

IMPROVED PHASED ARRAY CONTROLLER  
FOR ULTRASOUND HYPERTHERMIA APPLICATORS

BY

AARON SCOTT FIELD

B.S., University of Illinois, 1984

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 1985

Urbana, Illinois

## DEDICATION

With love and thanks to my mother, whose confidence in me through the years has given me the strength and perseverance that makes every achievement possible.

## ACKNOWLEDGEMENTS

The author gratefully acknowledges the following people for their valuable contributions to this thesis: Dr. Leon Frizzell, whose guidance and encouragement throughout the entire project was the primary motivation for this thesis; Dr. Richard Magin, who encouraged me to pursue graduate research on the hyperthermia project; Dr. Christopher Badger, who devoted a great deal of his time to assisting me on the project and without whom this thesis would not have been possible; graduate students Paul Benkeser, Robert Cargoni, Panos Hadjimitsos, James Novak, Kenneth Ocheltree, Steven Silverman, and Harold Underwood, who shared their knowledge and experience with me on many occasions throughout the project; and finally Mrs. Wanda Elliott and Mr. Robert Cicone, who were always willing to drop whatever they were doing to lend a helping hand.

The author would also like to thank a very dear friend, Ms. Marcia Arnopol, for her love and support throughout this endeavor.

## TABLE OF CONTENTS

	PAGE
CHAPTER 1. INTRODUCTION. . . . .	1
1.1. Hyperthermia and Ultrasonic Phased Arrays . . . . .	2
1.2. Hyperthermia System Overview. . . . .	2
1.3. Introduction to the Phased Array Controller . . . . .	4
CHAPTER 2. STACKED ARRAY CONTROL . . . . .	9
2.1. Theory. . . . .	9
2.2. Design Objectives. . . . .	12
2.3. Implementation . . . . .	15
CHAPTER 3. FREQUENCY SELECTION. . . . .	22
3.1. Introduction . . . . .	22
3.2. Design Objectives. . . . .	22
3.3. Implementation . . . . .	24
CHAPTER 4. CLOSED-LOOP CONTROL FOR PHASED ARRAY HYPERTHERMIA. . . . .	28
4.1. Introduction . . . . .	28
4.2. Organization of the Treatment Field. . . . .	29
4.3. Overview of a Typical Treatment. . . . .	34
CHAPTER 5. HARDWARE . . . . .	37
5.1. Introduction . . . . .	37
5.2. Design and Operation . . . . .	37
CHAPTER 6. SOFTWARE . . . . .	52
6.1. Introduction . . . . .	52

## TABLE OF CONTENTS (CONT.)

	PAGE
6.2. Sharing of External RAM and COMM/SCAN Communications . . . . .	52
6.3. "Double Buffering" of Intensity Data Storage Space . . . . .	55
6.4. Command Overview . . . . .	56
6.5. Communications Software. . . . .	57
6.6. Scan Control Software. . . . .	71
6.6.1. Main Program (MAIN). . . . .	78
6.6.2. Timer Interrupt Routine (TIMER). . . . .	82
6.6.3. Timing Considerations. . . . .	82
6.6.4. EXT0 Interrupt Routine . . . . .	84
6.6.5. EXT1 Interrupt Routine . . . . .	85
CHAPTER 7. SUMMARY AND FUTURE CONSIDERATIONS. . . . .	86
7.1. Summary. . . . .	86
7.2. Future Considerations. . . . .	86
APPENDIX. MODIFICATION OF "LOOK-UP-TABLE GENERATOR" PROGRAM . . . . .	89
LIST OF REFERENCES . . . . .	93

## LIST OF FIGURES

FIGURE	PAGE
1. Hyperthermia System Block Diagram. . . . .	3
2. Tapered Linear Phased Array. . . . .	6
3. Stacked Linear Phased Array . . . . .	10
4. Stack of Ten Linear Arrays—Side View Showing Excitation to Produce Eight Possible Y-Positions. .	11
5. Diagram Showing Control Signals Gating Driving Signals to Stacked Array Elements . . . . .	16
6. Stacked Array Control Hardware Schematic. . . . .	17
7. Clock/Stacked Array Control Board Component Layout. . . . .	19
8. Digitally Controlled Clock Source Schematic . . . . .	25
9. Diagram Showing Sub-fields of the Total Field . . . . .	32
10. Cross-sections of Standard Control Block Showing Focal Volumes . . . . .	33
11. Dual-8031 System Block Diagram. . . . .	38
12. Scan Control CPU Board Schematic. . . . .	40
13. Scan Control CPU Board Component Layout . . . . .	41
14. Communications CPU Board Schematic. . . . .	43
15. Communications CPU Board Component Layout . . . . .	44
16. External RAM Shared Access System Block Diagram . . . . .	45
17. Flow Diagrams Showing How COMM Accesses External RAM. . . . .	54
18. Flow Diagram for COMM—Main Program (Including power-up routine). . . . .	59
19. Flow Diagrams for COMM— TIMER and EXT0 Interrupt Routines . . . . .	60
20. Flow Diagram for COMM—General Flow of Serial Reception Processing . . . . .	61

## LIST OF FIGURES (CONT.)

FIGURE	PAGE
21. Flow Diagram for COMM—SERIAL Interrupt Routine . .	62
22. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	63
23. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	64
24. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	65
25. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	66
26. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	67
27. Flow Diagram for COMM—SERIAL (Cont.) . . . . .	68
28. Flow Diagram for SCAN— General Flow of Scan Algorithm. . . . .	72
29. Flow Diagram for SCAN—Main Program (MAIN). . . . .	73
30. Flow Diagram for SCAN—"Power-Up" Subroutine. . . . .	74
31. Flow Diagram for SCAN—Algorithm for Calculation of New Block Position . . . . .	75
32. Flow Diagram for SCAN—TIMER Interrupt Routine. . . . .	76
33. Flow Diagram for SCAN—EXT0 Interrupt Routine . . . . .	77

## CHAPTER 1. INTRODUCTION

### 1.1 Hyperthermia and Ultrasonic Phased Arrays

Hyperthermia as a form of cancer therapy involves the application of heat to malignant tumors. Elevated temperatures (in the approximate range of 42-45 degrees Centigrade) have been found not only to have a destructive effect on cells but to have an enhancing effect on other anti-cancer agents such as X-rays and certain drugs (Hahn, 1984).

Although hyperthermia has the potential to be an effective tool in the treatment of cancer, the development of clinical hyperthermia systems has been impeded by several problems. The foremost of these is that normal tissue surrounding a tumor must not be heated along with the tumor, as damage to the normal tissue results. This requires a selective, well-controlled method of delivering heat to a tumor, as well as an effective means to continuously monitor the temperature distribution in and around the tumor during treatment.

Several modalities for delivering heat to tumors are being investigated, including microwaves and ultrasound. Of these, ultrasound offers the advantage of being easily focused into a well-defined volume, and at a depth of several centimeters in soft tissue. Ultrasound has traditionally been applied by fixed focus applicator systems, which must be adjusted by hand prior to treatment and mechanically moved to



change the position of the focus (Lele, 1980). Another approach is to use an ultrasonic phased array transducer, which produces a focus whose position can be manipulated electronically by varying the relative phases of the signals driving the array. It has been suggested recently that a stacked or a tapered linear phased array instead of a two-dimensional phased array might be used to decrease the number of elements and associated amplifiers and phasing circuits required (Benkeser et al., 1985).

## 1.2 Hyperthermia System Overview

A clinical hyperthermia system is currently being developed by URI Therm-x, Inc., which will make use of an ultrasonic linear phased array. Figure 1 shows a block diagram of the system consisting of a central control computer (a Digital Equipment Corporation MICRO PDP-11), two Intel 8031 microcontroller-based peripheral subsystems, a series of amplifiers, and the phased array applicator. One of the subsystems is a 16-channel thermometry unit, which makes possible the closed-loop control of a hyperthermic treatment by providing the central computer with real-time temperature information from various locations in and around the tumor. The other subsystem acts as an interface between the central computer and the phased array applicator, and is hereafter referred to as the phased array controller (PAC).

The hyperthermia system's operation is as follows: The central computer calculates an optimal heating pattern for a

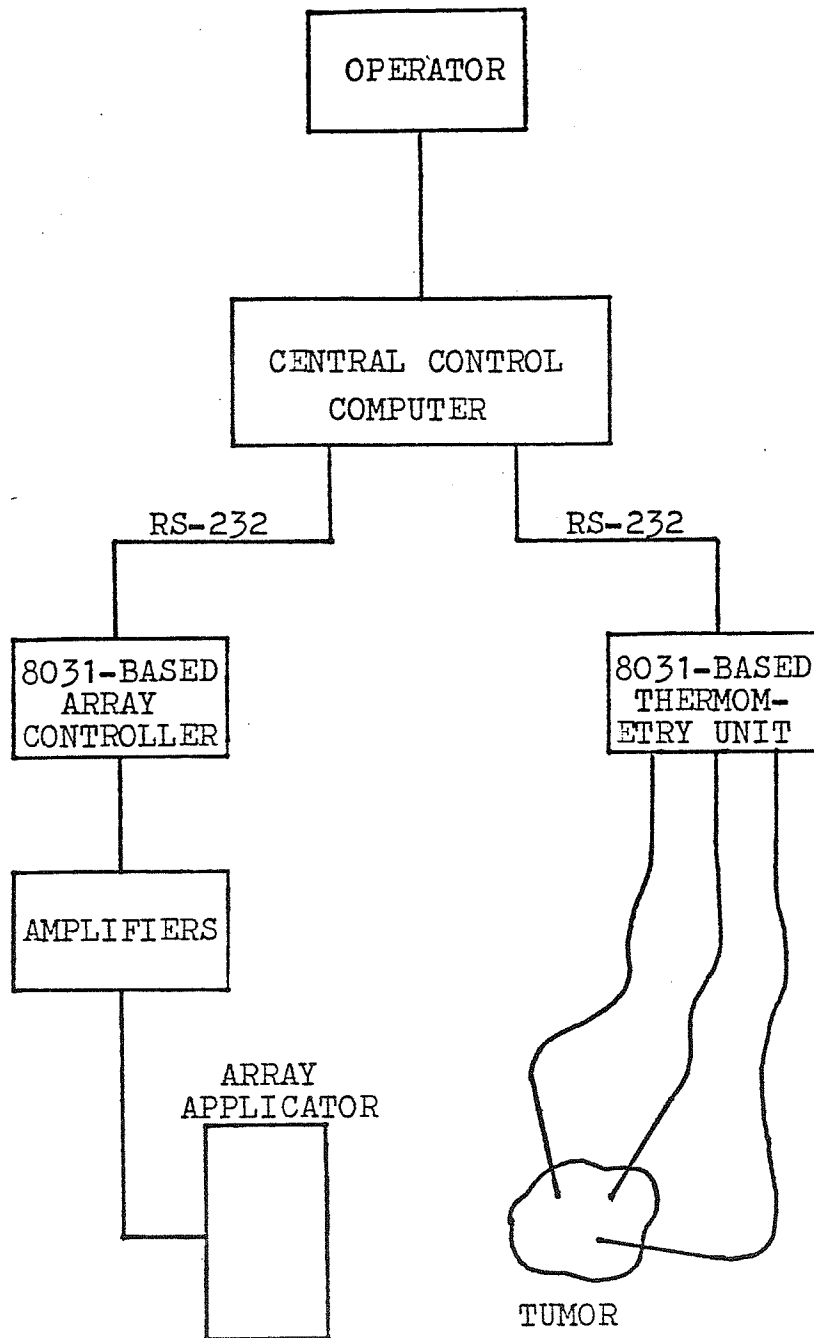


Figure 1. Hyperthermia System Block Diagram

particular tumor based on its size, shape, and various physiological factors, such as blood flow in the tumor region and absorption properties of the tumor tissue. This initial heating pattern is sent to the array controller in the form of desired time-averaged intensities for the ultrasonic focus at various locations within the tumor volume. The PAC, which creates the phased excitation signals that are amplified and applied to the array elements, sweeps a tightly focused beam throughout the tumor volume while adjusting the beam's time-averaged intensity according to the instructions from the central computer.

The resulting temperatures created in the tissue are monitored by the thermometry unit, which sends this information back to the central computer. These data are processed in an adaptive thermal modeling algorithm, and a revised heating pattern is sent to the array controller. In addition, the opportunity for operator input is provided at all times during this closed-loop process.

### 1.3 Introduction to the Phased Array Controller

An initial prototype PAC (PAC1) was designed and constructed previously (see Silverman, 1984) for use in the hyperthermia system described above. This thesis describes several modifications and further developments made on the PAC1. A brief introduction to the PAC is presented here.

The PAC generates excitation signals for the elements of an ultrasonic linear phased array transducer. These signals

are TTL-level square waves of a specified frequency with specific phasing relative to one another. They are amplified and transmitted to the array elements in order to create a focused beam of ultrasound, whose position relative to the array can be manipulated in two dimensions by varying the relative phases. Beam position in the third dimension can be adjusted electronically in a number of ways. One way is to taper each of the array elements along its length, as shown in Fig. 2. This causes the resonant frequency of each element to vary along its length, so that different excitation frequencies cause different regions of the array to produce the greatest output (Benkeser et al., 1985). This type of array is referred to as an ultrasonic tapered linear phased array (UTLPA), which PAC1 was designed to control. Another type of array is the ultrasonic stacked linear phased array (USLPA), which allows movement of the beam by switching excitation among groups of untapered linear phased arrays. The USLPA is described briefly in Chapter 2.

The treatment field of either array type is defined by a three-dimensional coordinate axis grid, whose origin is at the center of the array as illustrated in Fig. 2. The array can be focused at any of 4096 coordinates defined in this grid. The size and shape of the focus are determined by the type and physical parameters of the array (Ocheltree, 1984).

Each excitation signal from the PAC originates from the most significant output bit of a 4-bit digital counter, which runs continuously to drive its corresponding array element.

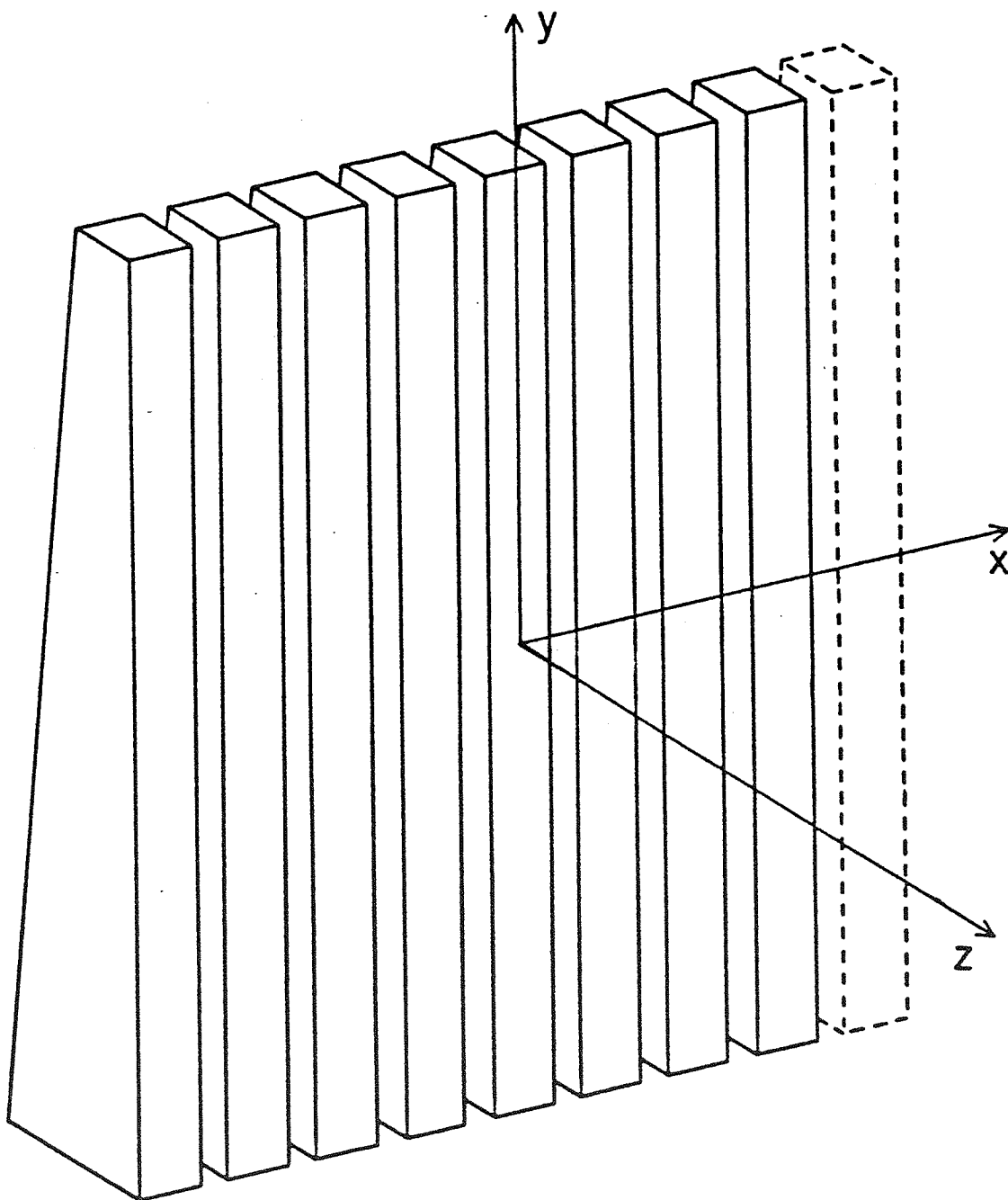


Figure 2. Tapered Linear Phased Array

The relative phase of each signal is determined by the initial 4-bit value with which its corresponding counter is loaded. The proper counter-load values for a given focal position are determined by its location in the X-Z plane, the frequency which corresponds to its location along the Y-axis (for the UTLPA), and the physical parameters of the array. The counter-load values for all 4096 possible focal positions are thus unique to a given array design and are calculated by an Apple-Basic program written specifically for this purpose (Silverman, 1984). They are then stored in EPROMs which reside within the PAC. To produce a focus at a specified location, the Intel 8031 microcontroller within the PAC sends these EPROMs the desired coordinates in the form of an address, where the proper counter-load values have been stored. These values are sent from the EPROMs to the counters, which are then activated to generate the proper excitation signals.

In order to heat a volume larger than the focus, a number of discrete X,Y,Z-coordinate positions are heated in sequence, forming a path which is scanned repetitively. This is done rapidly, so that each point is heated frequently enough to maintain a nearly constant temperature. The amount of heat delivered to each position is controlled by adjusting the fraction of time that the array is excited while held in that position before moving on to the next position. This "duty cycle" modulation of the array output was the only intensity control available in the PAC1. Modifications have

been made, however, which enable the controller to adjust the output level of the signal amplifiers as well. This is discussed in a later chapter.

This thesis is organized into seven chapters. Chapter 2 describes the design of new hardware that adapts the PAC for use with a USLPA as well as a UTLPA applicator. The hardware that was designed for frequency selection in the PAC is described in Chapter 3. Chapter 4 discusses some of the practical considerations for clinical use of the PAC and details a proposed closed-loop control scheme. Chapters 5 and 6 describe in detail the hardware and software, respectively, which were developed to implement the closed-loop control scheme. Finally, Chapter 7 contains suggestions for future consideration and a summary of the thesis work.

## CHAPTER 2. STACKED ARRAY CONTROL

2.1 Theory

A linear phased array has the ability to focus in two dimensions; that is, the position of the focus relative to the array can be manipulated within a single plane (the X-Z plane in Fig. 2). The focus can be positioned in the third dimension in a number of ways. The PACI was designed for use with a UTLPA, which uses frequency selection for positioning along the Y-axis. Another method is to use a series of untapered arrays stacked upon one another, as shown in Fig. 3. This configuration is referred to as an ultrasonic stacked linear phased array (USLPA). Movement of the focus in the Y-direction is accomplished by switching the excitation from one group of adjacent linear arrays to another.

Theoretical calculation of focal dimensions indicates that the optimal beam size for a stacked array is produced when three adjacent linear arrays are excited simultaneously (Ocheltree et al., 1984). Movement by one position in the Y-direction is achieved by turning off an array at one side of the group of three excited arrays and exciting the adjacent array on the other side. Thus, a stacked array with four linear arrays would have two possible positions in the Y-direction, while ten arrays would allow eight positions. This concept is illustrated in Fig. 4.



