SONAR SIGNAL SIMULATOR USING SDK-85 MICROCOMPUTER

by

Suk L. Chiu

A thesis
presented to the School of Graduate Studies and Research
of the University of Ottawa
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in
Electrical Engineering

OTTAWA, Canada, 1982

I hereby declare that I am the sole author of this thesis.

I authorize University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Suk L. Chiu

I further authorize University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Suk L. Chiu

ABSTRACT

A sonar signal simulator, based on the application of the SDK-85 microcomputer, has been developed to work as a training and test device with the current AN/SQS 505 Sonar System. It is entirely self contained. Development work consisted of both the programme and hardware development of the total package.

This thesis deals with a unified approach for the design and development. The mathematical principle for deriving the signal generation algorithm is presented. The software development includes a computational part for the generation of a target return signal. A control part is also included for input/output and timing control of the signal generation. Hardware design to interface the computer to the sonar system is also presented. A complete configuration is included.

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his thesis supervisor, Professor Willem Steenaart for his encouragement and guidance throughout this work.

Special thanks are also due to Dr Daniel Dubois for his work on the general problem analysis, Dr Mariusz Barski for his development of a major part of the program, and to Mohammed Master and Rowle Scott for their part in the design and construction of the hardware and in the overall system design and testing.

The financial assistance from the Department of Electrical Engineering of the University of Ottawa, and the Department of National Defense, is gratefully acknowledged.

CONTENTS

ABSTRACT iv

ACKNOWLEDGEMENTS v

Chapter page

I. INTRODUCTION 1

General 1
Problem Statement 3
Outline of Thesis 6

II. MATHEMATICAL DERIVATION OF SIGNAL GENERATION ALGORITHM 9

Introduction 9
Formulation of Simulation Algorithm 10
Assumptions 10
Reflection Time and Receive Time 13
Range 16
Doppler Effect 17
Sector 21
Half-Beam Delays 23
Amplitude of the Return Signal 25
Limitations 28

III. SOFTWARE DESIGN 31

Introduction 31
Computational Part for Signal Generation in Fortran 32
Fortran Subroutines 32
Generation of Input Numbers to the Hardware . 35
Sector Numbers 36
Time Delay Dividing Numbers 37
Frequency Dividing Number 39
Amplitude Numbers 40
Flowchart 40
Input/Output and Control Parts in 8085 Assembly Language 42
Programme Organization for Data Input/Output and Control 43
Repetition Time of Synchronization Pulses, REPTIM 48
Set Timer, SETTIM 49

Error Messages 50
Amplitude of Return Signal, AMP 51
Interrupts 53
Output, ECHO 56

IV. HARDWARE DESIGN 58

Introduction 58
Clock Circuit 61
Implementation of 10 Megahertz Frequency . . 64
Implementation of the Time Delays 66
Implementation of the Return Signal Frequency 70
Multiplying Digital-to-Analog Converter (MDAC) and Multiplexing Circuit 74
Implementation of the Multiplying Digital-to-Analog Converter (MDAC) 75
Implementation of the Multiplexing Circuit . 79
Implementation of Synchronization Pulses 82

V. CONCLUSIONS 83

Appendix page

A. OVERALL SCHEMATIC DIAGRAMS 85

B. HARDWARE INTERCONNECTIONS 89

C. LIST OF PARTS 99

D. OPERATION INSTRUCTIONS 102

E. COMPUTER PROGRAMMES 105

REFERENCES 131

LIST OF FIGURES

Figure page

1. Interface between the Simulator and the Sonar Receiver 8

2. Target Position relative to the Receiver 12

3. Frequency Modification for Moving Transmitter . . . 17

4. Frequency Modification for Moving Receiver 18

5. Diagram for Doppler Effect 20

6. Number Description of Transducer Staves 22

7. One Particular Sector for Half-Beam Delay 24

8. Diagram for the Amplitude of Return Signal 26

9. Delay from Synchronization Pulse to Output 35

10. Absolute Delays of Half Beams 38

11. Flowchart of Subroutine COMP2 41

12. Flowchart of the Structure of data Input/Output and Timing Control (Sheet 1 of 2) 46

13. Flowchart of the Structure of data Input/Output and Timing Control (Sheet 2 of 2) 47

14. Repetition Time of Synchronization Pulses 49

15. Configurations Explaining the Setting of Counts . . 50

16. Envelope Shape of Injected Signal 52

17. Timing Control of TSYNC and Timer Interrupts 55

18. Port Assignments for Output Data 57

19. Block Diagram of Hardware Design 60

20. Block Diagram of the Clock Circuit 63

21. Implementation of the 10 MHz Frequency Source . . . 65

22. Timing Diagram for the 10 MHz Frequency Source . . . 65

23. Implementation of the Time Delay of the Left Half Beam of the Even Sector 68

24. Timing Diagram for the Time Delay 69

25. Implementation of the Return Signal Frequency . . . 72

26. Timing Diagram for the Return Signal Frequency . . . 73

27. Block Diagram of Multiplying Digital-to-Analog Converter and Multiplexing Circuit 74

28. Implementation of the Multiplying Digital-to-Analog Converter (MDAC) 77

29. Implementation of the Injected Signal into the Sonar System 78

30. Block Diagram of the Multiplexing Circuit 80

31. Implementation of the Multiplexing Circuit 81

32. Implementation of Synchronization Pulses 82

33. Schematic Diagram of Clock Circuit (Sheet 1 of 2) . 86

34. Schematic Diagram of Clock Circuit (Sheet 2 of 2) . 87

35. Schematic Diagram of Multiplying D/A Converter and Multiplexing Circuit 88

36. IC Chips Layout (Sheet 1 of 3) 91

37. IC Chips Layout (Sheet 2 of 3) 92

38. IC Chips Layout (Sheet 3 of 3) 93

39. Interconnections between the Ribbon Cables and the Boards (Sheet 1 of 2) 94

40. Interconnections between the Ribbon Cables and the Boards (Sheet 2 of 2) 95

41. Interconnections between the Boards (Sheet 1 of 2) . 96

42. Interconnections between the Boards (Sheet 2 of 2) . 97

43. Front View of the Sonar Signal Simulator Package . . 98

Chapter I
INTRODUCTION

1.1 GENERAL

In order to provide training opportunities for sonar receiver operators and to evaluate receiver performance in the absence of actual sonar targets for active sonar systems, a sonar return signal simulator can be designed that will fulfill these requirements. It is the objective of this thesis to complete such a design, which is self contained, and is designed for one particular sonar system, the AN/SQS 505 system. The principles outlined in this design could however easily be adapted to supply simulated responses to other sonar receivers. A search of professional literature has not produced references to earlier work on sonar signal simulators. The work reported here is thus entirely the development of the author in conjunction with the members of the research team on this project as indicated in the acknowledgements.

The problem is to supply signal inputs to the sonar system, which simulate the actual return signals and which correspond to the assumed target vessel position, velocity and direction with respect to the position, velocity and direction of the receiver. The range R and angle θ of this relative position are computed at regular time intervals, corresponding to the periodic signal transmission from the transducer. The return signal parameters include also the doppler frequency shift and the normalized amplitude. The doppler frequency shift is derived from the velocity and direction of the target with respect to the receiver. The normalized return signal amplitude relates to the actual visible size of the target, and to the amplitude gain of the target which depends on the target position with respect to the center of the beam. The amplitude envelope is also chosen to simulate the multipath effect. The sectors on the transducer are numbered in a clockwise direction, each of width 10°. Therefore, there are altogether thirty-six transducer sectors. A response may occur in a combination of two adjacent sectors of the receiving transducer, due to the angular position of the target. Also the time delay between the right and left half beams of an excited sector, represents the phase difference between the two half beam signals due to the target location deviating from the center of the main beam. The indication on the display screen in the sonar system is to show the distance between the target and the receiver, doppler frequency shift, angular position and the size of the target.

The computer programmes written assume the initial velocity, distance and the angular position of the target with respect to the receiver. An Intel SDK-85 microcomputer is used to generate, according to these input parameters, the return signal in synchronization with the transmitted pulses. The role of the microcomputer will be further discussed in the next section. Control signals are also generated by the computer for the proper timing to generate the return signal. In order to interface this signal with the sonar system, hardware is used to convert the digital output data from the computer into analog signals. In addition, multiplexer circuits are built to select the appropriate circuits of the sonar system to represent the angular position of the target, corresponding to the sector of the receiving transducer.

1.2 PROBLEM STATEMENT

The specifications are provided to develope a sonar simulator for one particular sonar system, the AN/SQS 505 system. However, the principles can be generalized to supply simulated responses to other sonar receivers.

The following data were used as basic parameters, creating a prototype system that can simulate one single target in a range from 800 yards to 32000 yards. The target should appear in one beam and in one adjacent beam. According to the assumptions of the initial orientation between the target and the receiver, the simulated return signal is computed. The doppler frequency shift of the return signal should have a resolution of 100 nanoseconds. The return frequency is the 7.2 kilohertz transmitted frequency modified by the doppler effect. The receiving transducer consists of thirty-six sectors to identify the possible target positions within 10^{o} beams. Also, each sector contains right and left half beam signals. Therefore, the simulator should provide seventy-two possible outputs to be connected to the sonar system, four of which are active only at any given time.

The thesis consists of three parts - derivation of the algorithm, the software design and hardware design. An algorithm is developed, and programmed to have the microcomputer provide the digital data to be translated into return target signals to the sonar system.

The SDK-85 microcomputer was chosen for its sufficient applications for this particular design. The initial data for the algorithm can be entered into the computer via the keyboard by accessing the keyboard interrupt. The computer has to be synchronized by the transmitted pulses from the sonar system. The computer provides a feature to compute the period of these pulses by hardwiring these signals from the sonar system to the Serial Input Data of the CPU (Central Processor Unit) of the computer. Also, the computer can be programmed to provide a start pulse through the Serial Output Data to the hardware. There are also two other interrupts which can be hardwired to the CPU. They are the synchronization pulses and the timer interrupts. Furthermore, all the signals are TTL (Transistor Transistor Logic) compatible.

The software consists of two parts - a computational part and a timing control part. For personal preference and simplicity, the compuational part is programmed in Fortran to calculate the data for the target return signal. The timing control part is programmed in 8085 Assembly Language. It controls the input/output data flow and it also generates the control timing interrupt signals for the occurrance of the return signals. The entire programmes are stored into the Erasable Programmable Read Only Memories (EPROM's) by means of EPROM programmer which is availabe in the University of Ottawa.

