A FEASIBILITY STUDY IN APPLYING READ-ONLY MEMORIES TO THE ANALYSIS OF NONLINEAR ANALOG FUNCTION RESPONSES

A Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Edgar Lee Dohmann

December, 1972

ACKNOWLEDGMENT

I would like to gratefully acknowledge the encouragement and assistance of my major advisor, Dr. T. N. Whitaker, in the preparation of this thesis.

My sincere appreciation goes to my wife, Janis, and to my parents for their encouragement and understanding during my college career and particularly during the preparation of this thesis.

I would also like to thank Miss Laverne Spell for her excellent help in typing this thesis.

The research reported here was supported by Houston Engineering Research Corporation (HERCO). I would like to thank my supervisor, Mr. J. E. Coffman, for his assistance and the use of the HERCO computer facility.

A FEASIBILITY STUDY IN APPLYING READ-ONLY MEMORIES TO THE ANALYSIS OF NONLINEAR ANALOG FUNCTION RESPONSES

An Abstract of a Thesis

Presented to

the Faculty of the Department of Electrical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Electrical Engineering

---

by

Edgar Lee Dohmann

December, 1972

ABSTRACT

The measurement of physical qualities such as temperatures, pressures, and flow rates is both desirable and essential in many industrial processes. Devices such as thermocouples and other transducers have been developed which yield a predictable voltage or current response to the external stimulus. The use of such devices which produce voltage and current signals provide economical and convenient capabilities to measure the desired physical qualities because many signals may be terminated and monitored at a central location such as a control room.

Thermocouples for temperature measurement are the most widely used devices in such industrial process applications because of their simplicity, accuracy, and low cost. As most such devices which respond to an external stimulus, the relationship between the voltage generated between thermocouple junctions and the temperature of the junctions is nonlinear.

This nonlinear relationship poses the greatest problem in designing and constructing accurate temperature indicating devices.

Several analog-to-digital conversion techniques have been developed which compensate for the nonlinear function being measured and produce temperature readouts in proper engineering units. All techniques developed for commercial applications strive to attain satisfactory degrees

of speed, simplicity, accuracy, economics, and reliability. Most techniques offer some advantages in two or more of these qualities but no presently available techniques offers a clear advantage in all five qualities.

Recent developments in the field of semiconductor memory devices have made it possible to design a digital temperature indicator using read-only memories (ROMs) with accuracies approaching those attainable with computer based techniques. The speed, simplicity, and reliability of such a ROM technique would be better than any other presently available technique. Predicted trends in future semiconductor developments also make such a technique very attractive economically.

A computer program has been developed to generate ROM truth tables for a nonlinear response system. The truth tables generated by this program are based on the system of quadratic equations which describe the nonlinear function. The accuracies are limited only by the inherent accuracy of the equations used, the size of the ROM, and the dynamic range of the stimulus to be measured.

TABLE OF CONTENTS

PAGE

CHAPTER

I. INTRODUCTION. 1

II. CONVENTIONAL LINEARIZATION TECHNIQUES 3

III. READ-ONLY MEMORY CHARACTERISTICS AND APPLICATION 23

IV. READ-ONLY MEMORY LINEARIZATION APPLICATION. . . 31

V. GENERATING READ-ONLY MEMORY TRUTH TABLES. . . . 39

VI. RESULTS AND CONCLUSIONS 53

BIBLIOGRAPHY. 56

APPENDIX A - FORTRAN COMPUTER PROGRAM FOR ROM TRUTH TABLE GENERATOR. 58

APPENDIX B - DERIVATION OF ADC RESOLUTION AND ROM LINEARIZER ACCURACY. 64

APPENDIX C - TYPE K THERMOCOUPLE CURVE LINEARIZATION RESULTS. 73

CHAPTER I

INTRODUCTION

Many natural processes are neither linear nor quantitative but behave in continuously smooth, nonlinear ways. It is desirable and necessary to measure such nonlinear natural processes in many applications such as industrial process control and operation, laboratory experimentation, and data analysis.

Many devices have been developed which yield a predictable voltage or current response to an external stimulus. Thermocouples are widely used as temperature transducers because of high inherent accuracy, wide measurement range, fast thermal response, ruggedness, reliability, and low cost. Thermocouple circuits develop an electromotive force (emf) at the measurement terminals whose magnitude and polarity depend on properties of the metals used and the temperature difference between the thermocouple junctions.

Although many materials can be combined to produce a thermoelectric effect, certain pairs of metals have become standard. The four most commonly used types of thermocouples are: Type J (Iron/constantan), Type T (copper/constantan), Type K (chromel/alumel), and Type E (chromel/constantan). The type used depends on such factors as desired operating range, external environment, desired emf output range, and installation considerations.

As with most transducer devices which respond to an external stimulus, the relationship between the voltage generated between thermocouple junctions and the temperature of the junctions is nonlinear. Several techniques have been developed to measure thermocouple emf, compensate for the nonlinear function, and produce temperature readouts in proper engineering units. All techniques developed for commercial applications strive to attain satisfactory degrees of speed, simplicity, accuracy, economics, and reliability.

The purpose of this investigation was to evaluate the feasibility of using read-only memory (ROM) devices in the linearization of such quantities as thermocouple-produced emf and develop techniques of applying ROMs to these tasks. While many of the principles and techniques presented and discussed in this report apply to measurement and analysis of nonlinear functions in general, this investigation was limited to the primary field of thermocouple circuits due to their common and widespread use and the availability of background information.

CHAPTER II

CONVENTIONAL LINEARIZATION TECHNIQUES

The simplest method of measuring temperature using thermocouple circuits is to use an analog or digital voltmeter to measure the emf produced and then to refer to a National Bureau of Standards (NBS) thermocouple conversion table to determine the corresponding temperature.

While this manual table-look-up method may be simple it can also be extremely time consuming and impractical in industrial process applications where hundreds or thousands of thermocouple circuits may be terminated in a single control room and it is often nesessary to read and/or record many temperature values in a small amount of time.

Figure 2.1 shows a generalized block diagram of a data acquisition system. Such a system permits an operator to select any one of many input points to be monitored and displayed. The control logic of the data acquisition system multiplexes the selected input point onto a single analog channel then measures the selected quantity and displays the resultant value to the operator.

The generalized block diagram is simplified even further in Figure 2.2. This figure shows the basic component parts of the Analog Measurement/Computation Logic and the resultant display output. Figure 2.2 shows only one analog input value as would be the case after the selected point is

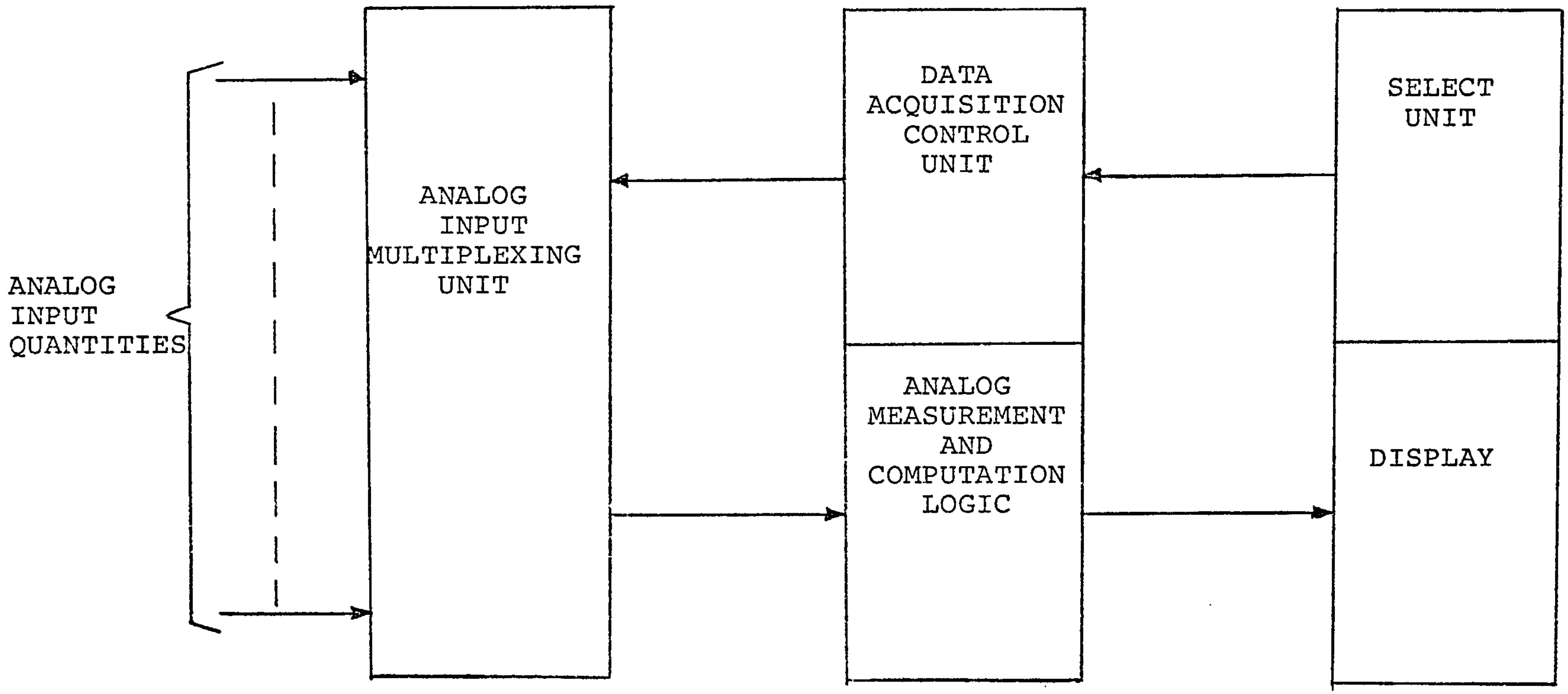


Figure 2.1

Typical data acquisition system

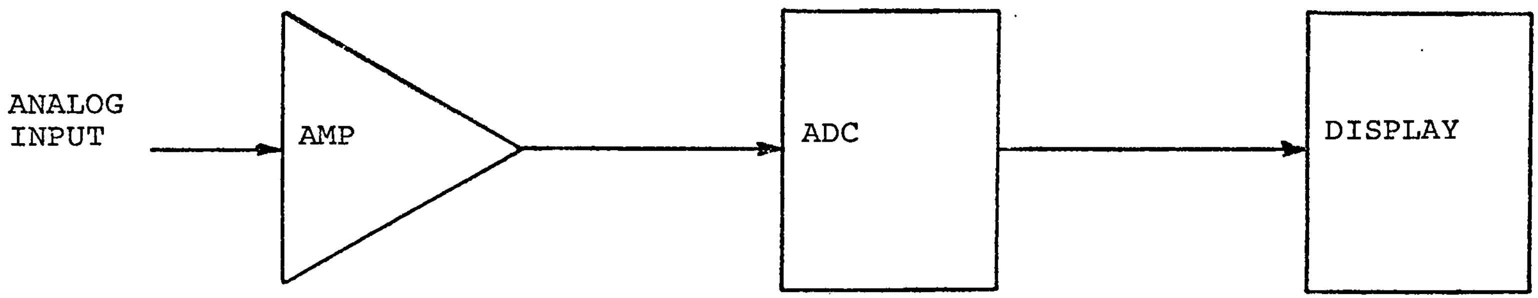


Figure 2.2

Basic analog-to-digital conversion/display system multiplexed onto a single analog input channel. This basic block diagram is referred to throughout the remainder of this chapter as some variation of this block diagram is used in almost every conventional linearization technique.

The two most basic means of linearizing an input function are by analog techniques or by digital techniques although some techniques employ a combination of both. It is not the intent or purpose of this chapter to present and discuss every linearization technique that has ever been developed but rather to discuss in general terms the characteristics of some of the most commonly used techniques to provide a basis for comparison of the read-only memory linearization technique to be discussed in a subsequent chapter.

The basic goals of a linearization technique should be to provide a system which is accurate, simple, economic, fast, and reliable. Each of these qualities is used as a basis for comparing the desirability of one technique over another. Often trade-offs must be made between two or more of these qualities to optimize the adherence to another.

Accuracy, when applied to linearization techniques, normally refers to the ability of the linearization network or device to conform to the published NBS tables without regard to the inherent inaccuracies of the other system elements such as the sensor itself, the signal conditioning amplifier, or the analog-to-digital converter.

Simplicity of a linearization technique includes the space required by the technique, the maintenance required, the versatility of the technique, and the difficulty to design and implement the technique. Versatility is very important because some applications require the use of a particular type of ADC or amplifier but if a linearization technique is highly dependent on some other type of ADC or amplifier, it may not be possible to use that technique.

Economics of a linearization technique includes both the cost of the components used and the cost of implementing the technique.

The speed of a linearization technique refers to the time required for the measured quantity to be adjusted to conform to the actual value of the stimulus.

The reliability of a linearization technique includes the expected life of the components used in the technique and the ability of the technique to continue to conform to the published NBS tables over an extended period of time.

The simplest linearization technique based on using a circuit like that shown in Figure 2.2 would be to scale the displayed data to a value that could be conveniently referenced in an NBS look-up table. This technique would be similar to that mentioned at the beginning of this chapter with the exception that automatic analog multiplexing is implied by the system shown in Figure 2.2. Even though automatic signal multiplexing is provided by this technique, the procedure of conversion by reference to a table is not simple and is certainly not reliable since each reference is subject to human error. The method is also very slow and costly in terms of time and effort expended. Since all five of the desired qualities are severely lacking in this method it is readily evident why automatic linearization techniques are preferred over manual methods.

AMPLIFIER, DIODE FUNCTION GENERATOR FEEDBACK TECHNIQUE

One of the most commonly used linearization techniques is the Amplifier DFG feedback technique. This technique uses the very popular piecewise linear approximation as do most other conventional linearization techniques. The accuracy of any method using piecewise linear approximation is determined by the number of line segments used and the inherent accuracy of the circuitry generating the line segments. Figure 2.3 illustrates the basic piecewise approximation of a nonlinear curve.

The Amplifier DFG feedback technique employs a diode function generator (DFG) network in the amplifier feedback circuit to generate a function which is the inverse to the input voltage. The block diagram of such a system is shown in Figure 2.4 and the typical response curves are shown in Figure 2.5.

