPOLY=2
GO TO 41
527 TEMP=(ABS(DR)+ABS(DI))/DEN
IF(TEMP-10.0)6,6,528
528 WR=0.5*WR
WI=0.5*WI
HR=HRO
HI=HIO
GO TO 53
6 KOUNT=KOUNT+1
C APPLY CONVERGENCE CRITERION
TEST=ABS(XR-XRO)+ABS(XI-XIO)
TEMP=ABS(XR)+ABS(XI)
IF(TEMP-1.0)62,62,61
61 TEST=TEST/TEMP
62 IF(TEST-TOL)7,7,64
64 P1R=P2R
P1I=P2I
P2R=P3R
P2I=P3I
P3R=PR
P3I=PI
IF(KOUNT-ITR8+4)52,541,541
541 IF(KOUNT-ITR8)52,542,542
C SET SS2 FOR NEW ITR8 AND NEW TOL IF DESIRED.
543 ITR8=ITR8+10
TOL=TOL*10.
GO TO 52

```

```

7      SAVE(INDEX-1)=XR
      SAVE(INDEX)=XI
      INDEX=INDEX+2
C      SUBROUTINE FOR CALC. COEFFS. OF POLY. DEGREE N-1 FROM
C      POLY. DEGREE N WHEN DIV. BY FACTOR  $X-(XR+I*XI)$ . ALL
C      NOS. COMPLEX. ARGUMENTS ARE N,XR,XI,AP(J),AI(J),J=1,
C      ...,N+1. RESULTS OF CALC. ARE AR(J),AI(J),J=1,...,N,AND
C      REMR,REMI.
      M=N+1
      DO 76 J=2,M
      AP(J)=AR(J)+XR*AP(J-1)-XI*AI(J-1)
76     AI(J)=AI(J)+XR*AI(J-1)+XI*AP(J-1)
C      END SUBROUTINE
      N=N-1
      IF(N-1)9,81,5
81     TEMP=AR(1)**2+AI(1)**2
      XR=-(AR(2)*AR(1)+AI(2)*AI(1))/TEMP
      XI=(AR(2)*AI(1)-AI(2)*AR(1))/TEMP
      SAVE(INDEX-1)=XR
      SAVE(INDEX)=XI
      9 RETURN
C      SUBROUTINE FOR THE CALC.
C      (RADR,RADI)=(Q,R)+SOR(U,V) OF GREATEST MAGNITUDE.
      500 TEMP=SQRT(U*U+V*V)
      IF(TEMP+U)5001,5001,5002
5001   RADR=C.O
      GO TO 5003
      5002 RADR=SQRT(TEMP+U)
5003   IF(TEMP-U)5004,5004,5005
5004   RADI=C.O
      GO TO 5006
      5005 RADI=SQRT(TEMP-U)
5006   TEMP=Q*RADR+R*RADI
      IF(TEMP)501,501,502
501    RADR=Q-RADR/1.4142135
      RADI=R-RADI/1.4142135
      GO TO 503
502    RADR=Q+RADR/1.4142135
      RADI=R+RADI/1.4142135
503    GO TO (50,526),KPAD
C      END SUBROUTINE
C      SUBROUTINE FOR EVALUATION OF P(X), X COMPLEX, THAT IS,
C      (PR,PI)=(...((A(1)*X+A(2))*X+A(3))...)*X+A(N+1)
C      ARGUMENTS ARE N,XR,XI,AR(J),AI(J).
41     M=N+1
      PR=AR(1)
      PI=AI(1)
      DO 42 J=2,M
      TEMP=XI*PR+XR*PI+AI(J)
      PR=XR*PR-XI*PI+AR(J)
      PI=TEMP
      IF(ABS(PR)+ABS(PI)-1.E27)42,42,420
42     CONTINUE

```

420 GO TO(504,527).KPOLY
C END SUBROUTINE
END

•1500E	01	0.0	0.0	0.0
•2000E	01	0.0	0.0	0.0
•3000E	01	0.0	0.0	0.0
•5000E	01	0.0	0.0	0.0
•1000E	02	0.0	0.0	0.0

APPENDIX B

FORTRAN SOURCE PROGRAM FOR BODE PLOT

THIS PROGRAM PLOTS THE MAGNITUDE IN DECIBELS AND PHASE IN DEGREES OF THE COMPENSATED OPEN-LOOP TRANSFER FUNCTION VERSUS FREQUENCY IN THE W-PLANE

SOURCE DECK