The hardware serves as an interface between the computer and the sonar system. It is designed to interpret the digital data from the computer to create the analog signals. The clock circuit, which is an entirely novel development, receives the digital data representing the frequency and the left half beam time delays. In synchronization with the transmitted pulses, the clock circuit generates the clock output signals representing the frequency of the return signals. These output signals are in turn used to clock the amplitude data through the Digital-to-Analog (D/A) converters into one or two of the thirty-six sectors of the sonar receiver.

The overall system is shown in Figure 1. The injection points for the analog signals into the sonar system are in the Preformed Beam (PFB) cards of the sonar system. There are thirty-six PFB cards to identify the thirty-six target sectors in 10° beams.

1.3 OUTLINE OF THESIS

In this thesis, the conceptual development of the sonar signal simulator is presented. This development is based on the application of the Intel SDK-85 microcomputer. The thesis consists of three parts - the development of the algorithm, the program development for the SDK-85 microcomputer and the hardware design necessary to change the digital computer outputs into the signal format required for interconnections to the sonar receiver.

In chapter two, the initial assumptions concerning the target and the receiver are stated. The algorithm to generate the target return signals is derived mathematically. This algorithm provides a target return signal represented by the normalized amplitude modified by the amplitude envelope chosen, and the doppler shifted return frequency. The two adjacent transducer sectors excited and the time delay between the right and left half beams of the return signal in each sector are also computed to indicate the correct location of the target.

In chapter three, the software is presented using Fortran and 8085 Assembly Languages. The computational part in Fortran is to compute the parameters representing the target return signal. The parameter data are computed to be compatible with the hardware. The control part, in 8085 Assembly Language, is formulated to provide proper timing for the occurrance of the return target signal. The input/output routines for the computer are also presented.

In chapter four, the hardware is presented to interface between the SDK-85 microcomputer and the sonar system. The design consists of three parts - the clock circuit, the Digital-to-Analog conversions and the multiplexer circuit. An account of the hardware development is given, which contains novel concepts. Finally, the interconnections between the SDK-85 microcomputer, the D/A converters, the clock circuit and the multiplexer outputs are given. This completes the overall design.

In chapter five, conclusions are given and a discussion of the results is presented.

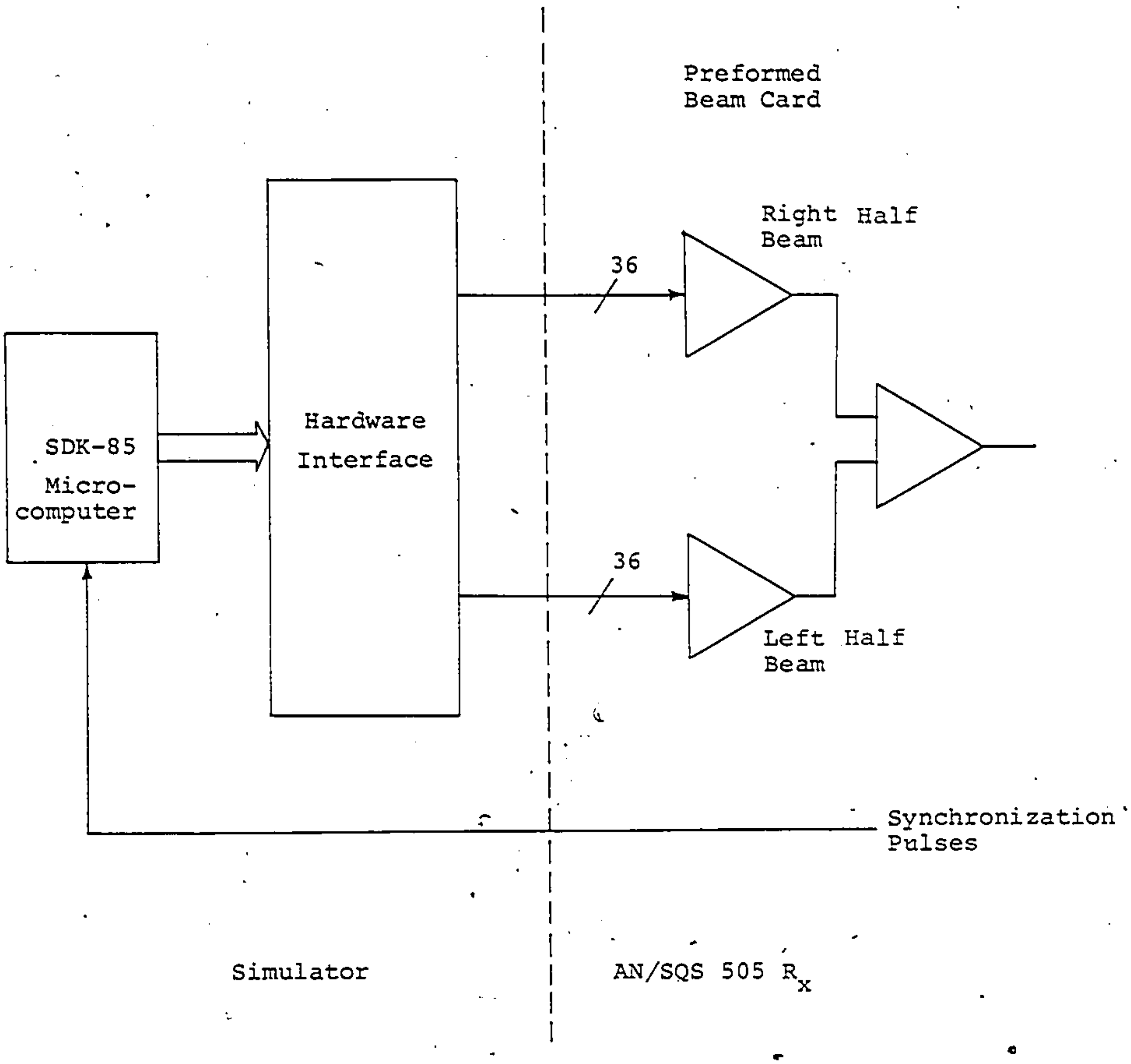


Figure 1: Interface between the Simulator and AN/SQS 505 Receiver

Chapter II

MATHEMATICAL DERIVATION OF SIGNAL GENERATION ALGORITHM

2.1 INTRODUCTION

The mathematical algorithm for the return target signals is generated after assumption of the initial position, velocity and direction of the target with respect to the receiver. Based on these input parameters, the signal parameters, required to represent the return target signal, are derived. They are the normalized amplitude, a selected amplitude envelope, and the return frequency in Hertz. In the following sections, a detailed derivation is given to compute the time of occurrance of the return target signal. This time of occurrance has to be taken relative to the transmitted pulses. In addition, the algorithm has to provide the sector in which the target is located, and the half beam delay in seconds corresponding to the phase difference between the right and left half beams of a particular sector. The algorithm is programmed to generate all the above output data (see chapter III), which are to be converted to analog form. Therefore the computer output data have to match the hardware design.

2.2 FORMULATION OF SIMULATION ALGORITHM

To analyse the problem of supplying test signal inputs for the sonar system, the required output parameters are calculated according to the assumed initial relative position and the relative motion of the target to the receiver.

2.2.1 Assumptions

Several assumptions are made:

(1) The coordinate system is chosen in such a way that the origin is the initial position(t=0) of the receiver, and the x-axis is the direction of motion of the receiver.

(2) Straight line motions are assumed for the target and the receiver.

(3) Constant speeds are also assumed for the target and the receiver.

(4) An initial position of the target relative to the receiver is assumed.

The diagram for one particular orientation between the target and the receiver is shown in Figure 2. The algorithm to be derived will be general for all possible orientations. Refer to Figure 2,

t_1 = time when the pulse is transmitted (seconds),

t_2 = time when the pulse is reflected (seconds),

t_3 = time when the pulse is received (seconds).

$$0 < t_1 < t_2 < t_3$$

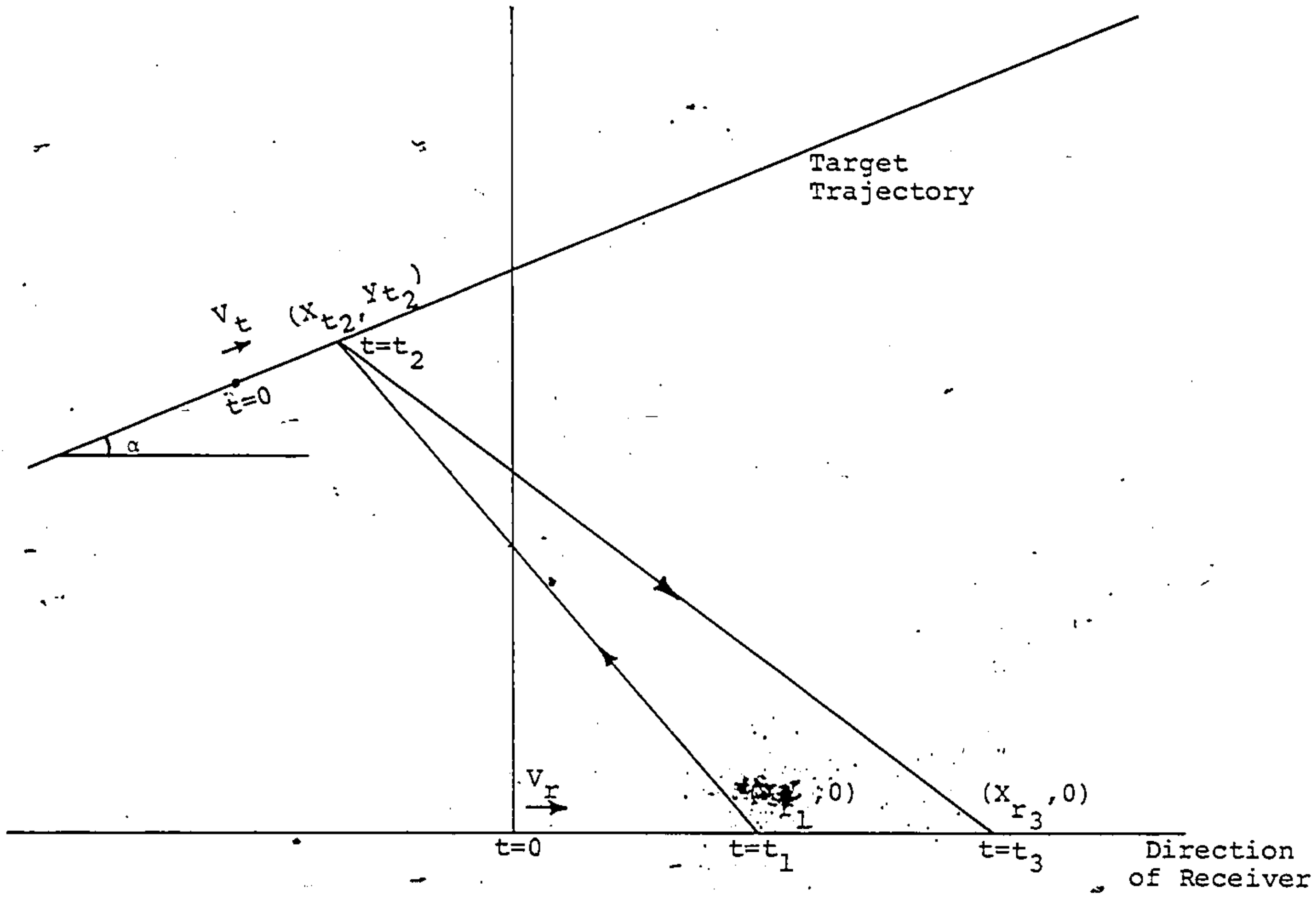


Figure 2: Target Position Relative to the Receiver

2.2.2 Reflection Time and Receive Time

Referring to Figure 2, the x and y components of the velocity of target are

$$V_{t_x} = V_t \cos \alpha \tag{1}$$

and

$$V_{t_y} = V_t \sin \alpha \tag{2}$$

where V_t is the velocity of target in yds/sec
V_{t_x} is the velocity of target in x-axis in yds/sec
V_{t_y} is the velocity of target in y-axis in yds/sec
α is the angle in radians between the target path and the x-axis.