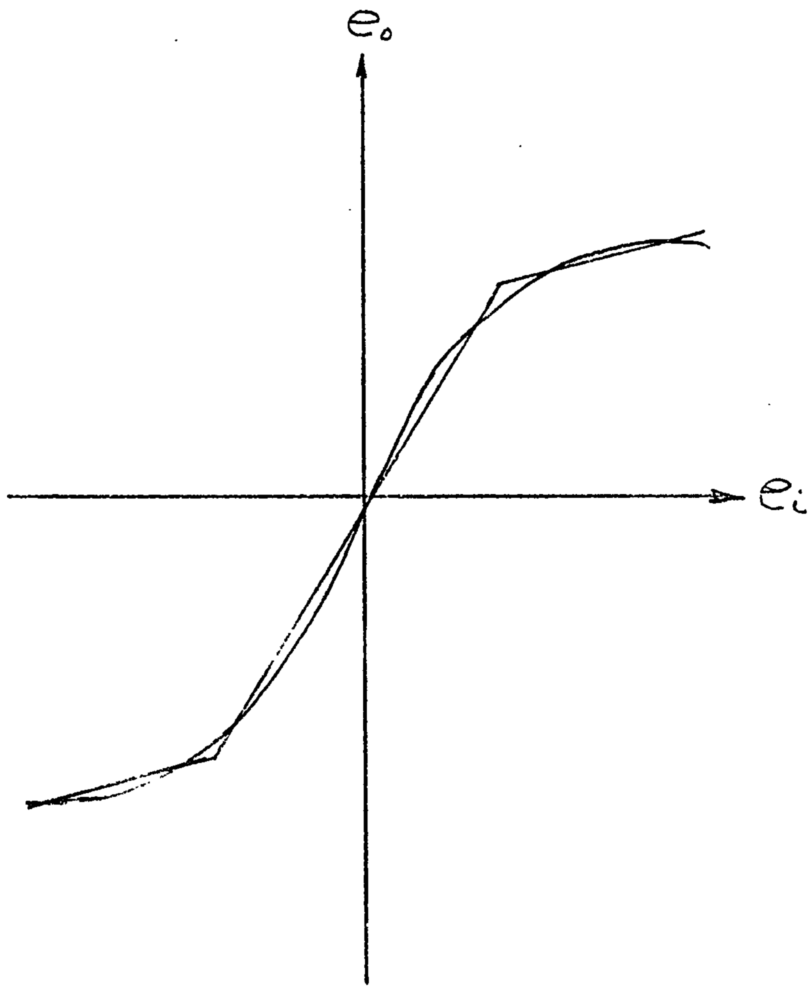


Figure 2.3

Piecewise approximation of a nonlinear function

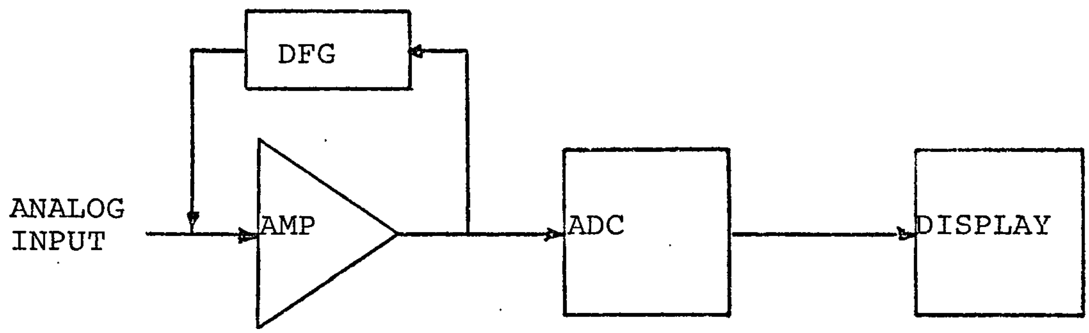
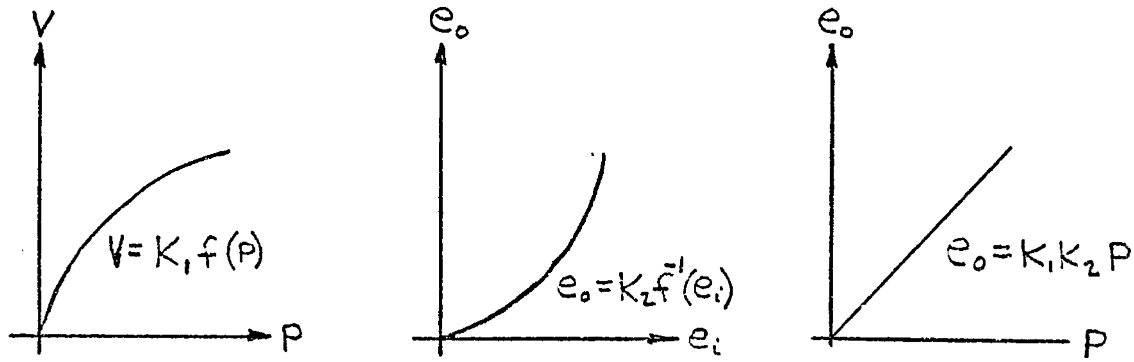


Figure 2.4

Amplifier diode-function-generation feedback technique



(a) Transducer output (b) DFG function (c) Composite transfer curve

Figure 2.5

Linearization of curves by a DFG circuit

The complexity of such a system depends on the number of line segments used and the capabilities provided for temperature-compensation of the breakpoint diodes. Precision limiter circuits using operational amplifiers to simulate ideal diodes can greatly improve the accuracy but would also greatly increase the cost and complexity of the system. This method has the significant advantage of being very versatile and essentially independent of the type of amplifier and ADC used. This system usually depends very much on the reference voltage used to bias the DFG circuit and, normally, each segment of the curve requires a separate adjustment because of inherent variations in diode characteristics. This method has the advantage of being very fast since the time required for the output voltage from the amplifier to stabilize depends only on such factors as amplifier slew rate and settling time and diode switching time.

RAMP LADDER LINEARIZATION TECHNIQUE

One analog-to-digital conversion technique is to allow a counter to accumulate pulses at a fixed rate while a ramp generator is running; then, when the ramp output matches the unknown input signal, the total count accumulated is proportional to that input signal. If the input current to the ramp generator is changed in steps depending on the number of pulses counted, a piecewise linear ramp can be generated as shown in Figure 2.6. If the polarity and magnitude of the ramp is properly calculated, the resultant ramp can approximate the transducer response curve as shown in Figure 2.7.

Figure 2.8 illustrates how such a nonlinear ramp generator might be used in a linearizing data acquisition system. The basic accuracy of such a system depends on the number of line segments used and the inherent accuracies of the components used. Precision components can be used in the ramp ladder circuit to simplify implementation and calibration and to increase reliability. The cost also depends on the desired accuracy and simplicity of calibration requirements.

The two major disadvantages of the ramp ladder linearizer are speed and flexibility. The ADC counter must start at zero at the beginning of the conversion cycle and must count up to the unknown value. For a full scale analog input voltage 2^n-1 clock pulses are required before the conversion is complete (n is the number of bits in the digital word). The frequency of the clock pulses driving the ADC counter is limited also by the response time of the switches in the ramp ladder circuit and the slew rate of the ramp generator.

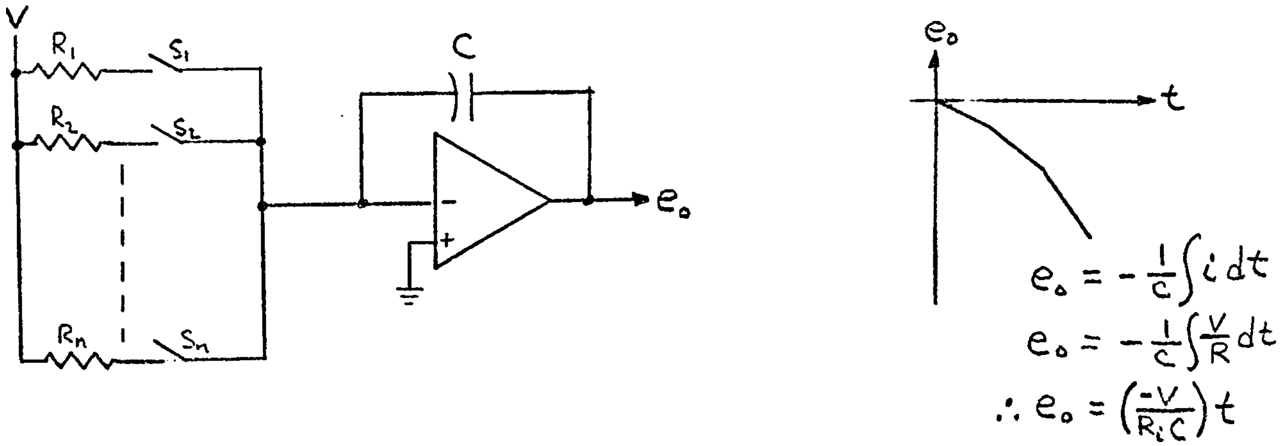


Figure 2.6

Typical nonlinear ramp generator

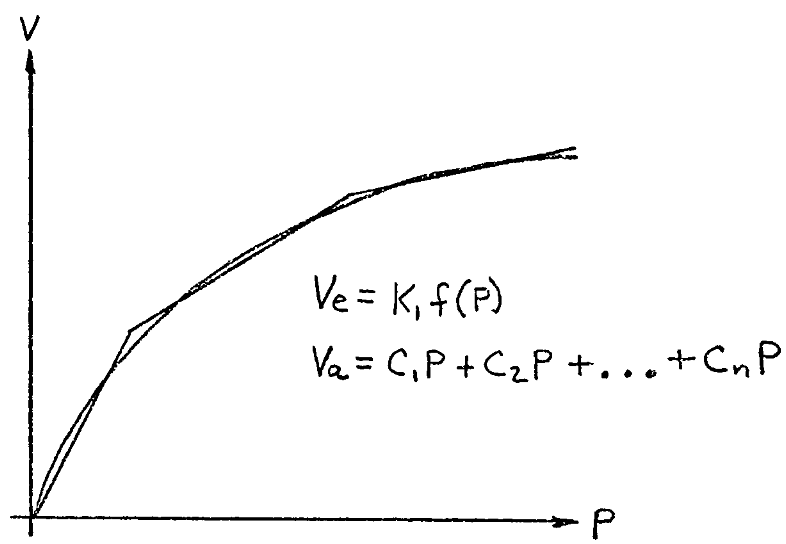


Figure 2.7

Piecewise linear approximation of transducer output

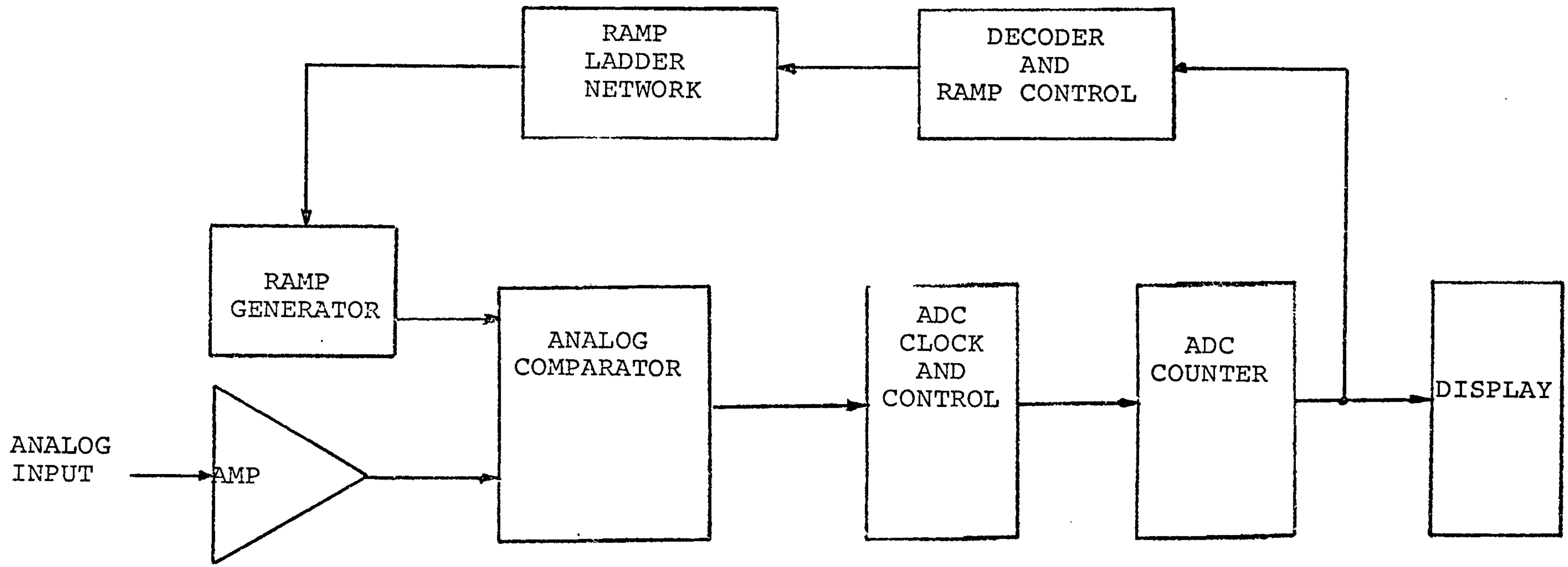


Figure 2.8

Ramp ladder network technique

The ramp ladder linearizer technique requires that a counter-ramp type of ADC be used. This type of ADC must be either a specially designed ADC or one which provides access to the digital count during the conversion cycle and access to the ramp generator input so the decoder/ramp control circuitry and ramp ladder circuitry can be added externally.

VARIABLE SPEED ADC COUNTER TECHNIQUE

Another linearization technique using the basic counter-ramp ADC is the variable speed ADC counter. This is a digital linearization technique as shown in Figure 2.9 using a fixed ramp to control the conversion time but using more than one clock frequency to control the speed of the ADC counter. The resultant output of such a device is shown in Figure 2.10. The number of pulses accumulated in the ADC counter varies with time to approximate the response function.

The accuracy and complexity of the circuitry used in this technique depends on the number of clock frequencies available and the complexity of the decoder circuit used to control the selection of the frequency to drive the counter. The theoretical accuracy obtainable by this method is dependent also on the number of line segments used in the piecewise approximation. The number of precision components needed to make a circuit which is easy to calibrate and

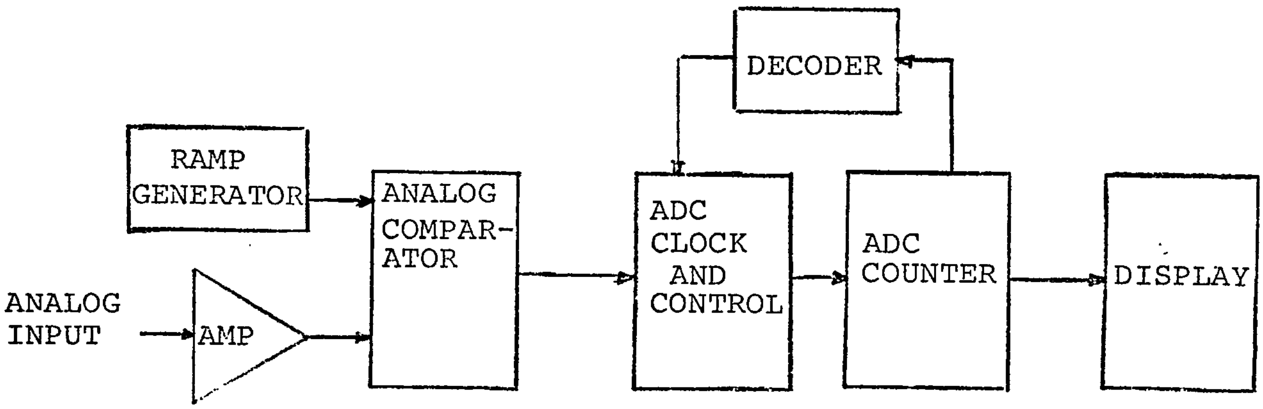


Figure 2.9

Variable speed ADC counter

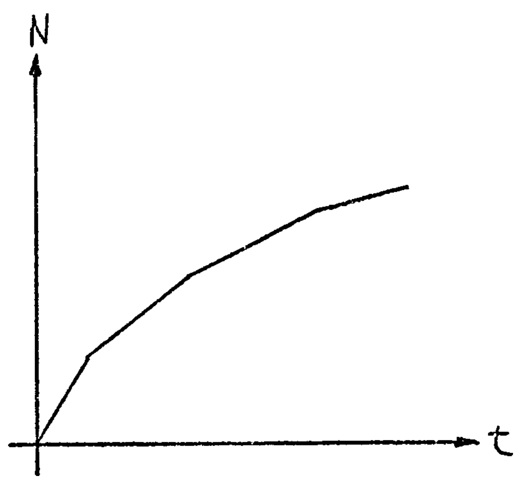


Figure 2.10

Piecewise linear approximation by varying counter speed maintain is much less than required by the ramp ladder technique. A severe limitation of this technique is the requirement that the ADC be specially designed for this application. Another limitation is the speed of the system. In general, however, it should be possible to operate the ADC counter for this method at faster speeds than for the ramp ladder method because the response time of the digital switches controlling the counter pulse frequency is less than the response time of the analog switches on the ramp ladder network. A one-digit ambiguity exists each time the oscillators are switched if discrete oscillators are used to vary the ADC counter speed. If a resistor is switched in the frequency determining network of a single oscillator, this ambiguity does not arise.