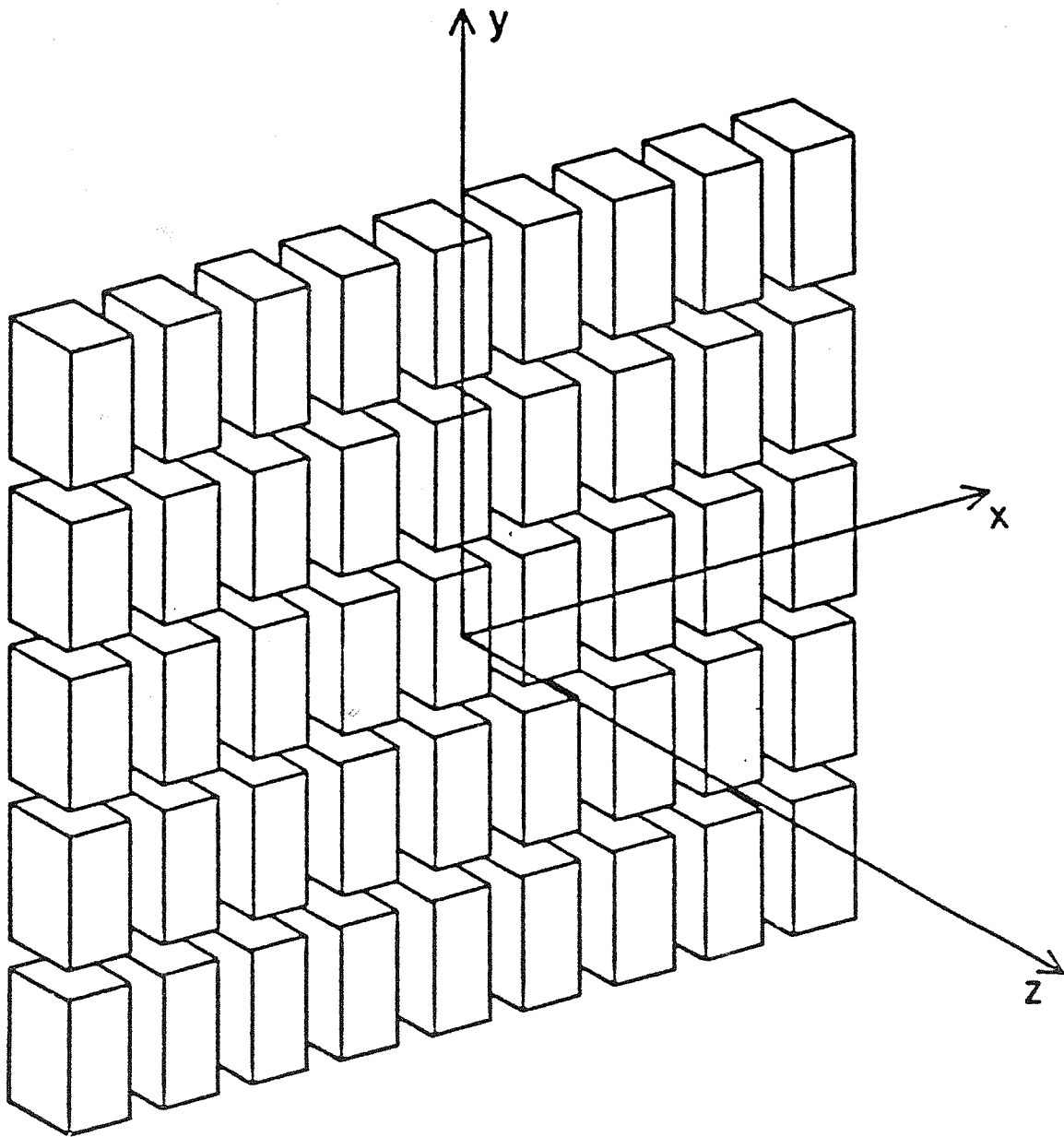


Figure 3. Stacked Linear Phased Array

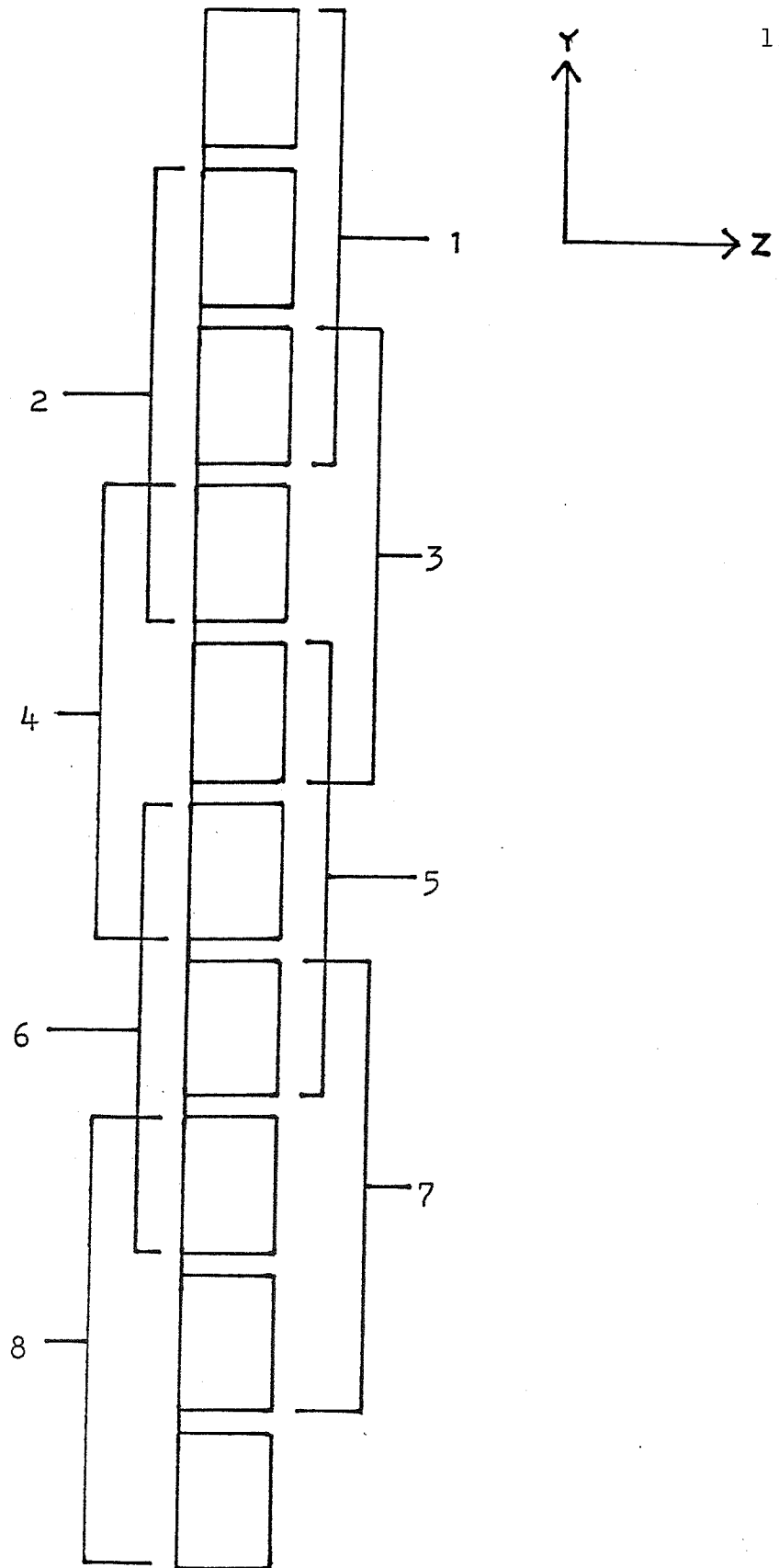


Figure 4. Stack of Ten Linear Arrays—Side View Showing Excitation to Produce Eight Possible Y-Positions

## 2.2 Design Objectives

The hardware used to control the stacked array, in addition to creating the required phased excitation signals, must be able to gate these signals to any set of three adjacent linear arrays. The group of three adjacent arrays which is excited at any given time is specified by the desired Y-coordinate. The primary design objective in meeting these requirements was to modify the PAC1, with minimal hardware changes, so that it could be used with either a tapered or a stacked linear phased array. When used with a tapered array, the Y-coordinate would select excitation frequency; when used with a stacked array, it would select the appropriate group of three arrays to be excited, and a fixed frequency would be used.

It is appropriate at this point to review briefly the nature of the coordinate specification at the hardware level in the PAC1 (refer to Silverman, 1984 for details). Sixteen coordinate positions were defined along each axis, so that four binary bits could be used to represent each coordinate along a single axis. This has been changed to 32, 8, and 16 for the number of X, Y and Z coordinate positions, respectively, leaving the total number of coordinate positions unchanged. The reason for this modification is discussed in Chapter 4. The change in coordinate positions required some minor modifications to the Apple-Basic "Look-Up-Table Generator" program. The modified program is

included in the Appendix.

Specification of a complete set of X,Y,Z-coordinate positions requires twelve binary bits (5 for X, 3 for Y, and 4 for Z), which are sent from the microcontroller board to all other boards via the backplane bus lines. These twelve bits are used by the element driver boards to address the look-up-table EPROMs, and are translated by an LED display driver board into a visual display of focus position. In addition, the Y-bits are used by the clock board to select excitation frequency. (The hardware to accomplish this was not part of the PAC1, but is described in Chapter 3).

For the array controller to be used with a stacked array, the Y-coordinate must function somewhat differently than described above. First of all, excitation frequency must now remain fixed for all values of Y. This means that the clock board, rather than using the Y-coordinate to select frequency, must generate a fixed frequency which is predetermined and unrelated to the Y-coordinate. This is accomplished by a simple switch, which is described along with frequency selection in Chapter 3.

Although no modifications are required for the element driver boards, the reason for this requires some explanation. Proper phasing is dependent upon excitation frequency, and for the tapered array there is a direct correspondence between Y-coordinate position and frequency, which is entered into the Apple-Basic "Look-Up-Table Generator" program.

Though frequencies are used in the calculations, the resulting counter-load values are stored in EPROMs according to coordinate positions. For the stacked array there is no longer a correspondence between Y-coordinate position and frequency, because a single frequency is used. This problem is solved by simply entering this frequency into the table generator for every Y-coordinate position, so that the table generator uses the same frequency for all of its calculations. The resulting counter-load values will then be independent of the Y-coordinate values, making the Y-coordinates received by the element driver boards irrelevant. Therefore, no hardware changes are required.

The LED display driver board also requires no modifications, since the visual display must reflect the actual coordinates of focus position, and frequency is not involved. It should be noted, however, that the visual LED display was designed for a 16x16x16 coordinate grid and will require some translation if it is to be used with the new 32x8x16 grid.

The primary function of the Y-coordinate value when using the stacked array is to specify a set of three adjacent linear arrays to receive the phased excitation signals, which are common to all three arrays. To accomplish this, each individual element must have a digitally-controllable switch that can gate it on or off. All the elements of a given linear array must be gated together by the same control

signal. This concept is illustrated for a stack of four arrays of three elements each in Fig. 5. Each element of the arrays is driven by a signal having a different phase, but these signals are common to each of the arrays. The goal of the stacked array control hardware is to receive the Y-coordinate value from the bus and translate it into the digital control signals required to turn on the appropriate set of three adjacent arrays. It is anticipated that there will be ten linear arrays in the prototype stacked array, giving eight Y-positions determined by a 3-bit coordinate specification number. Table 1 shows the eight possible Y-coordinate values along with the corresponding control signals. The table also shows corresponding decoder outputs, which are explained in the following section.

TABLE 1. LIST OF DECODER OUTPUTS AND CONTROL SIGNALS BY Y-COORDINATE VALUE

Y-COORDINATE	DECODER OUTPUTS								CONTROL SIGNALS										
	<u>Y0</u>							<u>Y7</u>	ROW:	1	2	3	4	5	6	7	8	9	10
0 0 0	0	1	1	1	1	1	1	1		1	1	1	0	0	0	0	0	0	0
0 0 1	1	0	1	1	1	1	1	1		0	1	1	1	0	0	0	0	0	0
0 1 0	1	1	0	1	1	1	1	1		0	0	1	1	1	0	0	0	0	0
0 1 1	1	1	1	0	1	1	1	1		0	0	0	1	1	1	0	0	0	0
1 0 0	1	1	1	1	0	1	1	1		0	0	0	0	1	1	1	0	0	0
1 0 1	1	1	1	1	1	0	1	1		0	0	0	0	0	1	1	1	0	0
1 1 0	1	1	1	1	1	1	0	1		0	0	0	0	0	0	1	1	1	0
1 1 1	1	1	1	1	1	1	1	0		0	0	0	0	0	0	0	1	1	1

### 2.3 Implementation

Figure 6 shows a schematic of the stacked array control hardware, which was constructed on an unused portion of the

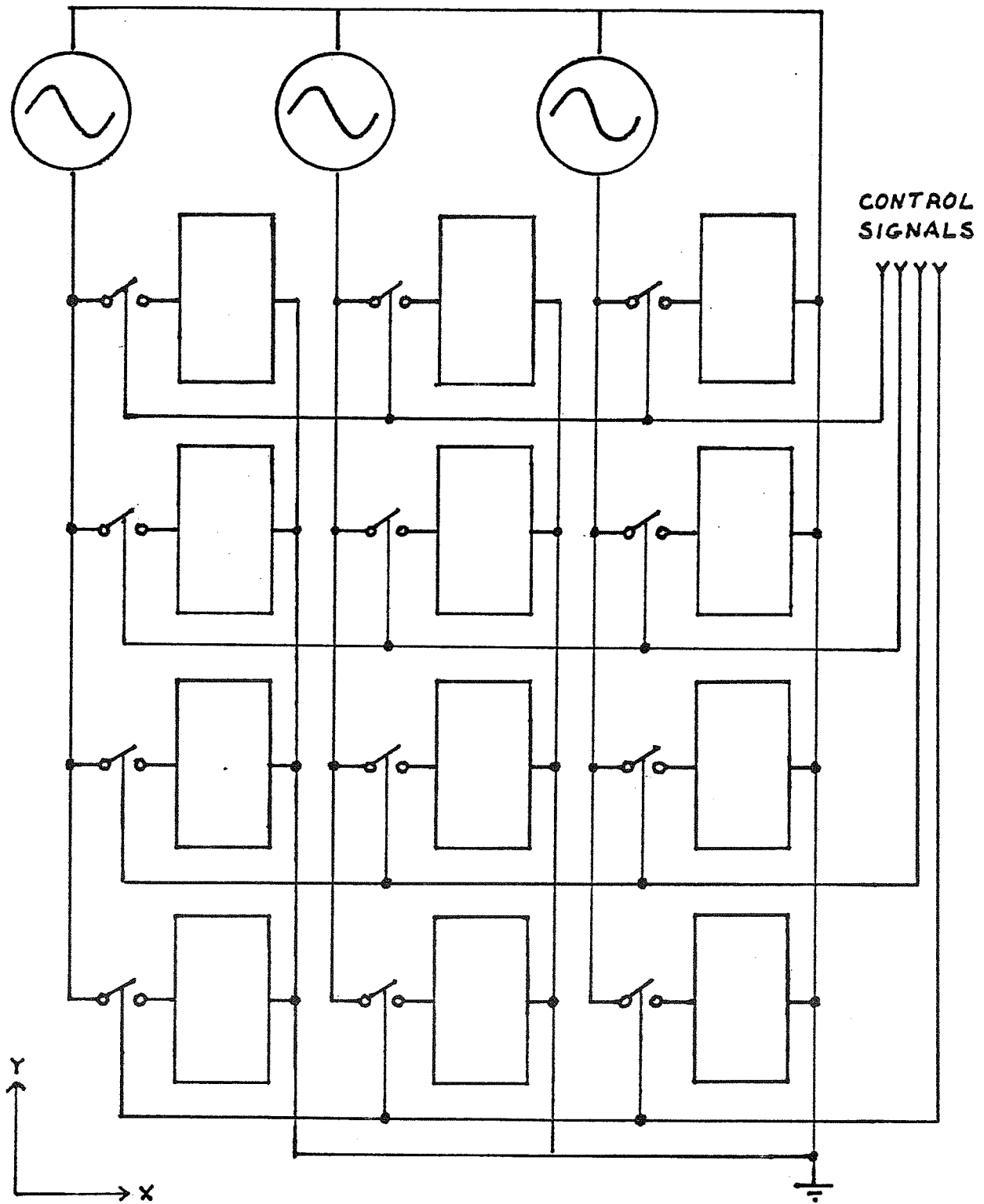


Figure 5. Diagram Showing Control Signals Gating Driving Signals to Stacked Array Elements

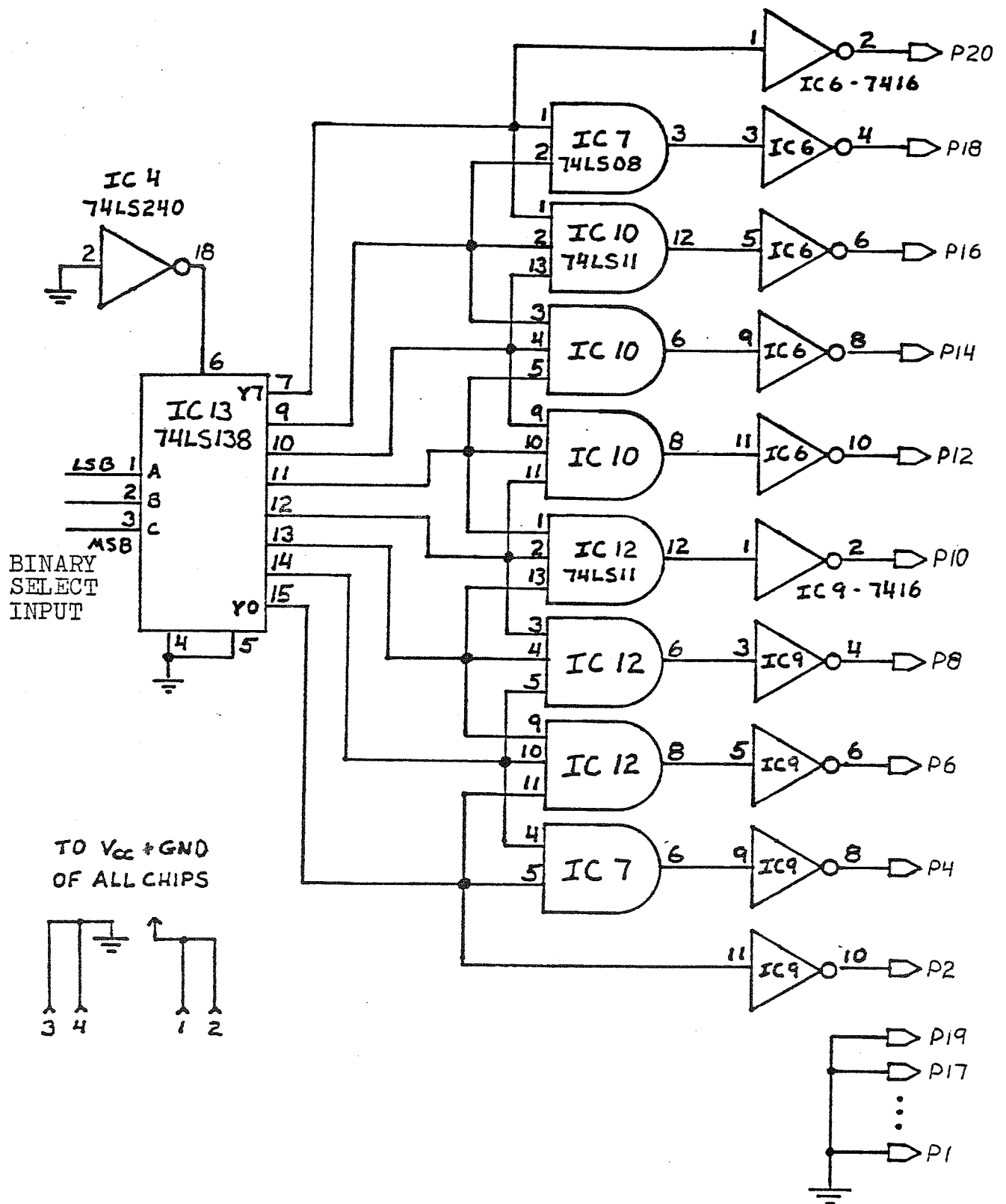


Figure 6. Stacked Array Control Hardware Schematic



array controller clock board. This board is now referred to as the clock/stacked array control board, and its component layout is shown in Fig. 7. The three bits for the Y-coordinate value are received on bus lines 25, 27, and 29. Line 23, which was used for the most significant bit (MSB) of the 4-bit Y-coordinate value in the PAC1, is now used for the least significant bit (LSB) of the new 5-bit X-coordinate value. This leaves line 25 for the MSB of the new 3-bit Y-coordinate value, line 27 for the middle bit, and line 29 for the LSB. Table 2 illustrates this change in bus line definitions.

TABLE 2. CHANGES IN BUS LINE DEFINITIONS FOR X- AND Y-COORDINATES

<u>BUS LINE</u>	<u>PREVIOUS DEFINITION</u>	<u>NEW DEFINITION</u>
15	X(3) MSB	X(4) MSB
17	X(2)	X(3)
19	X(1)	X(2)
21	X(0) LSB	X(1)
23	Y(3) MSB	X(0) LSB
25	Y(2)	Y(2) MSB
27	Y(1)	Y(1)
29	Y(0) LSB	Y(0) LSB

The stacked array control hardware uses the I/O Select (line 33) and Data Strobe (line 32) signals to latch the Y-coordinate in the same manner as the other boards in the controller. The three Y-bits are then decoded by a 3- to 8-line decoder, which outputs a zero in the position corresponding to the binary input, and a one at every other position. Ten control signals are required, and inspection of Table 1 will show that their logical relationship to the

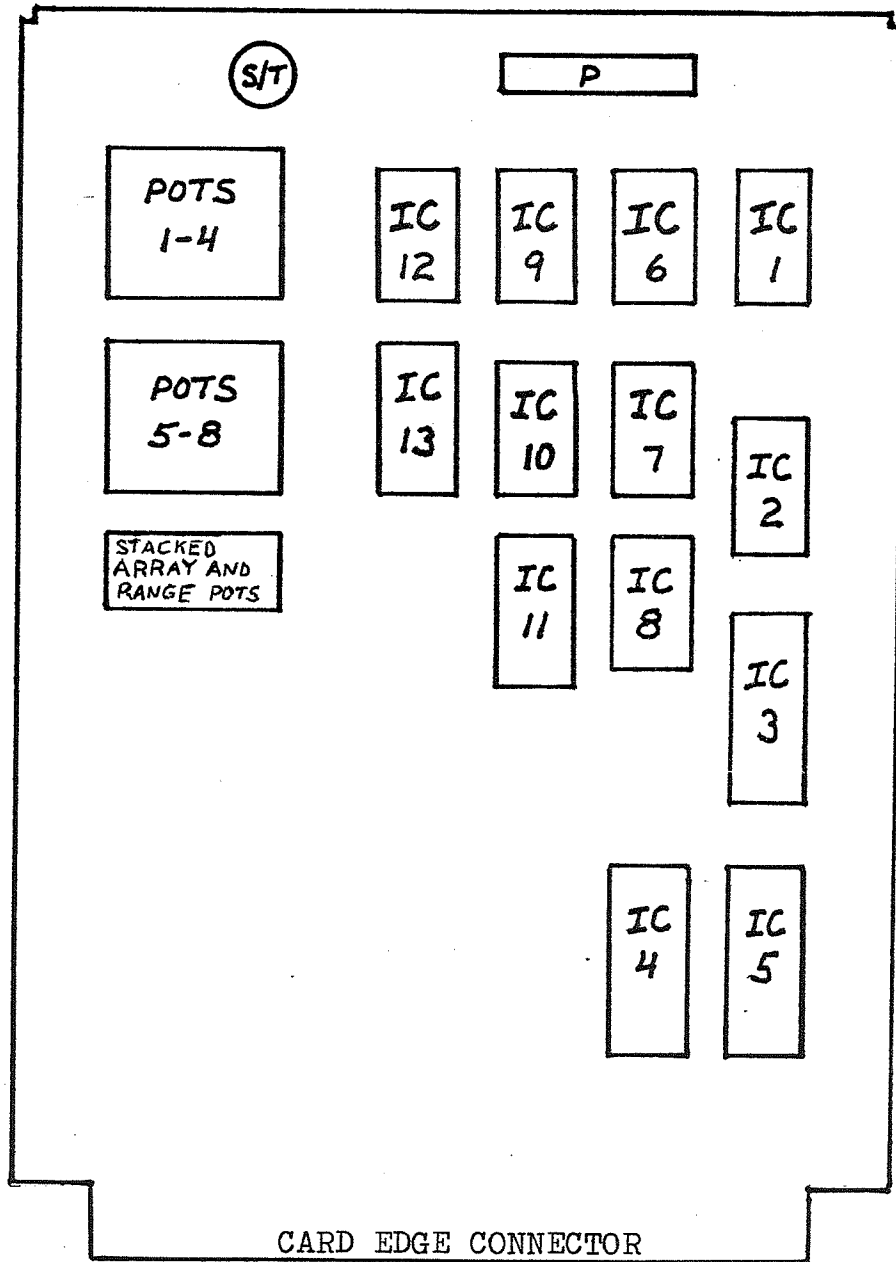


Figure 7. Clock/Stacked Array Control Board Component Layout (Component side of board)

decoder outputs is as follows:

$$\begin{array}{ll}
 \text{ROW 1} = \overline{Y_0} & \text{ROW 6} = \overline{Y_3 \cdot Y_4 \cdot Y_5} \\
 \text{ROW 2} = \overline{Y_0 \cdot Y_1} & \text{ROW 7} = \overline{Y_4 \cdot Y_5 \cdot Y_6} \\
 \text{ROW 3} = \overline{Y_0 \cdot Y_1 \cdot Y_2} & \text{ROW 8} = \overline{Y_5 \cdot Y_6 \cdot Y_7} \\
 \text{ROW 4} = \overline{Y_1 \cdot Y_2 \cdot Y_3} & \text{ROW 9} = \overline{Y_6 \cdot Y_7} \\
 \text{ROW 5} = \overline{Y_2 \cdot Y_3 \cdot Y_4} & \text{ROW 10} = \overline{Y_7}
 \end{array}$$

(where  $\cdot$  and  $\overline{\quad}$  represent the AND and inversion operations, respectively).

Each decoder output is ANDed with the immediately adjacent output(s) and then inverted to produce the control signals for rows 2 through 9. Rows 1 and 10 are controlled by inverting the  $Y_0$  and  $Y_7$  outputs, respectively.

The nature of the switches which will use these control signals will now be discussed. Since each individual element must have its own switch, the number of switches required will be equal to ten (the number of arrays in the stack) times the number of elements in each array. This becomes impractical for arrays with many elements (64 elements in each array, for example), unless the switches are small and the circuitry very simple.

One possibility is to use a single MOSFET for each switch. For an N-channel enhancement mode MOSFET, a positive bias of approximately ten volts applied to the gate (with respect to either the source or the drain, depending on how the switch is configured) will turn the switch on, while zero bias will leave the switch off. Although the design of these

switches is not part of this thesis, several considerations for implementing a MOSFET design are presented in Chapter 7.

In order to interface the digital control signals with analog switches, inverters with high voltage outputs (approximately 12 volts) are used. The control signals are sent to a ribbon connector on the edge of the clock/stacked array control board. The correct operation of the stacked array control circuitry has been verified with the use of a logic analyzer.

## CHAPTER 3. FREQUENCY SELECTION

### 3.1 Introduction

The purpose of the frequency selection circuitry in the phased array controller is to provide a clock source for the counters on the element driver boards, at a frequency determined by the Y-coordinate value. A portion of the hardware for this frequency selector was constructed on a board referred to as the "clock board" in the PAC1. The schematic for this clock board (Silverman, 1984; Fig. 7) shows a dashed box labelled "Digitally Controlled Clock Source." The hardware required to fill this space must accept the "binary select input" (the Y-coordinate value) and use it to select one of several frequencies for the clock source, which is placed on the line labelled "output." The hardware must also be able to output a single, predetermined frequency independent of the binary select input if a stacked array is being used. The design and construction of the digitally controlled clock source is the topic of this chapter.