At $t = t_1$, the receiver position is

$$x_{r_1} = V_r t_1 \tag{3}$$

At $t = t_2$, the target position is

$$x_{t_2} = x_{t_0} + V_{t_x} t_2 \tag{4}$$

$$y_{t_2} = y_{t_0} + V_{t_y} t_2 \tag{5}$$

The distance traveled by the transmitted wave is

$$(t_2-t_1)\ V_s = \sqrt{(x_{t_2} - x_{r_1})^2 + y_{t_2}^2}$$

$$= \sqrt{(x_{t_0} + V_{t_x}\ t_2 - V_r\ t_1)^2 + (y_{t_0} + V_{t_y}\ t_2)^2} \qquad (6)$$

where V_s is the velocity of sound (yds/sec) in water. Squaring both sides, we obtain

$$\begin{aligned}(t_2^2 + t_1^2 - 2t_1\ t_2)\ V_s^2 = {} & x_{t_0}^2 + V_{t_x}^2\ t_2^2 + V_r^2\ t_1^2 + 2\ x_{t_0}\ V_{t_x}\ t_2 \\ & - 2\ x_{t_0}\ V_r\ t_1 - 2\ V_{t_x}\ V_r\ t_1\ t_2 \\ & + y_{t_0}^2 + V_{t_y}^2\ t_2^2 + 2\ y_{t_0}\ V_{t_y}\ t_2 \end{aligned} \qquad (7)$$

Rearranging the terms, we have

$$\begin{aligned} & t_2^2\ (V_{t_x}^2 + V_{t_y}^2 - V_s^2) + t_2\ (2\ x_{t_0}\ V_{t_x} + 2\ y_{t_0}\ V_{t_y} \\ & - 2\ V_{t_x}\ V_r\ t_1 + 2\ t_1\ V_s^2) + (x_{t_0}^2 + y_{t_0}^2 + V_r^2\ t_1^2 \\ & - 2\ x_{t_0}\ V_r\ t_1 - t_1^2\ V_s^2) = 0 \end{aligned} \qquad (8)$$

t_2 is therefore solved by

$$t_2 = \frac{-b \pm \sqrt{b^2-4\ ac}}{2a} \qquad (9)$$

where

$$a = V_{t_x}^2 + V_{t_y}^2 - V_s^2 \tag{10}$$

$$b = 2\, x_{t_0}\, V_{t_x} + 2\, y_{t_0}\, V_{t_y} - 2\, V_{t_x}\, x_{r_1} + 2\, t_1\, V_s^2 \tag{11}$$

$$c = x_{t_0}^2 + y_{t_0}^2 + x_{r_1}^2 - 2\, x_{t_0}\, x_{r_1} - t_1^2\, V_s^2 \tag{12}$$

Thus t_2 ,the reflection time, is known and x_{r_1} , x_{t_2} and y_{t_2} are known also. At $t = t_3$, the receiver position is

$$x_{r_3} = V_r\, t_3 \tag{13}$$

The distance traveled by the reflected wave is

$$(t_3 - t_2)\, V_s = \sqrt{(x_{r_3} - x_{t_2})^2 + y_{t_2}^2}$$

$$= \sqrt{(V_r\, t_3 - x_{t_2})^2 + y_{t_2}^2} \tag{14}$$

Squaring both sides, we obtain

$$(t_3^2 + t_2^2 - 2\, t_2\, t_3)\, V_s^2 = V_r^2\, t_3^2 + x_{t_2}^2 - 2\, V_r\, t_3\, x_{t_2} + y_{t_2}^2 \tag{15}$$

Rearranging the terms, we have

$$t_3^2\, (V_r^2 - V_s^2) + t_3\, (\, 2\, t_2\, V_s^2 - 2\, V_r\, x_{t_2}) + (x_{t_2}^2 + y_{t_2}^2 - t_2^2\, V_s^2) = 0 \tag{16}$$

t_3 is solved by

$$t_3 = \frac{-b \pm \sqrt{b^2 - 4\,ac}}{2a} \tag{17}$$

where

$$a = V_r^{\,2} - V_s^{\,2} \tag{18}$$

$$b = 2\,t_2\,V_s^{\,2} - 2\,V_r\,x_{t_2} \tag{19}$$

$$c = x_{t_2}^{\,2} + y_{t_2}^{\,2} - t_2^{\,2}\,V_s^{\,2} \tag{20}$$

Thus t_3 , the receive time, is known.

2.2.3 Range

The range between the target and the receiver can be computed as

$$\text{RANGE} = \sqrt{(x_{r_3} - x_{t_2})^2 + y_{t_2}^{\,2}} \tag{21}$$

The time delay between the pulse transmitted and the signal received is determined by the range between the target and the receiver, and is computed by

$$T_{range} = t_3 - t_1 \tag{22}$$

2.2.4 Doppler Effect

The frequency of the pulse is modified by the speed of the moving vessel relative to the speed of sound in water [4]. The frequency of the pulse f is modified as shown in the Figure 3 and Equation (23).

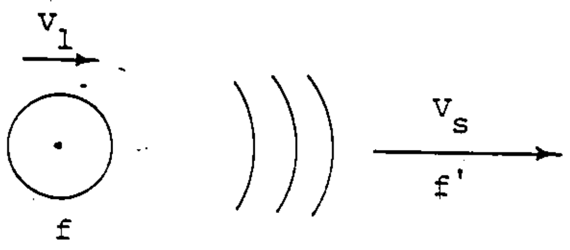


Figure 3: Frequency Modification for Moving Transmitter

$$f' = f \frac{V_s}{V_s - V_1} \quad , \quad f' > f \tag{23}$$

where V_1 is the speed of the transmitter,
f is the transmitted frequency in Hz,
f' is the modified frequency of the wave travelling in water, in Hz.

For a moving receiver, the relationship between the received frequency and the modified frequency f' is shown by

$$f'' = f' \frac{V_s - V_2}{V_s} , \quad f'' < f' \tag{24}$$

as described in Figure 4.

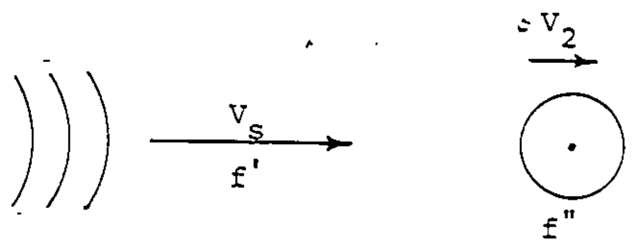


Figure 4: Frequency Modification for Moving Receiver

The return frequency f'' is found from Equations (23) and (24). The diagram in Figure 5 shows the overall doppler effect. The parameters are defined as follows:

f_1 = transmitted frequency of the pulse, in Hz

f_2 = frequency of the travelling pulse (receiver to target), in Hz

f_3 = frequency of the pulse ~~received at~~ the target, in Hz

f_4 = frequency of the travelling pulse (target to receiver), in Hz

f_5 = frequency of the pulse received at the receiver, in Hz

The angles θ_1, θ_2, θ_3 and θ_4, as defined in the figure, are easily computed by the cosine rule from the known distances a, b, c, d and e. Therefore the relationships between the frequencies are:

$$f_2 = f_1 \frac{V_s}{V_s - V_r \cos \theta_1} \tag{25}$$

$$f_3 = f_2 \frac{V_s - V_t \cos \theta_2}{V_s} \tag{26}$$

$$f_4 = f_3 \frac{V_s}{V_s - V_t \cos \theta_3} \tag{27}$$

$$f_5 = f_4 \frac{V_s - V_r \cos \theta_4}{V_s} \tag{28}$$

Then f_5 rewritten as function of f_1 is:

$$f_5 = f_1 \frac{(V_s - V_t \cos \theta_2)\ (V_s - V_r \cos \theta_4)}{(V_s - V_r \cos \theta_1)\ (V_s - V_t \cos \theta_3)} \tag{29}$$