```

C      BODE PLOT FOR COMPENSATED ACCELEROMETER LOOP
      COMPLEX Z,G,CMPLX
      WRITE (6,6)
      GAIN=0.1
      WT=1.84
      F=-2.0*(WT*COS(WT)-SIN(WT))/(WT-SIN(WT))
      F=-2.0*COS(WT)
      A=3.1415927/36.0
      PHI=0.0
      DO 5 I=1,35
      PHI=PHI+A
      IF(ABS(WT-PHI)-1.0E-20) 3,1,1
1  AX=COS(PHI)
      BX=SIN(PHI)
      Z=CMPLX(AX,BX)
      G=GAIN*(Z*Z+F*Z+1.0)/(Z-1.0)/(Z*Z+F*Z+1.0)*(Z+2.0)/Z
      ARSVAL=CABS(G)
      DR=20.0*ALOG10(ARSVAL)
      PHASE=57.29578*ATAN2(AIMAG(G),REAL(G))
      OMEGAW=TAN(PHI/2.0)
      WRITE(6,2) OMEGAW,ARSVAL,DR,PHASE,G
2  FORMAT(1H0,1P6E14.4)
      GO TO 5
3  OMEGAW=TAN(PHI/2.0)
      WRITE(6,4) OMEGAW
4  FORMAT(1H0,1P6E14.4,10X,11HSINGULARITY)
5  CONTINUE
6  FORMAT(1H1,6X,6HOMEGAW,8X,6HARSVAL,10X,2HDR,12X,
      F3HPHI,9X,6HREAL G,8X,6HIMAG G)
      STOP
      END

```

APPENDIX C

FORTRAN SOURCE PROGRAM FOR TIME DOMAIN ANALYSIS OF SYSTEM

THIS PROGRAM USES THE STATE TRANSITION EQUATIONS TO PLOT THE RESPONSE OF THE SYSTEM FOR A TEST SIGNAL

SOURCE DECK

```

C      TIME DOMAIN SIMULATION OF THE COMPENSATED SYSTEM
      DIMENSION X1(2),X2(2),X3(2),RQ(2),D(2)
1     FORMAT(1H0,12,5X,10A514,4)
2     FORMAT(1H1)
      WRITE(6,2)
      N=1
      H=1.0E+06
      WN=184.0
      T=0.01
      RVAX=2.5
      GP=562.0
      GT=0.417
      AN=8.0
      R=5.0E+04
      TEST=0.0
C      GDA IS THE TOTAL LOOP GAIN
      GDA=0.1
      W1=SIN(WN*T)/WN
      W2=COS(WN*T)
      W3=(1.0-W2)/WN**2
      W4=(T-W1)/H
      W5=(1.0-W2)/H
      W6=WN*SIN(WN*T)
      W7=W6/H
      AQ=GDA/GT/GP/(T-W1)*H
      RQC=0.0
      RQ(1)=0.0
      X1(1)=0.0
      X2(1)=0.0
      X3(1)=0.0
      D(1)=0.0
      DO 5 I=1,101
      IF(I-11) 4,3,3
3     R=TEST
4     WRITE(6,1) I,RQC,RQ(N),X1(N),X2(N),X3(N),D(N)
C      BEGIN ANALOG PORTION SIMULATION
      X1(N+1)=X1(N)+W1*X2(N)+W3*X3(N)+W4*(R-GT*D(N))
    
```

```

X2(N+1)=W2*X2(N)+W1*X3(N)+W5*(R-GT*D(N))
X3(N+1)=W2*X3(N)-W4*X2(N)+W7*(R-GT*D(N))
RDC=GP*X1(N+1)
C   END   ANALOG PORTION SIMULATION
C   BEGIN DIGITAL UNIT SIMULATION
CALL QUANT(RDC,RMAX,RQ(2),AN)
D(N+1)=AQ*(RQ(N+1)+2.0*RQ(N))
C   END   DIGITAL UNIT SIMULATION
RQ(N)=RQ(N+1)
X1(N)=X1(N+1)
X2(N)=X2(N+1)
X3(N)=X3(N+1)
5 D(N)=D(N+1)
STOP
END
$IFETC QUANT
SUBROUTINE QUANT(RDC,RMAX,RQ,AN)
C   SUBROUTINE FOR QUANTIZING BETA
F1=AN*RMAX/(AN-1.0)
TEST=-F1
N=2.0*AN-1.0
DO 1 K=1,N
TEST=TEST+F1/AN
IF(RDC-TEST) 2,2,1
1 CONTINUE
2 RQ=TEST
RETURN
END

```

APPENDIX D

FORTRAN SOURCE PROGRAM FOR TRUTH TABLE OF COMBINATIONAL NETWORK

THIS PROGRAM SIMULATES THE FUNCTION OF THE DIGITAL CONTROLLER AND PRODUCES A TRUTH TABLE FOR REALIZING A COMBINATIONAL CIRCUIT AS A PORTION OF THE CONTROLLER

SOURCE DECK