### 3.2 Design Objectives

Although the clock board schematic for the PAC1 (Silverman, 1984; Fig. 7) shows four lines being used for the binary select input, recall that the Y-coordinate value is now specified by only three bits. One of these four lines has therefore been removed (the one coming from bus line 23—

see Table 2). This means that eight frequencies must be available for a tapered array. Another frequency which will in general be different from these eight is required for a stacked array. There must be a simple way to switch the clock source between the tapered array and the stacked array modes.

The range of possible operating frequencies for a phased array is approximately 200 to 800 kHz. Since the most significant bit of a four-bit counter is used to produce each excitation signal, the frequency of the clock source must be sixteen times greater than the desired operating frequency. This results in a range of approximately 3 to 13 MHz required for the clock source. The output must have accuracy on the order of  $\pm 0.1\%$  at the maximum frequency and  $\pm 1\%$  at the minimum frequency, because a change of more than a few kHz in the excitation signals will produce a significant change in focal position. The output must also exhibit no temperature drift.

It is likely that a number of different arrays will be constructed, perhaps for tumors at different depths or in different regions of the body. Since a different set of frequencies will be required for each array, the frequency selection circuitry must be designed so that changing the set of available frequencies is a simple matter.

### 3.3 Implementation

Figure 8 shows a schematic of the digitally controlled clock source, with a dashed box around it to show that it fills the dashed box on the original clock board schematic (Silverman, 1984; Fig. 7). The hardware was constructed on the same board, now referred to as the clock/stacked array control board.

The design is simple, the main components being an eight-channel analog multiplexer and a voltage-controlled oscillator (VCO). The VCO outputs a TTL-level square wave whose frequency is dependent upon the voltage applied at the frequency control input (pin 13). The multiplexer selects one of eight input voltages to be directed to the VCO, depending on the value of the three-bit binary select input (the Y-coordinate value).

The multiplexer input voltages are provided by a bank of eight potentiometers (pots), each of which is connected between the five-volt supply and ground. The pots are set to provide whatever voltages are needed by the VCO to produce the clock frequencies required for a given array. Once set, the position of each center-tap is mechanically "frozen" with a drop of epoxy (or similar substance) on the turn-screw, to insure that all the voltages remain fixed. The pots are bonded together in two sets of four in order to form "plug-in modules" which are unique to a given tapered array. This makes reconfiguring the hardware for a different array as simple as replacing these modules with those for the new

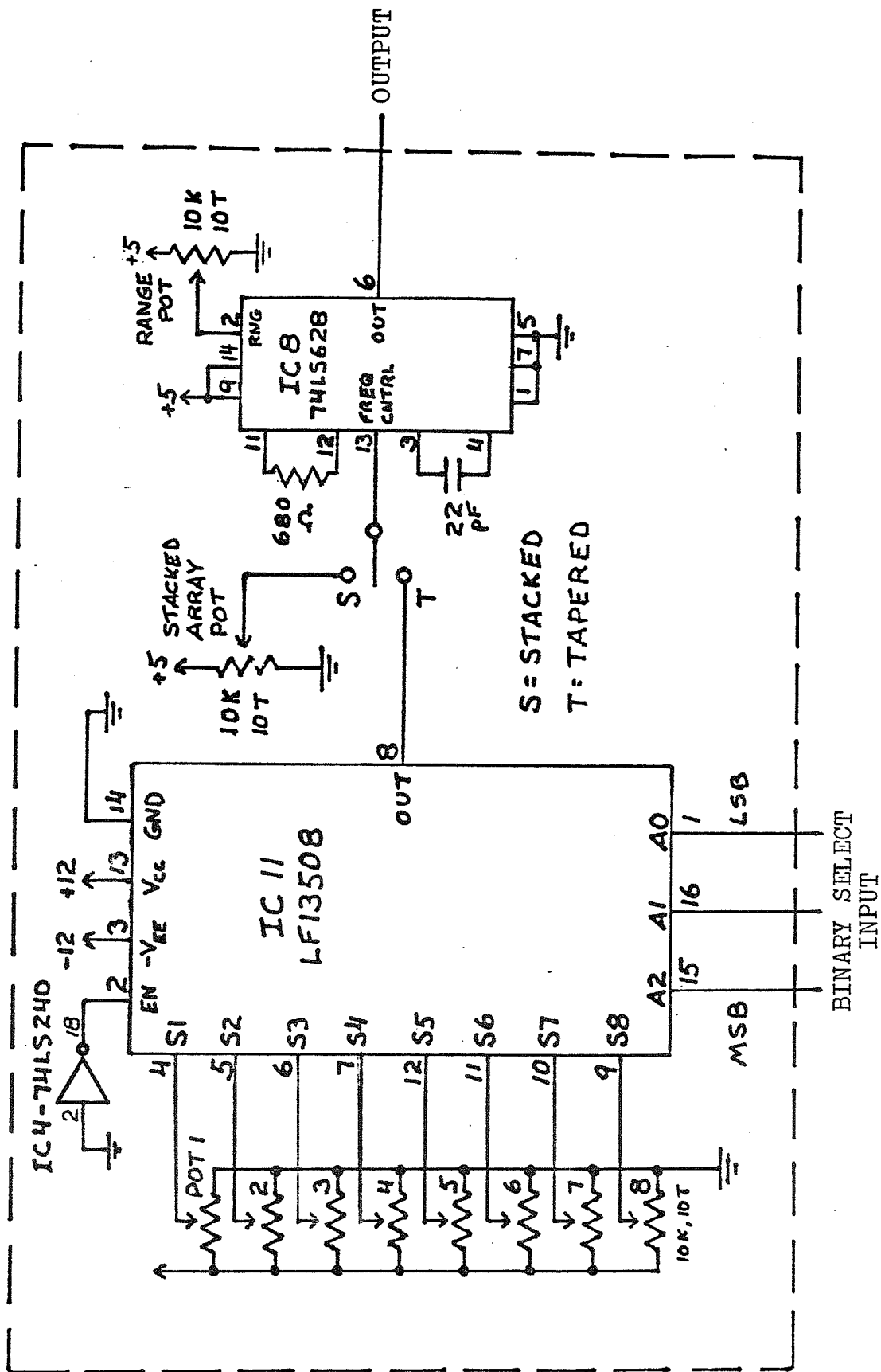


Figure 8. Digitally Controlled Clock Source Schematic



array.

For a stacked array, the VCO must output a single, fixed frequency independent of the Y-coordinate value. It must therefore receive a single, fixed voltage at its frequency control input rather than the output of the multiplexer. A simple toggle switch located on the edge of the clock/stacked array control board is used to select either the multiplexer output or a fixed voltage for frequency control. The fixed voltage is provided by another potentiometer, whose center-tap is set and frozen as described above. This pot is unique to a given stacked array and can be easily replaced.

The output frequency of the VCO depends not only on the frequency control voltage but also on the frequency range which is selected. The range is determined by the value of an external capacitor (connected between pins 3 and 4) and by an external voltage applied to the range input (pin 2). The desired range for the phased arrays (approximately 3 to 13 MHz) can be obtained with a 22 picofarad capacitor and 2-3 volts applied to the range input. (The reader may wish to refer to the characteristic frequency curves for the VCO in Texas Instruments' The TTL Data Book for Design Engineers, p. 7-463). The range voltage is provided by another potentiometer whose center-tap is set and frozen. Although it is likely to be unnecessary, this pot can be easily replaced if an array requiring frequencies outside the 3-13 MHz range is constructed.

Time constraints did not allow the implementation described above to be completely tested. The operation of the analog multiplexer is very straightforward and should not present any problems as long as the voltages provided by the potentiometers remain fixed. The concept of "freezing" the center-tap has been tested and appears to work very well. The only potential source of problems appears to be in the stability of the VCO. One VCO in the 74LS series of TTL ICs, the 74LS624, has been observed to have the required accuracy but unacceptable temperature drift. The VCO used in the design and construction of the clock source is the 74LS628, which is equivalent to the 624, but it allows more precise temperature compensation by the addition of an external resistor. If this compensation proves to be inadequate, an alternate design approach would be to use a crystal oscillator circuit instead of a VCO. Eight crystals would be needed, custom ground to the exact frequencies required. An analog multiplexer could then be used to switch one of these crystals into the circuit, as dictated by the Y-coordinate value.

## CHAPTER 4. CLOSED-LOOP CONTROL FOR PHASED ARRAY HYPERTHERMIA

4.1 Introduction

This chapter deals with the protocol which was designed for integrating the phased array controller (PAC) into a complete closed-loop system, described briefly in Chapter 1. Practical considerations for clinical use of the PAC are discussed, along with an explanation of how the central computer controls a hyperthermic treatment via the PAC.

Before discussing the closed-loop control scheme, the primary restraint on its design should be noted. The phased array control scheme was designed to be as similar as possible to that which was already designed for a fixed (non-phased) array applicator system (the SONOTHERM 1000 by URI Therm-x, Inc.). The reader may wish to consult reference materials for this system. The control scheme which was used in the PAC1 has for the most part been discarded and need not be referred to unless otherwise noted.

The central computer controls a hyperthermic treatment by telling the PAC which regions of the tumor should be scanned and at what time-averaged intensities, based on temperature information received from the 16-channel thermometry unit. In order to do this, the computer must view the tumor volume as a series of smaller volumes. At any given time, each of these smaller regions will be scanned with a time-averaged intensity ranging from 0 to 100%, depending on temperatures in and around that region. The

computer must know where the temperature probes are placed in relation to each of these regions, as well as the position of the tumor with respect to the treatment field of the array applicator. The following section deals with the definition of the treatment field and its division into regions of intensity control.

#### 4.2 Organization of the Treatment Field

The dimensions of the treatment field as well as its position relative to the array applicator must be defined before the counter-load value look-up-tables can be generated for that array. This is done by entering into the "Look-Up-Table Generator" program a physical distance from the X-axis (the array face) for each Z-coordinate, and from the Z-axis for each X-coordinate. Typically, the extreme focal distances (near and far, left and right) for a particular array would be entered for the first and last coordinates in both dimensions, and then the field distributed evenly over the remaining coordinates. This would center the field on the Z-axis and cover the maximum range of the array in both dimensions. It should be noted that this defines a new "origin." The position whose coordinates are (0,0,0) will be viewed as the "origin position" by the array controller, but it will not be located at the intersection of the X, Y, and Z-axes in Fig. 2.

The prototype phased arrays will be designed for tumors up to 5 cm in diameter and centered at a maximum depth of

approximately 10 cm. These arrays will produce a focal region approximately 3 mm x 1.5 cm x 2 cm (X x Y x Z, 3 dB widths). The reason for changing the coordinate grid dimensions from 16x16x16 to 32x8x16 can now be seen. Since the width of the focal volume in the X-dimension is so much smaller than in the Y- and Z-dimensions, a greater number of X-coordinate positions will be needed to effectively cover the entire width of a tumor. Eight coordinate positions are all that are required in the Y-dimension where the beam is quite broad.

In the X-dimension, thirty-two focus volumes placed "end-to-end" (with no overlap of the field within their 3 dB widths) would cover almost 10 cm ( $32 \times 3 \text{ mm} = 9.6 \text{ cm}$ ). However, since the width of the treatment field will be approximately 5 cm, thirty-two positions will cover the X-dimension with considerable overlap of adjacent focal regions. Having this overlap available will be very desirable in situations where the perfusion rate varies among different regions of the tumor. In the Y-dimension, eight focal volumes overlapped by one-half would cover approximately 7 cm, so again there is adequate coverage with good overlap.

In the Z-dimension, sixteen focal volumes overlapped by one-half would cover approximately 17 cm, which is more than twice the Z-dimension of a tumor to be treated. These extra positions are utilized to provide increased flexibility in the placement of the applicator in terms of distance from the tumor. The total field in the Z-dimension is divided into a

series of overlapping "sub-fields." Each sub-field is large enough to cover the entire treatment field (tumor) in the Z-dimension, but requires only eight of the sixteen positions available in the Z-dimension. Each sub-field is located one position deeper than the previous sub-field, giving a total of nine, as illustrated in Fig. 9. Having these distinct sub-fields allows the operator some flexibility in coupling the applicator to the patient. There is no need to place the applicator at a fixed distance from the tumor, as long as the tumor is situated somewhere within the extremes of the Z-coordinate positions. Once the applicator is positioned, the sub-field which best overlaps the tumor volume is used for treatment.

Whichever sub-field is selected is further divided into a series of smaller three-dimensional regions, which will be referred to as "standard control blocks," or just simply "blocks." The computer determines an intensity ranging from 0 to 100% for each of these blocks, based on the temperature information it receives. The standard control block is defined in terms of coordinates rather than physical distances. Each block consists of four positions in the X-dimension and a single position in the Y- and Z-dimensions, so that there are four focal positions per block (see Fig. 10). A sub-field therefore consists of eight blocks in each dimension for a total of 512 blocks. It should be noted that while the standard control blocks are discrete units at

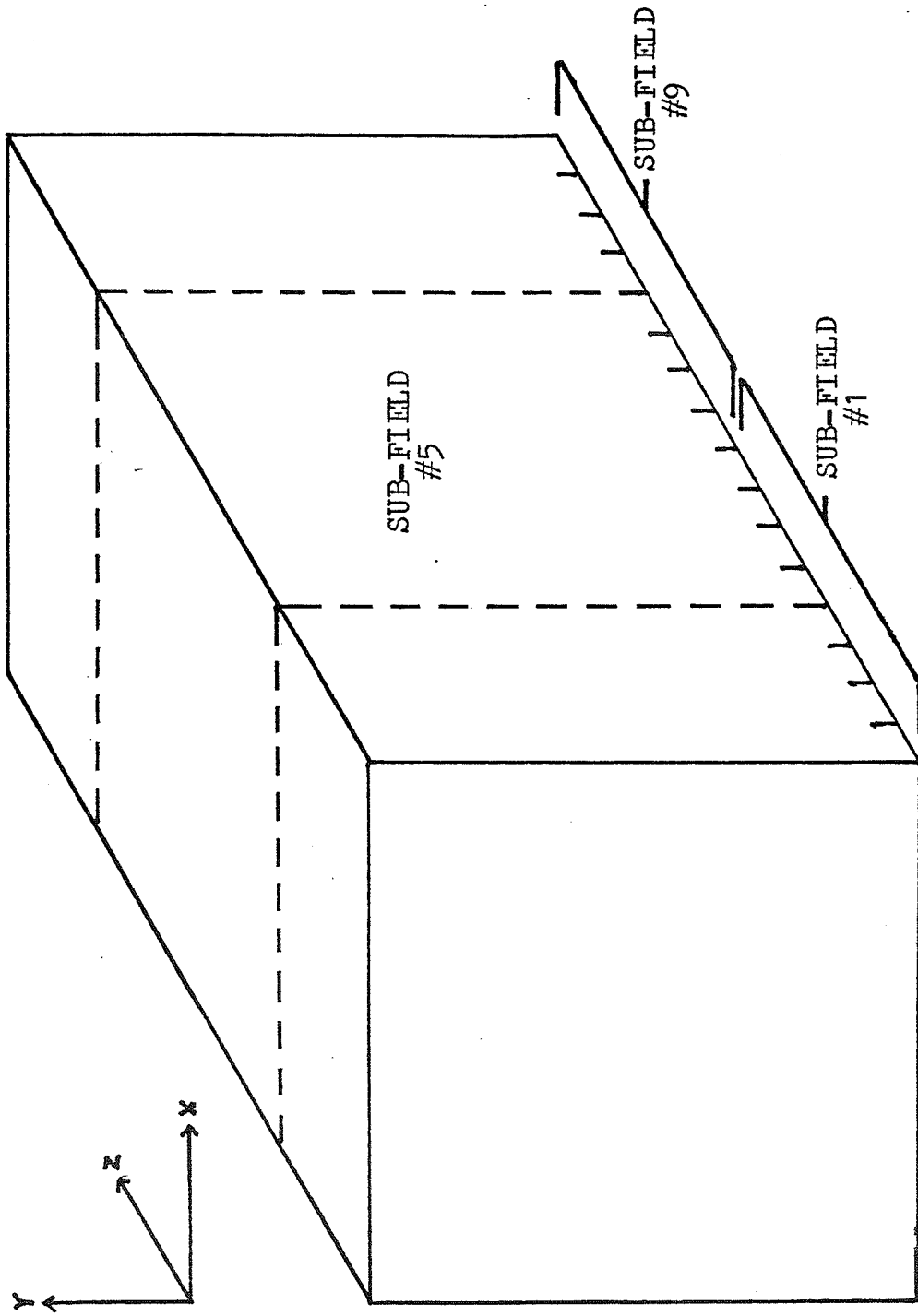


Figure 9. Diagram Showing Sub-fields of the Total Field

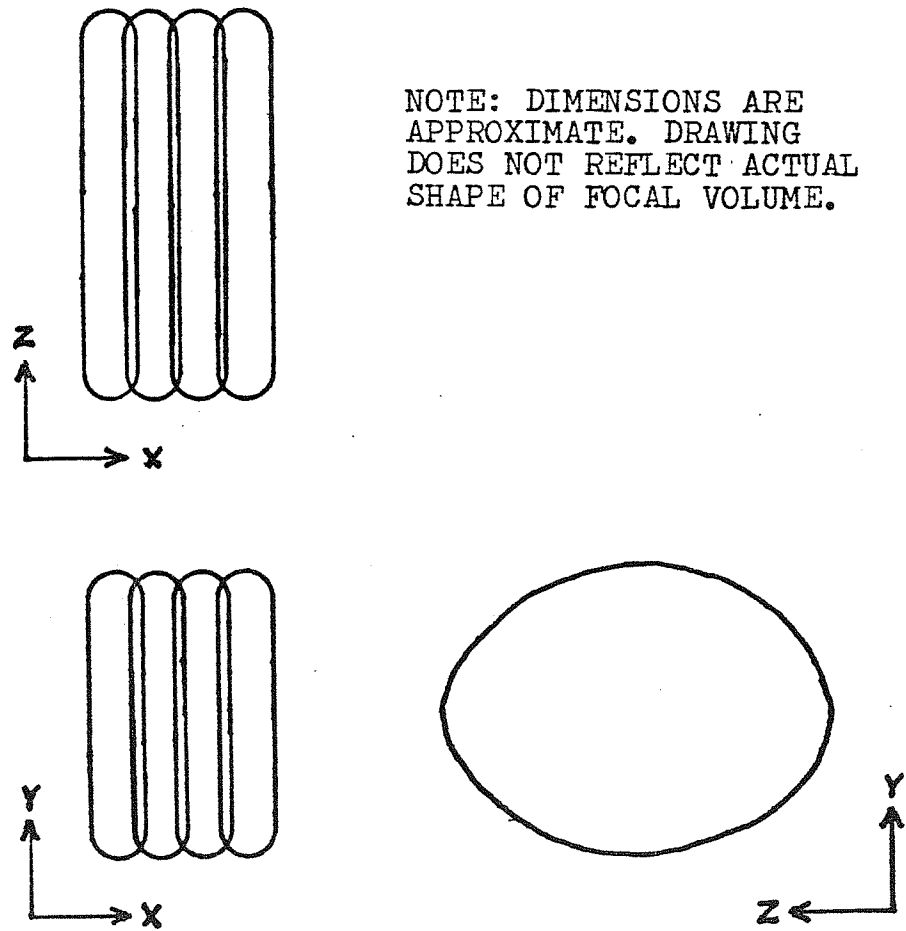


Figure 10. Cross-sections of Standard Control Block Showing Focal Volumes



the coordinate level, they overlap one another spatially due to the overlap of adjacent focal regions.

Although the PAC can adjust the time-averaged intensity at each individual focal position independently, all four positions within a block are grouped together and scanned at uniform intensity specified by the central computer for that block. If the computer specifies a zero intensity for a block, none of the positions in that block are scanned.

In the event that initial experiments indicate a need for the standard block to be defined differently than described above, the parameters which define the block can be easily changed. Not only can the dimensions be altered, but the focal positions which are scanned within the block can be spaced as desired (at every other position for example). This is explained more fully in Chapter 6.

The following section describes a typical treatment in general terms, while the details of the communications protocol, scan algorithm, and additional hardware for implementing the closed-loop control scheme are considered in the next two chapters.

### 4.3 Overview of a Typical Treatment

Before a scan can begin, several things must take place. The array controller must perform a self-check on its internal hardware and inform the central computer of the results of this check. The central computer must tell the array controller which of the nine sub-fields is to be used.

The array controller must set the amplifier output level according to instructions from the central computer, as is done in the SONOTHERM 1000. (This represents a significant change from the intensity control of the PAC1, which had no control over the amplifier output level; it instead combined "overall" and "relative" intensity factors in the implementation of software duty-cycle intensity control [see Silverman, 1984].) Finally, the central computer must send the PAC a list of time-averaged intensities for the standard blocks. "Intensity factors" are used for this purpose—they are translated by the PAC into duty-cycle modulation of array output, as in the PAC1. The intensity factors are sent in a specific sequence agreed upon by the computer and the PAC.

Once the intensity factors have been received by the PAC, scanning can be initiated. The PAC scans through the standard blocks in their predetermined sequence, skipping over blocks having a zero intensity factor. Temperatures are monitored as the PAC cycles through all the blocks repetitively. Every ten seconds the central computer revises the table of block intensities based on the temperatures it receives. It sends the new intensities to the PAC, and they are used as soon as they are received. This is done in a manner such that scanning is not interrupted. Thus, the computer exercises closed-loop control over a scan while it is in progress. If the operator wishes to halt the scan momentarily and then continue, he can do so.

The following two chapters discuss in detail the hardware and software currently being developed for the PAC to implement the control scheme just described.

## CHAPTER 5. HARDWARE

5.1 Introduction

In order to implement the control scheme discussed in the previous chapter, the phased array controller (PAC) must perform two major tasks. It must communicate with the central computer, and it must scan specified regions of a tumor with specified time-averaged intensities. The communications protocol which will be used is essentially the same as that in the SONOTHERM 1000 (see Cargoni, 1985). A great deal of "bookkeeping" is required of software in implementing this protocol—keeping track of redundant triplets, retransmission requests, timing functions and so forth. There is also a great deal involved in scan control—calculating the next position to focus at, keeping track of focal position within a block as well as block position within the sub-field, etc. This presents a problem for the 8031 microcontroller, because communications and scan control must be tended to simultaneously. This suggests the possibility of using two 8031s, one devoted solely to communications and the other to scan control. This chapter describes the design and operation of such a dual-controller system.

5.2 Design and Operation

Figure 11 shows a block diagram of the system using two 8031s. A separate ROM is used to hold each controller's

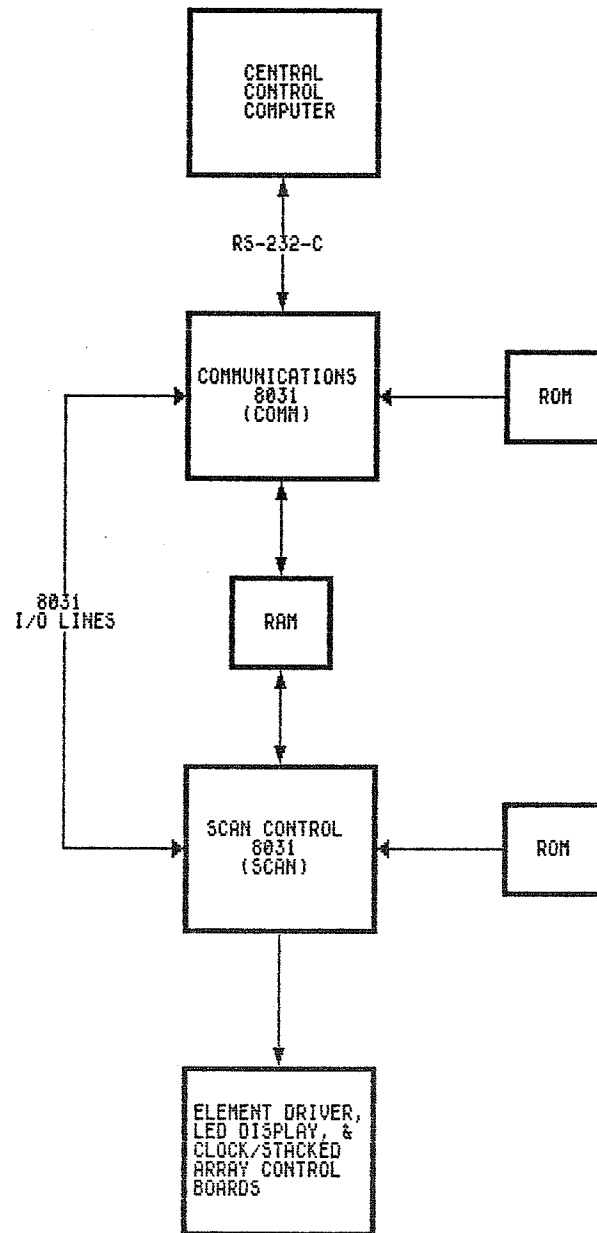


Figure 11. Dual-8031 System Block Diagram

software, while the processors share the external data memory (RAM). This is where the block intensity factors are stored by the communications 8031, and retrieved by the scan-control 8031 (hereafter referred to as COMM and SCAN, respectively). COMM controls communications in the PAC by interacting with the central computer, while SCAN controls scanning by interacting with all the other boards in the PAC. The two controllers communicate with one another via their I/O ports and through the external data memory.