The total doppler frequency shift is:

$$\Delta f = f_5 - f_1 \tag{30}$$

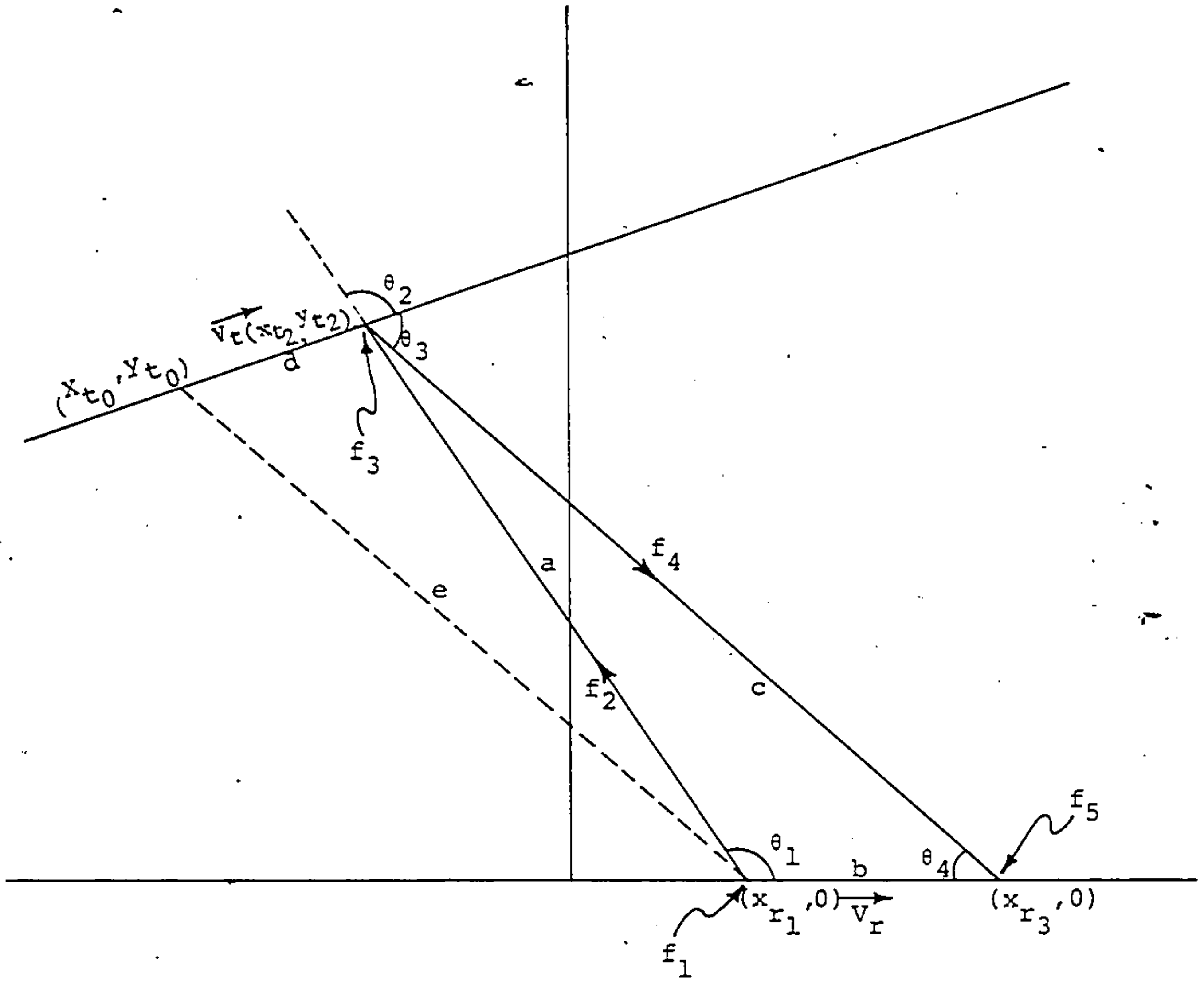


Figure 5: Diagram for Doppler Effect

Positive doppler frequency shift means an approaching target and negative doppler frequency shift means a receding target.

2.2.5 Sector

The transducer consists of thirty-six sectors to identify the possible thirty-six target positions in 10° beams. The sectors are numbered in a clockwise direction while looking down the vertical axis of the transducer. Stave number one is the element with its center displaced five degrees to the right of the ship's bow axis. Since the direction of the receiver is defined by the positive x-axis direction, the sector numbers are as shown in Figure 6.

θ_4 is defined in Figure 5. Therefore θ, which is defined as increasing with the sector number can be computed by

(a) if $y_{t_2} > 0$, $\theta = 180^\circ + \theta_4$ (31)

where y_{t_2} is the y-component of the target position;

(b) if $y_{t_2} \leq 0$, $\theta = 180^\circ - \theta_4$ (32)

Now the number of the sector in which the target lies can be computed from θ and numbering arrangement of Figure 6. When the target is located between two beam centers, the target should appear in two beams on the display screen.

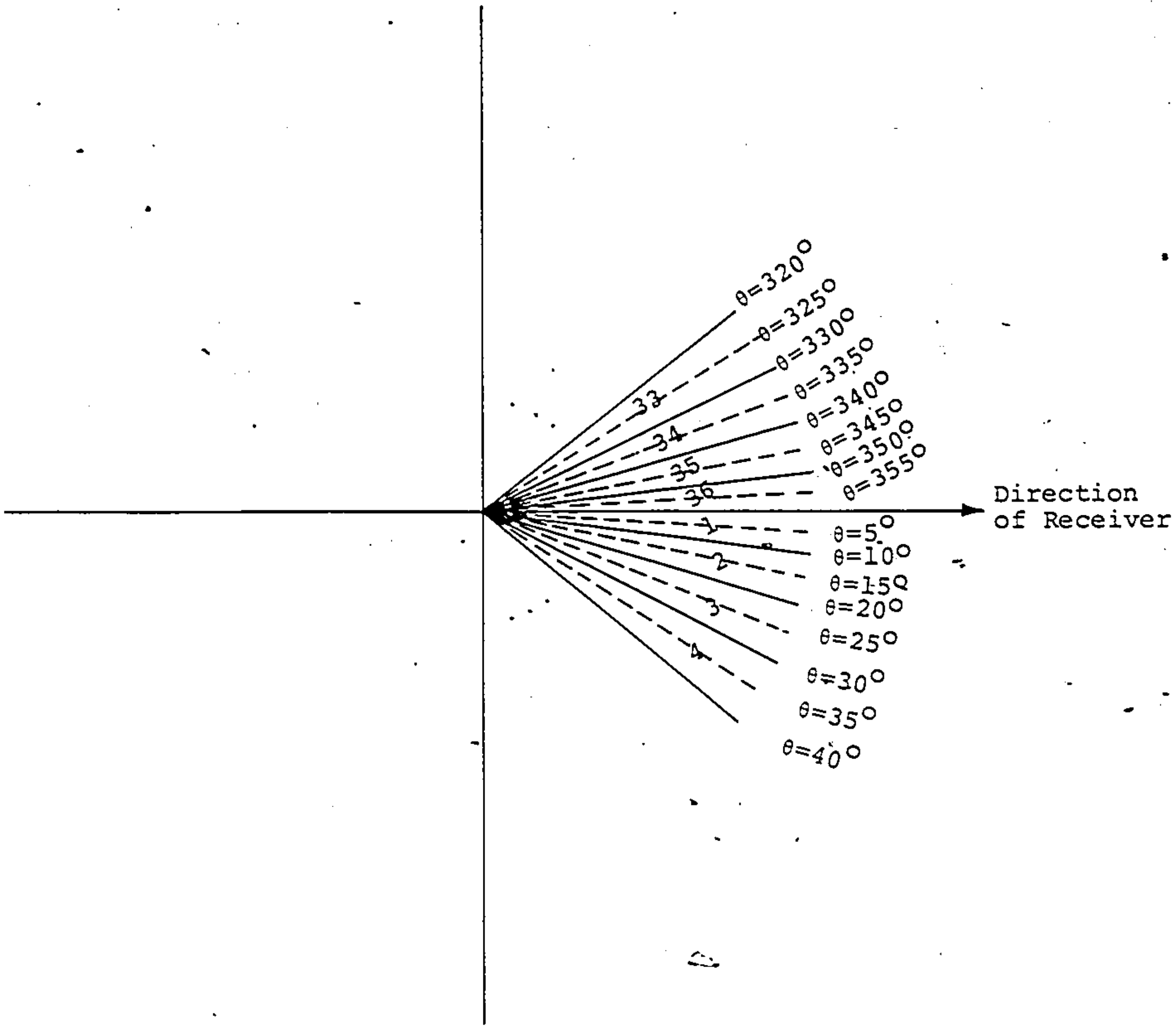


Figure 6: Number Description of Transducer Staves

2.2.6 Half-Beam Delays

Figure 7 shows one particular sector (not drawn to scale). Points a and b are the virtual acoustic centers of the right and left hand beams respectively and they are separated by 1.8 feet. The angle γ is the angle subtended by the target with respect to the center of the beam. Defining Δt as the relative time delay between the right and left hand beams, Δt can be computed from:

$$\Delta t = \frac{1.8 \quad x \sin \gamma}{V_s} \tag{33}$$

where V_s is the velocity of sound in water, in ft/sec.

In this particular case, the left hand beam is delayed with respect to the right hand beam by Δt time units. As shown, the relative time delay between the right and left hand beams can be calculated from γ.

The relative time delay of the adjacent sector is calculated from the angle of deviation of the target, from the center of the adjacent beam, which equals $(10^{o} - \gamma)$.

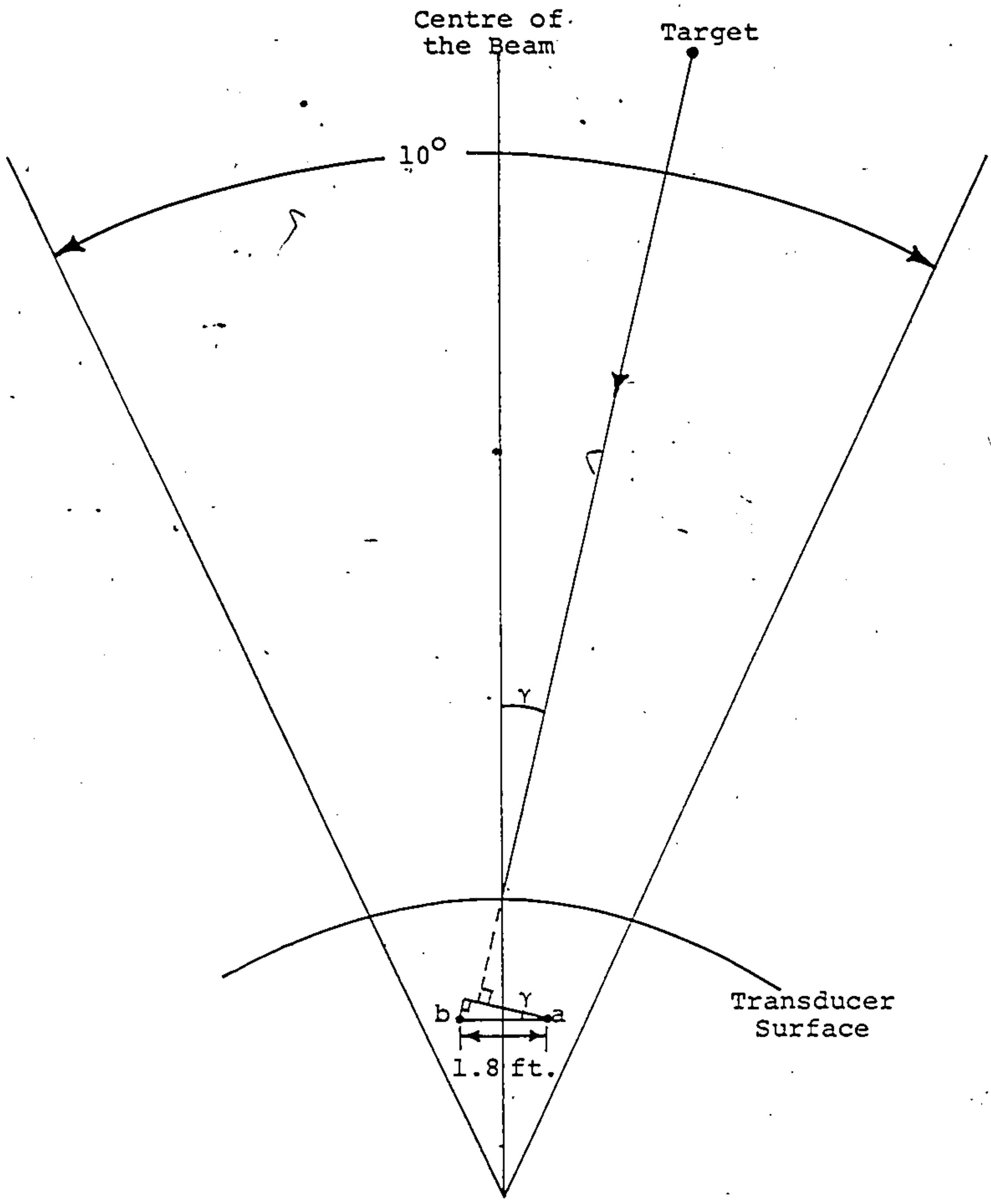


Figure 7: One Particular Sector for Half-Beam Delay

2.2.7 Amplitude of the Return Signal

The amplitude of the return signal is a function of

(a) the range between the target and the receiver,

(b) the actual visible size of the target, and

(c) the amplitude percentage gain of the target which depends on the target position with respect to the center of the beam. (See table on page 27.)