ACCUMULATED ADC COUNT CORRECTION TECHNIQUE

Another digital linearization technique accomplishes the same overall effect in a slightly different manner. This method which corrects the accumulated pulse count of the ADC counter has essentially the same general composition as the method shown in Figure 2.9. A fixed ramp generator produces a linear output until a comparison is made with the unknown input. During the conversion cycle the ADC counter is driven at a constant rate but when the decoder/control circuitry detects a "correction point", a pulse is either added to or subtracted from the total accumulated count.

The counter correction circuit can either increment or decrement the ADC counter when the correction point is detected if an up/down counter is used or the total of all additions and subtractions can be maintained until the end of the conversion cycle and the final accumulated value corrected before presentation to the display. The net effect of either method of correcting the accumulated value of the ADC counter is to approximate the response curve by shifting line segments with the same slope.

The major advantage of the two digital techniques is that for the same or less degree of complexity and cost than analog techniques, a piecewise approximation can be realized with many more line segments thereby yielding a more accurate device.

The reliability of the digital techniques discussed here is very good since highly accurate, stable, and reliable components are available to construct these devices and very little calibration of the system would be required.

Even though the ADCs discussed for the counter-ramp conversion techniques used a ramp generator to compare with the unknown input the dual-slope ADC technique could be applied to all 3 linearization methods discussed here. The dual-slope method employs an integrating ramp on the analog input to charge a capacitor to the unknown value. The capacitor is then discharged at a constant rate until the unknown voltage reaches zero and the accumulated pulses in the ADC counter during this discharge time is proportional to the unknown input. If the unknown value were discharged at a non-linear rate or if the clock pulse frequency were varied, or if the accumulated value were adjusted by appropriate values this ADC technique could be used in essentially the same manner as the counter-ramp method.

DIGITAL COMPUTER PROGRAMMING TECHNIQUES

Another commonly used technique is to use a digital computer program to perform the linearization. Such a method as shown in Figure 2.11 would merely examine the digitized value of the analog input and either evaluate a series of polynomial or straight-line equations describing the response function or use a table look-up technique.

The computer program linearization offers the advantage of being very versatile. The method is essentially independent of the type of amplifier and ADC used.

To justify the computer program linearization approach economically, however, the computer would have to be an inherent part of an overall system and the measured data would have to be required by other programs in the overall system. The cost of a stand-alone system as shown in Figure 2.11 would, in general, be prohibitive.

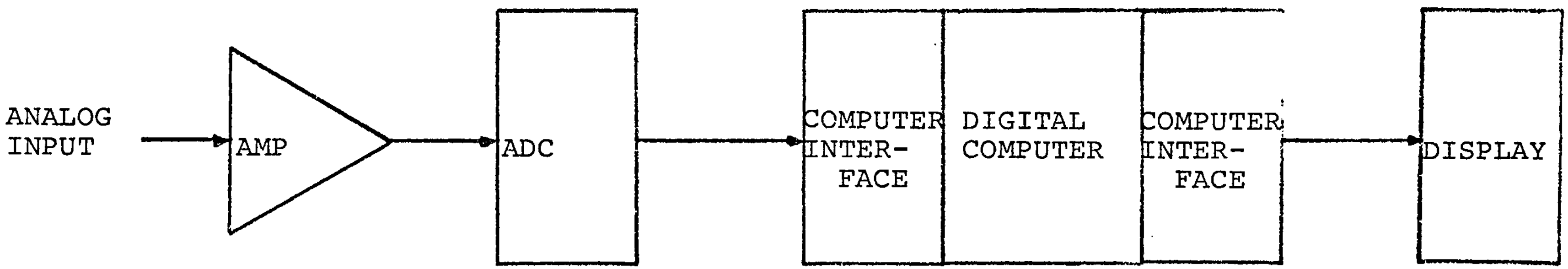


Figure 2.11

Computer program linearization

In addition to the conversion time of the ADC used in the method of Figure 2.11, the device access time and program execution time would be prime factors in the overall speed of the system. In general a table look-up technique would be faster than a polynomial equation evaluation technique but would also occupy more memory space.

The straight forward processing technique of the computer program linearizer presents a very simple design concept but the actual implementation of the system may be very complicated depending on the structure and purpose of the overall system.

Of all the commonly used linearization techniques presented here, none possesses extreme advantages over any other. Perhaps the most commonly used techniques in the past have been manual table-look-up and DFG feedback circuits. The ramp ladder and variable clock techniques have been used when simpler circuitry and less calibration were desired. The computer program technique has been widely used in systems in which computers are employed for other uses. The digital techniques have become increasingly popular in the last two years because continuing advances in digital technology make the circuitry required to support such techniques simpler and more economical to use.

Indications are that the application requirements will continue to dictate the technique used but continuing advances in digital technology should make digital techniques more and more attractive in the future.

Recent innovations in economical modular analog circuitry components such as power supplies, multiplexers, sample and hold devices, analog-to-digital converters, digital-to-analog converters, and amplifiers make it increasingly more attractive to search for linearization techniques which are as independent of the other circuit modules as possible.

CHAPTER III

READ-ONLY MEMORY CHARACTERISTICS AND APPLICATIONS

A read-only memory (ROM) is a circuit which can accept a digital code at its input terminals and provide a unique digital code on its output terminals. The relationship between the input and output codes is fixed and is usually alterable only by replacing all or part of the circuitry.

Many advances in read-only memory technology have been made since the mid-1960's and although ROMs have received much publicity recently, they certainly are not new devices. Read-only memories have been used in digital circuitry for many years but their usage in the past has been rather limited.

The most common form of read-only memory is the diode matrix which has been used since the early days of electronic digital computers. A typical example of such a diode matrix is shown in Figure 3.1. The matrix shown has 4 input bits and 5 output bits which makes the circuit a 20-bit ROM, meaning that the circuit has 20 memory calls which may be programmed with a logic "0" (no diode) or a logic "1" (diode present).

The most common use of such diode matrix circuits has been the implementation of Boolean logic equations. The truth table shown in Figure 3.2 illustrates how the matrix

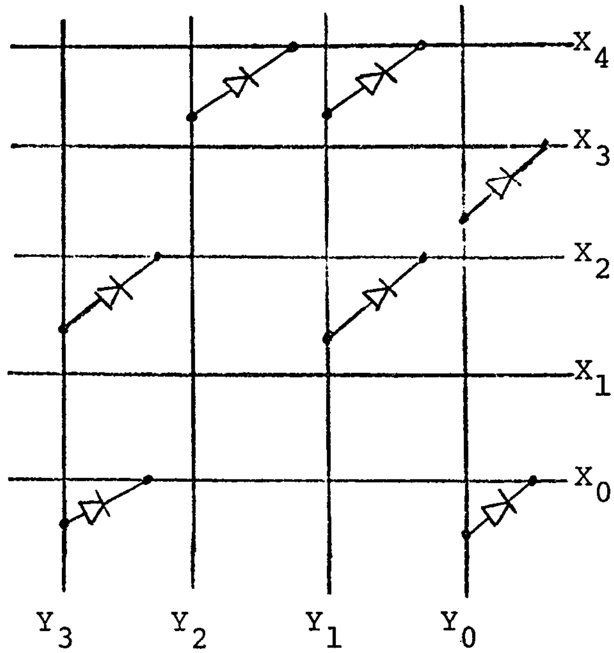


Figure 3.1

Diode matrix read-only memory

INPUT BIT PATTERN				OUTPUT BIT PATTERN				
Y_3	Y_2	Y_1	Y_0	X_4	X_3	X_2	X_1	X_0
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	1
0	0	1	0	1	0	1	0	0
0	0	1	1	1	1	1	0	1
0	1	0	0	1	0	0	0	0
0	1	0	1	1	1	0	0	1
0	1	1	0	1	0	1	0	0
0	1	1	1	1	1	1	0	1
1	0	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0	1
1	0	1	0	1	0	1	0	1
1	0	1	1	1	1	1	0	1
1	1	0	0	1	0	1	0	1
1	1	0	1	1	1	1	0	1
1	1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	0	1

Figure 3.2

A read-only memory truth table

of Figure 3.1 solves the following set of logic equations:

$$X_0 = Y_0 + Y_3$$
$$X_1 = 0$$
$$X_2 = Y_1 + Y_3$$
$$X_3 = Y_0$$
$$X_4 = Y_1 + Y_2$$

Such diode matrix circuits have also been popularly used in code-conversion circuits such as BCD-to-decimal, binary-to-BCD, and BCD-to-binary. Each of these examples and many others qualify as read-only memory circuits since a unique digital code can be applied to the input terminals and a unique digital code obtained on the output terminals.

Typical applications for devices which can generate a group of unique output bits for every group of unique input bits include implementing logic and counting functions, performing code conversion, function generating, character generating, micro-programming of digital processes, and performing table look-up functions.

Many devices which qualify as read-only memories have been developed to perform these tasks, although most of them, such as the diode matrix, are not commonly referred to as ROMs. A very popular technique is to store the bit pattern of a particular function in a common read-write memory (core memory, semiconductor memory, drum storage, etc) then activate a write-disable circuit to prevent inadvertent destruction of the stored information but still permit access to the stored information. Such a device would properly be classified as a ROM while the write-disable circuit remained active but would still provide the option of changing the contents of the memory at a later time.

Even though many devices qualify as read-only memories, the term "read-only memory" or ROM, is in general used to refer to semiconductor integrated circuit devices satisfying the definition of a read-only memory. Semiconductor memories incorporate high-gain and high-speed digital elements. Consequently they are a faster and more ideal storage device than their magnetic counterparts which depend on the analog properties of a slower device and exhibit no gain.

Memories built with semiconductor devices are becoming cost-competitive with ferrite-core memories. In addition they offer superior performance and greater freedom of organization. Many types of semiconductor memories, based on both metal-oxide silicon (MOS) and bipolar technologies, are becoming available. These memories differ primarily in cost, organization, and performance.

MOS technology is presently the most popular for the fabrication of read-only memories. The high density possible in this memory matrix cannot be achieved by any other present technology. In addition the circuitry required for address decoding and input-output buffering can be included on the same substrate as that which contains the memory matrix. The result is a very dense, low-cost array that can be used easily.

A general diagram of a MOS read-only memory is shown in Figure 3.3. The ROM is organized as a 16-word 2-bits-per-word memory. The basic storage element of the MOS ROM is the MOS transistor. The ROM shown in Figure 3.3 has 32 possible transistor locations or memory "cells". A logic "1" is coded by inserting a transistor in a particular position and a logic "0" by omitting the transistor. The truth table for this ROM is illustrated in Figure 3.4.

Semiconductor memories (except read-only memories) are, in general, volatile storage devices. For a volatile storage device to retain stored data, power must be applied continuously. Read-only memories, while difficult to alter, eliminate this storage volatility problem. Most ROMs are fabricated by using a photographic mask with a customized memory matrix on it which stores the desired bit patterns. ROMs manufactured in this manner must be replaced if the bit pattern is to be changed later. Some ROMs are termed "programmable" read-only memories (PROMs) meaning that they can be programmed after the integrated circuit wafer has been packaged. Most PROMs are manufactured with logic "1"s in all bit positions and the programming consists of removing certain "1"s by applying a large reverse-biased voltage

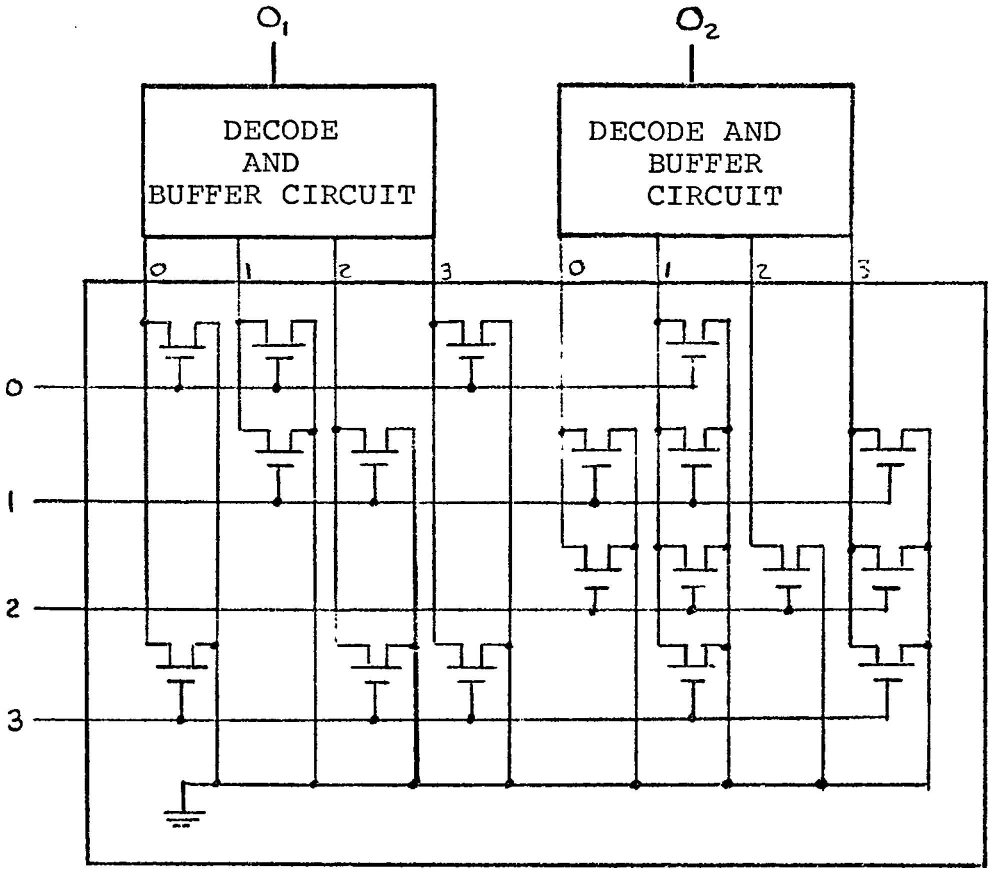


Figure 3.3

MOS read-only memory (16 word x 2 bit)

ROM WORD NO.	0_1	0_2
1	1	0
2	1	1
3	0	0
4	1	0
5	0	1
6	1	1
7	1	0
8	0	1
9	0	1
10	0	1
11	0	1
12	0	1
13	1	0
14	0	1
15	1	0
16	1	1

Figure 3.4

MOS read-only memory truth table

to certain terminals in order to destroy the corresponding transistor. Some PROMs are even erasable (E-PROMs) by x-rays or ultra-violet rays. Applying such energy sources to an E-PROM returns all bit positions to logic "1" permitting the device to be re-programmed.

Some factors significantly affecting the cost of ROM devices are the type, size, method of programming, and quantity purchased. All of these factors must be considered when purchasing ROMs because an absolute comparison between devices cannot be made unless these factors are defined. MOS memories are, in general, less expensive than bipolar memories and can be obtained with more memory capacity in the same physical area. Memory costs are often compared on a price-per-bit basis. Read-only memories in the 1024 to 4096 bit range may be priced as low as 1¢ per bit in quantities of 1000 devices but may cost 5¢ per bit in unit quantities. Also a ROM in the 64 to 256 bit range may cost more per bit than a 1024 to 4096 bit ROM, because the actual device costs may be approximately equal. Another factor greatly affecting the cost of ROMs is the programming of the ROM. ROMs which are produced by photomasks are in general less expensive than equivalent ROMs which are PROMs or E-PROMs. If a custom mask must be made for a particular application, about 50 to 100 such ROMs would have to be produced from the mask to bring the total overall cost below that of a PROM or E-PROM. In general, for very small quantities of custom memory patterns the PROM or E-PROM is the most economical. It is necessary to evaluate the actual costs and the projected requirements of a particular memory configuration to determine the most economical type of ROM to use in a particular application.

CHAPTER IV

READ-ONLY MEMORY LINEARIZATION APPLICATION

The basic characteristic of read-only memories (ROMs) is that a unique digital code on the input terminals results in a unique digital code on the output terminals. This makes them extremely useful for performing code conversion and table look-up functions. One of the many applications of such functions is the conversion of digitized values of non-linear analog functions to suitably coded engineering units.