```

      READ(5, 10) C, X1M, X2M, X3M
10  FORMAT(4E11.4)
      WRITE(6, 11) C, X1M, X2M, X3M
11  FORMAT(1H1, 4E11.4)
      E1 = 8.0*X1M/7.0
      E2 = 8.0*X2M/7.0
      E3 = 8.0*X3M/7.0
      X1 = -E1
      DO 4 I = 1, 15
      CALL TABLE (I, NA, NB, NC, ND)
      X1 = X1 + E1/8.0
      X2 = -E2
      DO 4 J = 1, 15
      CALL TABLE (J, NE, NF, NG, NH)
      X2 = X2 + E2/8.0
      X3 = X1 + C*X2
      TEST = -E3
      DO 1 K = 1, 15
      IF(K - 15) 7, 2, 2
7  TEST = TEST + E3/8.0
      IF(X3 - TEST) 2, 2, 1
1  CONTINUE
2  CALL TABLE (K, NZ1, NZ2, NZ3, NZ4)
      WRITE(6, 3) I, J, K, NA, NB, NC, ND, NE, NF, NG, NH, NZ1, NZ2,
      CNZ3, NZ4
3  FORMAT(1H0, 3I5, 3X, 4I3, 4X, 4I3, 4X, 4I3)
4  CONTINUE
      STOP
      END
SIBFTC TABLE
      SUBROUTINE TABLE(L, MA, MB, MC, MD)
      MA=0
      MB=0
      MC=0
      MD=0
      IF(L.GE.8) MA=1

```

86

```

IF(L.GE.4.AND.L.LT.8.OR.L.GE.12)MB=1
IF(L.GT.1.AND.L.LT.4)MC=1
IF(L.GT.5.AND.L.LT.8)MC=1
IF(L.GT.9.AND.L.LT.12)MC=1
IF(L.GT.13)MC=1
M = 2*(L/2)
IF(M.NE.1)MD=1
RETURN
END
    
```

2.0000E 00 0.1000E 01 0.1000E 01 3.0000E 00

REFERENCES

1. Benjamin C. Kuo, Analysis and Synthesis of Sampled-Data Control Systems, Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1963.
2. Richard C. Dorf, Time-Domain Analysis and Design of Control Systems, Reading, Mass.: Addison-Wesley Publishing Co., 1965.
3. Montgomery Phister, Jr., Logical Design of Digital Computers, New York, New York: John Wiley and Sons, Inc., 1958, pp. 81-94.
4. H. C. Torng, Logical Design of Switching Systems, Reading, Mass.: Addison-Wesley Publishing Co., 1965.
5. Thomas C. Bartee, Digital Computer Fundamentals, New York, New York: McGraw Hill, 1960.
6. Yaohan Chu, Digital Computer Design Fundamentals, New York, New York: McGraw-Hill, 1962, pp. 10-24.
7. D. J. Gawlowicz, C. K. Taft, and M. J. Wijnschenk, "Compensating Pulse-Data Servos", Control Engineering, Vol. 2, pp. 6-8, June 1965.
8. μ-PAC Integrated Circuit Modules Instruction Manual, Framingham, Massachusetts: Computer Control Company, Inc. June 1965.
9. Data Conversion Circuits and Subsystems, Framingham, Massachusetts: Computer Control Company, Inc., 1964.
10. The Digital Logic Handbook, Maynard, Massachusetts: Digital Equipment Corporation, 1966.
11. David Colin, "High Speed Techniques in Analog-to-Digital Conversion," Electronic Instrument Digest, Vol. 2, No. 4, pp. 49-54.
12. R. J. Widlar, "The Operation and Use of a Fast Integrated Circuit Comparator", Application Bulletin of Fairchild Semiconductor, Fairchild Semiconductor, Mountain View, California, 1966.
13. W. W. Peterson, Error-Correcting Codes, Cambridge, Massachusetts: M.I.T. Press, 1961, pp. 107-117.
14. A Gill, Introduction to the Theory of Finite-State Machines, New York, New York: McGraw-Hill, 1962, pp. 107-117.