However, the Automatic Gain Control (AGC) circuit, which is internal to the receiver system, limits the dynamic range necessary to simulate a target return signal. This will be further discussed in the hardware design section.

Referring to Figure 8, the power of the return signal is proportional to

$$\text{Emitted power} \times G^2(\gamma) \times \sigma(\beta, L_T, W_T) \tag{34}$$

where Emitted power is a constant,

G is the amplitude percentage gain of the target appearing in the beam,

σ is the actual visible size of the target,

L_T is the length of the target,

W_T is the width of the target.

The actual visible size of the target is computed by

$$\sigma(\beta, L_T, W_T) = L_T \cos\beta + W_T \sin\beta \,, \quad 0 \leq \beta \leq 90^\circ \tag{35}$$

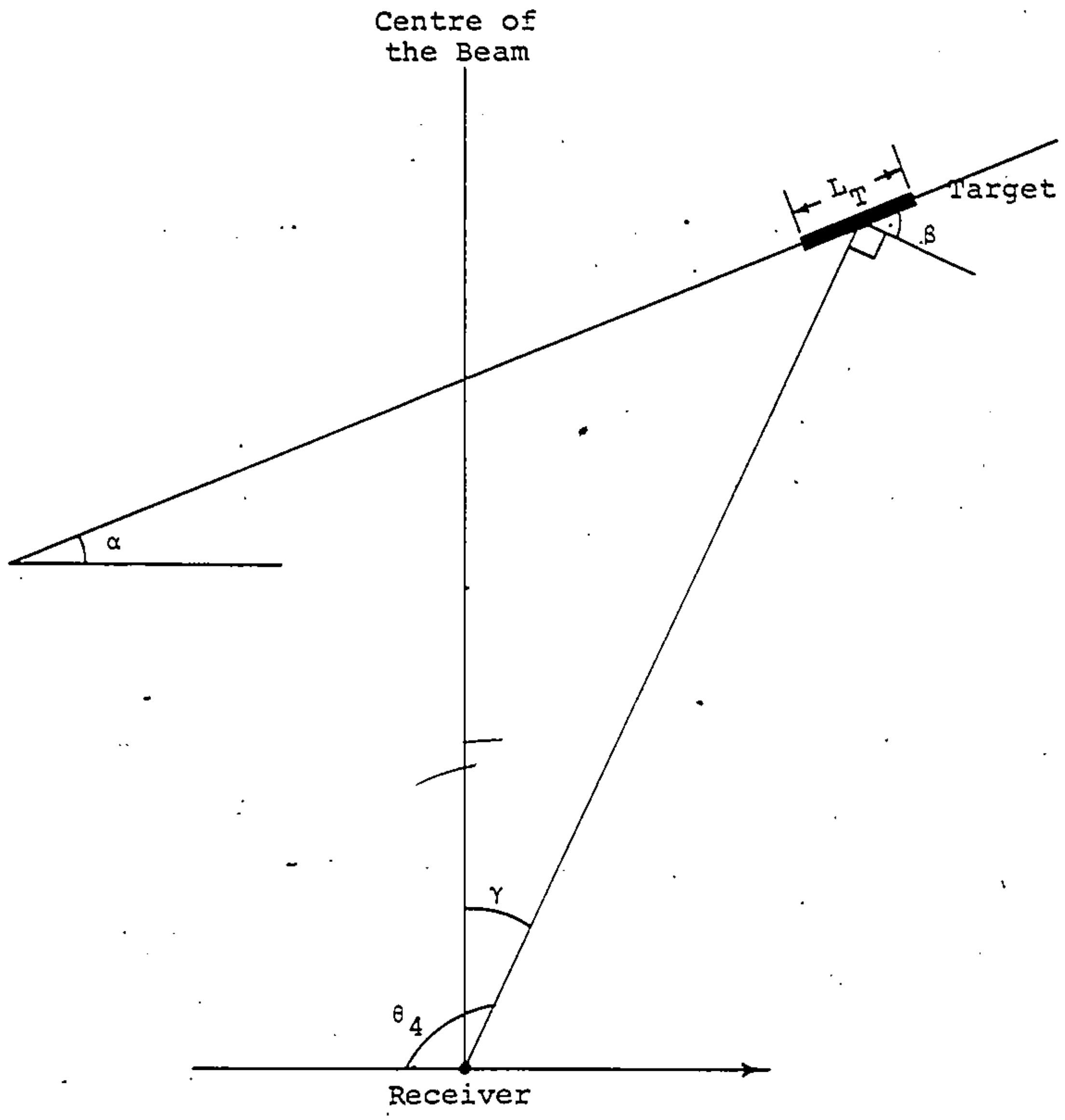


Figure 8: Diagram for the Amplitude of Return Signal

According to the transmitter book [3] any full beam has an aperture of 10° with a 3 db attenuation at $\pm 5^\circ$ from the center. Therefore $G(\gamma)$ is defined as

$$G(\gamma) = \cos(9\gamma) \qquad (36)$$

When the target is located between two beam centers, the target should appear in two beams on the display screen us-using the following algorithm :

TABLE

γ Degree Deviation from Center of Main Beam	$\cos(9\gamma)$ Main Beam		$\cos(9(10-\gamma))$ Adjacent Beam	
	Amplitude %	Power Attenuation (db)	Amplitude %	Power Attenuation (dB)
0	100	0	0	$-\infty$
2	95	-0.43	31	-10.2
4	81	-1.84	59	- 4.61
5	71	-3.0	71	- 3.0
6	59	-4.61	81	- 1.84
8	31	-10.2	95	- 0.43
10	0	$-\infty$	100	0

Three required envelope shapes for the target signals injected into the sonar receiver are to be implemented. The design normalises the maximum calculated signal amplitude.

For a given position of the target, the amplitude (envelope) of the return signal is (by Equations (34), (35) and (36))

$$A = \sqrt{\cos^2 (9\gamma) \times \sigma} \qquad (37)$$

Now σ is redefined as $\cos\beta + 0.1 \sin\beta$, providing that we take the width as one-tenth the length of the submarine.

2.3 LIMITATIONS

The sonar signal injector should simulate a single target only. Target realism is achieved by varying the simulated return frequency, amplitude envelope, and signal level. Operation is in a range limited from 800 yards to 32000 yards. A target is considered to be in no more than two adjacent beams at once to simulate smear at high levels of signal or the movement of a target from one beam to an adjacent beam.

The return amplitude signal is realized by Equation (37). The minimum amplitude of the return signal is calculated when $\gamma = 9^\circ$ and $\sigma = 0.1$.

$$A_{min} = \sqrt{\cos^2 (9 \times 9) \times 0.1}$$
$$= 0.049 \qquad (38)$$

Therefore the total amplitude variation is

$$\frac{A_{max}}{A_{min}} = \frac{1}{0.049}$$

$$= \frac{20}{1} \qquad (39)$$

A wordlength of four bits is considered to be sufficient to simulate this dynamic range of return amplitude to obtain an adequate display appearance.

The total range of variation of the doppler frequency shift has to be estimated for the designs of the software and hardware. The doppler frequency is maximum for radial relative movement between the target and the receiver. Positive doppler frequency implies an approaching target, while negative doppler frequency implies a receding target. Now, let us consider the case when the target and the receiver are approaching each other on the same path. Referring to Figure 5 and Equations (25) to (28) ,

$$\theta_1 = 0^\circ \qquad (40a)$$

$$\theta_2 = 0^\circ \qquad (40b)$$

$$\theta_3 = 180^\circ \qquad (40c)$$

$$\theta_4 = 180^\circ \qquad (40d)$$

Therefore

$$f_2 = f_1 \frac{V_s}{V_s - V_r} \tag{41a}$$

$$f_3 = f_2 \frac{V_s - V_t}{V_s} \tag{41b}$$

$$f_4 = f_3 \frac{V_s}{V_s + V_t} \tag{41c}$$

$$f_5 = f_4 \frac{V_s + V_r}{V_s} \tag{41d}$$

The maximum doppler frequency shift (Δf) is computed as

$$\max \Delta f = f_5 - f_1$$

$$\approx 250 \text{ Hz} \tag{42}$$

Similarly, for a receding target (radial movement), the maximum doppler frequency is -250 Hz.

Zero doppler shift is encountered at the instant when the paths are perpendicular to the line joining the ship and the target.

b

Chapter III
SOFTWARE DESIGN

3.1 INTRODUCTION

The entire software consists of the computational part, and input/output control and timing control part. The computational part generates the time delay between the synchronization pulse and the target return signal; the sector(s) that the simulated target is situated in; the time delay between the right and left half beams; the return frequency and the normalized return amplitude of the signal. The amplitude envelope is chosen to simulate the multipath effect. These output data from the computational part of the program are to be converted into analog form by the hardware. Therefore the data is computed to be compatible with the hardware design.

The other section of this chapter describes the input/output control and timing control part of the software. This part is written in 8085 Assembly Language. Initial data are typed in via the keyboard of the SDK-85 microcomputer. The input data are converted from BCD (Binary Coded Decimal) to floating point format and stored in memory, ready to be used in the computational part of the programmes. The output data is also stored and ready to be delivered to the hardware. The system is organized to carry out all the procedures in sequence. Interrupt signals are inserted. The return signals are synchronized with the transmitted pulses. The RAM timer is programmed to provide interrupt signals for the occurrance of the return target signals. Error messages are also detected.