By effectively programming a NBS thermocouple table into a ROM, a temperature can be determined by measuring the emf produced by a thermocouple. Referring to an NBS "manually" (as discussed in Chapter 2) can be performed automatically. A simplified block diagram of such an automatic device is shown in Figure 4.1. If in such a device as shown in Figure 4.1, the amplifier and ADC are used to generate a binary code proportional to the input analog value, the ROM can be used to generate a corresponding human-oriented engineering unit value for each discrete ADC value. The display is used to present the ROM output in alpha-numerical form to the person desiring to measure the physical quantity.

A technique such as that shown in Figure 4.1 offers many of the qualities desirable in linearization techniques. The technique is very simple in concept and is fairly simple to implement once it is designed. There are several important

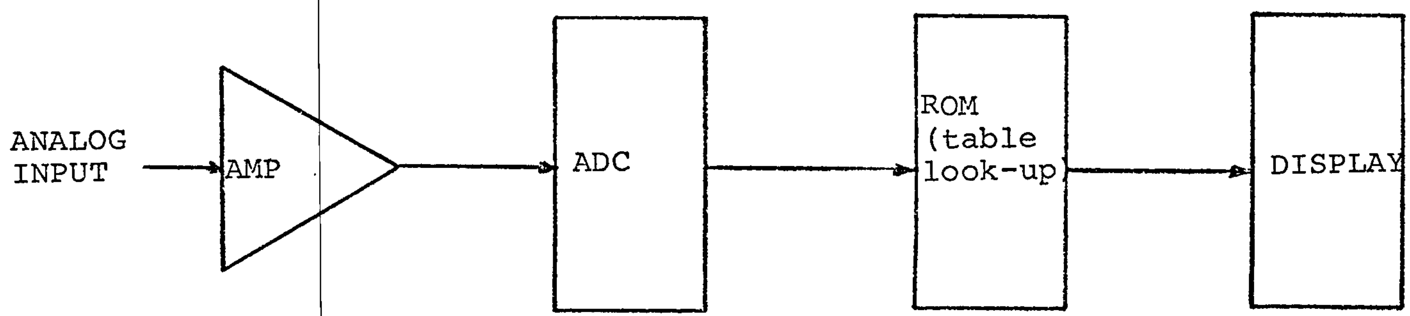


Figure 4.1

ROM linearizer block diagram

considerations in the design of the system and the programming of the ROM; however, these considerations are not extremely difficult and not altogether different from those required in the design of systems using other linearization techniques. The considerations peculiar to the ROM linearizer are discussed in detail in Chapter 5.

The system is very versatile since the type of amplifier and ADC used may be almost any type as required by the particular application the system is to be used in. The ROM may require different programming for different types of ADCs but the technique in general is virtually ADC-independent. A variety of display devices may also be used as the output code of the ROM may be programmed in many ways although some methods require more bits per word, hence a "larger" ROM which may result in a more costly ROM.

Another advantage of the ROM linearizer is the simplicity of maintenance and calibration. An important characteristic of this system is the absence of feedback loops in the overall block diagram. This eliminates most of the calibration and maintenance problems associated with many of the techniques discussed in Chapter 2. Another important characteristic of the system is that once a ROM has been programmed (truth table generated) many identical ROMs can be manufactured with very little testing time needed. Once an overall system is designed, many can be built requiring only minor calibration of the amplifier and ADC.

The ROM linearizer is probably the fastest linearizing technique available. Since there are ROMs available with access speeds of one microsecond or less, the response time of the overall system is essentially dependent only on the amplifier, ADC, and display response times.

The use of a ROM linearizer is an extremely reliable technique. One inherent characteristic of read-only memories is that they never forget and have a normal life expectancy of many years. Extremely reliable amplifier, ADC, and display devices are also available so it is possible to construct highly reliable ROM linearizer systems.

The accuracy of a ROM linearizer is dependent on the inherent accuracies of the amplifier and the ADC and the ROM itself. The programming of the ROM is the most important consideration in the overall accuracy of the system. If a ROM is chosen which has the same number of "words" as the ADC has discrete values and each "word" has an adequate number of bits to code the engineering unit value to be represented, then the overall accuracy depends on the number of discrete ADC values used to represent the analog input and the method used to generate the ROM truth table. The considerations involved in determining the accuracy of a ROM linearizer are discussed in detail in Chapter 5.

The economics of the ROM linearizer may range from a very economical system to a very expensive system depending on a number of factors. The two most important considerations in determining the cost of a ROM are the programming technique used and the size of the ROM. The size of the ROM is usually determined by the range and accuracy desired and the method of coding the output data.

The two most popular methods of coding the output data would probably be binary and BCD. Binary is the most commonly used code in digital applications so the ROM addresses (word numbers) are usually coded in binary. BCD is a very popular code for display purposes since each BCD quad may be decoded to drive a numeric display using the familiar decimal numbers. There are a number of BCD-to-7 segment and BCD-to-decimal decoder/driver/display devices that are very economical, reliable, and simple to use, and which can be adapted to a wide variety of applications.

The ROM output data can be coded in binary or BCD format with essentially the same degree of effort. Coding the data in BCD eliminates the necessity of converting the binary data to BCD for display purposes but also requires more bits per word for any numbers greater than 9. For example a 2048 ROM arranged 256 words by 8 bits per word, could be used to generate output data from 0 to 99 if coded in BCD but could generate output data from 0 to 255 if coded in binary.

If it is necessary to generate output displays from 0 to 99, 8 bits would be required for BCD coding but only 7 for binary coding. If the ROM has to have 256 words, then a 2048-bit ROM would be required for BCD output code and a 1792-bit ROM would be required for binary output. If the ROM used costs 5 cents per bit then the cost difference is $12.80. If it is necessary to convert the binary data to BCD, the cost of the ROM should include the cost of performing the conversion externally if the binary coded ROM is used.

Another consideration in performing the binary-to-BCD conversion externally to the ROM is the fact that doing so introduces more components and more complexity into the system. However, as mentioned previously, ROMs are ideally suited for code conversion applications so the binary-to-BCD conversion could be another highly-reliable easily-designed ROM. There are also some very economical 4-bit binary-to-BCD and BCD-to-binary converter integrated circuits available which can be configured in arrays to perform n-bit binary-to-BCD and BCD-to-binary conversion.

Figure 4.2 shows a typical application where it might be desirable to use a binary-to-BCD conversion external to the ROM performing the linearization. The system shown has four analog input types each with a different response curve therefore requiring four ROMs for linearizing. If the ROMs

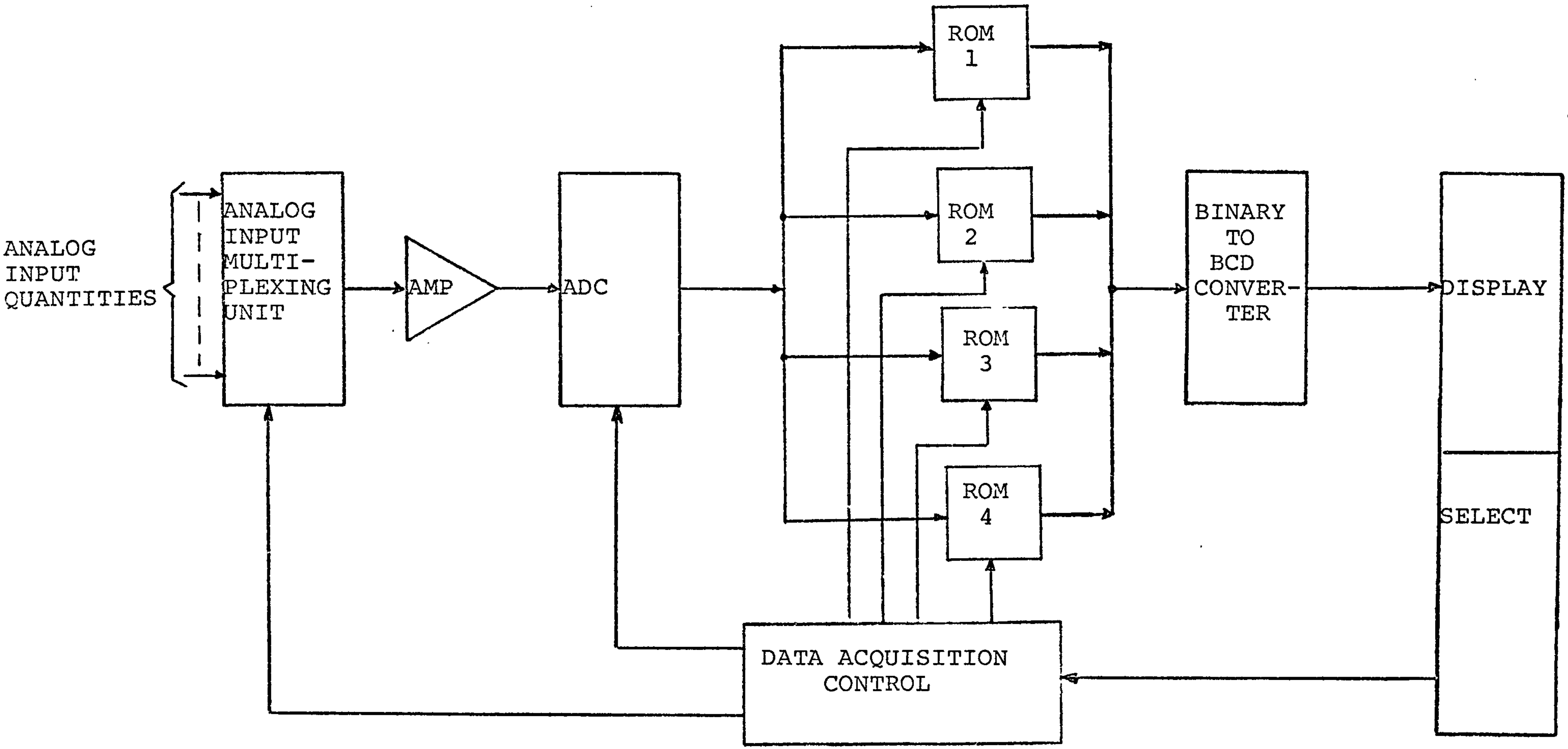


Figure 4.2

ROM linearizing data acquisiton system

used permit having their outputs tied together, they may all be used to drive a common binary-to-BCD converter and the number of bits per word saved in each linearizing ROM may be more than enough to justify the external binary-to-BCD conversion.

CHAPTER V

GENERATING READ-ONLY MEMORY TRUTH TABLES

The procedure of using read-only memories (ROMs) to convert digitized values of nonlinear analog functions to corresponding engineering units is very similar to the procedure described in Chapter II using a computer program to perform the linearization. The ROM actually performs as a special-purpose computer program for the purpose of linearizing a particular function.

There are two very popular techniques used in the computer program linearization method to determine the physical value to attach to the measured analog quantity. The first procedure involves comparison of voltages by tabular look-up in an appropriate table (such as an NBS thermocouple table)which has been stored in the computer memory. In the second procedure, the table or response curve may be approxmiated by polynomial equations either by a single power series or a set of second order equations.

The single equation power series technique uses an equation of the form:

$$T=C_0+C_1E+C_2E^2+C_3E^4+C_4E^5+C_6E^6 \qquad (5.1)$$

The string of second order equations uses equations of the form:

$$T=C_0+C_1E+C_2E^2, \text{ each equation having a} \qquad (5.2)$$

a specified range of applicability.

For each measuring device as thermocouples, the accepted standard is the set of tables published in NBS circular 561. Accuracies of linearization techniques for thermocouples are usually measured in terms of degrees of variation from these tables. For thermocouple curve approximation by equations (5.1) and (5.2) the values of the coefficients are determined by curve-fit techniques. Equation (5.1) is usually used when it is desired to use one equation over a broad range of the thermocouple curve. Equation (5.2) is usually used over narrow ranges of thermocouple curves either when the values of interest are confined to a narrow range or when several second order equations are more desirable to evaluate than one high ordered equation. Coefficients for equations (5.1) and (5.2) can be calculated to conform to NBS table values within 0.5°F for most common thermocouple types.

For thermocouple curves the most accurate technique would be to store the NBS look-up tables in a computer program. However, due to the amount of memory storage and time required to store the tables in memory this may not be a very economical approach. Using equations like (5.1) or (5.2) to calculate temperatures from measured emf is a more general approach to nonlinear function measurement. Many nonlinear functions can be easily expressed easily by polynomial equations as in equations (5.1) or (5.2). By merely changing a few coefficients the same program can be used to evaluate a different nonlinear function. If the table look-up technique is used the entire table would have to be recalculated if a new function is to be evaluated.

Even though the ROM linearizer employs a table look-up technique to perform the linearization, the same arguments apply to calculating the ROM truth table. The truth table for a thermocouple curve can be calculated by hand for a read-only memory but the same procedure would have to be used for every truth table to be generated.

The Fortran program in Appendix A was developed to generate truth tables for ROMs by evaluating a series of equations like equation (5.2). Data required by the program includes the coefficients C_0, C_1, and C_2 of equation (5.2), the range for which the equation is valid, and the number of equations used to approximate the curve.

The values for the coefficients and the best ranges for the equations used to describe the function may be determined by any of several commonly used curve-fit techniques. Tables 5.1 and 5.2 give the results obtained by using the simple trial-and-error technique of entering breakpoints into a computer program, permitting the program to generate a second-order equation, then adjusting breakpoints and recalculating until an acceptable small maximum error was reached. The best technique to use may depend on the type

COEFFICIENTS			RANGE OF EQUATION (°F)	MAX. DEV. FROM NBS TABLES (°F)
C_0	C_1	C_2		
31.94469	35.63730	-0.28722	0-200	0.308
33.91171	34.33704	-0.10397	200-400	0.301
43.19788	32.26427	0.00762	400-700	0.289
18.28229	34.47177	-0.04064	700-960	0.291
-46.70140	39.30495	-0.13029	960-1220	0.335
-40.58614	39.18289	-0.13151	1220-1400	0.297
308.71375	22.99346	0.05607	1400-1600	0.395

Table 5.1

Type J (iron/constantan) thermocouple calibration curve data

COEFFICIENTS			RANGE OF EQUATION (°F)	MAX. DEV. FROM NBS TABLES (°F)
C_0	C_1	C_2		
31.68816	46.15694	-0.65195	0-140	0.356
36.65155	42.12250	0.16779	140-280	0.292
18.50809	47.10442	-0.14561	280-700	0.313
45.33263	43.60689	-0.03162	700-1080	0.371
114.41992	37.89775	0.08633	1080-1380	0.347
169.81726	34.45381	0.13980	1380-1800	0.370
272.45447	29.44804	-0.20083	1800-2240	0.432
447.60522	21.93095	0.28114	2240-2480	0.349

Table 5.2

Type K (chromel/alumel) thermocouple calibration curve data of curve to be linearized. Some functions which respond as, for instance, the square root of a stimulus may have a very simple characteristic equation which can be described without requiring use of a curve-fit technique.

The data generated by the ROM Truth Table Generator of Appendix I is a ROM truth table with the ROM address given in decimal and binary format and the data for each ROM word given in decimal and BCD. The decimal representation is provided for easy reference and the binary and BCD codes are provided as the actual bit pattern that would be programmed into the ROM. BCD is chosen for this discussion since it readily lends itself to numeric display in decimal form. The truth table could be easily changed from BCD to binary values by changing just one statement in the main program from a call to a BCD conversion routine to a call to a binary conversion routine.

Besides the calibration curve data, other input data to the ROM Truth Table Generator is the number of ROM words and the ADC resolution (counts per analog input unit). The number of ROM words (NWDS) is used by the program to determine the stopping point of the program. The ADC resolution (NCTS) is used as a scaling factor for the set of equations used to approximate the response curve. For example, the coefficients for Types J and K thermocouples given in Tables 5.1 and 5.2 are based on the units of E for equation (5.2) being in millivolts. The value of NCTS used in the program would represent the number of ADC counts per millivolt of analog input.

The value chosen for NCTS is the key to the accuracy of the ROM table and the eventual size of ROM required for the desired range of the physical stimulus to be measured. The derivation of the choice for NCTS is given in Appendix B along with some examples of how the choice of NCTS affects the inherent system accuracy.