The dual-controller system was designed to require only minimal modifications to the microcontroller board used in the PAC1. A second microcontroller board was designed to have COMM mounted on it and will be referred to as the "communications CPU board." The original microcontroller board with scan mounted on it will now be referred to as the "scan control CPU board." It should be noted that the schematic for the original microcontroller board (Silverman, 1984) does not reflect even the original design actually used, and it should not be referred to.

Figures 12 and 13 show a schematic and component layout, respectively, of the scan control CPU board, complete with all modifications made to the original board. Three components (ICs 1, 2, and 14) are drawn with broken lines to indicate that they should be removed from the board in order to implement the dual-controller system. These components constitute the RS-232 drivers and the 2K RAM, which now reside on the communications CPU board. If it is desired to

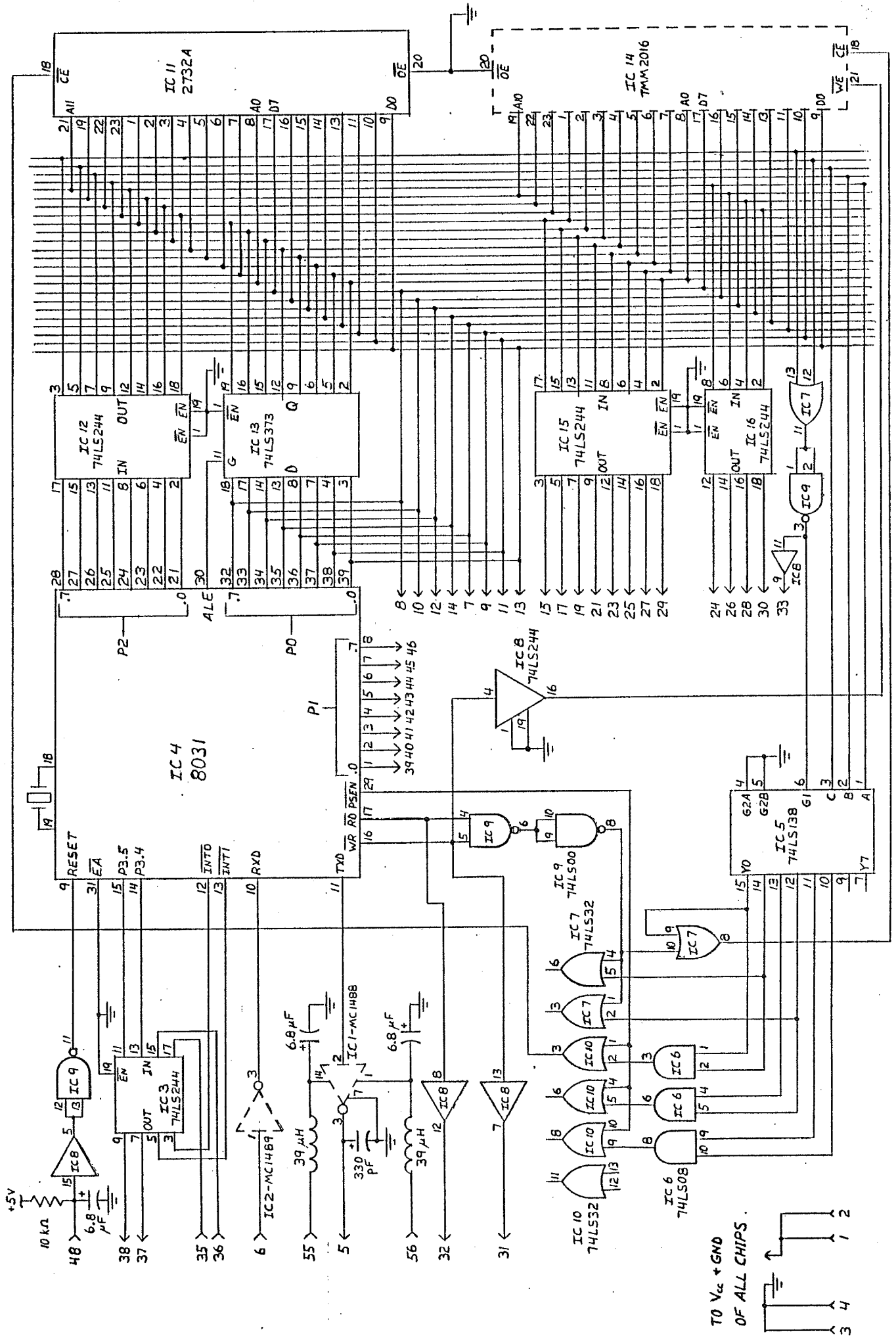


Figure 12. Scan Control CPU Board Schematic

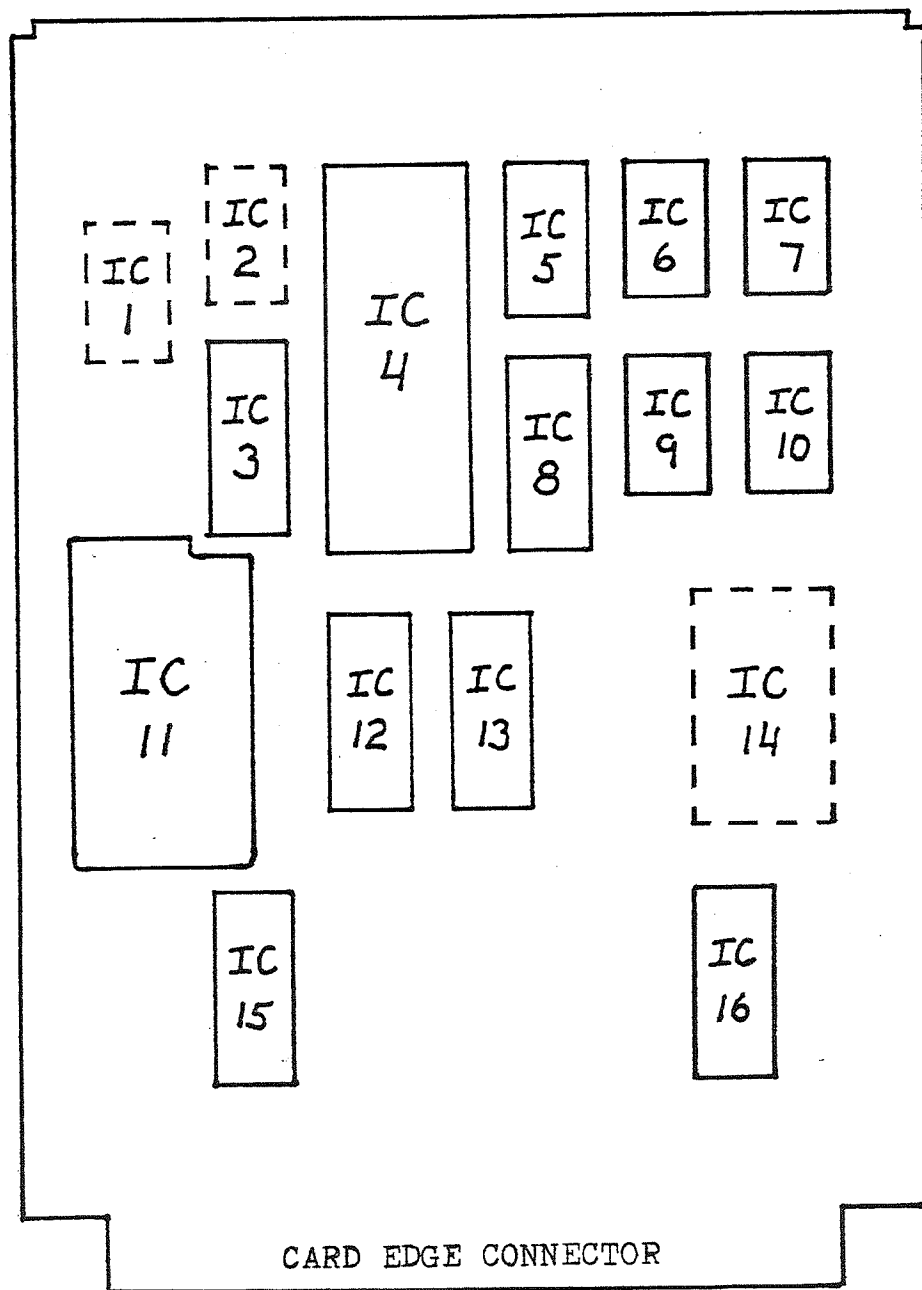


Figure 13. Scan Control CPU Board Component Layout  
(Component side of board)



use the scan control CPU board as it was used in the PAC1 (with the original software), this can be done by leaving these components in place and removing the communications CPU board from the card cage.

Figures 14 and 15 show a schematic and component layout, respectively, of the communications CPU board, which consists of one 8031 (IC 4), one 8-bit latch (IC 13), one 4K EPROM (IC 11), one 2K RAM (IC 15), two RS-232 interface chips (ICs 1 and 2), two octal tri-state bus transceivers (ICs 3 and 8), four quad 2- to 1-line data selectors (ICs 5, 6, 9, and 10), five octal tri-state buffers (ICs 12, 14, 16, 17, and 18), and a number of AND gates (IC 7). Figure 16 shows a block diagram illustrating the manner in which the two 8031s use separate program memories (ROM) but share the same data memory (RAM). The reader may wish to refer to this diagram throughout the following discussion.

In order to share the same RAM, the two 8031s must both have access to the RAM address and control inputs. This is accomplished by sending the address buses of both 8031s, along with the read and write signals ( $\overline{RD}$  and  $\overline{WR}$ ), to a series of 2- to 1-line data selectors. The data selectors allow only one of the two 8031s access to RAM at a time. The select function is performed by SCAN's Port 1 No.7 I/O line (P1.7). SCAN holds this line low so that it normally has access to RAM itself. When COMM requires access to RAM, it must request it from SCAN. Since it is impossible for both 8031s to access RAM simultaneously, the software for

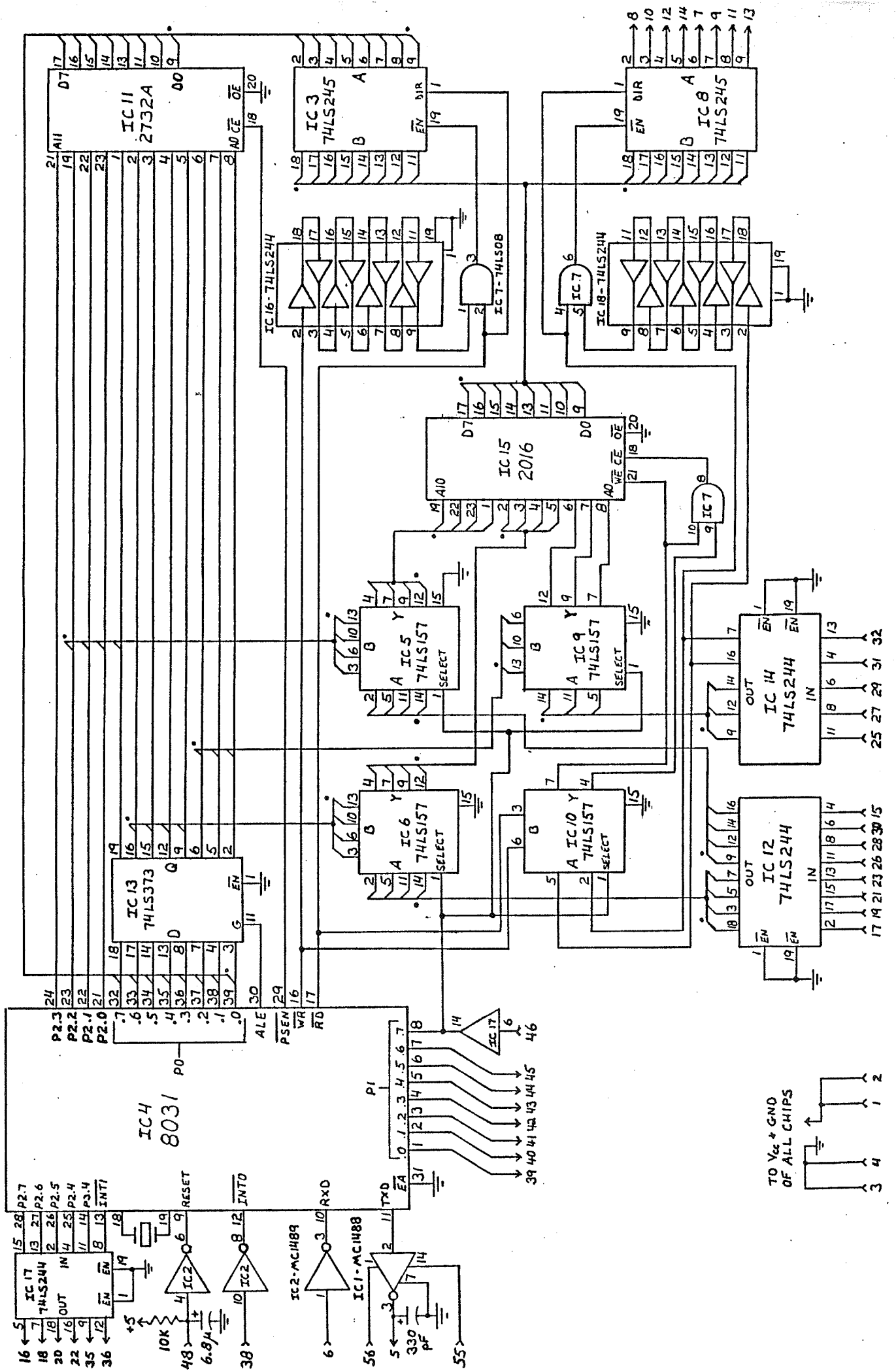


Figure 14. Communications CPU Board Schematic

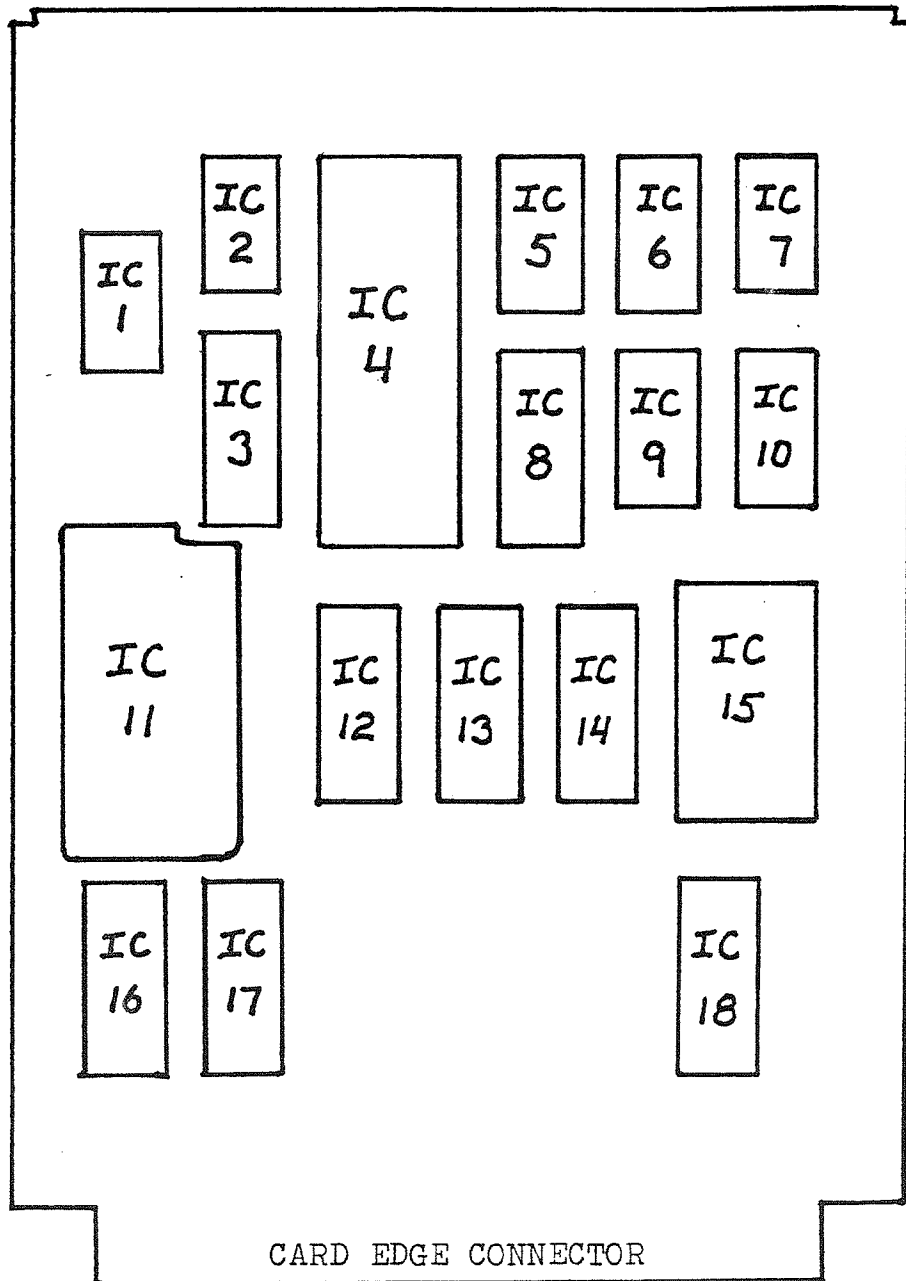
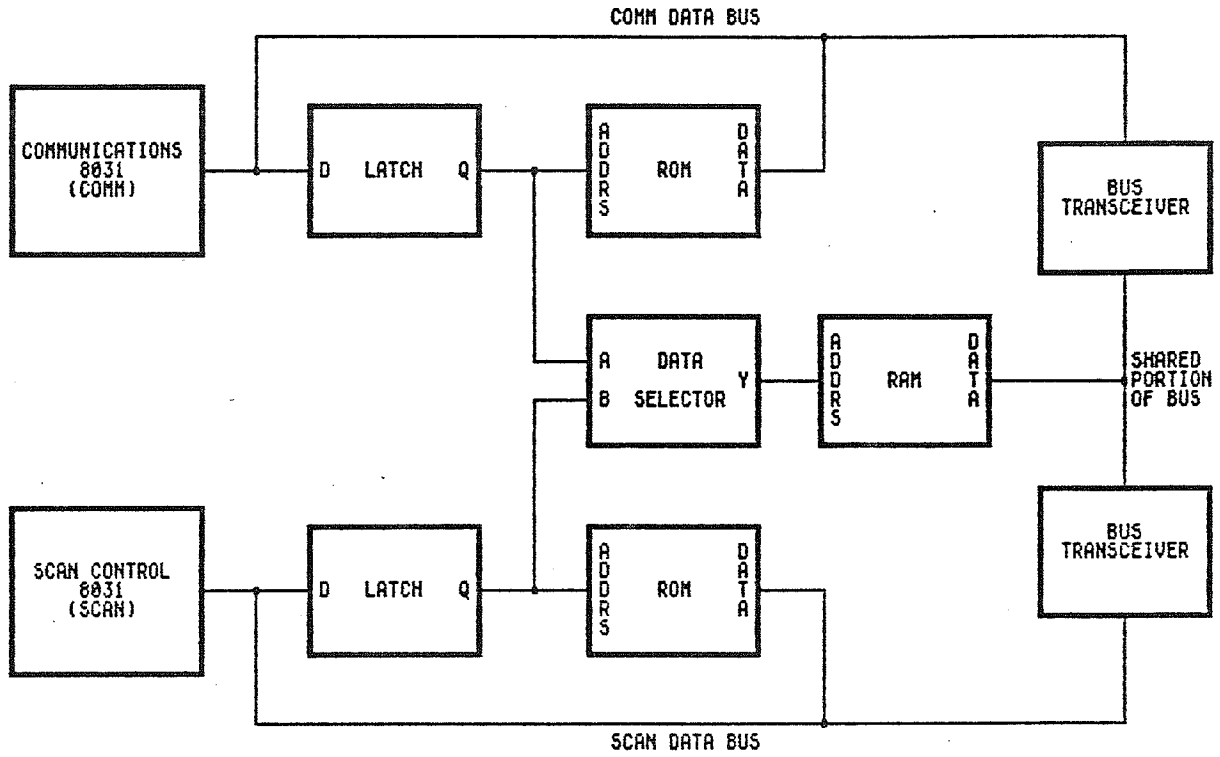


Figure 15. Communications CPU Board Component Layout (Component side of board)



NOTE: CONTROL LINES ( $\overline{P1.7}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ) NOT SHOWN.

Figure 16. External RAM Shared Access System Block Diagram

both 8031s must implement a system that prevents simultaneous access attempts from being made. This is discussed in the next chapter.

Sharing the same RAM also means the two 8031s must share a portion of their data buses (the RAM data lines). This is accomplished without bus conflicts by placing a tri-state bus transceiver between each data bus and the RAM data lines, as shown in Fig. 16. Each 8031 controls the transceiver on its own side of the shared bus section. Normally, while the 8031s are running their programs out of their respective ROMs, the transceivers are disabled and the two data buses are effectively isolated from one another. When one of the 8031s accesses RAM for a read or write operation, it enables the appropriate transceiver and sets up the proper transceiver direction for the data transfer. The other transceiver remains disabled, maintaining isolation between the two data buses.

Each transceiver is controlled as follows: When either the  $\overline{RD}$  or  $\overline{WR}$  signal from the 8031 is asserted, the transceiver is enabled ( $\overline{EN} = \overline{RD} \cdot \overline{WR}$ ). The direction is controlled by the  $\overline{RD}$  signal, which is normally high in its inactive state. This means that transceiver direction is normally set up for a write operation until the  $\overline{RD}$  signal is taken low. Thus for a write operation, the assertion of  $\overline{WR}$  enables the transceiver, and the direction is already set up properly. For a read, the assertion of  $\overline{RD}$  both enables the transceiver and sets up the proper direction.

One additional aspect to this transceiver control should be noted. During a write operation, the data presented to RAM by the 8031 must remain valid until it is latched by RAM on the low-to-high transition of  $\overline{WR}$ . Since it is the assertion of  $\overline{WR}$  which enables the transceiver, the low-to-high transition will disable the transceiver and may cause the data to be removed before being latched. This possibility is avoided by simply delaying the  $\overline{WR}$  signal approximately one clock period, so that the transceiver remains enabled until just after the low-to-high transition of  $\overline{WR}$ . The delay is created by sending  $\overline{WR}$  through several buffers. The enabling of the transceiver when  $\overline{WR}$  is first asserted will also be delayed, but this does not pose a problem because the delay is relatively small. (The reader may wish to refer to the bus cycle timing diagrams for the 8031 in Intel's MCS-51 Family of Single Chip Microcomputers User's Manual, pp. 2-10, 2-11).

What follows is an explanation of how each line of the backplane is used in the new PAC (PAC2). This should provide most of the information necessary to understand the operation of the hardware.