3.2 COMPUTATIONAL PART FOR SIGNAL GENERATION IN FORTRAN

The generation of the return signal has to be started by punching in the input data, converting and storing them in the floating point format in some assigned memory locations. These numbers are ready to be used as soon as the Fortran subroutines are called.

3.2.1 Fortran Subroutines

The algorithm for the generation of the return target signal is programmed in Fortran Language. This programme is divided into three parts to allow convenient calculation of the next return signal after transmitting the next synchronization pulse. The three Fortran subroutines are:

```
SUBROUTINE  RECPOL

SUBROUTINE  COMP1

SUBROUTINE  COMP2
```

RECPOL serves to convert the input data from polar to rectangular coordinates. It computes the velocity of the target and the relative range of the target.

COMP1 is called immediately after the transmission of a pulse; when this event occurs, the starting reference time is set to zero. COMP1 calculates the time delay of the return signal from the time of transmission. The delay corresponds to the round trips of transmission and return of the signal. This is shown in Figure 9. COMP1 is designed in such a way that the timer for the delay can be set as soon as the time delay information associated with the range is available. The next call of COMP1 occurs at the next transmission time. This time corresponds to a single period of the synchronization pulses.

COMP2 performs the last stage of the computations. It first checks if the range between the target and the receiver lies between 800 and 32000 yards. If it is, the computations will continue; otherwise, error messages are generated in the following manner:

```
IF (RANGE .LT. 800.) GO TO 450

IF (RANGE .GT. 32000.) GO TO 451

FLG=0

RETURN
```

```
450    FLG=1
       RETURN
451    FLG=2
       RETURN
       END
```

The flag is set to zero if the range is within the limit. The flag is set to one if the range is less than 800 yards. The flag is set to two if the range is greater than 32000 yards. This flag will then be checked in the Assembly programme and different error messages will be generated accordingly.

The sector numbers, time delays between the right and left half beams, frequency modified by the doppler effect, and normalized amplitude are also computed to complete the programme of COMP2.

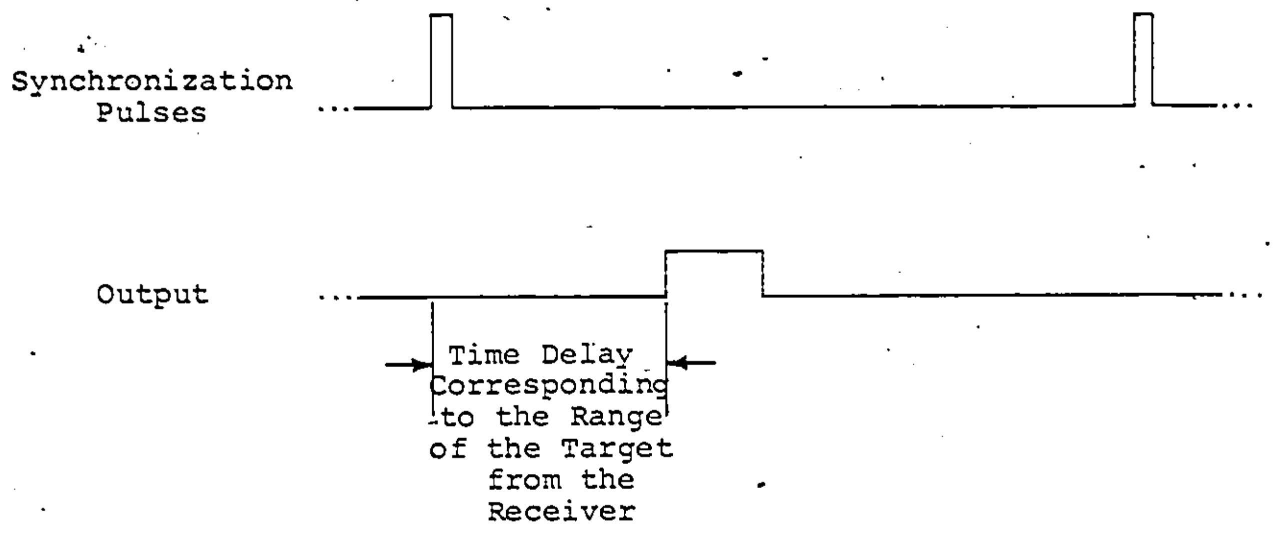

Synchronization Pulses
Output
Time Delay Corresponding to the Range of the Target from the Receiver

Figure 9: **Delay from Synchronization Pulse to Output**

3.2.2 <u>Generation of Input Numbers to the Hardware</u>

COMP2 calculates the actual realizable values of the parameters required as output signals. However, these actual values computed will not be delivered to the hardware. These values are modified to be interpreted by the hardware. Therefore, COMP2 has to translate these actual numbers to integers which can be coded into a required number of bits. Then these values can be transferred to the output ports ready to be delivered to the hardware.

3.2.2.1 Sector Numbers

The target should always appear in two beams on the display screen. A method has to be determined to reduce the numbers of addresses for the input sectors. A target must appear in one sector and the adjacent sector. Since the center of one beam is 10 degrees apart from the center of the adjacent beam, there are altogether 36 sectors to be excited. The method of reducing the number of addresses is based on considering the sectors as being grouped in sets of two, with even and odd sector numbers; so that an odd sector and one of its adjacent even sectors will be addressed in all cases. Thus, the two different groups of even and odd sectors have 18 different sectors each. Thus, two sets of address lines of five bits each are sufficient to address the two sets of 18 sectors (even and odd).

Even sector	2	4	6	8	10	...	28	30	32	34	36
Computer output #	0	1	2	3	4	...	13	14	15	16	17

Odd sector	1	3	5	7	9	...	27	29	31	33	35
Computer output #	0	1	2	3	4	...	13	14	15	16	17

If EVSEC = even sector number

and ODSEC = odd sector number, the equations to compute these values can be implemented in the following programming form:

```
                EVSEC = (EVSEC - 2)/2
and             ODSEC = (ODSEC - 1)/2
```

3.2.2.2 Time Delay Dividing Numbers

Since the target must appear in two beams and each sector excited consists of right and left half signals, there are a total of four signals to be injected into the sonar system at the half beam level for every return signal. COMP2 calculates the relative delays, used for fine tracking, between the right and left half beams both in the even and odd sectors. A method is proposed to delay the right half beams of both sectors by 30 microseconds as shown in Figure 10, so that the delays are described solely as positive numbers. The advantage of doing this is to eliminate a possible negative delay. COMP2 controls the variable left half beam delay by determining a value in the range between 0 and 75 microseconds. This produces the following effects:

a. the appearance of the variable delayed signals 30 microseconds ahead of the fixed delay when the variable delay is set to zero microsecond;

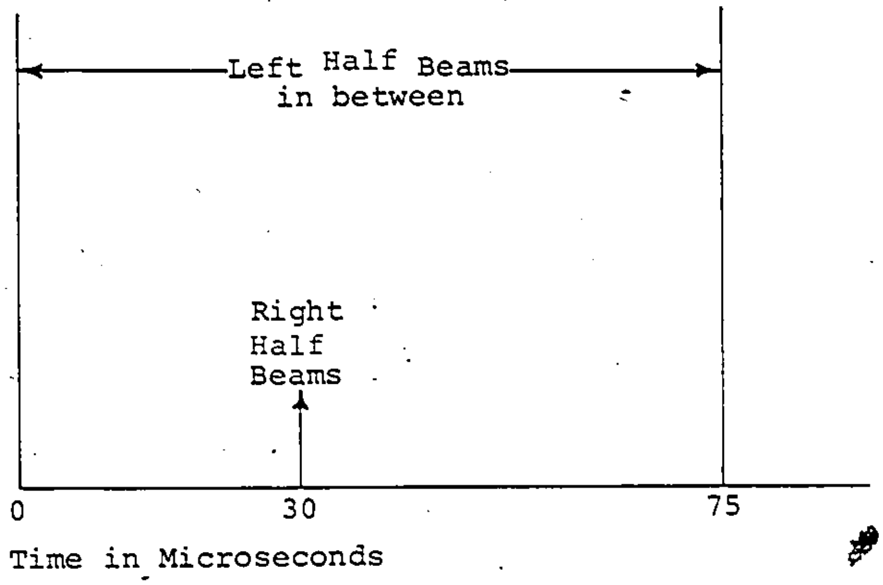


Figure 10: Absolute Delays of Half Beams

b. the arrival of the variable delay signals 45 microseconds after the fixed delay when the variable delay is set to the maximum 75 microseconds; and

c. all values in between the above extremes in increments of 5 microseconds by adjusting the variable delay between zero and 75 microseconds with a 5 microsecond resolution.

Further computations must be made to determine the absolute delays of the left half beam signals of both even and odd sectors. Four bits are sufficient to divide down (will be described in Chapter IV 'Hardware Design') to the required frequency with a 5 microsecond resolution. If

ELDEL = delay of left half signal of even sector,

NELDEL = computer output dividing number,

OLDEL = delay of left half signal of odd sector,

NOLDEL = computer output dividing number,

the equations to compute these values can be implemented in the following programming form:

```
 ELDEL = (ELDEL/5.) + 0.5

NELDEL = IFIX(ELDEL)

 OLDEL = (OLDEL/5.) + 0.5

NOLDEL = IFIX(OLDEL)
```

Thus the delays can be quantized into the closest integers associated with the 16 different levels.

3.2.2.3 Frèquency Dividing Number

The frequency of the return signal was calculated in a range of 7.2 KHz + 250 Hz. The hardware is designed in a way to divide the 10 MHz crystal (will be discussed in Chapter IV) down to twice the required frequency. If

FREQ = frequency of the return signal,
NFREQ = computer output dividing number,

the frequency dividing number is computed in the following form:

```
 FREQ = 10000./(2.*FREQ) + 0.5
NFREQ = IFIX(FREQ)
```

3.2.2.4 Amplitude Numbers

COMP2 only computes the maximum normalizing amplitudes of the return signal in both even and odd sectors according to the algorithm derived. The points describing the amplitude envelope will be calculated in the Assembly programme.

3.2.3 <u>Flowchart</u>

The three different Fortran subroutines complete the generation of the return signal. Once the input parameters are initialized, RECPOL is not repeated again. However, COMP1 and COMP2 are repeatedly called for every consecutive computation of the return signal. COMP2 does the rest of the computations. Figure 11 describes the mechanism of COMP2.