As an analysis example, assume that a Type K thermocouple is to be used in the range of 32°F to 150°F and that it is desired to measure temperatures to the nearest degree in this range. The calibration curve data given in Table 5.2 could be used to calculate the ROM truth table. A Type K thermocouple circuit referenced to 32°F will generate a nonlinear emf from 0 millivolts to 2.66 millivolts for this range.

If the response curve of a Type K thermocouple were linear, only 119 discrete ADC values would be required to display each temperature value between 32°F and 150°F with the amplifier and ADC of Figure 4.1 scaled so 0 ADC counts would represent 32°F and 118 ADC counts would represent 150°F.

However, the thermocouple curve is nonlinear for this range so a ROM is chosen to adjust the ADC values from 0 to 118 to "linearize" the response curve. The derivation in

Appendix B reveals that the best choice of the approximate slope S used to determine the value of NCTS for the program of Appendix A is one which approximates the curve as closely as possible with all points of the curve falling below the approximation slope.

Figure 5.1 shows how such a curve might be selected for a Type K thermocouple. The slope of the curve shown in Figure 5.1 would result in an NCTS value of 45 ADC counts per millivolt. The results of using this value for a 512 word ROM are given in Appendix C.

Tables 5.3 through 5.7 show the ROM truth tables for Type K thermocouples that would be obtained for 36-word ROMs using values of 20, 40, 45, 50, and 60 for NCTS. The Type K thermocouple response curve is nonlinear between 32°F and 68°F but the nonlinearity is very slight and the results obtained by using a value of 45 for NCTS shows that the temperature over this small range could be measured within one degree of accuracy without any linearization techniques at all. However, most practical application require temperature measurement over a much broader range than 46°F. Sometimes accuracies greater than one degree may be required and many transducer response curves deviate more from a straight line approximation than thermocouples do. Tables 5.3 through 5.7 are presented for purposes of illustrating the importance of a good choice for the value of NCTS.

Using a value too low such as 20 ADC counts per millivolt results in a number of "gaps" in the truth table output data. For example an emf of 0.37 millivolts which should correspond exactly to 49°F would be displayed as 48°F. It is readily obvious that many temperatures could not be displayed since they are not programmed into the ROM.

Using a value too high such as 60 ADC counts per millivolt results in a number of "repeated" values in the truth table output data. While this increases the accuracy of the ROM rather than decreases it, using excessively high values for NCTS may increase the size of the ROM unnecessarily and may increase the cost of the ROM linearizer significantly.

As a comparison, a value of 20 for NCTS permits a temperature range of 32°F to 113°F to be linearized by a 36-word ROM. A value of 45 for NCTS permits a temperature range of 32°F to 68°F to be linearized by a 36-word ROM. A value of 60 for NCTS permits a temperature range of 32°F to 59°F to be linearized by a 36-word ROM. Thus it is readily obvious that the desired inherent accuracy directly affects the required "size" of the ROM and often the cost of the ROM is heavily dependent on the size of the ROM.

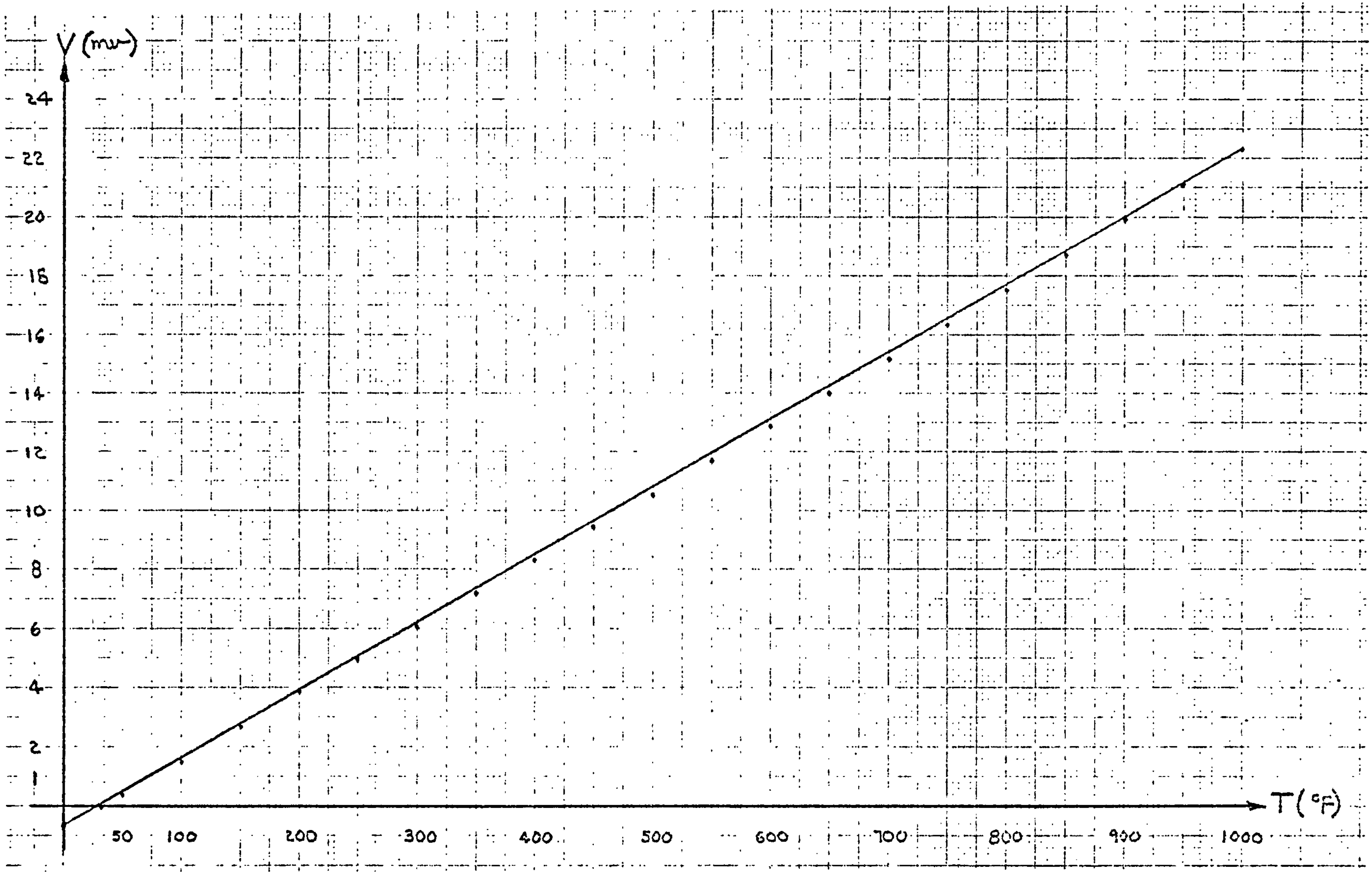


Figure 5.1

Type K (chromel alumel) response curve approximation

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	34	0000000001	0000000000110100
2	36	0000000010	0000000000110110
3	39	0000000011	0000000000111001
4	41	0000000100	0000000001000001
5	43	0000000101	0000000001000011
6	45	0000000110	0000000001000101
7	48	0000000111	0000000001001000
8	50	0000001000	0000000001010000
9	52	0000001001	0000000001010010
10	55	0000001010	0000000001010101
11	57	0000001011	0000000001010111
12	59	0000001100	0000000001011001
13	61	0000001101	0000000001100001
14	64	0000001110	0000000001100100
15	66	0000001111	0000000001100110
16	68	0000010000	0000000001101000
17	70	0000010001	0000000001110000
18	73	0000010010	0000000001110011
19	75	0000010011	0000000001110101
20	77	0000010100	0000000001110111
21	79	0000010101	0000000001111001
22	82	0000010110	0000000010000010
23	84	0000010111	0000000010000100
24	86	0000011000	0000000010000110
25	88	0000011001	0000000010001000
26	91	0000011010	0000000010010001
27	93	0000011011	0000000010010011
28	95	0000011100	0000000010010101
29	97	0000011101	0000000010010111
30	99	0000011110	0000000010011001
31	102	0000011111	0000000100000010
32	104	0000100000	0000000100000100
33	106	0000100001	0000000100000110
34	108	0000100010	0000000100001000
35	110	0000100011	0000000100010000

Table 5.3

NCTS = 20

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	33	0000000001	0000000000110011
2	34	0000000010	0000000000110100
3	35	0000000011	0000000000110101
4	36	0000000100	0000000000110110
5	37	0000000101	0000000000110111
6	39	0000000110	0000000000111001
7	40	0000000111	0000000001000000
8	41	0000001000	0000000001000001
9	42	0000001001	0000000001000010
10	43	0000001010	0000000001000011
11	44	0000001011	0000000001000100
12	45	0000001100	0000000001000101
13	47	0000001101	0000000001000111
14	48	0000001110	0000000001001000
15	49	0000001111	0000000001001001
16	50	0000010000	0000000001010000
17	51	0000010001	0000000001010001
18	52	0000010010	0000000001010010
19	53	0000010011	0000000001010011
20	55	0000010100	0000000001010101
21	56	0000010101	0000000001010110
22	57	0000010110	0000000001010111
23	58	0000010111	0000000001011000
24	59	0000011000	0000000001011001
25	60	0000011001	0000000001100000
26	61	0000011010	0000000001100001
27	63	0000011011	0000000001100011
28	64	0000011100	0000000001100100
29	65	0000011101	0000000001100101
30	66	0000011110	0000000001100110
31	67	0000011111	0000000001100111
32	68	0000100000	0000000001101000
33	69	0000100001	0000000001101001
34	70	0000100010	0000000001110000
35	72	0000100011	0000000001110010

Table 5.4

NCTS = 40

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	33	0000000001	0000000000110011
2	34	0000000010	0000000000110100
3	35	0000000011	0000000000110101
4	36	0000000100	0000000000110110
5	37	0000000101	0000000000110111
6	38	0000000110	0000000000111000
7	39	0000000111	0000000000111001
8	40	0000001000	0000000001000000
9	41	0000001001	0000000001000001
10	42	0000001010	0000000001000010
11	43	0000001011	0000000001000011
12	44	0000001100	0000000001000100
13	45	0000001101	0000000001000101
14	46	0000001110	0000000001000110
15	47	0000001111	0000000001000111
16	48	0000010000	0000000001001000
17	49	0000010001	0000000001001001
18	50	0000010010	0000000001010000
19	51	0000010011	0000000001010001
20	52	0000010100	0000000001010010
21	53	0000010101	0000000001010011
22	54	0000010110	0000000001010100
23	55	0000010111	0000000001010101
24	56	0000011000	0000000001010110
25	57	0000011001	0000000001010111
26	58	0000011010	0000000001011000
27	59	0000011011	0000000001011001
28	60	0000011100	0000000001100000
29	61	0000011101	0000000001100001
30	62	0000011110	0000000001100010
31	63	0000011111	0000000001100011
32	64	0000100000	0000000001100100
33	65	0000100001	0000000001100101
34	66	0000100010	0000000001100110
35	67	0000100011	0000000001100111

Table 5.5

NCTS = 45

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	33	0000000001	0000000000110011
2	34	0000000010	0000000000110100
3	34	0000000011	0000000000110100
4	35	0000000100	0000000000110101
5	36	0000000101	0000000000110110
6	37	0000000110	0000000000110111
7	38	0000000111	0000000000111000
8	39	0000001000	0000000000111001
9	40	0000001001	0000000001000000
10	41	0000001010	0000000001000001
11	42	0000001011	0000000001000010
12	43	0000001100	0000000001000011
13	44	0000001101	0000000001000100
14	45	0000001110	0000000001000101
15	45	0000001111	0000000001000101
16	46	0000010000	0000000001000110
17	47	0000010001	0000000001000111
18	48	0000010010	0000000001001000
19	49	0000010011	0000000001001001
20	50	0000010100	0000000001010000
21	51	0000010101	0000000001010001
22	52	0000010110	0000000001010010
23	53	0000010111	0000000001010011
24	54	0000011000	0000000001010100
25	55	0000011001	0000000001010101
26	56	0000011010	0000000001010110
27	56	0000011011	0000000001010110
28	57	0000011100	0000000001010111
29	58	0000011101	0000000001011000
30	59	0000011110	0000000001011001
31	60	0000011111	0000000001100000
32	61	0000100000	0000000001100001
33	62	0000100001	0000000001100010
34	63	0000100010	0000000001100011
35	64	0000100011	0000000001100100

Table 5.6

NCTS = 50

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	32	0000000001	0000000000110010
2	33	0000000010	0000000000110011
3	34	0000000011	0000000000110100
4	35	0000000100	0000000000110101
5	36	0000000101	0000000000110110
6	36	0000000110	0000000000110110
7	37	0000000111	0000000000110111
8	38	0000001000	0000000000111000
9	39	0000001001	0000000000111001
10	39	0000001010	0000000000111001
11	40	0000001011	0000000001000000
12	41	0000001100	0000000001000001
13	42	0000001101	0000000001000010
14	42	0000001110	0000000001000010
15	43	0000001111	0000000001000011
16	44	0000010000	0000000001000100
17	45	0000010001	0000000001000101
18	45	0000010010	0000000001000101
19	46	0000010011	0000000001000110
20	47	0000010100	0000000001000111
21	48	0000010101	0000000001001000
22	49	0000010110	0000000001001001
23	49	0000010111	0000000001001001
24	50	0000011000	0000000001010000
25	51	0000011001	0000000001010001
26	52	0000011010	0000000001010010
27	52	0000011011	0000000001010010
28	53	0000011100	0000000001010011
29	54	0000011101	0000000001010100
30	55	0000011110	0000000001010101
31	55	0000011111	0000000001010101
32	56	0000100000	0000000001010110
33	57	0000100001	0000000001010111
34	58	0000100010	0000000001011000
35	58	0000100011	0000000001011000

Table 5.7

NCTS = 60

CHAPTER VI

RESULTS AND CONCLUSIONS

Using the formulas derived in Appendix B and the results discussed in Chapter V it can be determined readily that to "linearize" a Type K thermocouple curve from 32°F to 500°F requires a 512 word by 11 bits per word if the result is to be in BCD to the nearest degree. A truth table for such a ROM is given in Appendix C. The required 512 x 11 ROM is a 5632-bit ROM. If the output data of the ROM were in binary and converted to BCD externally to the linearizing ROM, the required ROM would be 512 words by 9 bits per word or a 4608-bit ROM.

It is relatively simple to implement either a 512 x 11 or a 512 x 9 ROM using one or more presently available semiconductor ROM integrated circuits. Some of the more popular types available are 1024-bit ROMs arranged 128 x 8; 2048-bit ROMs arranged 256 x 8 or 512 x 4; 4096-bit ROMs arranged 512 x 8 or 1024 x 4; and 8192-bit ROMs arranged 1024 x 8.

The present cost of such semiconductor ROMs is between 1¢ and 5¢ per bit in quantities of 100 or more while unit quantities may range from 3¢ to 10¢ per bit.

Measurement accuracies within 1°F for Type K thermocouples can be obtained with one straight-line approximation segment between 0°F and 500°F with such conventional techniques as the ramp ladder network. This same technique can attain the same accuracy from 0°F to 2000°F for Type K thermocouples with 6 straight-line approximation segments.

The ROM linearization technique presented here requires 512 words x 11 bits (BCD output) or 512 words x 9 bits (binary output) for the same degree of accuracy between 0°F and 500°F and a 2048 x 14 ROM (BCD output) or 2048 x 11 ROM (binary output) for the same degree of accuracy between 0°F and 2000°F.

In most industrial process applications accuracies within one or two degrees Farenheit of the actual value are acceptable. In such cases it is probably better in most applications to use a conventional linearization technique, such as one of those discussed in Chapter II, rather than the ROM linearization technique presented here. This is particularly true since the largest single-chip ROM presently available is 8192 bits and the cost of ROMs is still high enough that conventional linearization techniques are more economical for wide-range, low degree of accuracy applications.