LINE 1 - Logic Power (+5V)  
LINE 2 - Logic Power (+5V)  
LINE 3 - Logic Ground  
LINE 4 - Logic Ground  
  
LINE 5 - Serial Transmit (TxD)  
LINE 6 - Serial Receive (RxD)

Note: Both CPU boards are wired for serial communications. For this reason the RS-232 drivers must be removed from the scan control CPU board, as mentioned earlier.

LINE 7 - D3  
 LINE 8 - D7  
 LINE 9 - D2  
 LINE 10 - D6  
 LINE 11 - D1  
 LINE 12 - D5  
 LINE 13 - D0  
 LINE 14 - D4

Lines 7 through 14 are used to link the 8031 data buses between the two CPU boards. Port 0 of SCAN is wired directly to these lines on the scan control CPU board, while one bank of bus transceivers is between these lines and the RAM data lines on the communications CPU board.

LINE 15 -	SCAN A7, X(4)	MSB of X-coordinate
LINE 17 -	" A6, X(3)	
LINE 19 -	" A5, X(2)	
LINE 21 -	" A4, X(1)	
LINE 23 -	" A3, X(0)	LSB of X-coordinate
LINE 25 -	" A2, Y(2)	MSB of Y-coordinate
LINE 27 -	" A1, Y(1)	
LINE 29 -	" A0, Y(0)	LSB of Y-coordinate
LINE 24 -	" A11, Z(3)	MSB of Z-coordinate
LINE 26 -	" A10, Z(2)	
LINE 28 -	" A9, Z(1)	
LINE 30 -	" A8, Z(0)	LSB of Z-coordinate

The above lines are used to transfer X,Y,Z-coordinate data from the scan control CPU board to the other boards in the system, in the same manner as in the PAC1. They actually carry address bits A0-A11 of SCAN. Address bits A0-A10 are now sent over these lines to the communications CPU board, where the data selectors determine whether A0-A10 of SCAN or COMM will have access to RAM.

LINE 16 - COMM A15  
 LINE 18 - " A14  
 LINE 20 - " A13  
 LINE 22 - " A12

The four high-order address bits of COMM are used to control the output level of the amplifiers, as in the SONOTHERM 1000. A 4-bit value corresponding to the desired DC power supply voltage is sent over these lines to the control hardware in response to the "Voltage (V)" command from the central computer. This is done by COMM rather than SCAN, which represents one exception to the rule that COMM tends only to communications with the central computer. There is no connection to these lines on the scan control CPU board.

LINE 31 - SCAN  $\overline{WR}$  (active low)  
 LINE 32 - SCAN  $\overline{RD}$ , Data Strobe (active low)

Lines 31 and 32 carry SCAN's read and write signals to the communications CPU board, where they go to the data selectors along with the address bits A0-A10.  $\overline{RD}$  and  $\overline{WR}$  of COMM are not connected to these lines.  $\overline{RD}$  of SCAN is used as the "Data Strobe" for the other boards as in the PAC1.

LINE 33 - I/O Select (active low)

Line 33 carries the result of SCAN A15 NOR'ed with SCAN A14. This signal performs the "I/O Select" function for strobing coordinates out to the other boards, as in the PAC1. There is no connection to this line on the communications CPU board.



LINE 34 - Not Used  
LINE 35 - Interrupt Request 0

Line 35 is used by COMM to send an external interrupt request to SCAN's INT0 pin (P3.2). COMM's I/O line P3.4 is used to send the request when COMM must tell SCAN to do something. It does so by placing a command at a specified location in RAM, then sending the interrupt request. SCAN responds to the interrupt by retrieving the command from RAM and acting on it. This is explained more fully in the next chapter.

LINE 36 - Interrupt Request 1

Line 36 is used by COMM to send an external interrupt request to SCAN's INT1 pin (P3.3). COMM's I/O line P3.3 is used to send the request when COMM requires access to RAM. SCAN responds to the interrupt by configuring the data selectors accordingly. This is explained more fully in the next chapter. Line P3.3 is also used for a handshaking signal during the 8031 power-up routines, before any interrupts are enabled (see Chapter 6, Figs. 18 and 30).

LINE 37 - "EXCITE" Flag (SCAN P3.4)

Line 37 performs the same function as in the PAC1—it is set by SCAN to enable the excitation signal outputs. There is no connection to this line on the communications CPU board.

LINE 38 - "ERROR" Flag (SCAN P3.5)

Line 38 performs the same function as in the PAC1—it

is set by SCAN on the occurrence of any error. When set, it now sends an external interrupt request to COMM's INT0 pin (P3.2) in order to inform COMM that an error has occurred.

LINE 39-LINE 46 - Pl.0-Pl.7, SCAN and COMM

Lines 39 through 46 are used to link the Port 1 I/O lines of SCAN and COMM between the two CPU boards. Line 46 (Pl.7) is used by SCAN to send a control signal for the data selectors to the communications CPU board. COMM's Pl.7 pin is configured as an input (this is done by the software) for monitoring the state of this control signal. Line 45 (Pl.6) is used for a handshaking signal during the 8031 power-up routines (see Chapter 6, Figs. 18 and 30).

LINE 47 - Not Used

LINE 48 - Reset (active low)

Line 48, when taken low, performs a reset on both 8031s.

LINE 49 - Not Used

LINE 50 - Clock

Line 50 carries the clock signal generated on the clock/stacked array control board to the element driver boards, as in the PAC1. There is no connection to this line on either of the CPU boards.

LINE 51 - Not Available

LINE 52 - Not Available

LINE 53 - Auxiliary Ground (not used)

LINE 54 - Auxiliary Ground (not used)

LINE 55 - +12V

LINE 56 - -12V

## CHAPTER 6. SOFTWARE

### 6.1 Introduction

Although time constraints did not allow assembly code to be written for the phased array controller (PAC), the software flow for both CPUs has been identified. This chapter is an attempt to provide the reader with the information he would need to complete the software development. Most of this information is in the form of flow diagrams, which begin with Fig. 17. Several aspects of the software require some explanation, however, and are fully discussed in this chapter.

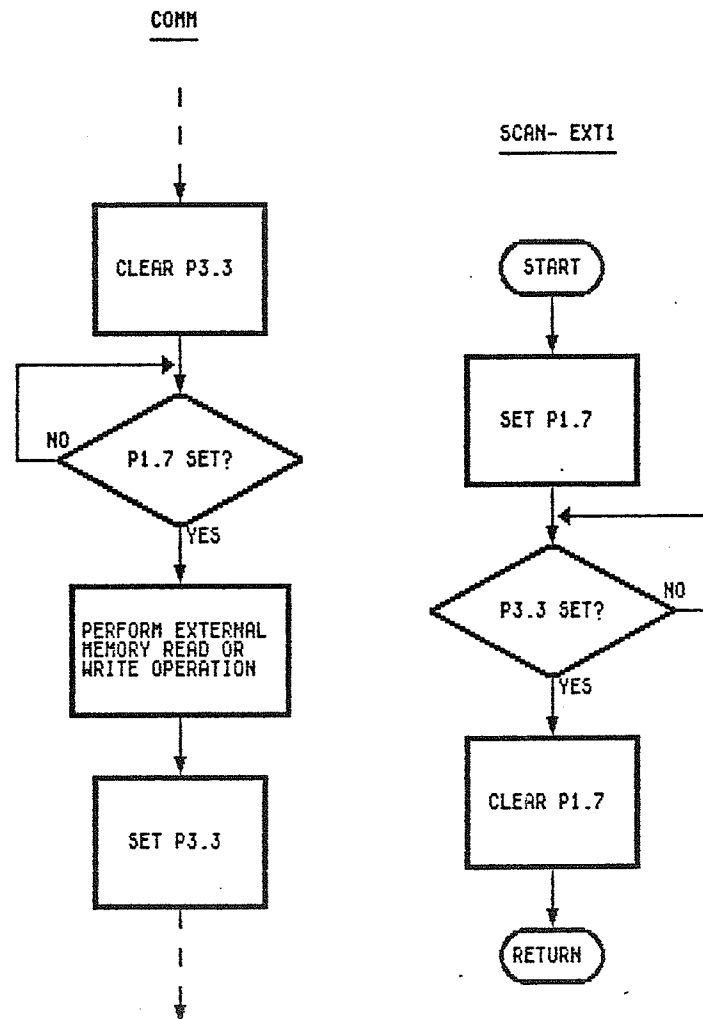
### 6.2 Sharing of External RAM and COMM/SCAN Communications

It was mentioned in the previous chapter that software must prevent both 8031s from trying to access RAM at the same time. This can be accomplished as follows: SCAN sets I/O line P1.7 so that SCAN normally has access to RAM itself. SCAN can therefore access RAM at any time. Whenever COMM requires access to RAM, it generates an external interrupt request at SCAN's INT1 pin (P3.3), using its own P3.3 I/O line. COMM then begins to poll the status of P1.7, the data selector control signal, waiting for access to be granted. SCAN's EXT1 interrupt routine gives COMM access to RAM by toggling P1.7. It then begins to poll the status of P3.3 (the interrupt request pin) waiting for COMM to remove the request. COMM does so immediately after completing its

external data transfer. SCAN then toggles Pl.7 again and returns from the interrupt routine. The flow of this sequence for both COMM and SCAN is shown in Fig. 17. It should be noted that this is not the only way to prevent the 8031s from attempting simultaneous external data transfers. Certain timing considerations appear to make it the best way, however. These will be discussed later in this chapter.

There is one exception to the scheme described above. Immediately following power-up (or a reset) all I/O lines are initialized to the high state, which gives COMM access to RAM. COMM performs its "self-check" while it has this access. SCAN takes Pl.7 low, giving itself access to RAM, as soon as COMM finishes its self-check. Figures 18 and 30 show flow diagrams for the power-up routines. (Flow diagrams for COMM are shown in Figs. 18-27, beginning on p. 59. Flow diagrams for SCAN are shown in Figs. 28-33, beginning on p. 72).

When COMM receives a command from the central computer, it must have a way to relay the command to SCAN, since most commands affect scanning in some way. The data which accompanies some of the commands must be relayed as well. Also, COMM must have access to a "status byte" that is maintained by SCAN. This exchange of information between the two 8031s can be accomplished through the shared external data memory (RAM), as follows: The two 8031s agree upon a specified location in RAM for each type of information that must be transferred between them. When COMM has a command



NOTE: THESE DIAGRAMS ASSUME THAT  
 P1.7 = 1 MAKES RAM ACCESSIBLE TO  
 COMM, AND P1.7 = 0 MAKES RAM  
 ACCESSIBLE TO SCAN.

Figure 17. Flow Diagrams Showing How COMM Accesses External RAM

that it must relay to SCAN, it places the code byte for the command at the RAM location specified for commands. If there is data accompanying the command, it is placed at the location specified for that type of data. (In the case of intensity data, an entire 1 kilobyte section of RAM is designated for intensity storage. This is divided into two sections—see Section 6.3). COMM then generates an external interrupt request at SCAN's INT0 pin (P3.2), using its own P3.4 I/O line (via bus line 35—see schematics, Figs. 12 and 14). SCAN's EXT0 interrupt routine retrieves the command, determines what it is, and acts accordingly. This may involve retrieving data, beginning a scan, stopping a scan, etc. SCAN maintains its status byte at a specified location in RAM at all times, and COMM can retrieve it without interrupting SCAN.

### 6.3 "Double Buffering" of Intensity Data Storage Space

Approximately every ten seconds during a scan, COMM receives a new list of 512 intensity factors (one for each standard block) from the central computer. COMM must store this list in RAM, and SCAN must refer to it each time it moves the focus to a new block. Since the transmission of intensity data occurs during a scan, it may be desirable to have two areas in RAM for intensity data—one for the data currently being used by SCAN, and another for the new data being stored by COMM. These two areas of RAM would then be "swapped" with each new set of intensity factors. This

"double buffering" of the intensity data storage space may prove to be unnecessary upon further study. It is, however, included in the software flow diagrams to show how it would be implemented.

#### 6.4 Command Overview

The fundamental aspects of the communications protocol, such as the command/reply structures and error detection, will be essentially the same as in the SONOTHERM 1000 (see Cargoni, 1985 for a detailed description of this protocol). However, some of the SONOTHERM 1000 commands must be used differently in the phased array system, or not used at all. There is also one new command required. What follows is a brief description of each command as it will be used in the phased array system.

NAME/STATUS (N) - Same function as in the SONOTHERM 1000—sent by the central computer to inquire about the PAC's status (ready to scan, scan in progress, etc.).

VOLTAGE (V) - Same function as in the SONOTHERM 1000—sent by the central computer, followed by a number from 0-8 corresponding to the desired voltage level for the amplifiers.

ORIGIN (O) - This is the new command, sent by the central computer followed by a number from 0-8, to inform the PAC which sub-field to use. It has a Class 4 structure (see Cargoni, 1985).

RECEIVE INTENSITIES (E) - This is the equivalent of the

"Receive Duty Cycles" command in the SONOTHERM 1000. It is used for sending the intensity factors for the standard blocks to the PAC. Since 512 bytes are required for the data, this command has a Class 7 structure.

INITIALIZE AND GO (I) - Used to initiate scanning after the first set of data is sent, or to reinitiate scanning after the "Wait" command. Class 2 structure.

WAIT (W) - Suspends scanning. Scan can be reinitiated with the "Initialize and Go" command. Class 2 structure.

RETRANSMIT (R) - Same function as in the SONOTHERM 1000—sent by either the central computer or the PAC to request a retransmission when an invalid character is received.

SHUTDOWN (S) - Same function as in the SONOTHERM 1000—sent by either the central computer or the PAC to initiate a complete system shutdown.

HELP (H) - This command may or may not be implemented in the phased array system. It would be sent by the PAC to the central computer in order to initiate a status check.

## 6.5 Communications Software

The communications software is divided into four major sections: the main program (MAIN), a serial interrupt routine (SERIAL), a timer interrupt routine (TIMER), and an external interrupt routine (EXT0).

MAIN performs a self-check on memory, initializes the 8031, and waits for an interrupt. SERIAL is the essence of



the communications software. It processes all the commands received from the central computer, performs error detection, sends replies back to the central computer, and relays commands and data to SCAN. TIMER performs the timing functions used in the SONOTHERM 1000—the line viability, expected retransmission, and triplet timers (see Cargoni, 1985). EXT0 responds to the setting of SCAN's "ERROR" flag by sending the "Shutdown" command to the central computer and then shutting down. (If the "Help" command is implemented it may be sent instead of the "Shutdown" command. The central computer would respond by sending the "Name/Status" command to find out what caused the error).

Flow diagrams for the communications software are presented in Figs. 18-27. Since the essential function of the communications software is to process commands received from the central computer, the following explanations of how COMM responds to each command is provided as a supplement to the flow diagrams. The general flow of serial reception processing is shown in Fig. 20.

NAME/STATUS (N) - SCAN maintains a status byte at a specified location in RAM which indicates its status at all times. The following statuses are possible:

- 1) Awaiting Data (origin factor, intensity factors, or both)
- 2) Ready to Scan (awaiting the "Initialize and Go" command)
- 3) Scan in Progress

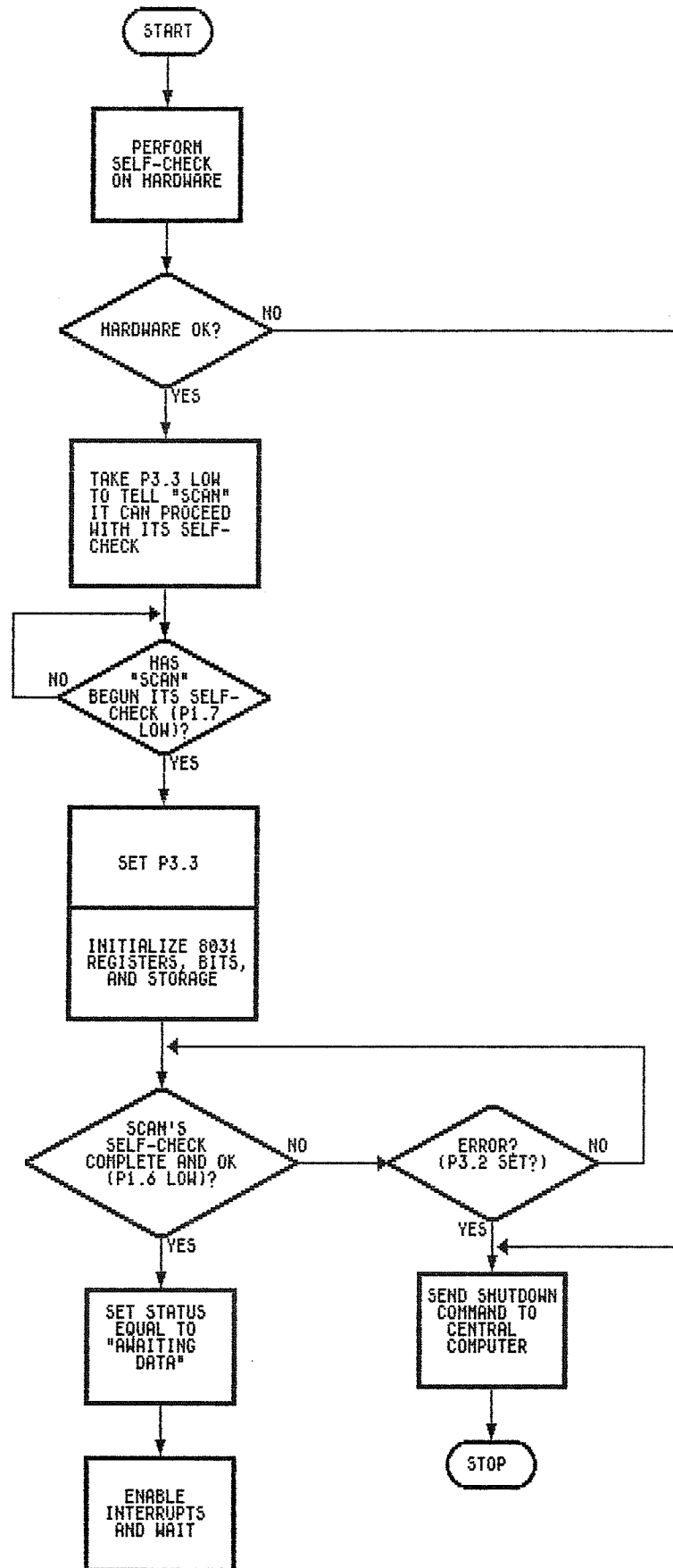


Figure 18: Flow Diagram for COMM—Main Program (Including power-up routine)