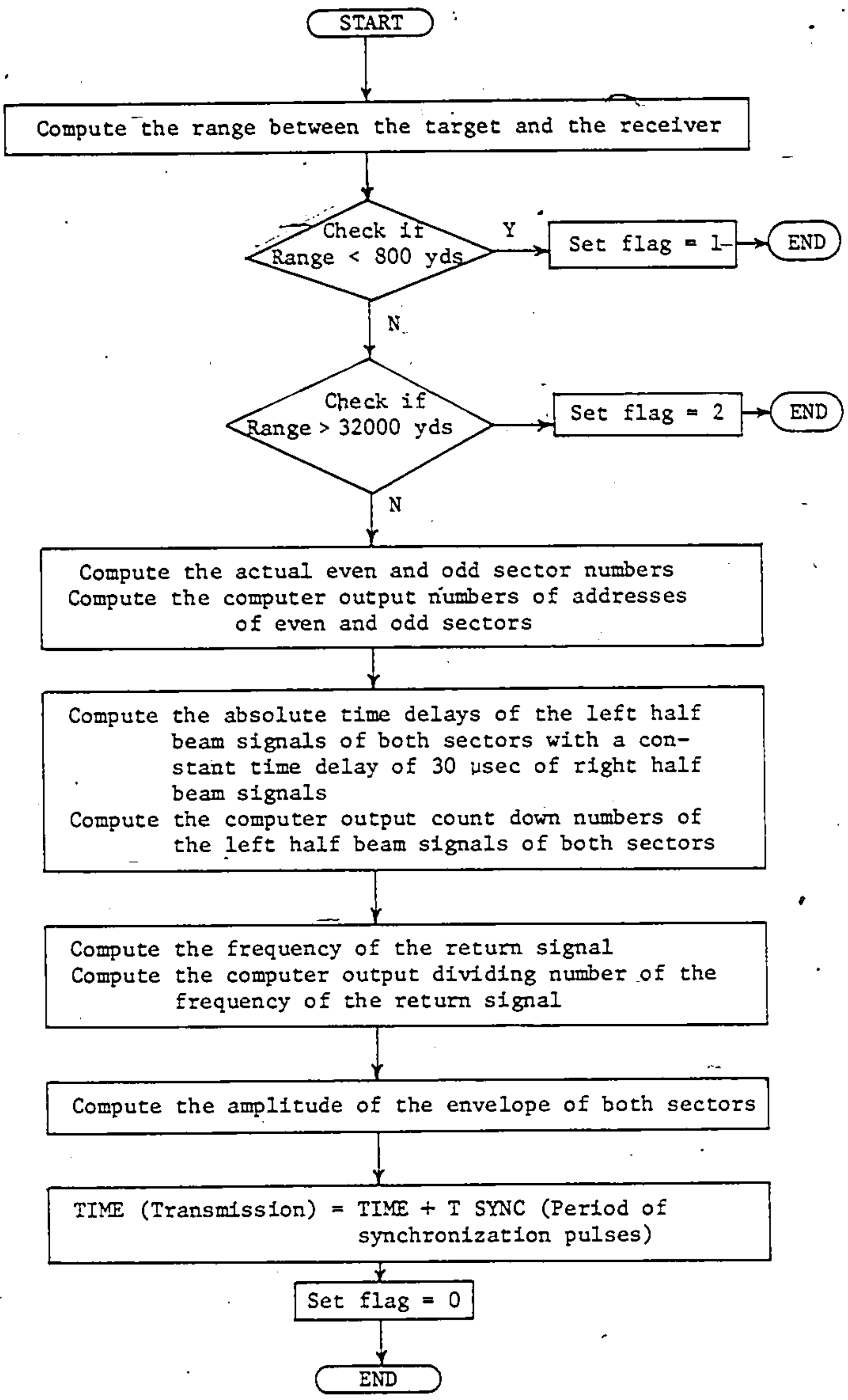

START
Compute the range between the target and the receiver
Check if Range < 800 yds
Y
Set flag = 1
END
N
Check if Range > 32000 yds
Set flag = 2
END
N
Compute the actual even and odd sector numbers
Compute the computer output numbers of addresses of even and odd sectors
Compute the absolute time delays of the left half beam signals of both sectors with a constant time delay of 30 μsec of right half beam signals
Compute the computer output count down numbers of the left half beam signals of both sectors
Compute the frequency of the return signal
Compute the computer output dividing number of the frequency of the return signal
Compute the amplitude of the envelope of both sectors
TIME (Transmission) = TIME + T SYNC (Period of synchronization pulses)
Set flag = 0
END

Figure 11: Flowchart of Subroutine COMP2

3.3 INPUT/OUTPUT AND CONTROL PARTS IN 8085 ASSEMBLY LANGUAGE

The input/output routines are written in Assembly Language. Parameters are typed in by the operator using the keyboard. The input data are checked for valid ranges. Parameter inputs are displayed and checked by the operater. The sequence of executions of all the subroutines has to be ordered to control the sequence of the output parameters to the hardware interface. Programmes in Fortran are properly inserted for signal generation. The interrupts control the timing of injected signals. Output data are then delivered to the output ports ready to be transferred to the hardware.

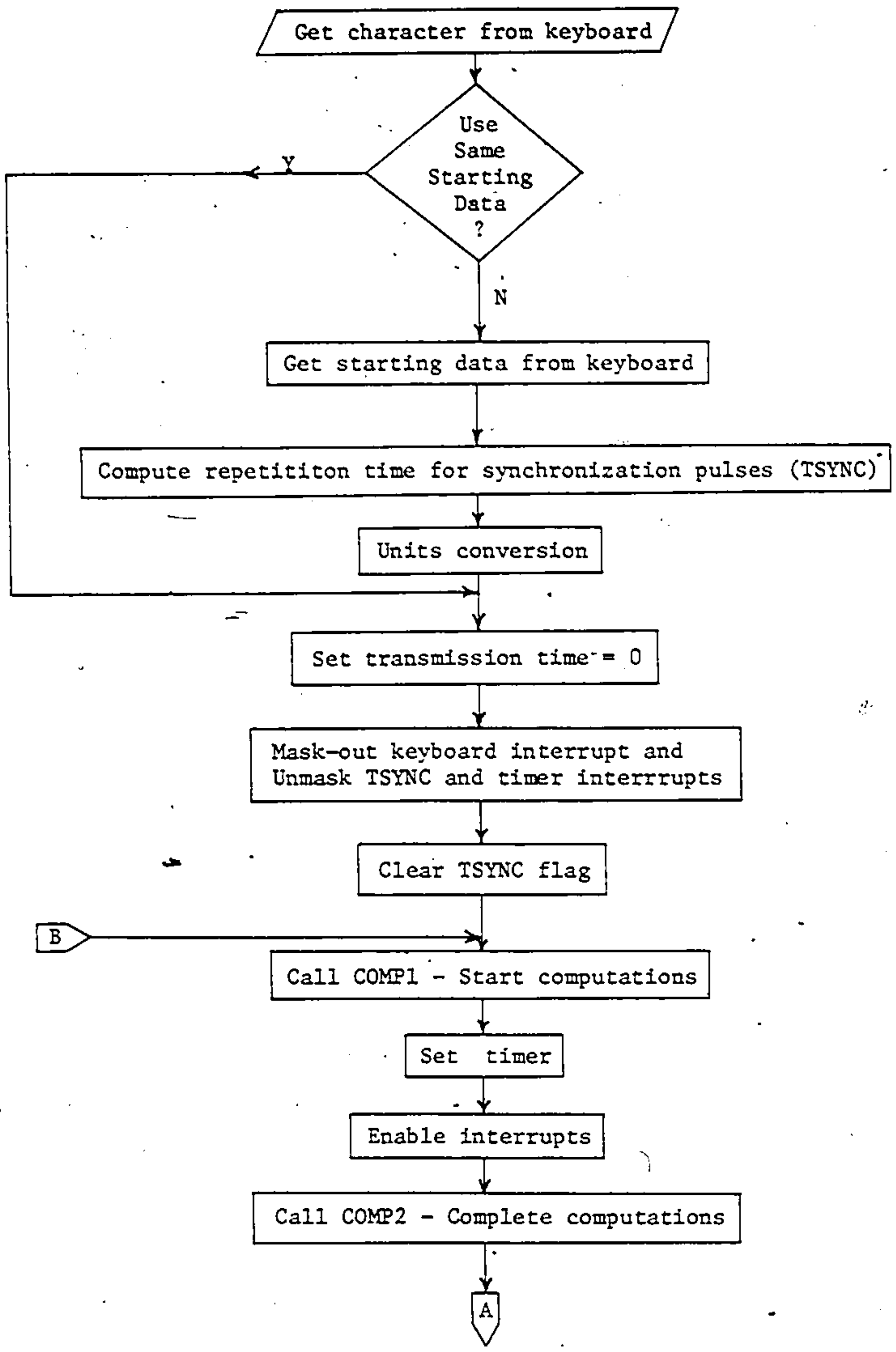
The entire system software is written in Fortran and 8085 Assembly Language. The SDK memory, however, is not sufficient for the entire compiled version of the programmes written and so additional memory chips are mounted. An additional 14K bytes of EPROM are used for the Assembly and Fortran programmes and an additional 1K bytes of RAM are used for the STACK.

One initialization is the resolution counting for the time delay which corresponds to the range between the target and the receiver. The RAM timer is to be used to count to the proper time delay for the range after the synchronization pulse. The timer uses the Central Processor Unit (CPU) clock which is 3072 KHz. This is a high clock rate and we do not require such a high resolution. Therefore a count down has to be done, in advance, to the required frequency corresponding to a suitable resolution. Thus the BASIC timer is used to count down the CPU clock from 3072 KHz to 400 Hz which is equivalent to 2.5 msec corresponding to 2 yards resolution. Now the 400 Hz is in turn the clock frequency to count down for the appropriate delay.

3.3.1 Programme Organization for Data Input/Output and Control

The structure of the system is designed in such a way that after the reset entry point it executes all subroutines in sequence, computations for repetition time of synchronization pulses and signal target return, error detection, interrupt signals and data to the output ports. A flowchart for the sequence is given in Figures 12 and 13.