Semiconductor industry predictions estimate that by 1975 single chip memories of 16,000 bits and greater will be available between 0.1¢ and 1¢ per bit. If these predictions prove to be correct the ROM linearization technique presented here may become economically attractive for some applications between now and 1975.

Presently for most industrial process applications the advantages of the ROM linearization technique are probably outweighed by the cost of implementing such a system. While this may be true in particular for thermocouples and other "almost linear" response curves, there may be some applications that more readily lend themselves to the characteristics of the ROM linearizer.

Several articles have been written about techniques using ROMs to synthesize complex waveforms. Such devices are very similar to implementing the ROM linearizer in reverse by using an up/down counter to sequentially address the inputs of a ROM programmed with a complex function look-up table and using the ROM output to drive a digital-to-analog converter.

The derivation given in Appendix B shows how a ROM can easily be programmed to analyze complex equations. The program developed here for generating ROM truth tables is based on describing the response function by one or more quadratic equations. The program could easily be changed to evaluate almost any type of simple or complex characteristic equation of complex function to generate truth tables.

If it is desired to analyze highly complex functions it may be more desirable to use a ROM linearizer rather than a conventional technique just as it has already been shown that is is sometimes more desirable to use ROMs to synthesize the functions.

BIBLIOGRAPHY

1. Bowers, John O. "Modern Analog-to-Digital Converters For Instrumentation Systems," Electronic Instrument Digest, Vol. 6, pp. 7-12, November, 1970.

2. Conversion Tables for Thermocouples. Pennsylvania: Leeds and Northrup Company, 1964.

3. Graeme, Jerald G., Gene E. Tobey, and Dr. Lawrence P. Huelsman. Operational Amplifiers - Design and Application. New York: McGraw-Hill Book Company, Inc., 1971.

4. Hoeschele, David F., Jr. Analog-to-Digital/Digital-to-Analog Conversion Techniques. New York: John Wiley & Sons, Inc., 1968.

5. Hoff, Marcian E. "Application Considerations for Semiconductor Memories," EEE, Vol. 18, pp. 62-69, June, 1970.

6. Howard, Harry T. "Semiconductor Memories," EDN, Vol. 15, pp. 22-32, February, 1970.

7. Huehne, Karl. "Programmable ROMs Offer Digital Approach to Waveform Synthesis," EDN, Vol. 17, pp. 38-41, August, 1972.

8. Keyser, David R. "Optimized Thermocouple Calibration," Instruments and Control Systems, Vol 44, pp. 81-84, June, 1971.

9. McCoy, Michael R. "MOS Read-Only Memories," EEE, Vol. 18, pp. 46-50, August, 1970.

10. Reiner, Robert. "Digitally Controlled Function Sythesizer," Instruments and Control Systems, Vol. 43, pp. 103-105, August, 1970.

11. Schmoll, Ronald H. "Temperature Measurements with Thermocouples," Instruments and Control Systems, Vol. 44, pp. 87-90, June, 1971.

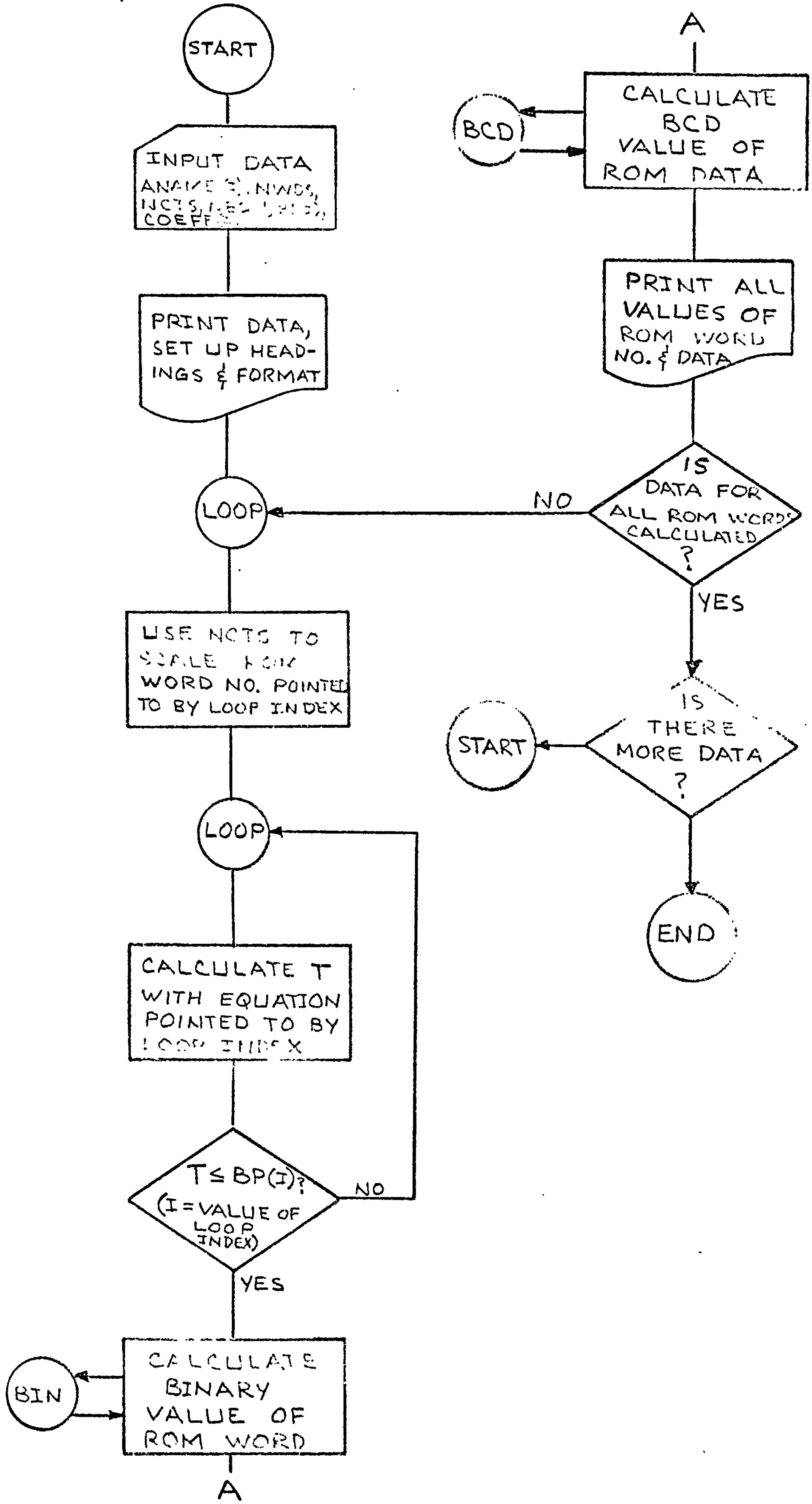
12. Sclater, Neil. "Handbook on A/D-D/A Converters," Electro Procurement, Vol. 14, pp. 17-22, September, 1972.

13. Scott, R. E. Linear Circuits. Massachusetts: Addison-Wesley Publishing Company, Inc., 1964.

14. Sheingold, Daniel H. Analog-Digital Conversion Handbook. Massachusetts: Analog Devices, Inc., 1972.

APPENDIX A

FORTRAN COMPUTER PROGRAM FOR ROM TRUTH TABLE GENERATOR


START
INPUT DATA
NWDS,
COEFF
PRINT DATA, SET UP HEADINGS & FORMAT
LOOP
USE NCTS TO
WORD NO. POINTED TO BY LOOP INDEX
LOOP
CALCULATE T WITH EQUATION POINTED TO BY
T ≤ BP(I)? (I = VALUE OF LOOP INDEX)
NO
YES
CALCULATE BINARY VALUE OF ROM WORD
BIN
A
A
CALCULATE BCD VALUE OF ROM DATA
BCD
PRINT ALL VALUES OF ROM WORD NO. & DATA
IS DATA FOR ALL ROM WORDS CALCULATED ?
NO
YES
IS THERE MORE DATA ?
START
END

```
C
C                          ROM TRUTH TABLE GENERATOR
C
C
C      ANAME DESCRIBES FUNCTION TO LINEARIZE
C      NWDS IS NUMBER OF ROM WORDS
C      NCTS SPECIFIES SCALING OF ADC
C      NEQN IS NUMBER OF EQUATIONS REPRESENTING CURVE
C      COEFF 1RE THE A,B,,C COEFFICIENTS OF THE EQUATIONS
C      T IS THE CALCULATED VALUE OF THE EQUATIONS
C
       DIMENSION NBIN(10),NBCD(16),BP(10),COEFF(30),ANAME(20)
1      READ (3,105) (ANAME(I),I=1,20)
       WRITE (4,105) ANAME
       READ(3,100) NWDS,NCTS,NEQN
       WRITE (4,107) NEQN
       DO 2 I=1,NEQN
       JJ=3*I-2
       JK=3*I
       READ(3,102)(COEFF(J),J=JJ,JK)
       WRITE(4,108) (COEFF(J),J=JJ,JK)
2      CONTINUE
       WRITE (4,109)
       READ(3,101) (BP(I),I=1,10)
       WRITE(4,101) (BP(I),I=1,10)
       WRITE (4,110)NCTS
       WRITE (4,111) NWDS
       CALL FORM
       WRITE (4,105)ANAME
       WRITE (4,103)
       WRITE (4,106)
       DO 5    I=1,NWDS
       N=I-1
       X=FLOAT(N)/FLOAT(NCTS)
       DO 3 J=1,10
       JBP=J*3
       T=COEFF(JBP-2)*X**2+COEFF(JBP-1)*X+COEFF(JBP)
       IF(BP(J)-T)3,4,4
3      CONTINUE
4      CONTINUE
       NT=IFIX(T+.5)
       CALL BIN(N,NBIN)
       CALL BCD(NT,NBCD)
       WRITE(4,104) N,NT,(NBIN(K),K=1,10),(NBCD(K),K=1,16)
5      CONTINUE
       GO TO 1
100    FORMAT (3I5)
101    FORMAT(10F8.1)
102    FORMAT(3F20.10)
103    FORMAT(///20X,38H*****READ ONLY MEMORY TRUTH TABLE*****//)
104    FORMAT (8X,I4,8X,I4,8X,10I1,10X,16I1)
105    FORMAT (20A4)
106    FORMAT(7X,8HROM WORD,4X,8HROM DATA,4X,13HROM WORD(BIN),8X,
     1 13HROM DATA(BCD)/)
```

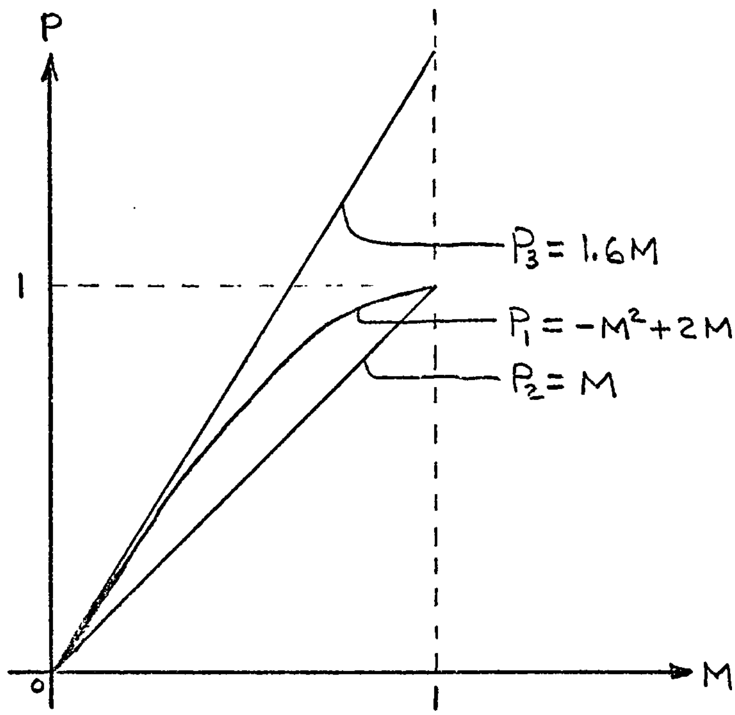
```
107    FORMAT(///3X,I4,1X,9HEQUATIONS//)
108    FORMAT(3X,F14.8,6HX**2 +,F14.8,3HX +,F14.8)
109    FORMAT(///14HBREAKPOINTS AT/)
110    FORMAT(///3X,I4,26H ADC COUNTS PER INPUT UNIT)
111    FORMAT(///5X,13HSIZE OF ROM =,I4,6H WORDS)
       END
       END
0,*SIZE  001323
1,*SIZE  000104
```

```
C
C                INTEGER TO 10-BIT BINARY CONVERTER
C
C
C       NN IS NUMBER TO CONVERT
C       NBIN IS BINARY VALUE OF NUMBER
C
        SUBROUTINE BIN(NN,NBIN)
        DIMENSION NBIN(10)
        KR=NN
        DO 40 I=1,10
        IJ=11-I
        K=KR/2
        KN=K*2
        NBIN(IJ)=KR-KN
        KR=K
40      CONTINUE
        RETURN
        END
        END
0,*SIZE  000076
1,*SIZE  000024
```

```
C
C              INTEGER TO 16-BIT BCD (4 DIGIT) CONVERTER
C
C
C       NTN IS NUMBER TO CONVERT
C       NBCD IS BCD VALUE OF NUMBER IN BIT FORMAT
C
        SUBROUTINE BCD(NTN,NBCD)
        DIMENSION NBCD(16)
        NTTH=NTN/1000
        N1=NTTH
        DO 51 I=1,4
        IJ=5-I
        K=N1/2
        K1=K*2
        NBCD(IJ)=N1-K1
        N1=K
51      CONTINUE
        NTTH1=NTTH*1000
        NTTH2=NTN-NTTH1
        NTH=NTTH2/100
        N1=NTH
        DO 52 I=1,4
        IJ=9-I
        K=N1/2
        K1=K*2
        NBCD(IJ)=N1-K1
        N1=K
52      CONTINUE
        NTH1=NTH*100
        NTH2=NTTH2-NTH1
        NTT=NTH2/10
        N1=NTT
        DO 53 I=1,4
        IJ=13-I
        K=N1/2
        K1=K*2
        NBCD(IJ)=N1-K1
        N1=K
53      CONTINUE
        NTT1=NTT*10
        NTO=NTH2-NTT1
        N1=NTO
        DO 54 I=1,4
        IJ=17-I
        K=N1/2
        K1=K*2
        NBCD(IJ)=N1-K1
        N1=K
54      CONTINUE
        RETURN
        END
        END
0,*SIZE  000344
```

APPENDIX B

DERIVATION OF ADC RESOLUTION AND ROM LINEARIZER ACCURACY



$P = -M^2+2M$

P=range of physical stimulus	(physical units-°F,°C,GPM, etc)
M=range of measured quantity	(measured units-mw,ma, etc)
S=approximate slope of curve	(slope of P_2 or P_3)
R = ADC resolution	(ADC counts per physical unit)
V = value of ADC counts	(ADC counts per measured unit)
T = total ADC counts required	(ADC counts, ROM words)
N = number of ADC bits	(binary coding)

1. Assume that P is the range of the physical stimulus to be measured and some device responds to this stimulus generating a measurable quantity M and that the characteristic equation for this response from 0 to 1 units of P is

$$P = -M^2 + 2M$$

Assume also for analysis purposes that the units of M are in millivolts and the units of P are in gallons per minute (GPM).

The appropriate slope of this curve is given by

$$S = P/M$$

where P is a full scale value of P and M is a full scale value of M. The figure given shows two approximation for S, $S_1 = 1$ and $S_2 = 1.6$. Both approximations are highly inaccurate but one shows all true values of P above the approximation and the other shows all values of P below the approximation. Any value of S between 1 and 1.6 would yield an approximate curve with some real values of P above and some values below the approximation line.

2. Let R represent the resolution of the ADC in counts per physical unit. In this case the resolution would be expressed in counts per GPM. Assume that it is desired to display values of GPM to the nearest tenth between 0 and 1, then the value of R would be 10 counts per GPM.