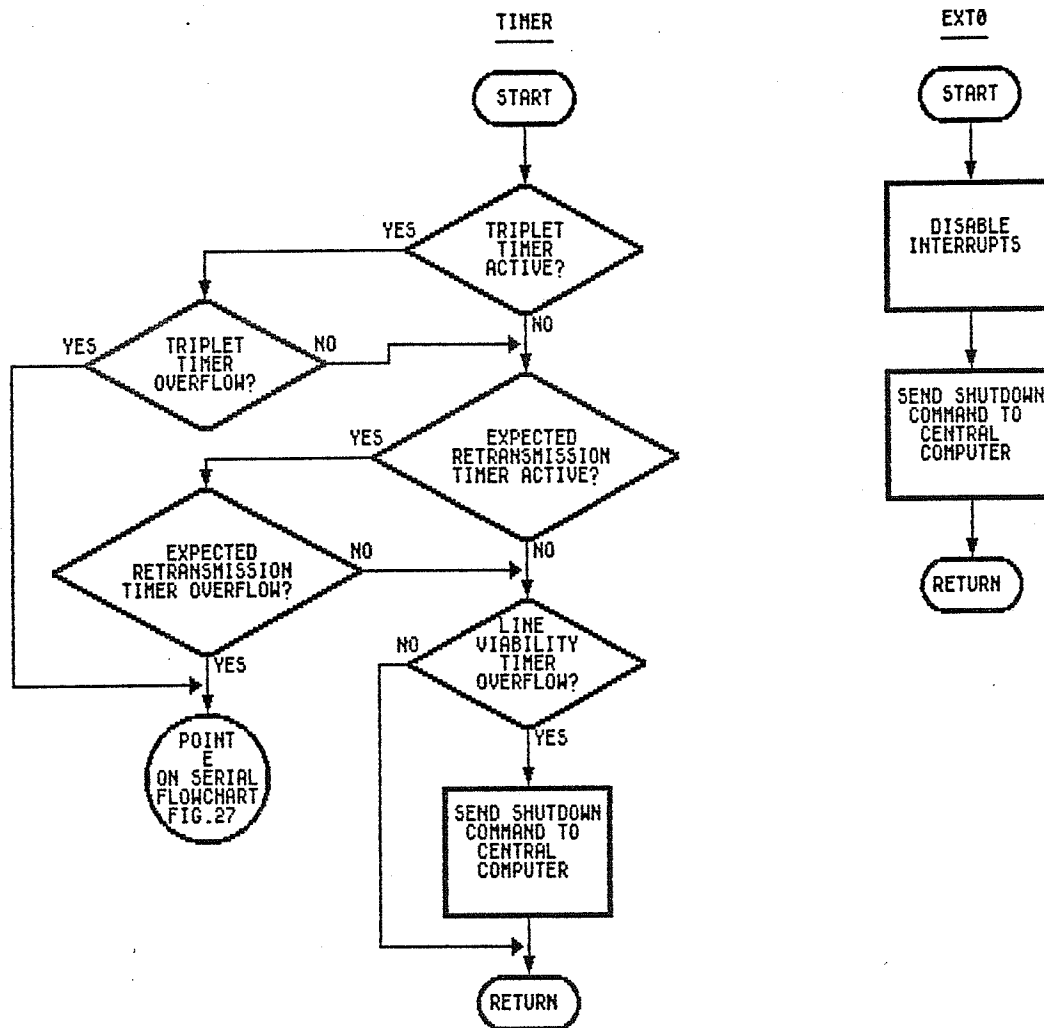


Figure 19. Flow Diagrams for COMM—TIMER and EXT0 Interrupt Routines

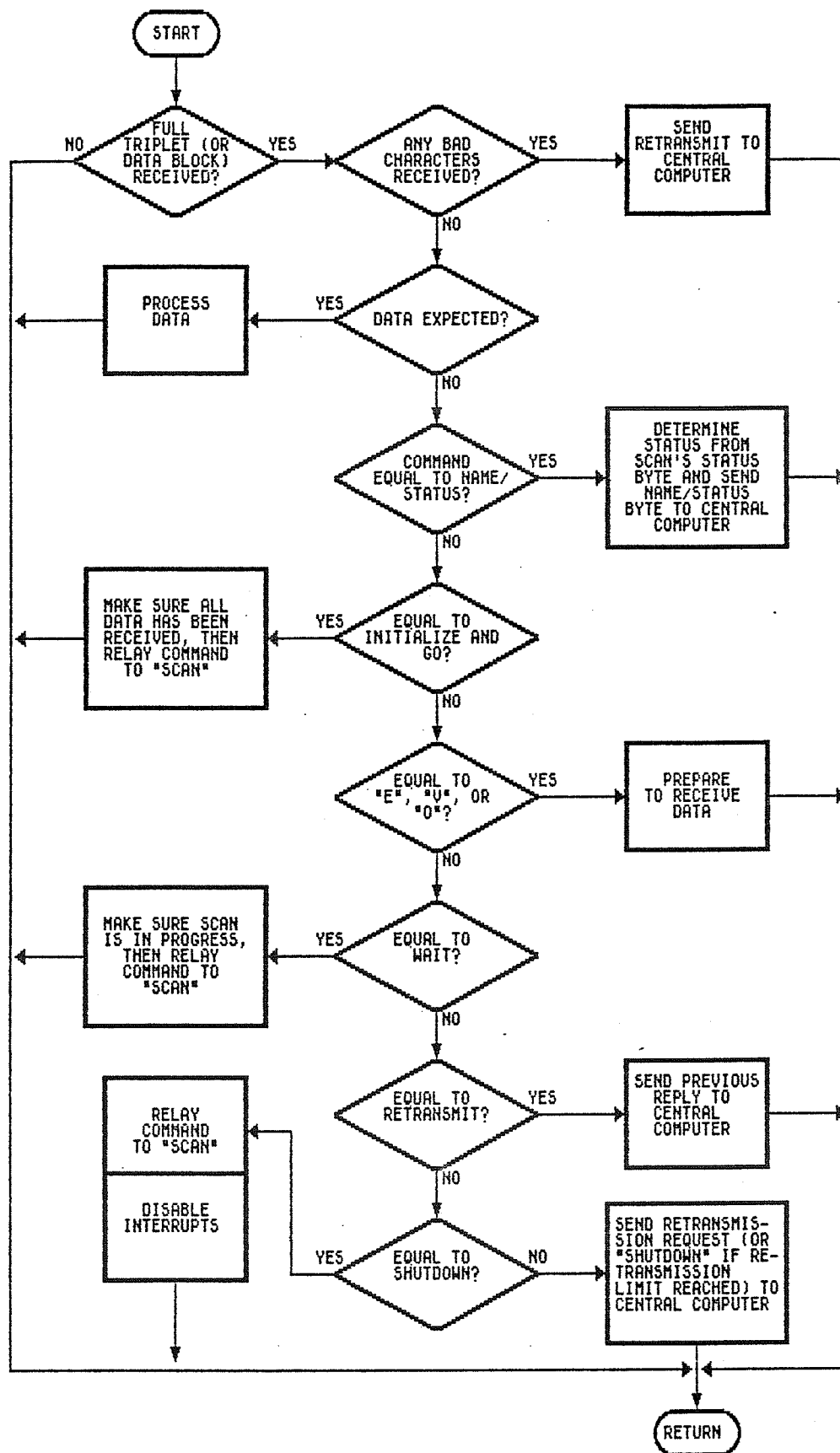


Figure 20. Flow Diagram for COMM—General Flow of Serial Reception Processing

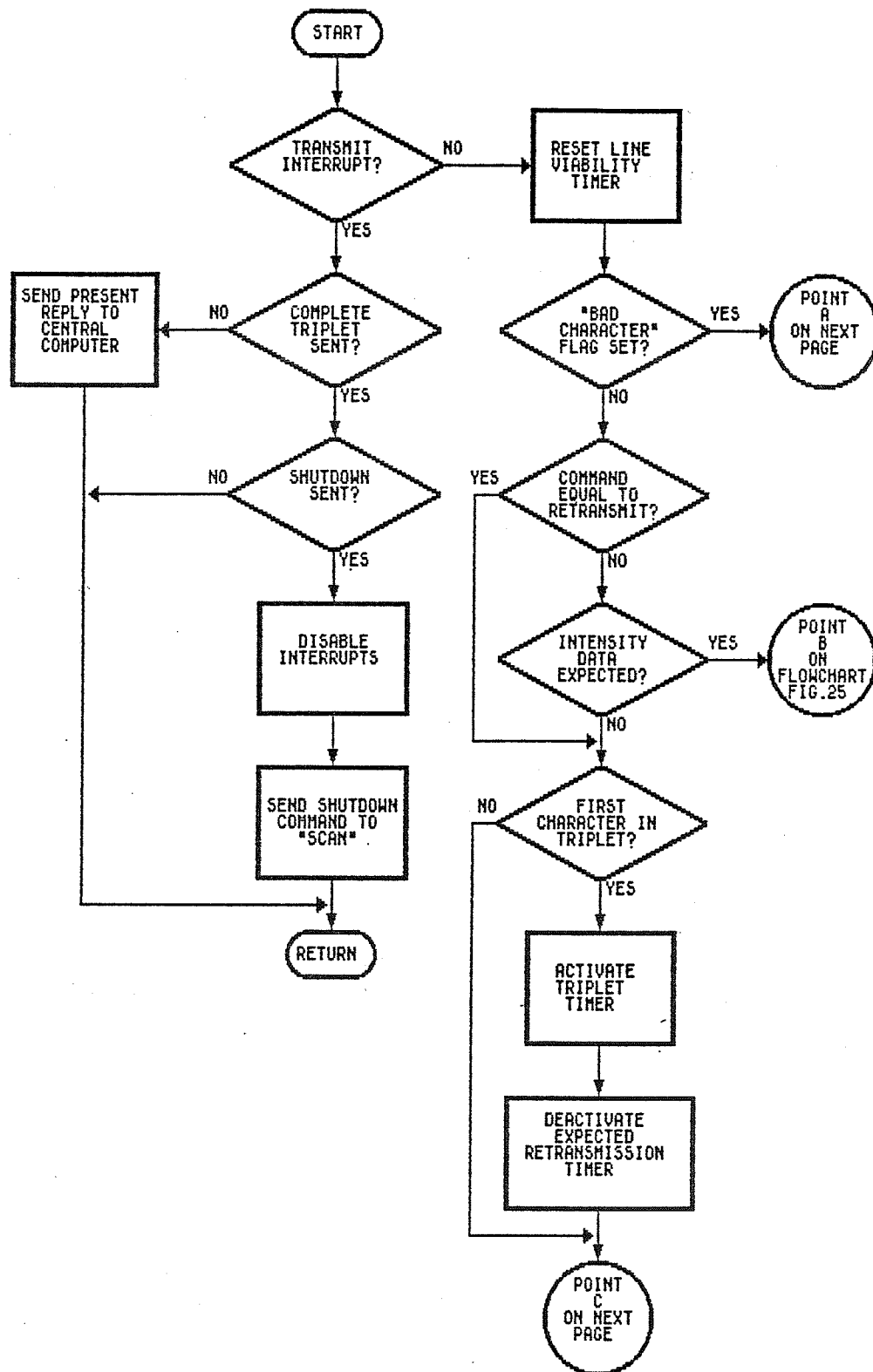


Figure 21. Flow Diagram for COMM—SERIAL Interrupt Routine

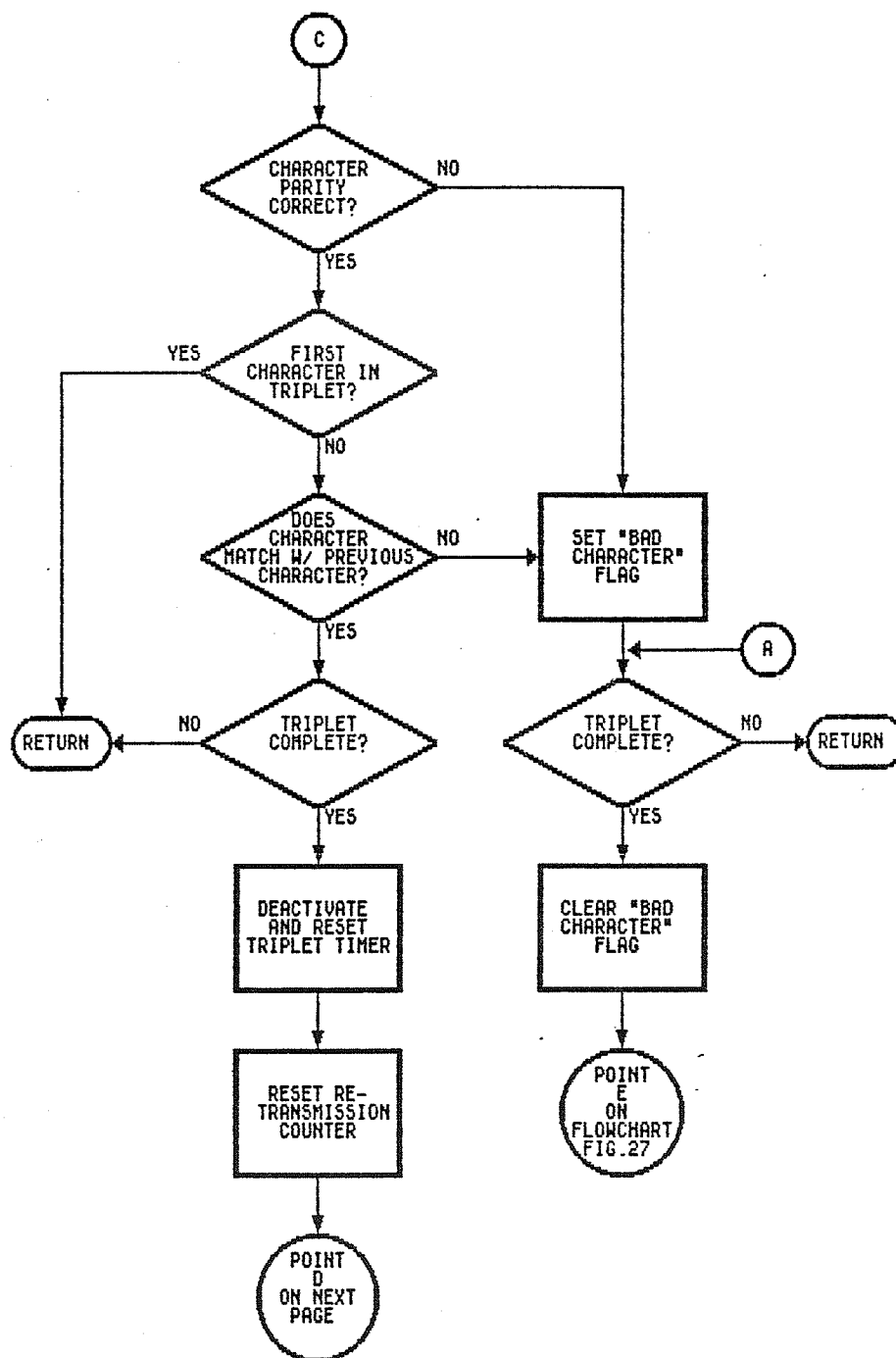


Figure 22. Flow Diagram for COMM—SERIAL (Cont.)

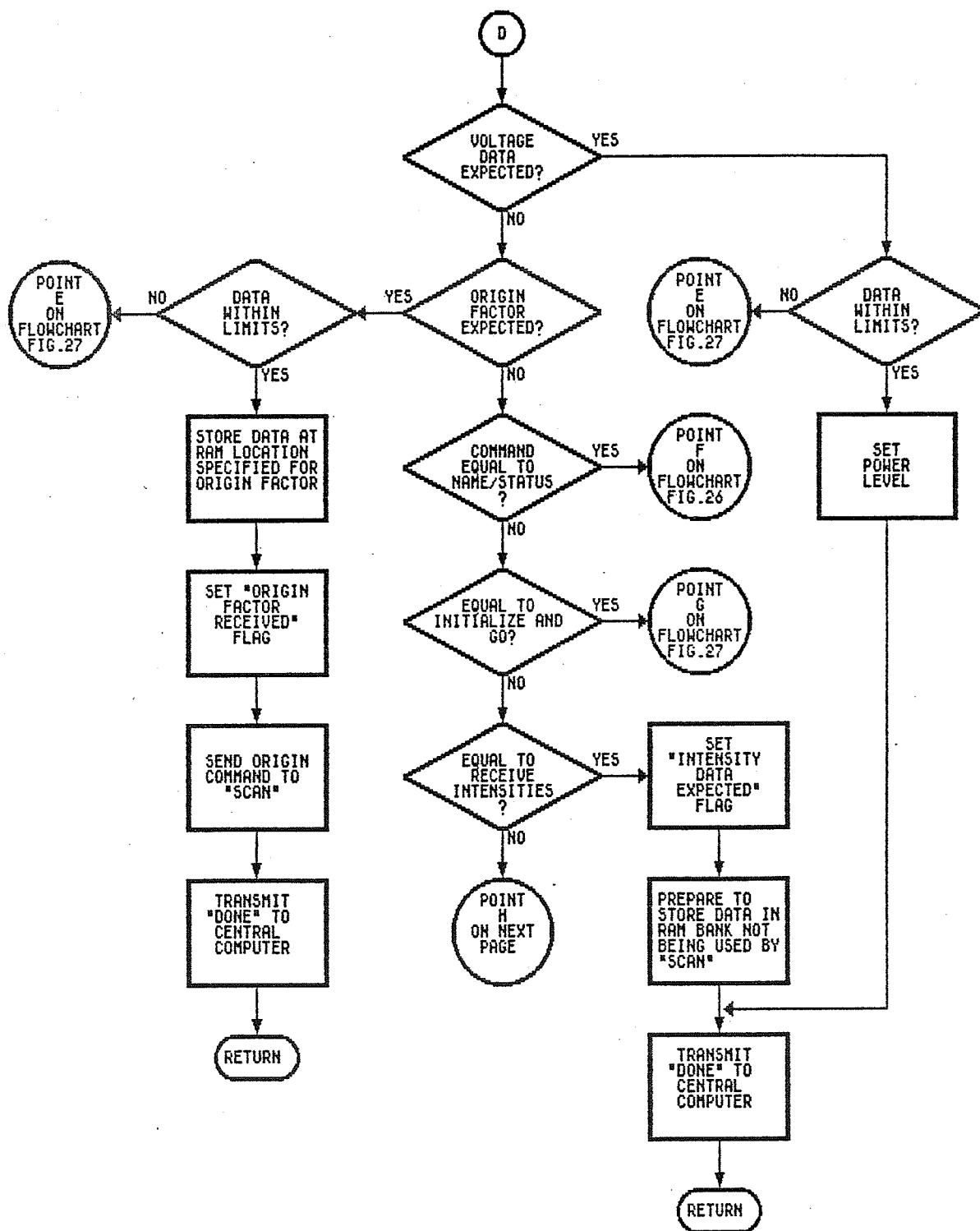


Figure 23. Flow Diagram for COMM—SERIAL (Cont.)

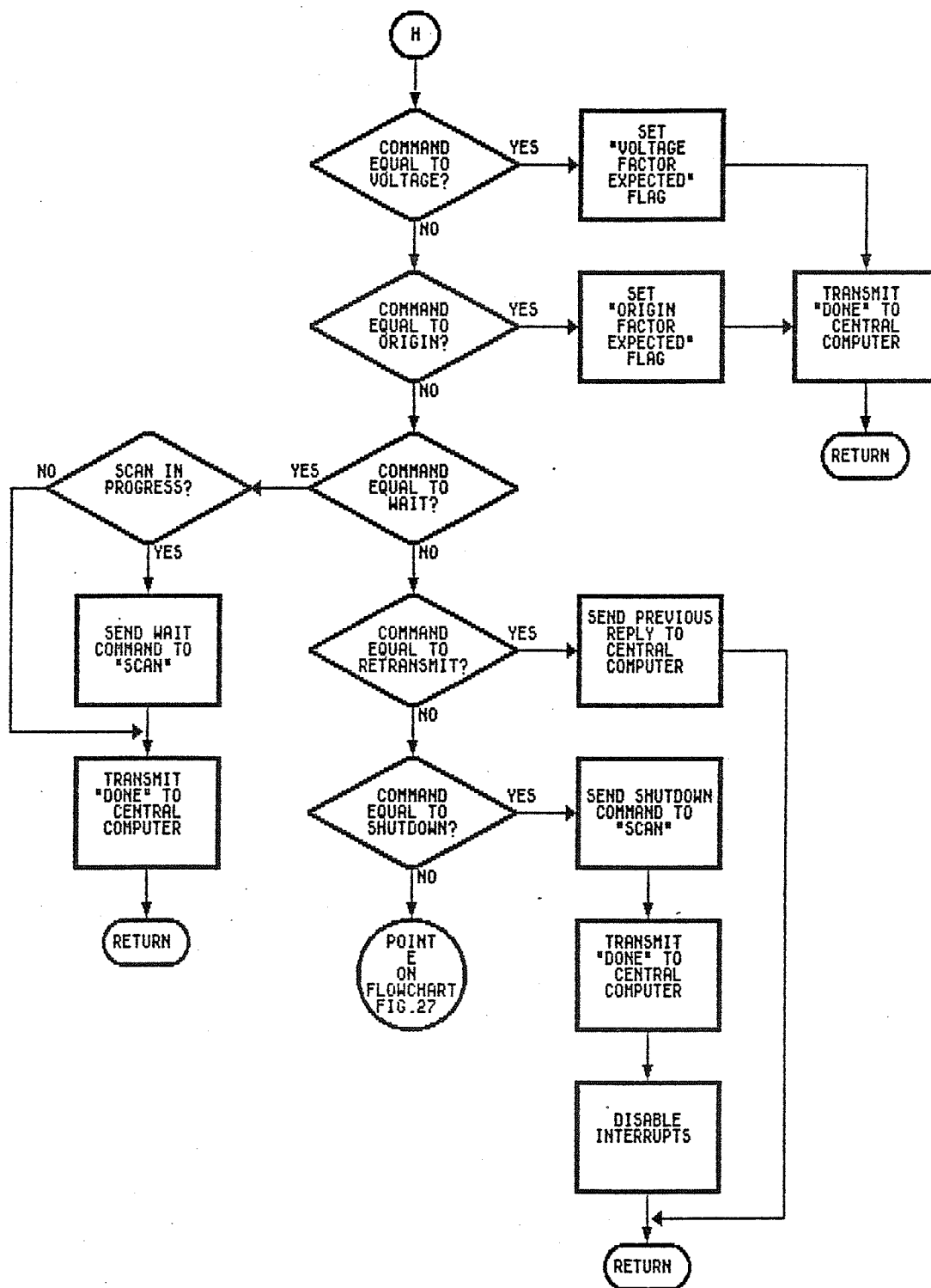


Figure 24. Flow Diagram for COMM-SERIAL (Cont.)



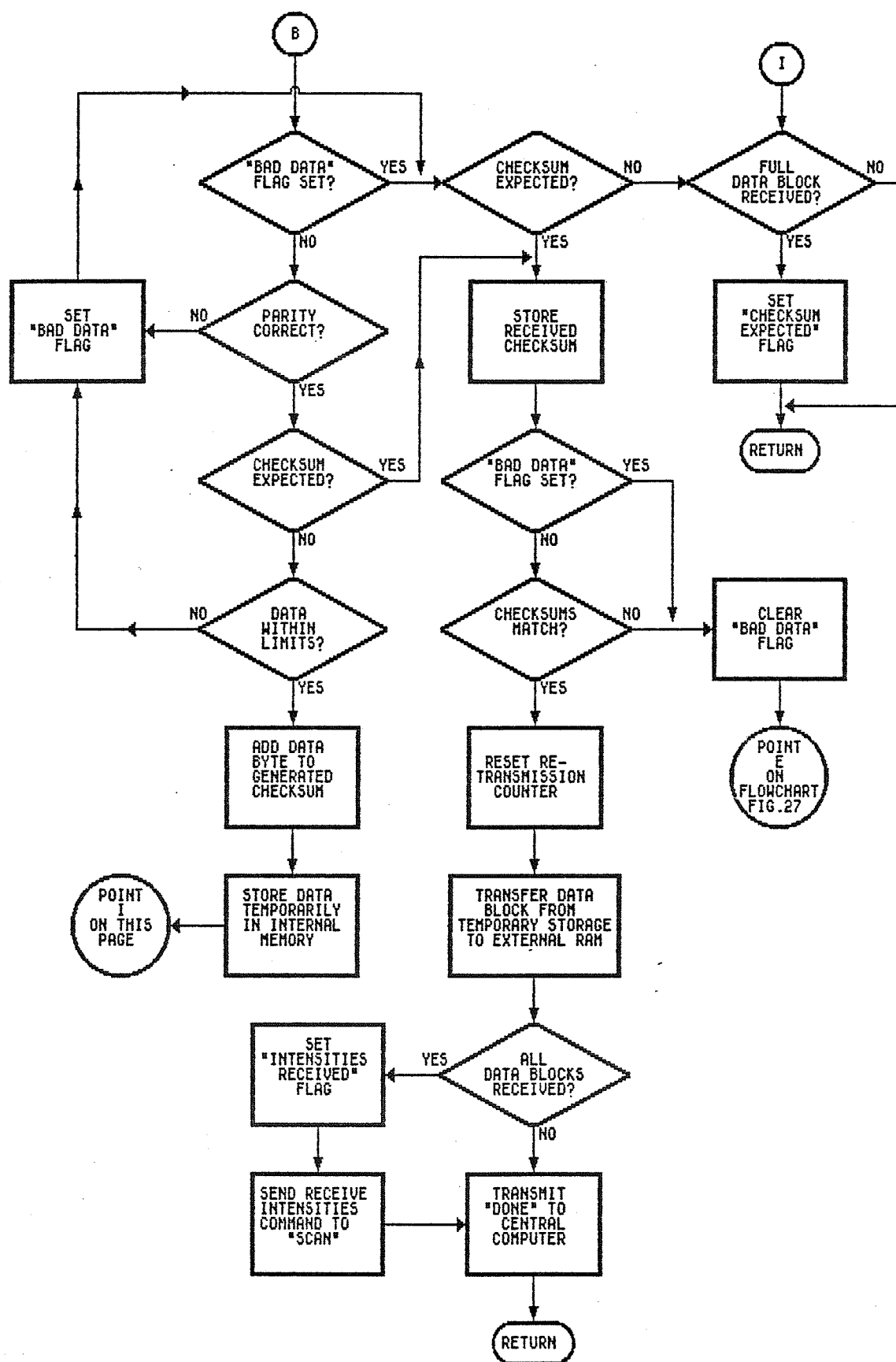


Figure 25. Flow Diagram for COMM—SERIAL (Cont.)

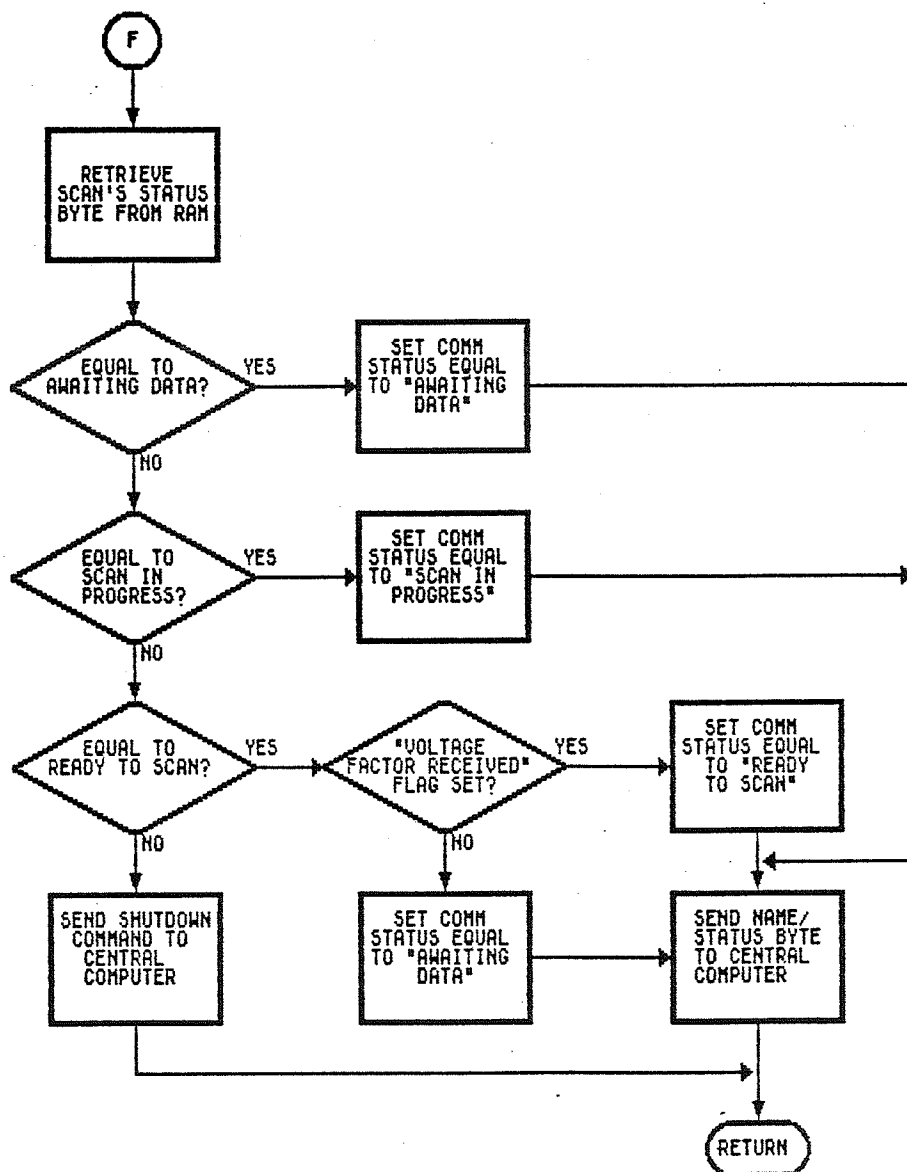


Figure 26. Flow Diagram for COMM—SERIAL (Cont.)