3. Let V represent the measurable value of each ADC count.

This is given by

$$V = RS$$

which has the units of counts per measured value or in this case counts per millivolt.

Thus it follows that

$$V_1 = S_1R = (1)(10) = 10 \text{ counts per mv}$$
$$V_2 = S_2R = (1.6)(10) = 16 \text{ counts per mv}$$

4. Let T represent the number of ADC counts required to represent the maximum desired value of P. This is given by

$$T = VM$$

or in this case the value of T is

$$T_1 = V_1M = 10 \text{ counts}$$
$$T_2 = V_2M = 16 \text{ counts}$$

This points out the fact that as the slope of the approximation curve gets steeper more ADC counts, and hence, the more ROM words will be required to generate the displayed values. But also the accuracy of the ROM linearizer is more accurate with a 16 count ADC than with 10.

5. Let N = the number of bits representing a full-scale ADC word. This is given by

$$N = \log_2 T$$

For this example

$$N_1 = \log_2 T_1 = \log_2 10 = 3.322$$
$$N_2 = \log_2 T_2 = \log_2 16 = 4.0$$

Hence, to fully represent the desired values of P, using slope S_1 would require a 4-bit ADC and a 10 word ROM and using slope S_2 would require a 4-bit ADC and a 16 word ROM.

6. The results of using a 10-word and a 16-word ROM to "linearize" the curve of this example are given in Tables B.1 and B.2. Note that in Table B.1 the approximation does not permit displays to the nearest tenth of the full scale range but as shown in Table B.2, using a 16-word ROM does permit displays to the nearest tenth.

Figures B.1 and B.2 illustrate the relative accuracies of using 10 and 16-word ROMs for this example. Note that a 10-word ROM results in a maximum error of about 0.12 while a 16-word ROM results in a maximum error of 0.08. Thus the 16-word ROM with a 4-bit ADC satisfies the design criteria for this example.

The accuracy of the ROM linearizer can be improved by using a larger ADC to get more counts per input unit and a larger ROM with more words and perhaps also with another BCD quad per word to display the nearest hundredth of the linearized data.

TYPICAL QUADRATIC RESPONSE CURVE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	0	0000000000	0000000000000000
1	2	0000000001	0000000000000010
2	4	0000000010	0000000000000100
3	5	0000000011	0000000000000101
4	6	0000000100	0000000000000110
5	8	0000000101	0000000000001000
6	8	0000000110	0000000000001000
7	9	0000000111	0000000000001001
8	10	0000001000	0000000000010000
9	10	0000001001	0000000000010000

Table B.1

NCTS = 10

TYPICAL QUADRATIC RESPONSE CURVE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	0	0000000000	0000000000000000
1	1	0000000001	0000000000000001
2	2	0000000010	0000000000000010
3	3	0000000011	0000000000000011
4	4	0000000100	0000000000000100
5	5	0000000101	0000000000000101
6	6	0000000110	0000000000000110
7	7	0000000111	0000000000000111
8	8	0000001000	0000000000001000
9	8	0000001001	0000000000001000
10	9	0000001010	0000000000001001
11	9	0000001011	0000000000001001
12	9	0000001100	0000000000001001
13	10	0000001101	0000000000010000
14	10	0000001110	0000000000010000
15	10	0000001111	0000000000010000

Table B.2

NCTS = 16

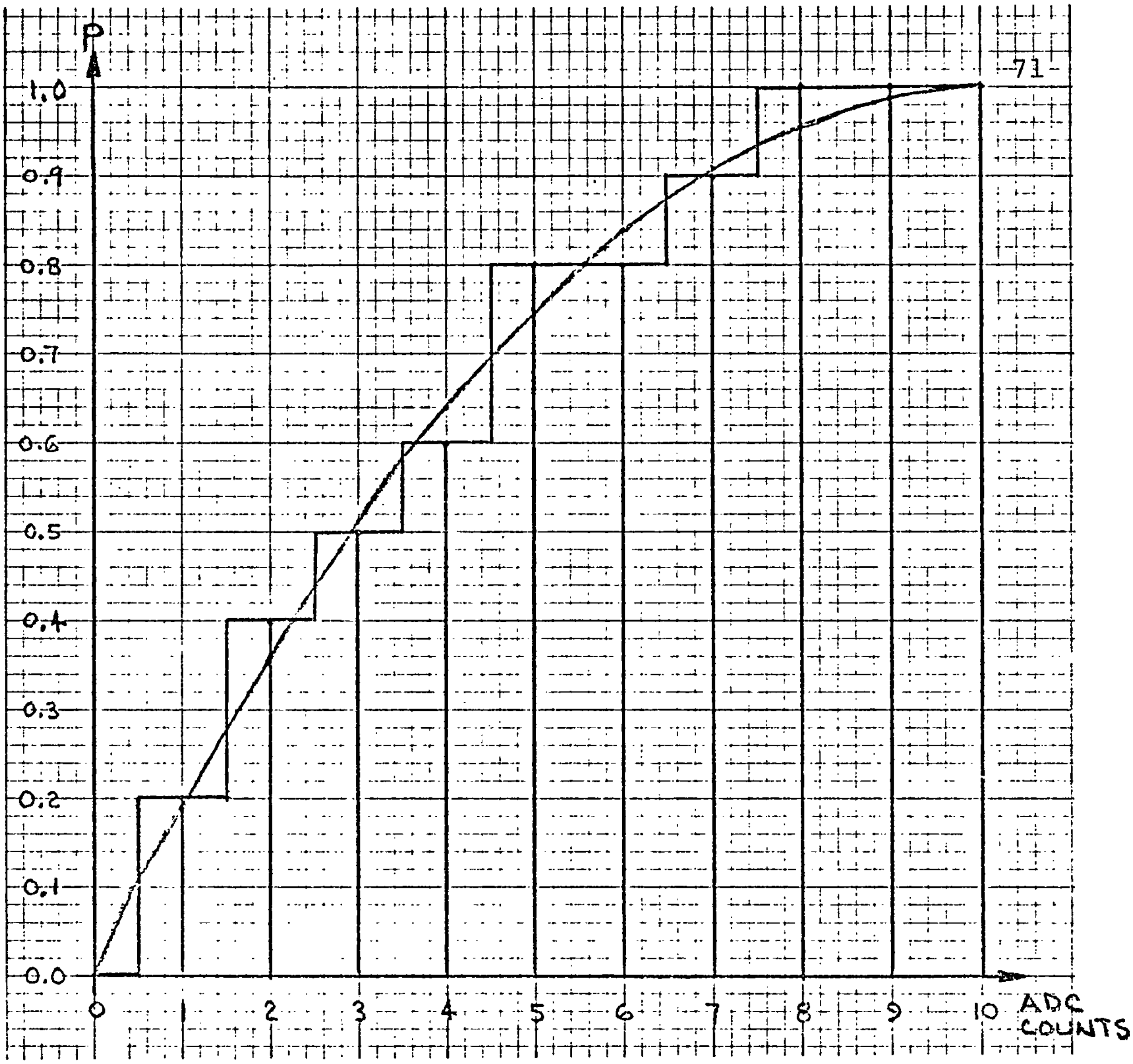


Figure B.1

ROM linearizer results for NCTS = 10

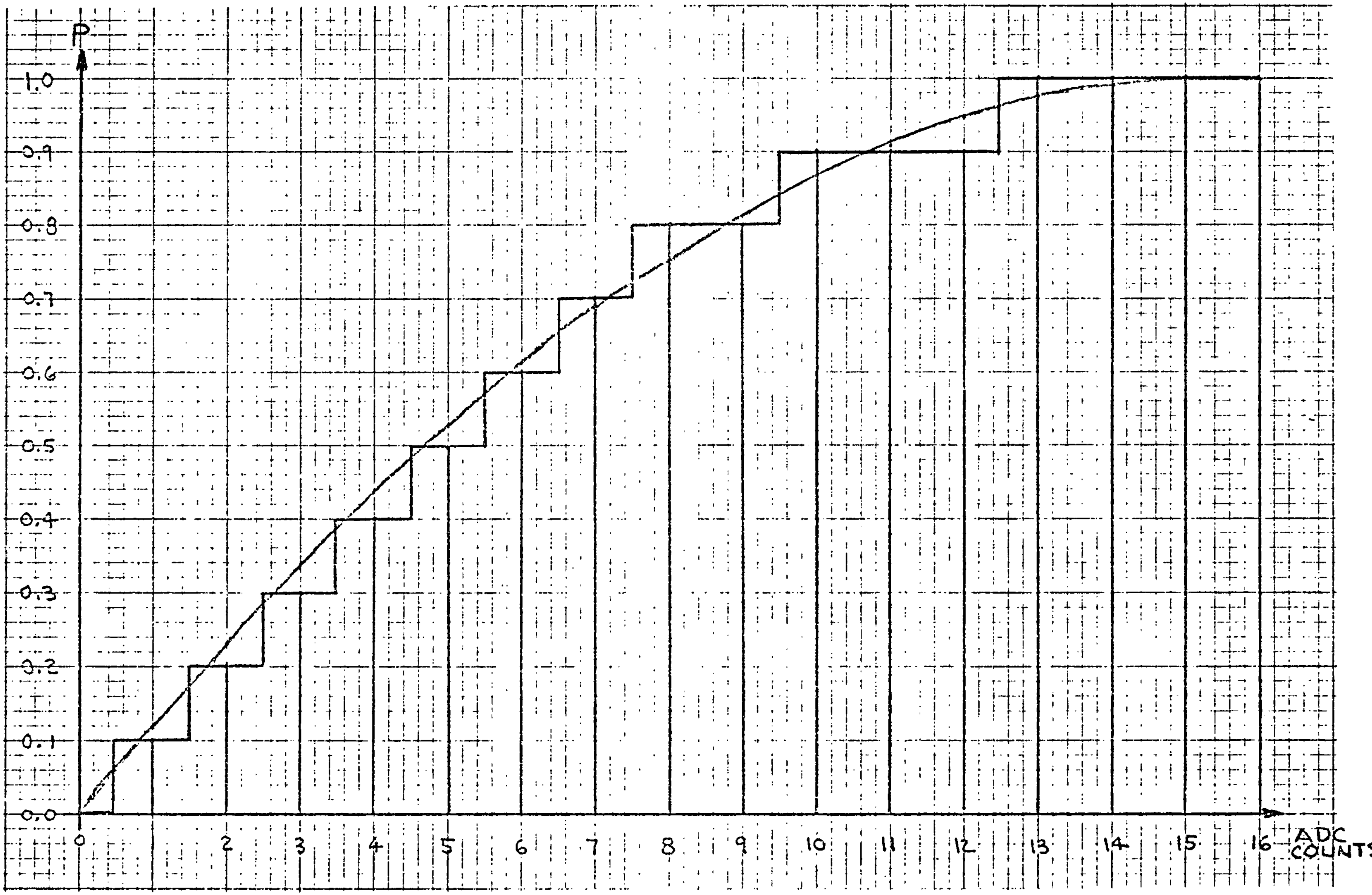


Figure B.2

ROM linearizer results for NCTS = 16

APPENDIX C

TYPE K THERMOCOUPLE LINEARIZATION RESULTS

TYPE K THERMOCOUPLE

8 EQUATIONS

```
-.65194988X**2 +   46.15693665X +    31.68815613
 .16779000X**2 +   42.12249756X +    36.65153503
-.14560997X**2 +   47.10441589X +    18.50808716
-.03162000X**2 +   43.60688782X +    45.33262634
 .08633000X**2 +   37.89773560X +   114.41989136
 .13979995X**2 +   34.45379639X +   169.81719971
 .20082998X**2 +   29.44803619X +   272.45446777
 .28113997X**2 +   21.93094635X +   447.60510254
```

BREAKPOINTS AT

```
140.0   280.0   700.0   1080.0   1380.0   1800.0   2240.0   2480.0      0.0
```

45 ADC COUNTS PER INPUT UNIT

SIZE OF ROM = 512 WORDS

TYPE K THERMOCOUPLE

*****READ ONLY MEMORY TRUTH TABLE*****

ROM WORD	ROM DATA	ROM WORD(BIN)	ROM DATA(BCD)
0	32	0000000000	0000000000110010
1	33	0000000001	0000000000110011
2	34	0000000010	0000000000110100
3	35	0000000011	0000000000110101
4	36	0000000100	0000000000110110
5	37	0000000101	0000000000110111
6	38	0000000110	0000000000111000
7	39	0000000111	0000000000111001
8	40	0000001000	0000000001000000
9	41	0000001001	0000000001000001
10	42	0000001010	0000000001000010
11	43	0000001011	0000000001000011
12	44	0000001100	0000000001000100
13	45	0000001101	0000000001000101
14	46	0000001110	0000000001000110
15	47	0000001111	0000000001000111
16	48	0000010000	0000000001001000
17	49	0000010001	0000000001001001
18	50	0000010010	0000000001010000
19	51	0000010011	0000000001010001
20	52	0000010100	0000000001010010
21	53	0000010101	0000000001010011
22	54	0000010110	0000000001010100
23	55	0000010111	0000000001010101
24	56	0000011000	0000000001010110
25	57	0000011001	0000000001010111
26	58	0000011010	0000000001011000
27	59	0000011011	0000000001011001
28	60	0000011100	0000000001100000
29	61	0000011101	0000000001100001
30	62	0000011110	0000000001100010
31	63	0000011111	0000000001100011
32	64	0000100000	0000000001100100
33	65	0000100001	0000000001100101
34	66	0000100010	0000000001100110
35	67	0000100011	0000000001100111
36	68	0000100100	0000000001101000
37	69	0000100101	0000000001101001
38	70	0000100110	0000000001110000
39	71	0000100111	0000000001110001
40	72	0000101000	0000000001110010

41	73	0000101001	0000000001110011
42	74	0000101010	0000000001110100
43	75	0000101011	0000000001110101
44	76	0000101100	0000000001110110
45	77	0000101101	0000000001110111
46	78	0000101110	0000000001111000
47	79	0000101111	0000000001111001
48	80	0000110000	0000000010000000
49	81	0000110001	0000000010000001
50	82	0000110010	0000000010000010
51	83	0000110011	0000000010000011
52	84	0000110100	0000000010000100
53	85	0000110101	0000000010000101
54	86	0000110110	0000000010000110
55	87	0000110111	0000000010000111
56	88	0000111000	0000000010001000
57	89	0000111001	0000000010001001
58	90	0000111010	0000000010010000
59	91	0000111011	0000000010010001
60	92	0000111100	0000000010010010
61	93	0000111101	0000000010010011
62	94	0000111110	0000000010010100
63	95	0000111111	0000000010010101
64	96	0001000000	0000000010010110
65	97	0001000001	0000000010010111
66	98	0001000010	0000000010011000
67	99	0001000011	0000000010011001
68	100	0001000100	0000000100000000
69	101	0001000101	0000000100000001
70	102	0001000110	0000000100000010
71	103	0001000111	0000000100000011
72	104	0001001000	0000000100000100
73	105	0001001001	0000000100000101
74	106	0001001010	0000000100000110
75	107	0001001011	0000000100000111
76	108	0001001100	0000000100001000
77	109	0001001101	0000000100001001
78	110	0001001110	0000000100010000
79	111	0001001111	0000000100010001
80	112	0001010000	0000000100010010
81	113	0001010001	0000000100010011
82	114	0001010010	0000000100010100
83	115	0001010011	0000000100010101
84	116	0001010100	0000000100010110
85	117	0001010101	0000000100010111