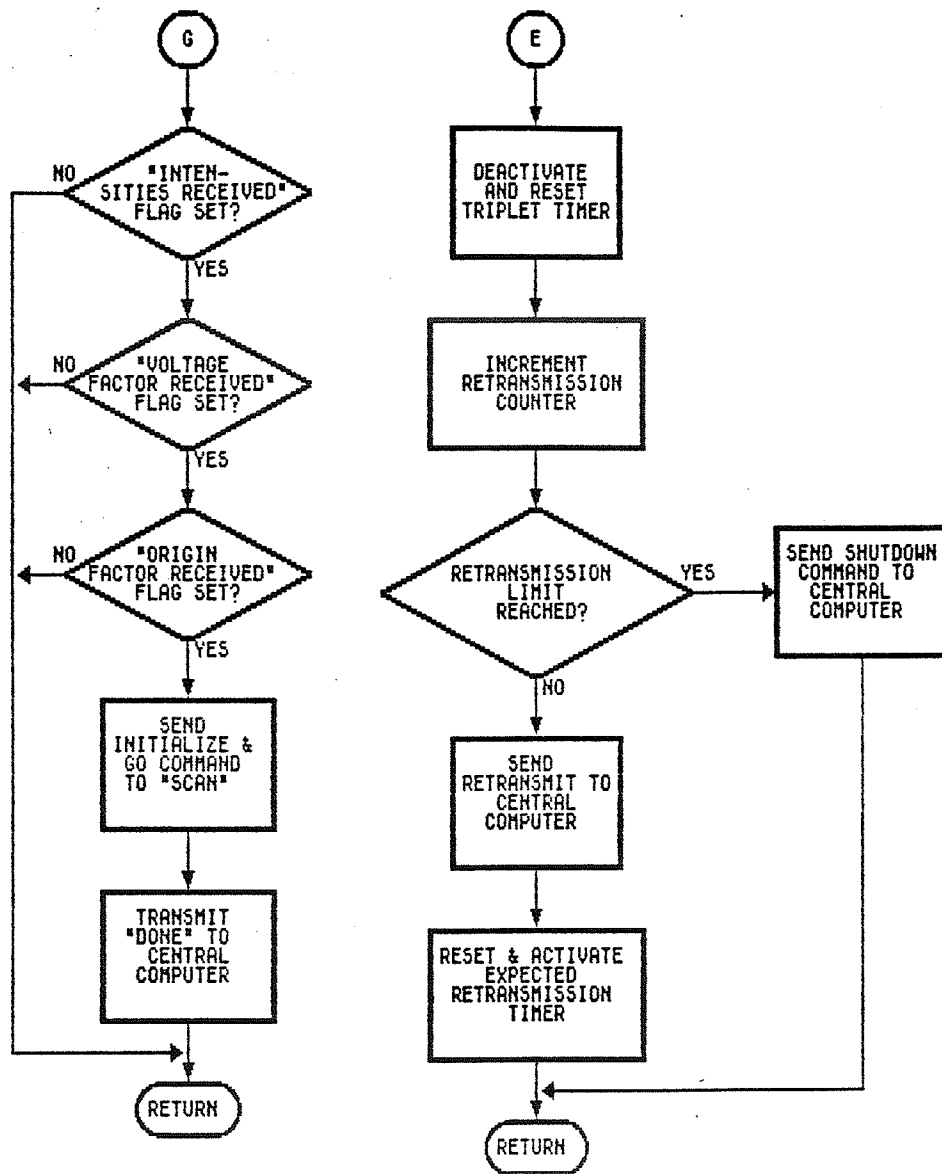


Figure 27. Flow Diagram for COMM—SERIAL (Cont.)

(No "Error" status is required, since setting the "ERROR" flag generates an interrupt in COMM immediately and shutdown occurs. If the "Help" command is implemented, different types of errors may be encoded in the status byte and the decision to shut down left to the central computer). When COMM gets the "N" command, it retrieves SCAN's status byte. If SCAN's status is either (1) or (3) above, COMM sends this status in the form of a "Name/Status" byte to the central computer. If SCAN is awaiting the "I" command, COMM must check to be sure that a voltage factor has been received before it sends this status. If the voltage factor has not yet been received, COMM sends the "Awaiting Data" status. If SCAN's status matches none of the three above, an error condition is assumed, and the "Shutdown" command is sent to the central computer instead of a Name/Status byte.

VOLTAGE (V) - COMM sets a flag indicating that a voltage factor is expected and sends the "Done" reply to the central computer. When the factor is received, it is sent to the power supply control circuitry in the same manner as in the SONOTHERM 1000. The "Done" reply is again sent to the central computer.

ORIGIN (O) - COMM sets a flag indicating that an origin factor is expected and sends the "Done" reply to the central computer. When the factor is received, it is stored at a specified location in external RAM. COMM then relays the "O" command to SCAN as described earlier and again sends the

"Done" reply to the central computer.

RECEIVE INTENSITIES (E) - COMM sets a flag indicating that intensity factors are expected, prepares to store the data in one of two banks of external RAM designated for intensity data storage (it uses whichever bank it did not use for the previous set of intensity factors, as mentioned earlier), and sends the "Done" reply to the central computer. The data are broken up into thirty-two data blocks, each consisting of sixteen data bytes and two checksum bytes (see Cargoni, 1985). COMM stores the data bytes as they arrive in its internal memory and generates a checksum, until a full data block is received. If the received checksum for the data block matches the generated checksum, the data block is transferred to external RAM and the "Done" reply is sent to the central computer. This is repeated for each data block until all the data are stored in external RAM. COMM then relays the "E" command to SCAN.

INITIALIZE AND GO (I) - COMM checks to be sure that all data required for scanning—voltage, origin, and intensity factors—have been received. If not, the "I" command is ignored. Otherwise, the command is relayed to SCAN and the "Done" reply sent to the central computer. (SCAN also checks to be sure it has the data it needs before it begins scanning, so that there is a double-check to prevent erroneous scanning).

WAIT (W) - COMM checks SCAN's status. If it is "Scan In Progress," the "W" command is relayed and the "Done" reply

sent to the central computer. Otherwise the "W" command is ignored.

RETRANSMIT (R) - COMM sends its previous reply to the central computer. It should be noted that sequence number control of retransmission requests is not included in the flow diagrams. (Cargoni, 1985 contains all the information necessary to implement this feature).

SHUTDOWN (S) - COMM disables its interrupts and relays the "S" command to SCAN.

#### 6.6 Scan Control Software

The scan control software consists of four major sections: the main program (MAIN), a timer interrupt routine (TIMER), and two external interrupt routines (EXT0 AND EXT1). MAIN does all the "bookkeeping" for scan control—calculating focal positions during a scan, keeping track of the standard blocks, etc. TIMER performs the same function as TINT in the PAC1—it establishes a time base for scanning. It controls the duty-cycle modulation of array output (which determines time-averaged intensity) and it determines when it is time to move to a new focal position. EXT0 is the "command center" of the scan control software. It retrieves commands placed in external RAM by COMM and implements them. EXT1 gives COMM access to RAM, as discussed earlier.

Flow diagrams for the scan control software are shown in Figs. 28-33. Figure 28 illustrates the general flow of the

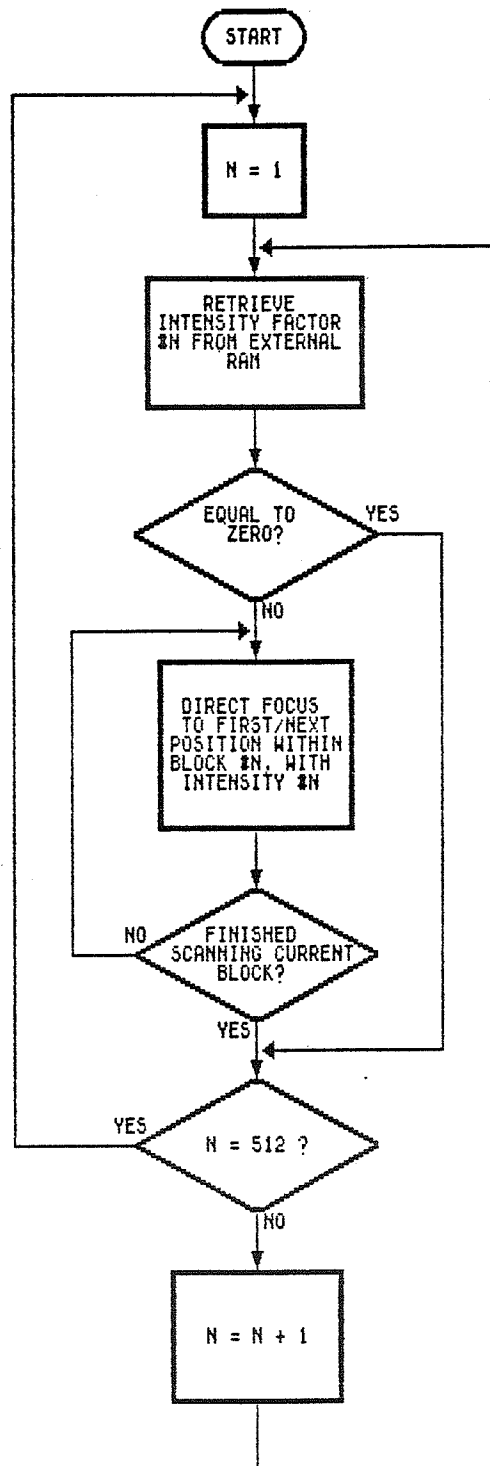


Figure 28. Flow Diagram for SCAN—General Flow of Scan Algorithm

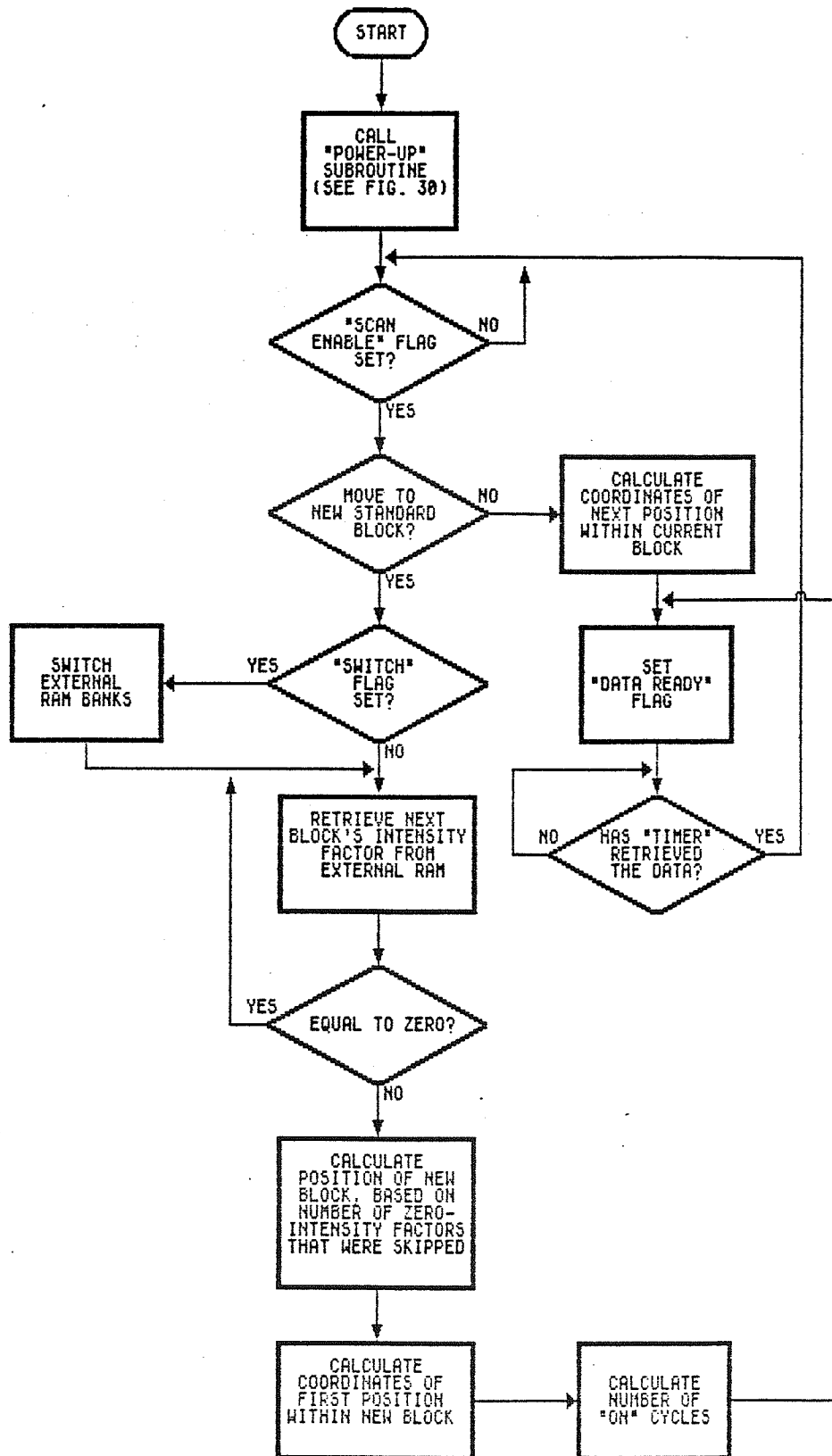


Figure 29. Flow Diagram for SCAN—Main Program (MAIN)



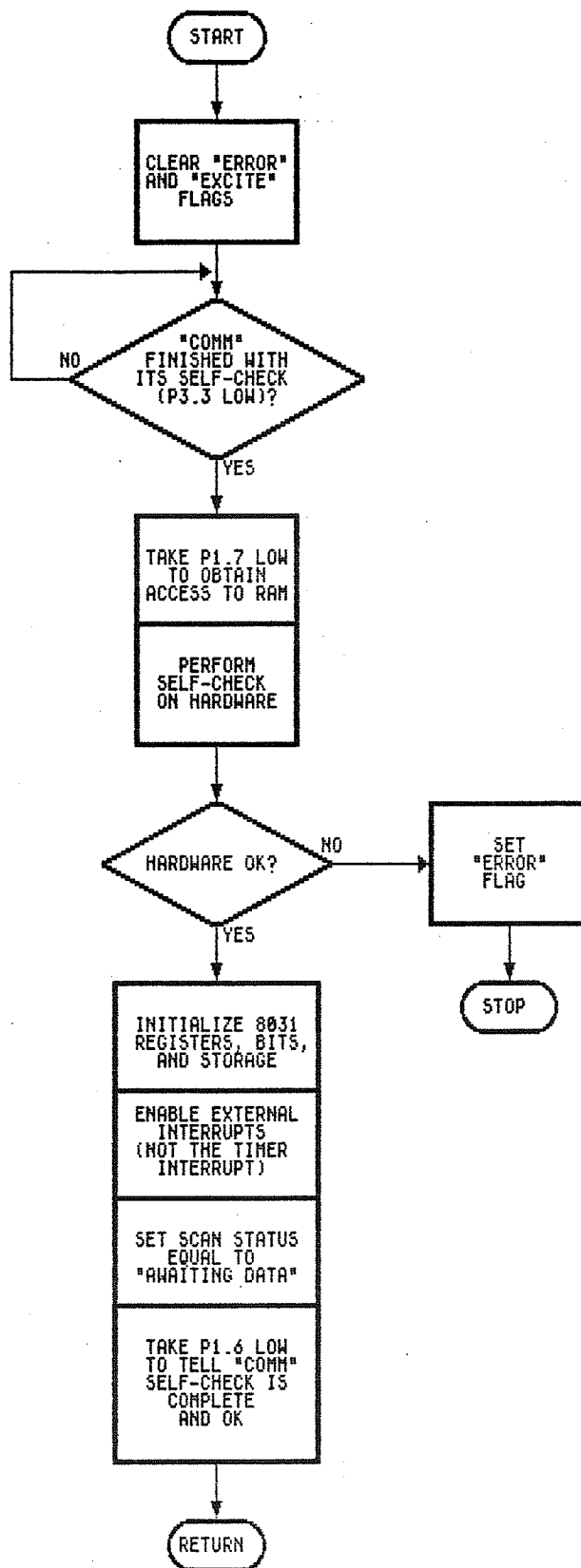


Figure 30. Flow Diagram for SCAN—"Power-Up" Subroutine

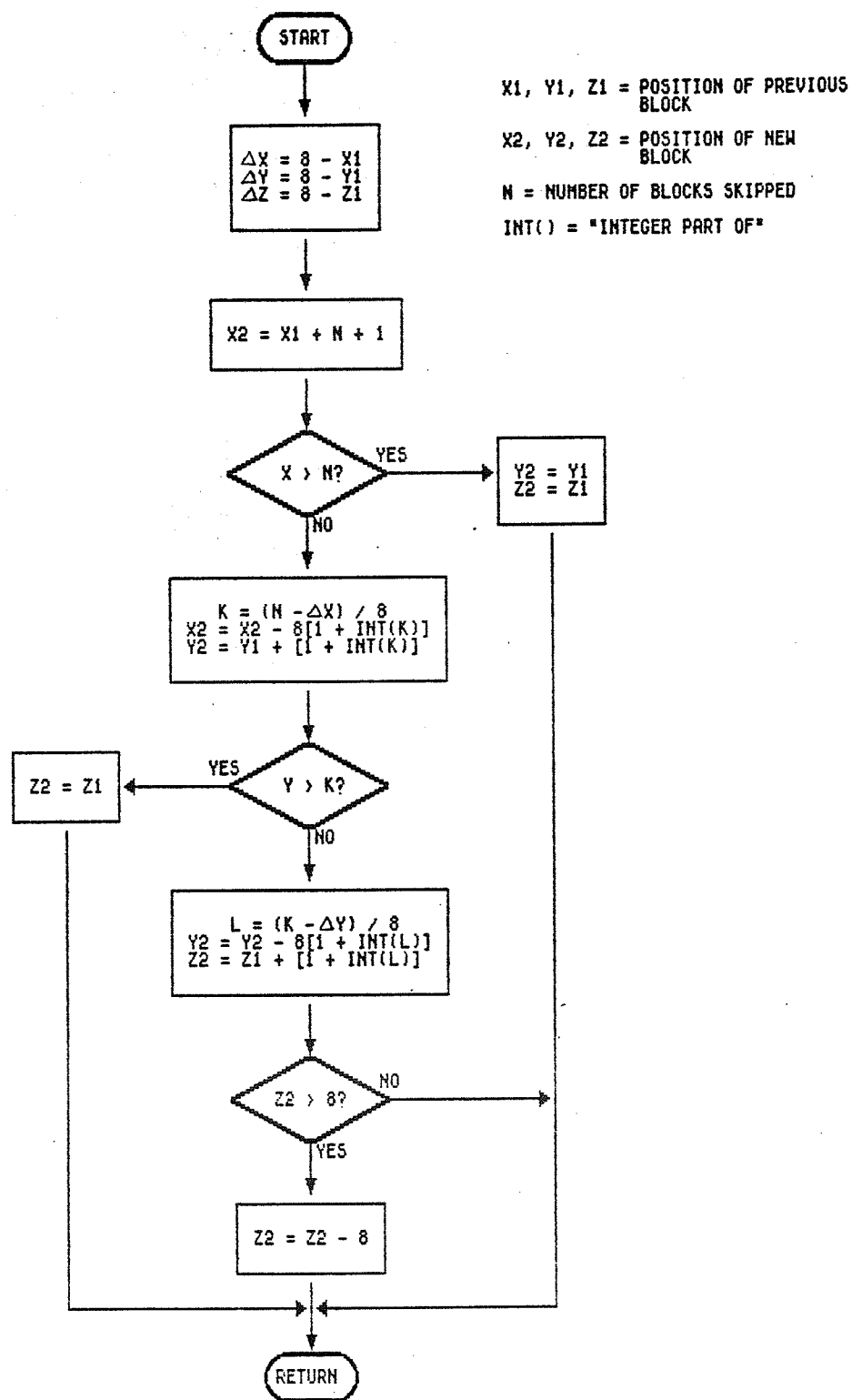


Figure 31. Flow Diagram for SCAN—Algorithm for Calculation of New Block Position

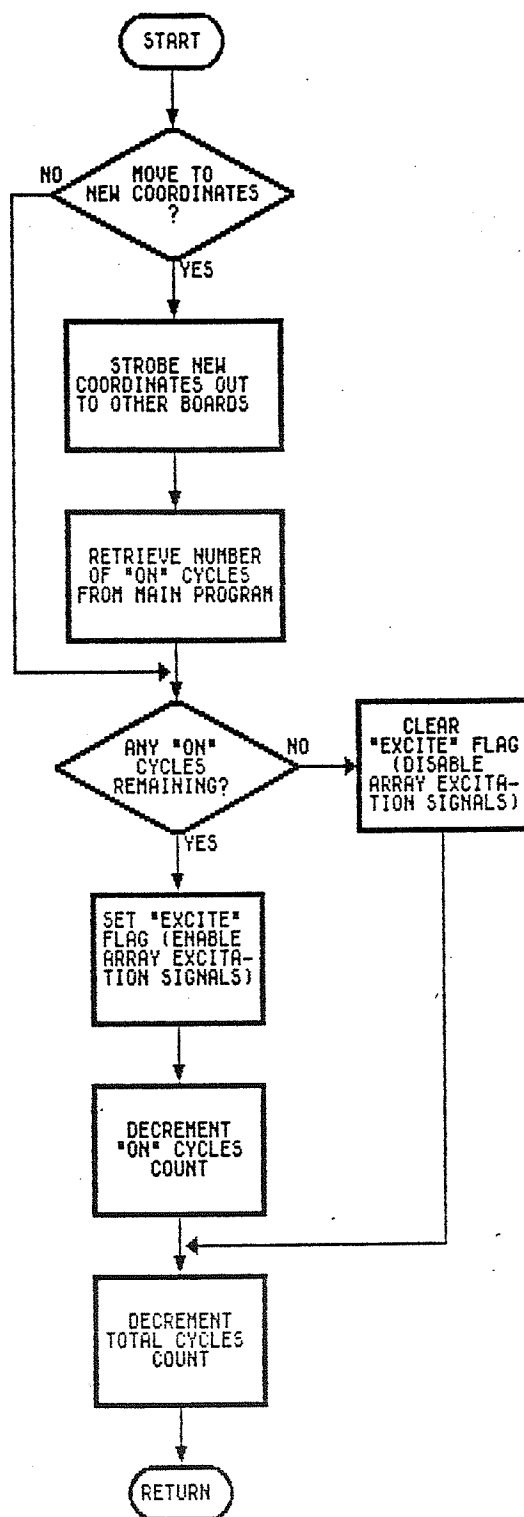


Figure 32. Flow Diagram for SCAN—TIMER Interrupt Routine

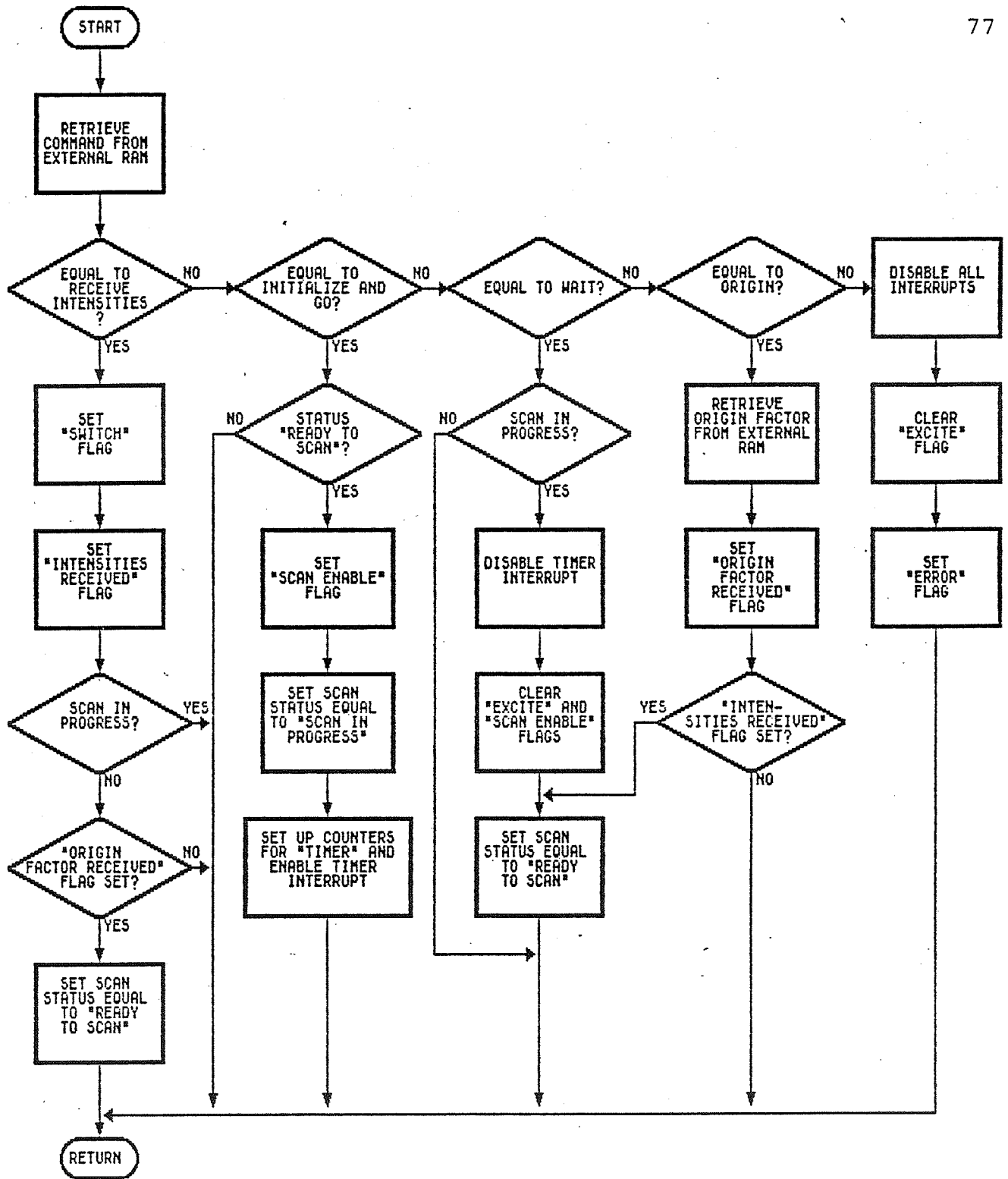


Figure 33. Flow Diagram for SCAN—EXT0 Interrupt Routine

scan algorithm. The controller scans the standard blocks sequentially, skipping over those having a zero intensity factor. Recall that SCAN knows before it begins scanning how the standard block is defined—this is part of the initialization routine, which will be discussed shortly. The following discussions of each major section of the scan control software are provided as an aid to understanding the flow diagrams.

#### 6.6.1 Main Program (MAIN)

MAIN (see Fig. 29) begins by calling the "Power-Up" subroutine, which performs a self-check on memory and data and address lines. If any errors are detected the "ERROR" flag is set and SCAN "hangs." Otherwise SCAN begins to initialize its registers, flags, etc.

Part of the initialization process involves defining the parameters of the standard block. Two constant parameters must be defined in each dimension—the number of focal positions comprising the block in that dimension, and the desired spacing of the positions that will actually be focused at. When SCAN scans a standard block, it begins at the "origin position" of the block (the position with the smallest coordinate values, i.e., closest to coordinates 0,0,0). It then "ratchets" through the block row by row, plane by plane, until each dimension is covered with the specified spacing. (If the block is defined as described in Chapter 4, this entire process simply involves incrementing

the X-coordinate by one, three times).

When initialization is complete, the external interrupts are enabled, SCAN status is set equal to "Awaiting Data," and SCAN informs COMM that its self-check is complete and OK. Flow then returns to MAIN, which waits for a "SCAN ENABLE" flag to be set before proceeding. This flag is set by the EXT0 interrupt routine after all the data required for scanning have been received, followed by the "Initialize and Go" command. MAIN then proceeds to determine the first focal position.

When a new focal position requires moving to a new standard block, which of course is the case for the very first position, MAIN must know the intensity factor of the new block. The blocks are sequenced within the sub-field in the same manner as the positions within each block—starting at the corner with the smallest coordinates and going row by row, plane by plane. MAIN retrieves the intensity factor for the next block in this sequence from one of two areas in RAM (recall that each time a new list of intensities is received SCAN and COMM swap the areas they use). If the intensity factor is zero, MAIN starts going through RAM until it finds a non-zero intensity factor. This may require checking a large number of intensity factors—especially if the tumor being treated is much smaller than the sub-field, in which case many blocks will be unused.

Once MAIN has a non-zero intensity factor, it must determine the position of the block to which this factor

belongs. It may be the very next block or it may be in another row and another plane. The information MAIN must use to determine the new block's position is the position of the previous block, the number of blocks in sequence that must be skipped, and the total number of blocks in each dimension of the sub-field.

Figure 31 shows a flow diagram of an algorithm that will determine the position of the new block, given the above information. For purposes of this algorithm, each block is assigned a set of X,Y,Z-"position numbers." (The term "position number" is used instead of "coordinate value" to avoid confusion with actual focus coordinates). These position numbers range from 1 to 8 in each dimension (recall that the sub-field is 8x8x8 blocks), so that the positions of the first and last blocks in sequence are (1,1,1) and (8,8,8), respectively. X1, Y1 and Z1 are used to represent the previous block's position numbers, and X2, Y2 and Z2 the new block's (unknown) position numbers. N represents the number of blocks which are to be skipped (the number of consecutive zeros found in RAM).

Once MAIN has determined the position of the new standard block it must calculate the coordinates of that block's "origin position," the first position within that block to be focused at. This is a simple calculation: Let X2 be the X-position of the new block that has just been determined, and let "XORIG" be the X-coordinate of the new block's origin position. Let "XDIM" be the number of focal

positions comprising the standard block in the X-dimension (defined equal to four in Chapter 4). Then

$$XORIG = XDIM (X2 - 1).$$

This equation applies to the Y-coordinate as well. For the Z-coordinate, the origin factor (a number from 0-8) is added to the right-hand side. This is how sub-field placement is implemented.

With the new coordinates ready to be sent to the other boards in the system, the last remaining calculation to be done is to translate the new intensity factor into the proper duty cycle information required by the timer interrupt routine. In the PAC1 a total of sixty-four timer cycles were spent at each position, and the fraction of these that were "ON" cycles was determined by both the "relative" and "overall" intensity factors. Since "overall" intensity factors are no longer used (the voltage factor is used instead), only the relative intensity factor will determine the number of "ON" cycles.

One-hundred rather than sixty-four cycles will be spent at each position. The intensity factors will range from 0-10 (although "ON" cycles are never calculated for the 0 factor) and will represent increments of 10%. Thus the number of "ON" cycles will simply be equal to the intensity factor times ten.

When MAIN finishes its calculations, it sets a flag to indicate that the coordinate and duty cycle data are ready for the timer interrupt routine. It then waits for TIMER to



retrieve the data, at which time it loops back to begin calculating the next set of focal coordinates.

#### 6.6.2 Timer Interrupt Routine (TIMER)

TIMER (see Fig. 32) is executed each time TIMER0 overflows while the interrupt is enabled. TIMER0 must be the only high-priority interrupt source in order to establish a fixed time base for scanning. The timer overflow rate is determined in software. TIMER keeps a running count of how many timer cycles are spent at each focal position ("total cycles count"). Every one-hundred cycles it strobes a new set of coordinates out to the other boards and activates the array (by setting the "EXCITE" flag). It also keeps track of "ON" cycles ("ON cycles count"), and deactivates the array at the proper time for duty cycle intensity control.

There are a number of crucial timing considerations which should be noted at this point. These are discussed in the following section.

#### 6.6.3 Timing Considerations

The amount of time spent at each focal position is equal to one-hundred times the period between timer overflows, which is determined in software. In order to heat a tumor effectively, each focal position must be heated frequently enough to avoid "thermal ripple" at that location. This places an important restriction on the amount of time that can be spent at each point, and care should be taken in determining the timer overflow rate.

Once it is decided how much time will be spent at each focal position, another timing restriction must be observed. In order to avoid a period of dead time between successive focal positions during a scan, MAIN must have a new set of coordinates and an "ON" cycles value ready for TIMER every one-hundred timer cycles. Recall that MAIN can be interrupted by any of three interrupt sources. The maximum time available to MAIN for performing its calculations is therefore determined as follows: Begin with the amount of time spent at each position, which is one-hundred times the timer period. Subtract the amount of time required for TIMER to execute one-hundred times. Then subtract the maximum amount of time that the two external interrupt routines will require (this will have to be estimated). The time remaining is the time available to MAIN for calculating the next set of coordinates and the "ON"-cycles value. Several suggestions for satisfying this timing requirement are provided in the next chapter to assist those who will translate the flow diagrams presented here into actual software for the array controller.

One further timing consideration should be mentioned. Scanning is initiated by enabling the timer interrupt and setting the "SCAN ENABLE" flag, which informs MAIN it can go ahead and start calculating coordinates. When TIMER is executed for the first time, MAIN will not have had a chance to calculate the first set of coordinates, since it normally has one-hundred timer cycles to do so. The counters used by

TIMER must therefore be initialized so that TIMER performs one-hundred "OFF" cycles without strobing any coordinates out to the other boards. This will give MAIN a chance to get the first set of coordinates ready.

#### 6.6.4 EXT0 Interrupt Routine

The flow diagram for this routine appears in Fig. 33. EXT0 retrieves the command placed in external RAM by COMM, determines what it is, and if it is valid, implements it. The commands are implemented as follows:

E - A flag is toggled to indicate that SCAN should switch memory banks the next time it retrieves an intensity factor. The status byte is then changed to "Ready to Scan," unless a scan is already in progress or the origin factor has not yet been received.

I - This command initiates scanning. The "SCAN ENABLE" flag is set to inform MAIN it can proceed to calculate coordinates. The status byte is changed to "Scan In Progress," and finally the timer interrupt is enabled. The "I" command is ignored if a scan is already in progress or if any information required for scanning has not yet been received.

W - This command suspends a scan without destroying any of its "bookkeeping" information, so that the scan can be easily reinitiated by the "I" command. The "SCAN ENABLE" and "EXCITE" flags are cleared, and the timer interrupt is disabled. The status byte is changed to "Ready to Scan."

The "W" command is ignored if a scan is not in progress.

0 - The origin factor is retrieved from its specified location in RAM and made available to the main program for its coordinate calculations. If intensity factors have already been received, the status byte is changed to "Ready to Scan."

If the command does not match any of these four, the "S" command is assumed. All interrupts are disabled, the "EXCITE" flag cleared, and the "ERROR" flag set.

#### 6.6.5 EXT1 Interrupt Routine

This routine was discussed in Section 6.2. Its flow diagram appears in Fig. 17.

## CHAPTER 7. SUMMARY AND FUTURE CONSIDERATIONS

### 7.1 Summary

This thesis dealt with several aspects of the development of a general purpose ultrasound phased array controller. The adaptation of the PAC1 for use with an ultrasonic stacked linear phased array was described. The design of additional hardware for automatic frequency selection was documented. Finally, a scheme for integrating the PAC into a complete, closed-loop hyperthermia system was proposed along with discussions of the additional hardware and software required to implement this scheme.

### 7.2 Future Considerations

In Chapter 2 it was mentioned that a single MOSFET might be used for each switch required to control the stacked array. Although the design of these switches is not part of this thesis, some considerations for implementing a MOSFET design are presented here.

The first restriction on the switch design is that one face of each array element must be grounded for purposes of electrical safety. This means that the switch must be located between the signal source and the array element, rather than between the element and ground. The result of this restriction is that the bias applied to the gate to control the switch must be with respect to the signal being switched rather than with respect to ground. A method of

offsetting the control signals provided by the PAC so that they provide the proper bias will therefore be required.

Since the signals which will drive the array elements contain no DC offset, the switches must be capable of switching both positive and negative voltages. This presents a problem for a single MOSFET, due to a parasitic diode that is typically placed between the FET's source and drain during manufacture. This diode prevents the FET from switching both polarities. While two FETs placed "back-to-back" would eliminate this problem, using two FETs for each element would be impractical for a large number of elements. One possibility to circumvent the problem with a minimum of additional hardware is to place an FET in the middle of a full-wave bridge rectifier (consisting of four diodes). This would allow a single FET to switch both polarities, because the signal would appear to the FET as having only positive polarity.

In Chapter 6, several timing considerations for the scan control software were discussed, one of which was concerned with the time available to the main program (MAIN) for calculating coordinate and "ON"-cycle values. Several suggestions for satisfying this timing requirement are now presented:

- 1) The timer period should be as long as possible without causing undesirable thermal ripple.

- 2) TIMER should be written as concisely as possible, with no unnecessary instructions. For example, pushing the

accumulator onto the stack at the beginning of the routine and then popping it off at the end requires four instruction cycles, whereas moving it to and from an unused register requires only two instruction cycles.

3) The portion of MAIN that will use up the most time is likely to be the loop that goes through RAM looking for a non-zero intensity factor, because a large number of consecutive zeros are possible. In fact, the worst-case condition (all 512 factors being zero) should be allowed for if possible. The loop must therefore be written very concisely. For example, incrementing a register to keep count of consecutive zeros could take much longer than using the addresses of the first and last zeros to calculate the number. Also, recall that SCAN has control over which 8031 has access to external RAM at any given time. SCAN normally maintains its own access, and COMM must request access from SCAN. The reason for setting up the shared access system this way can now be seen—SCAN can access RAM without any delays, thereby keeping the loop as short as possible.

## APPENDIX. MODIFICATION OF "LOOK-UP-TABLE GENERATOR" PROGRAM

It was mentioned in Chapter 2 that the dimensions of the coordinate-axis grid were changed from 16x16x16 to 32x8x16 (XxYxZ). This requires some minor modifications to the Apple-Basic "Look-Up-Table Generator" program (see Silverman, 1984). This appendix contains the modified program. Its file name is LUTG2 (short for Look-Up-Table Generator, version 2).



```
10 REM THIS APPLE II PROGRAM IS AN UPDATED VERSION OF THE ORIGINAL
12 REM LUTG PROGRAM. IT CONTAINS THE MODIFICATIONS NECESSARY TO
13 REM CREATE LOOK-UP-TABLES FOR A 32x8x16 COORDINATE GRID.
14 REM
15 REM
50 REM THE FOLLOWING PROGRAM CREATES THE LOOK-UP-TABLES FOR THE EPROMS
51 REM OF THE PHASED ARRAY CONTROLLER ON THE ELEMENT DRIVER BOARD(S).
52 REM THE PROGRAM CAN HANDLE UP TO 64 ELEMENTS. IT GENERATES A GROUP
53 REM OF BINARY FILES ON DISK WHICH CAN THEN BE USED TO BURN LOOK-UP-
54 REM TABLE EPROMS.
55 REM
56 REM THE FIRST PART OF THE PROGRAM PROMPTS THE USER FOR ALL
57 REM NECESSARY INFORMATION ABOUT THE ARRAY CONFIGURATION.
100 D$ = CHR$(4)
150 PRINT D$;"PR#3"
200 DIM F(8),Z(16),X(32)
300 INPUT "NUMBER OF ELEMENTS ? ";N
350 T = INT(N)
355 IF T < > N THEN GOTO 30300
360 IF N < 1 THEN GOTO 30300
365 IF N > 64 THEN GOTO 30300
400 INPUT "ELEMENT SPACING (MM) ?";S
450 IF S < 0 THEN GOTO 30400
455 IF S > 50 THEN GOTO 30400
500 FOR H = 0 TO 7
510 PRINT
600 PRINT "Y=";H;" CORRESPONDS TO ? (KHZ) "
700 INPUT F(H)
710 PRINT
750 IF F(H) < 400 THEN GOTO 30000
755 IF F(H) > 1000 THEN GOTO 30000
800 NEXT H
900 PRINT "X AXIS DISTANCES RELATIVE TO THE CENTER OF THE ARRAY"
910 PRINT
1000 FOR H = 0 TO 31
1010 PRINT
1100 PRINT "X=";H;" CORRESPONDS TO ? (MM)".
1200 INPUT X(H)
1210 PRINT
1250 IF X(H) < - 100 THEN GOTO 30100
1255 IF X(H) > 100 THEN GOTO 30100
1300 NEXT H
1500 PRINT "Z AXIS DISTANCES RELATIVE TO THE CENTER OF THE ARRAY"
1505 PRINT
1510 FOR H = 0 TO 15
1515 PRINT
1600 PRINT "Z=";H;" CORRESPONDS TO ? (MM)"
1700 INPUT Z(H)
1710 PRINT
1750 IF Z(H) < 50 THEN GOTO 30200
1755 IF Z(H) > 200 THEN GOTO 30200
1800 NEXT H
1900 HOME
1955 REM
1956 REM
```

```

1960 PRINT "NUMBER OF ELEMENTS IS ";N
1962 PRINT "ELEMENT SPACING IS ";S;" (MM.)"
1964 PRINT
2000 PRINT TAB( 10);"X (IN MM)"; TAB( 15);"Y (IN KHz)"; TAB( 15);"Z (IN
MM)"
2100 PRINT "-----"
-----
2200 FOR H = 0 TO 7
2300 PRINT H; TAB( 12);X(H); TAB( 23);F(H); TAB( 21);Z(H)
2400 NEXT H
2410 FOR H = 8 TO 9
2420 PRINT H; TAB( 12);X(H); TAB( 46);Z(H)
2430 NEXT H
2431 FOR H = 10 TO 15
2432 PRINT H; TAB( 11);X(H); TAB( 46);Z(H)
2433 NEXT H
2435 PRINT "CHECK THESE VALUES, THEN HIT RETURN TO SEE"
2436 INPUT "REMAINDER OF TABLE";A$
2440 FOR H = 16 TO 31
2450 PRINT H; TAB( 11);X(H)
2460 NEXT H
2500 INPUT "ARE THESE VALUES CORRECT ? (Y/N) ";A$
2600 IF A$ = "N" THEN GOTO 300
2700 IF A$ = "Y" THEN GOTO 2900
2800 GOTO 2500
2840 REM
2850 REM THE FOLLOWING SERIES OF NESTED LOOPS WILL GENERATE THE BINARY
2851 REM LOOK-UP-TABLE FILES, AND STORE THEM ON THE DISK. THE FILES
2852 REM WILL ALL BE NAMED "ELEMENTS" WITH A POSTSCRIPT INDICATING
2853 REM WHICH PAIR OF ELEMENTS THE INDIVIDUAL FILES ARE FOR.
2854 REM
2855 REM THE DATA FOR ONE ELEMENT IS STORED IN THE UPPER NIBBLE OF
2856 REM THE X,Y,AND Z COORDINATE VALUES. THE DATA FOR THE NEXT
2857 REM ELEMENT FOR THE SAME COORDINATES IS STORED IN THE LOWER
2858 REM NIBBLE OF THE SAME ADDRESS.
2859 REM
2860 REM THE EQUATION USED TO CALCULATE THE BINARY VALUES IS:
2861 REM  $\text{MOD } 16 < 16 * F(J) / 1500 * \text{SQRT} \langle (X(I) - X(E))^2 + Z(K)^2 \rangle >$ 
2862 REM WHERE (I,J,K) IS THE COORDINATE FOCUS, AND X(E) IS THE
2863 REM DISTANCE FROM THE ORIGIN TO THE CENTER OF ELEMENT # E.
2900 FOR H = 0 TO 7
3000 F(H) = F(H) * 16 / 1500
3010 NEXT H
3020 FOR H = 0 TO 15
3100 Z(H) = Z(H) ^ 2
3200 NEXT H
3205 REM OFFSET IS THE DISTANCE FROM THE ORIGIN TO THE CENTER
3206 REM OF ELEMENT NUMBER 0
3210 OFFSET = (N - 1) * S / 2
3250 FOR E = 0 TO N - 1 STEP 2
3260 PRINT "GENERATING LOOK-UP-TABLE FOR ELEMENTS ";E;"&";E + 1
3300 FOR I = 0 TO 31
3400 FOR J = 0 TO 7
3500 FOR K = 0 TO 15

```

```
4000 ARGUMENT = (X(I) + OFFSET - (E * S)) ^ 2 + Z(K)
4100 HIGHNIBBLE = F(J) * SQR (ARGUMENT)
4200 ARGUMENT = (X(I) + OFFSET - ((E + 1) * S)) ^ 2 + Z(K)
4300 LOWNIBBLE = F(J) * SQR (ARGUMENT)
4400 TEMP = LOWNIBBLE
4500 GOSUB 20000
4600 BUFFER = TEMP
4700 TEMP = HIGHNIBBLE
4800 GOSUB 20000
4900 BUFFER = (TEMP * 16) + BUFFER
4950 REM THE ADDRESS FOR DATA WITHIN THE LOOK-UP-TABLE IS
4952 REM OF THE FORM ZXY
5000 POKE 16384 + (256 * K) + (8 * I) + J,BUFFER
5100 NEXT K
5200 NEXT J
5300 NEXT I
5400 L$ = STR$ (E)
5500 L1$ = STR$ (E + 1)
5600 PRINT D$;"BSAVE ELEMENTS ";L$;"&";L1$;" ,A$4000,L$1000"
5700 NEXT E
19499 PRINT CHR$ (12); CHR$ (21)
19500 END
19900 REM THIS SUBROUTINE PERFORMS A MOD 16 FUNCTION ON THE
19901 REM CONTENTS OF TEMP. THE RESULTING 4 BIT VALUE IS RETURNED
19902 REM IN THE LOW NIBBLE OF TEMP.
20000 D = INT (TEMP / 16)
20100 TEMP = TEMP - (D * 16)
20200 TEMP = INT (TEMP)
20300 RETURN
30000 PRINT "VALUE MUST BE BETWEEN 400 AND 1000, TRY AGAIN"
30010 PRINT
30020 GOTO 600
30100 PRINT "VALUE MUST BE BETWEEN -100 AND 100, TRY AGAIN"
30110 PRINT
30120 GOTO 1100
30200 PRINT "VALUE MUST BE BETWEEN 50 AND 200, TRY AGAIN"
30210 PRINT
30220 GOTO 1600
30300 PRINT "PLEASE ENTER INTEGER VALUE BETWEEN 0 AND 64"
30310 PRINT
30320 GOTO 300
30400 PRINT "ELEMENT SPACING MUST BE BETWEEN 0 AND 50 MM. , TRY AGAIN"
30410 PRINT
30420 GOTO 400
```

## LIST OF REFERENCES

- Badger, C.W., private communication.
- Benkeser, P.J., Ocheltree, K.B., Frizzell, L.A., and Cain, C.A., "Ultrasound Phased Array Hyperthermia Applicator," Proc. IEEE EMBS Conf., Chicago, Sept. 27-30, 1985.
- Cargoni, R.A., "The Design of a Digital Communication Protocol for Use in a Modular Hyperthermia System," M.S. Thesis, University of Illinois, Urbana, Illinois, 1985.
- Hahn, G.M., "Hyperthermia for the Engineer: A Short Biological Primer," IEEE Trans. Biomed. Eng., vol. BME-31, pp. 3-8, 1984.
- Intel Corporation, MCS-51 Family of Single Chip Microcomputers User's Manual, Santa Clara, California, January, 1981.
- Intel Corporation, Microcontroller User's Manual, Santa Clara, California, May, 1982.
- Lele, P.P., "Induction of Deep, Local Hyperthermia by Ultrasound and Electromagnetic Fields," Radiation and Environmental Biophysics, 17, 205-217, 1980.
- Ocheltree, K.B., "Theoretical Analysis of Ultrasonic Phased Arrays for Hyperthermia Treatment," M.S. Thesis, University of Illinois, Urbana, Illinois, 1984.
- Ocheltree, K.B., Benkeser, P.J., Frizzell, L.A., and Cain, C.A., "An Ultrasonic Phased Array Applicator for Hyperthermia," IEEE Trans. Sonics Ultrasonics, vol. 31, pp. 526-531, 1984.
- Silverman, S.G., "The Design of an Ultrasound Phased Array Controller for Use in Hyperthermia," M.S. Thesis, University of Illinois, Urbana, Illinois, 1984.
- Texas Instruments Incorporated, The TTL Data Book for Design Engineers, Second Edition, Dallas, Texas, 1981.