86	118	0001010110	0000000100011000
87	118	0001010111	0000000100011000
88	119	0001011000	0000000100011001
89	120	0001011001	0000000100100000
90	121	0001011010	0000000100100001
91	122	0001011011	0000000100100010
92	123	0001011100	0000000100100011
93	124	0001011101	0000000100100100
94	125	0001011110	0000000100100101
95	126	0001011111	0000000100100110
96	127	0001100000	0000000100100111
97	128	0001100001	0000000100101000
98	129	0001100010	0000000100101001
99	130	0001100011	0000000100110000
100	131	0001100100	0000000100110001
101	132	0001100101	0000000100110010
102	133	0001100110	0000000100110011
103	134	0001100111	0000000100110100
104	135	0001101000	0000000100110101
105	136	0001101001	0000000100110110
106	137	0001101010	0000000100110111
107	138	0001101011	0000000100111000
108	139	0001101100	0000000100111001
109	140	0001101101	0000000101000000
110	141	0001101110	0000000101000001
111	142	0001101111	0000000101000010
112	143	0001110000	0000000101000011
113	143	0001110001	0000000101000011
114	144	0001110010	0000000101000100
115	145	0001110011	0000000101000101
116	146	0001110100	0000000101000110
117	147	0001110101	0000000101000111
118	148	0001110110	0000000101001000
119	149	0001110111	0000000101001001
120	150	0001111000	0000000101010000
121	151	0001111001	0000000101010001
122	152	0001111010	0000000101010010
123	153	0001111011	0000000101010011
124	154	0001111100	0000000101010100
125	155	0001111101	0000000101010101
126	156	0001111110	0000000101010110
127	157	0001111111	0000000101010111
128	158	0010000000	0000000101011000
129	159	0010000001	0000000101011001
130	160	0010000010	0000000101100000

131	161	0010000011	0000000101100001
132	162	0010000100	0000000101100010
133	163	0010000101	0000000101100011
134	164	0010000110	0000000101100100
135	165	0010000111	0000000101100101
136	165	0010001000	0000000101100101
137	166	0010001001	0000000101100110
138	167	0010001010	0000000101100111
139	168	0010001011	0000000101101000
140	169	0010001100	0000000101101001
141	170	0010001101	0000000101110000
142	171	0010001110	0000000101110001
143	172	0010001111	0000000101110010
144	173	0010010000	0000000101110011
145	174	0010010001	0000000101110100
146	175	0010010010	0000000101110101
147	176	0010010011	0000000101110110
148	177	0010010100	0000000101110111
149	178	0010010101	0000000101111000
150	179	0010010110	0000000101111001
151	180	0010010111	0000000110000000
152	181	0010011000	0000000110000001
153	182	0010011001	0000000110000010
154	183	0010011010	0000000110000011
155	184	0010011011	0000000110000100
156	185	0010011100	0000000110000101
157	186	0010011101	0000000110000110
158	187	0010011110	0000000110000111
159	188	0010011111	0000000110001000
160	189	0010100000	0000000110001001
161	190	0010100001	0000000110010000
162	190	0010100010	0000000110010000
163	191	0010100011	0000000110010001
164	192	0010100100	0000000110010010
165	193	0010100101	0000000110010011
166	194	0010100110	0000000110010100
167	195	0010100111	0000000110010101
168	196	0010101000	0000000110010110
169	197	0010101001	0000000110010111
170	198	0010101010	0000000110011000
171	199	0010101011	0000000110011001
172	200	0010101100	0000001000000000
173	201	0010101101	0000001000000001
174	202	0010101110	0000001000000010
175	203	0010101111	0000001000000011

176	204	0010110000	0000001000000100
177	205	0010110001	0000001000000101
178	206	0010110010	0000001000000110
179	207	0010110011	0000001000000111
180	208	0010110100	0000001000001000
181	209	0010110101	0000001000001001
182	210	0010110110	0000001000010000
183	211	0010110111	0000001000010001
184	212	0010111000	0000001000010010
185	213	0010111001	0000001000010011
186	214	0010111010	0000001000010100
187	215	0010111011	0000001000010101
188	216	0010111100	0000001000010110
189	217	0010111101	0000001000010111
190	217	0010111110	0000001000010111
191	218	0010111111	0000001000011000
192	219	0011000000	0000001000011001
193	220	0011000001	0000001000100000
194	221	0011000010	0000001000100001
195	222	0011000011	0000001000100010
196	223	0011000100	0000001000100011
197	224	0011000101	0000001000100100
198	225	0011000110	0000001000100101
199	226	0011000111	0000001000100110
200	227	0011001000	0000001000100111
201	228	0011001001	0000001000101000
202	229	0011001010	0000001000101001
203	230	0011001011	0000001000110000
204	231	0011001100	0000001000110001
205	232	0011001101	0000001000110010
206	233	0011001110	0000001000110011
207	234	0011001111	0000001000110100
208	235	0011010000	0000001000110101
209	236	0011010001	0000001000110110
210	237	0011010010	0000001000110111
211	238	0011010011	0000001000111000
212	239	0011010100	0000001000111001
213	240	0011010101	0000001001000000
214	241	0011010110	0000001001000001
215	242	0011010111	0000001001000010
216	243	0011011000	0000001001000011
217	244	0011011001	0000001001000100
218	245	0011011010	0000001001000101
219	246	0011011011	0000001001000110
220	247	0011011100	0000001001000111

221	248	0011011101	0000001001001000
222	249	0011011110	0000001001001001
223	250	0011011111	0000001001010000
224	250	0011100000	0000001001010000
225	251	0011100001	0000001001010001
226	252	0011100010	0000001001010010
227	253	0011100011	0000001001010011
228	254	0011100100	0000001001010100
229	255	0011100101	0000001001010101
230	256	0011100110	0000001001010110
231	257	0011100111	0000001001010111
232	258	0011101000	0000001001011000
233	259	0011101001	0000001001011001
234	260	0011101010	0000001001100000
235	261	0011101011	0000001001100001
236	262	0011101100	0000001001100010
237	263	0011101101	0000001001100011
238	264	0011101110	0000001001100100
239	265	0011101111	0000001001100101
240	266	0011110000	0000001001100110
241	267	0011110001	0000001001100111
242	268	0011110010	0000001001101000
243	269	0011110011	0000001001101001
244	270	0011110100	0000001001110000
245	271	0011110101	0000001001110001
246	272	0011110110	0000001001110010
247	273	0011110111	0000001001110011
248	274	0011111000	0000001001110100
249	275	0011111001	0000001001110101
250	276	0011111010	0000001001110110
251	277	0011111011	0000001001110111
252	278	0011111100	0000001001111000
253	279	0011111101	0000001001111001
254	280	0011111110	0000001010000000
255	281	0011111111	0000001010000001
256	282	0100000000	0000001010000010
257	283	0100000001	0000001010000011
258	284	0100000010	0000001010000100
259	285	0100000011	0000001010000101
260	286	0100000100	0000001010000110
261	287	0100000101	0000001010000111
262	288	0100000110	0000001010001000
263	289	0100000111	0000001010001001
264	290	0100001000	0000001010010000
265	291	0100001001	0000001010010001

266	292	0100001010	0000001010010010
267	293	0100001011	0000001010010011
268	294	0100001100	0000001010010100
269	295	0100001101	0000001010010101
270	296	0100001110	0000001010010110
271	297	0100001111	0000001010010111
272	298	0100010000	0000001010011000
273	299	0100010001	0000001010011001
274	300	0100010010	0000001100000000
275	301	0100010011	0000001100000001
276	302	0100010100	0000001100000010
277	303	0100010101	0000001100000011
278	304	0100010110	0000001100000100
279	305	0100010111	0000001100000101
280	306	0100011000	0000001100000110
281	307	0100011001	0000001100000111
282	308	0100011010	0000001100001000
283	309	0100011011	0000001100001001
284	310	0100011100	0000001100010000
285	311	0100011101	0000001100010001
286	312	0100011110	0000001100010010
287	313	0100011111	0000001100010011
288	314	0100100000	0000001100010100
289	315	0100100001	0000001100010101
290	316	0100100010	0000001100010110
291	317	0100100011	0000001100010111
292	318	0100100100	0000001100011000
293	319	0100100101	0000001100011001
294	320	0100100110	0000001100100000
295	321	0100100111	0000001100100001
296	322	0100101000	0000001100100010
297	323	0100101001	0000001100100011
298	324	0100101010	0000001100100100
299	325	0100101011	0000001100100101
300	326	0100101100	0000001100100110
301	327	0100101101	0000001100100111
302	328	0100101110	0000001100101000
303	329	0100101111	0000001100101001
304	330	0100110000	0000001100110000
305	331	0100110001	0000001100110001
306	332	0100110010	0000001100110010
307	333	0100110011	0000001100110011
308	334	0100110100	0000001100110100
309	335	0100110101	0000001100110101
310	336	0100110110	0000001100110110

311	337	0100110111	0000001100110111
312	338	0100111000	0000001100111000
313	339	0100111001	0000001100111001
314	340	0100111010	0000001101000000
315	341	0100111011	0000001101000001
316	342	0100111100	0000001101000010
317	343	0100111101	0000001101000011
318	344	0100111110	0000001101000100
319	345	0100111111	0000001101000101
320	346	0101000000	0000001101000110
321	347	0101000001	0000001101000111
322	348	0101000010	0000001101001000
323	349	0101000011	0000001101001001
324	350	0101000100	0000001101010000
325	351	0101000101	0000001101010001
326	352	0101000110	0000001101010010
327	353	0101000111	0000001101010011
328	354	0101001000	0000001101010100
329	355	0101001001	0000001101010101
330	356	0101001010	0000001101010110
331	357	0101001011	0000001101010111
332	358	0101001100	0000001101011000
333	359	0101001101	0000001101011001
334	360	0101001110	0000001101100000
335	361	0101001111	0000001101100001
336	362	0101010000	0000001101100010
337	363	0101010001	0000001101100011
338	364	0101010010	0000001101100100
339	365	0101010011	0000001101100101
340	366	0101010100	0000001101100110
341	367	0101010101	0000001101100111
342	368	0101010110	0000001101101000
343	369	0101010111	0000001101101001
344	370	0101011000	0000001101110000
345	371	0101011001	0000001101110001
346	372	0101011010	0000001101110010
347	373	0101011011	0000001101110011
348	374	0101011100	0000001101110100
349	375	0101011101	0000001101110101
350	376	0101011110	0000001101110110
351	377	0101011111	0000001101110111
352	378	0101100000	0000001101111000
353	379	0101100001	0000001101111001
354	380	0101100010	0000001110000000
355	381	0101100011	0000001110000001

356	382	0101100100	0000001110000010
357	383	0101100101	0000001110000011
358	384	0101100110	0000001110000100
359	385	0101100111	0000001110000101
360	386	0101101000	0000001110000110
361	387	0101101001	0000001110000111
362	388	0101101010	0000001110001000
363	389	0101101011	0000001110001001
364	390	0101101100	0000001110010000
365	391	0101101101	0000001110010001
366	392	0101101110	0000001110010010
367	393	0101101111	0000001110010011
368	394	0101110000	0000001110010100
369	395	0101110001	0000001110010101
370	396	0101110010	0000001110010110
371	397	0101110011	0000001110010111
372	398	0101110100	0000001110011000
373	399	0101110101	0000001110011001
374	400	0101110110	0000010000000000
375	401	0101110111	0000010000000001
376	402	0101111000	0000010000000010
377	403	0101111001	0000010000000011
378	404	0101111010	0000010000000100
379	405	0101111011	0000010000000101
380	406	0101111100	0000010000000110
381	407	0101111101	0000010000000111
382	408	0101111110	0000010000001000
383	409	0101111111	0000010000001001
384	410	0110000000	0000010000010000
385	411	0110000001	0000010000010001
386	412	0110000010	0000010000010010
387	413	0110000011	0000010000010011
388	414	0110000100	0000010000010100
389	415	0110000101	0000010000010101
390	416	0110000110	0000010000010110
391	417	0110000111	0000010000010111
392	418	0110001000	0000010000011000
393	419	0110001001	0000010000011001
394	420	0110001010	0000010000100000
395	421	0110001011	0000010000100001
396	422	0110001100	0000010000100010
397	423	0110001101	0000010000100011
398	424	0110001110	0000010000100100
399	425	0110001111	0000010000100101
400	426	0110010000	0000010000100110

401	427	0110010001	0000010000100111
402	428	0110010010	0000010000101000
403	429	0110010011	0000010000101001
404	430	0110010100	0000010000110000
405	431	0110010101	0000010000110001
406	432	0110010110	0000010000110010
407	433	0110010111	0000010000110011
408	434	0110011000	0000010000110100
409	435	0110011001	0000010000110101
410	436	0110011010	0000010000110110
411	437	0110011011	0000010000110111
412	438	0110011100	0000010000111000
413	439	0110011101	0000010000111001
414	440	0110011110	0000010001000000
415	441	0110011111	0000010001000001
416	442	0110100000	0000010001000010
417	443	0110100001	0000010001000011
418	443	0110100010	0000010001000011
419	444	0110100011	0000010001000100
420	445	0110100100	0000010001000101
421	446	0110100101	0000010001000110
422	447	0110100110	0000010001000111
423	448	0110100111	0000010001001000
424	449	0110101000	0000010001001001
425	450	0110101001	0000010001010000
426	451	0110101010	0000010001010001
427	452	0110101011	0000010001010010
428	453	0110101100	0000010001010011
429	454	0110101101	0000010001010100
430	455	0110101110	0000010001010101
431	456	0110101111	0000010001010110
432	457	0110110000	0000010001010111
433	458	0110110001	0000010001011000
434	459	0110110010	0000010001011001
435	460	0110110011	0000010001100000
436	461	0110110100	0000010001100001
437	462	0110110101	0000010001100010
438	463	0110110110	0000010001100011
439	464	0110110111	0000010001100100
440	465	0110111000	0000010001100101
441	466	0110111001	0000010001100110
442	467	0110111010	0000010001100111
443	468	0110111011	0000010001101000
444	469	0110111100	0000010001101001
445	470	0110111101	0000010001110000
446	471	0110111110	0000010001110001
447	472	0110111111	0000010001110010
448	473	0111000000	0000010001110011
449	474	0111000001	0000010001110100
450	475	0111000010	0000010001110101

451	476	0111000011	0000010001110110
452	477	0111000100	0000010001110111
453	478	0111000101	0000010001111000
454	479	0111000110	0000010001111001
455	480	0111000111	0000010010000000
456	481	0111001000	0000010010000001
457	482	0111001001	0000010010000010
458	483	0111001010	0000010010000011
459	484	0111001011	0000010010000100
460	485	0111001100	0000010010000101
461	486	0111001101	0000010010000110
462	487	0111001110	0000010010000111
463	488	0111001111	0000010010001000
464	489	0111010000	0000010010001001
465	490	0111010001	0000010010010000
466	491	0111010010	0000010010010001
467	492	0111010011	0000010010010010
468	493	0111010100	0000010010010011
469	494	0111010101	0000010010010100
470	495	0111010110	0000010010010101
471	496	0111010111	0000010010010110
472	497	0111011000	0000010010010111
473	498	0111011001	0000010010011000
474	499	0111011010	0000010010011001
475	499	0111011011	0000010010011001
476	500	0111011100	0000010100000000
477	501	0111011101	0000010100000001
478	502	0111011110	0000010100000010
479	503	0111011111	0000010100000011
480	504	0111100000	0000010100000100
481	505	0111100001	0000010100000101
482	506	0111100010	0000010100000110
483	507	0111100011	0000010100000111
484	508	0111100100	0000010100001000
485	509	0111100101	0000010100001001
486	510	0111100110	0000010100010000
487	511	0111100111	0000010100010001
488	512	0111101000	0000010100010010
489	513	0111101001	0000010100010011
490	514	0111101010	0000010100010100
491	515	0111101011	0000010100010101
492	516	0111101100	0000010100010110
493	517	0111101101	0000010100010111
494	518	0111101110	0000010100011000
495	519	0111101111	0000010100011001
496	520	0111110000	0000010100100000
497	521	0111110001	0000010100100001
498	522	0111110010	0000010100100010
499	523	0111110011	0000010100100011
500	524	0111110100	0000010100100100
501	525	0111110101	0000010100100101
502	526	0111110110	0000010100100110
503	527	0111110111	0000010100100111
504	528	0111111000	0000010100101000
505	529	0111111001	0000010100101001
506	530	0111111010	0000010100110000
507	531	0111111011	0000010100110001
508	532	0111111100	0000010100110010
509	533	0111111101	0000010100110011
510	534	0111111110	0000010100110100
511	535	0111111111	0000010100110101