A character from the keyboard, either a GO or NEXT, is pressed by the operator (see Appendix D 'Operation Instructions'). GO is an indication that the same starting data should be used to run the following programme. Some procedures are skipped as indicated in the flowchart. NEXT indicates new values are to be input. DATAIN (see Appendix E) is called to get the starting data from the keyboard. After this, all seven-segment LED's are turned on until the completion of REPTIM (see Section 3.3.2) and UNITS (see Appendix E). REPTIM computes the repetition time for synchronization pulses (TSYNC) and UNITS converts the starting data to the proper units ready for the computations of the return signals. At this time, reference time is set to zero for the first transmission. The entire display is then cleared to indicate the completion of all the conversions and GO is now displayed in the data field. Data introduction has been completed, thus the keyboard interrupt has to be masked-out and TSYNC and timer interrupts (see Section 3.3.6) are unmasked. The TSYNC flag is cleared at the same time. After all the preparations have been done, COMP1 is called to start the computations. The time corresponding to the range traveled back and forth is available at this moment, therefore the timer is set (see Section 3.3.3) to simulate this time traveled. This timer does not start to count until the TSYNC interrupt comes. The interrupt system is now enabled. COMP2 is then called to complete the computations. Error messages (see Section 3.3.4) are displayed if either the range is less than 800 yards or greater than 32000 yards. The error flag is checked. ERR 2 displayed indicates that the target is too close, while ERR 3 displayed indicates that the target is too far. Either one will cause the system to stop. If the error flag is zero, it means the range falls in its valid limitation. AMP (see Section 3.3.5) is therefore called. It computes, according to the envelope chosen, the points describing the amplitude envelope and stores all of these in the consecutive memory locations. The interrupt enable flag is now checked to see if the interrupts are enabled. If yes, the system has to wait until the TSYNC interrupt comes. If no, the interrupt system is again enabled and wait for the timer interrupt to come. If it happens that the TSYNC interrupt comes without the timer interrupt, error message (ERR 1) will be displayed to indicate that the TSYNC period is too short. This will also cause the system to stop. If everything comes in order, the system will go back to the computation of COMP1 and repeat the same procedures again for the next return of the target signal. As will be mentioned in Section 3.3.6 , the TSYNC interrupt will start the timer and the timer interrupt will stop the timer and produce a start pulse to the hardware interface. It also transfers all the data to the output ports (see Section 3.3.7) and finally resets the TSYNC flag.



Get character from keyboard
Use Same Starting Data ?
Y
N
Get starting data from keyboard
Compute repetititon time for synchronization pulses (TSYNC)
Units conversion
Set transmission time = 0
Mask-out keyboard interrupt and Unmask TSYNC and timer interrrupts
Clear TSYNC flag
B
Call COMP1 - Start computations
Set timer
Enable interrupts
Call COMP2 - Complete computations
A

Figure 12: Flowchart of the Structure of Data Input/Output and Timing Control (Sheet 1 of 2)

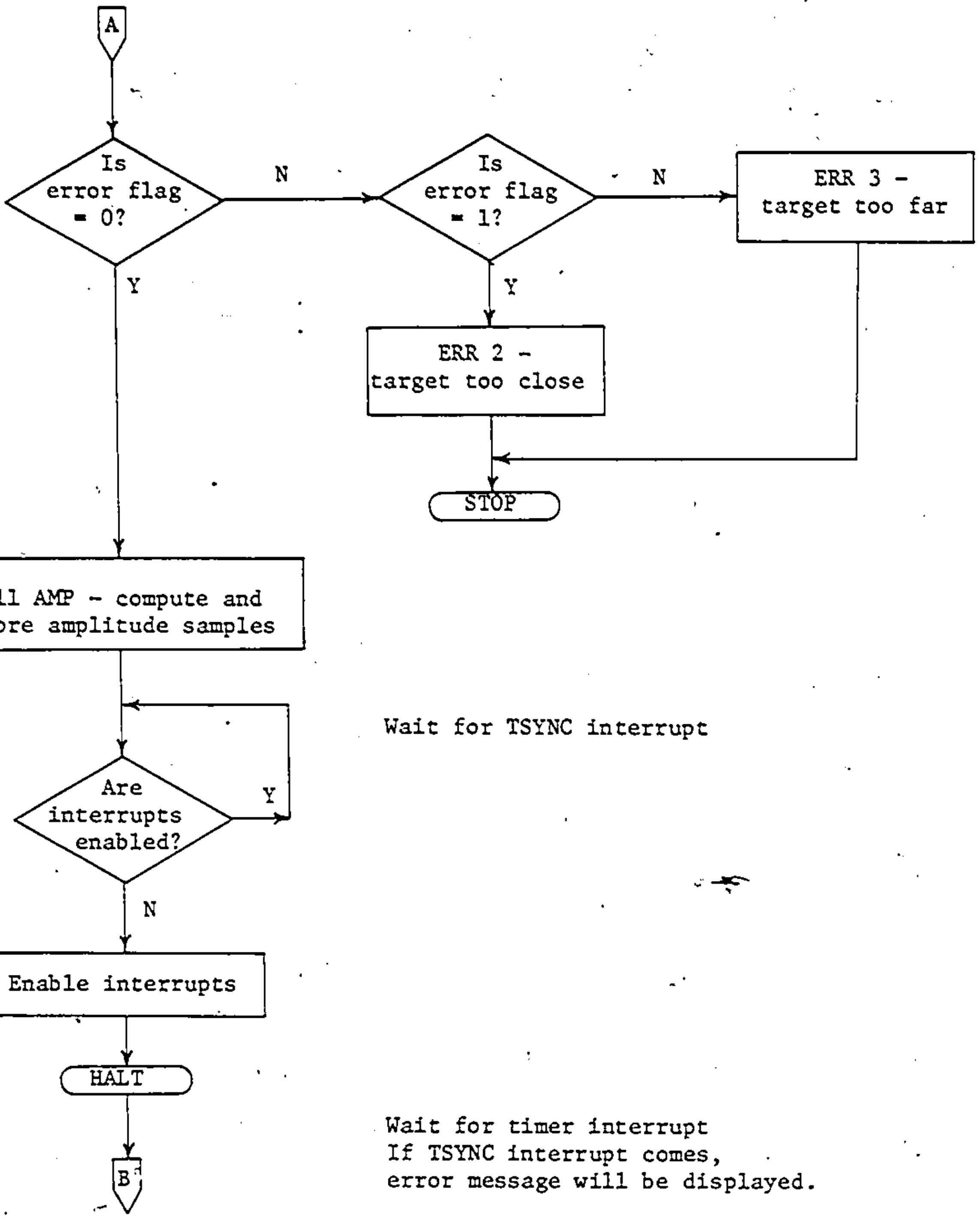


Figure 13: Flowchart of the Structure of Data Input/Output and Timing Control (Sheet 2 of 2)

3.3.2 Repetition Time of Synchronization Pulses, REPTIM

One of the subroutines which is called for is REPTIM. It computes the repitition time of the synchronization pulses presented to the SID (Serial Input Data) at the CPU. The pulses are provided from the sonar receiver and serve two purposes, one for the subroutine REPTIM and the other to provide interrupt signals for synchronization.

REPTIM checks whether a pulse has occurred. If not, it waits until a pulse arrives. It then continues to count milliseconds starting from the rising edge of the pulse until the next pulse arrives. Figure 14 describes this. The total number of milliseconds is then converted to floating point format and again divided by 1000 to get TSYNC in seconds resulting in an accuracy of one millisecond. The number is now stored in memory and is one of the parameters ready to be used for computations.

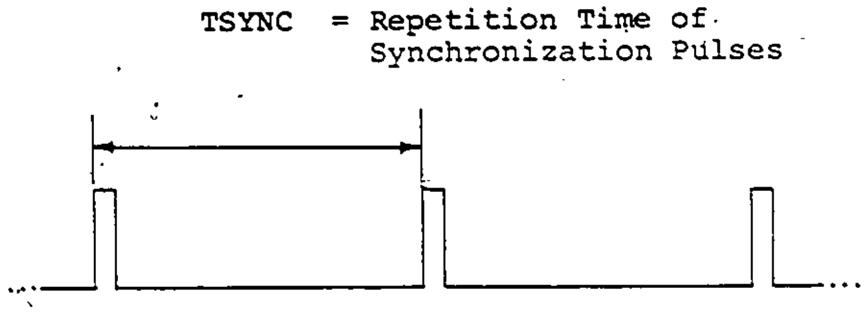


Figure 14: Repetition Time of Synchronization Pulses

3.3.3 Set Timer, SETTIM

As soon as COMP1 is completed, the information concerning the transmission time and the corresponding time of the return signal is available. Hence the timer can be set to count the time which simulates the distance traveled between the target and receiver. The EXPANSION RAM timer is therefore used to count this time. It has a clock frequency of 400 Hz. Figure 15 shows two configurations that count for N seconds. Therefore the actual time computed for the range has to be multiplied by 400 to count with the 400 Hz clock. This multiplied number is then converted from floating point to integer in order to set the timer. The single square wave mode [9] is also set to provide the necessary interrupt signal when the terminal count is reached.

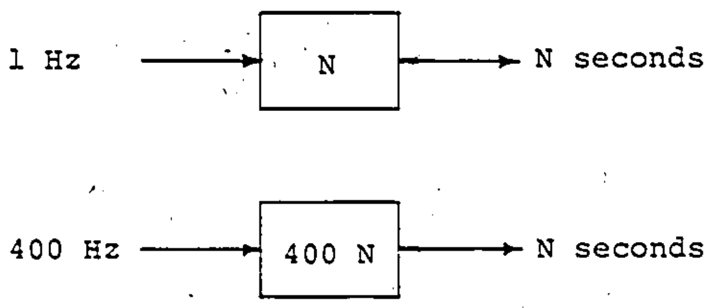


Figure 15: Configurations Explaining the Setting of Counts

3.3.4 Error Messages

Error messages have to be displayed as required. There are three error messages, named ERR 1, ERR 2 and ERR 3. For any one of them, ERR is displayed in the address field and either 1, 2 or 3 is displayed in the data field. The explanations of the error messages are:

ERR 1 - repetition time of TSYNC too short

ERR 2 - target too close, less than 800 yards

ERR 3 - target too far, greater than 32000 yards

3.3.5 Amplitude of Return Signal, AMP

Computations in COMP2 normalize the maximum calculated amplitude of the signal. Then the normalized signal is modulated by a wave envelope shape that is selected from computer memory by the operator. Since the duration of a transmitted pulse is 40 milliseconds, the envelope shape has a time span more than 40 milliseconds to account for the spreading effect. Three possible envelope shapes for the target signals are given in Figure 16. Five-milliseconds step size is used to generate the envelope shapes. AMP determines which envelope is chosen, and computes and stores all the points describing the envelope in consecutive memory locations. Since the maximum time span is 80 milliseconds in waveform 2, seventeen memory locations are therefore reserved for storage. If the entire memory is not going to be occupied as in waveforms 1 and 3, the remaining locations are set to zero.

Four bits are sufficient for the dynamic range of the signal amplitude. The normalized amplitude is translated to the maximum amplitude of the envelope of a particular waveform. The subroutine AMP does this by multiplying the amplitude with 15 and the resulting number is rounded off to the closest one of the sixteen levels. Consecutive points on the envelope are then stored and ready to be transferred to the output ports every five milliseconds.

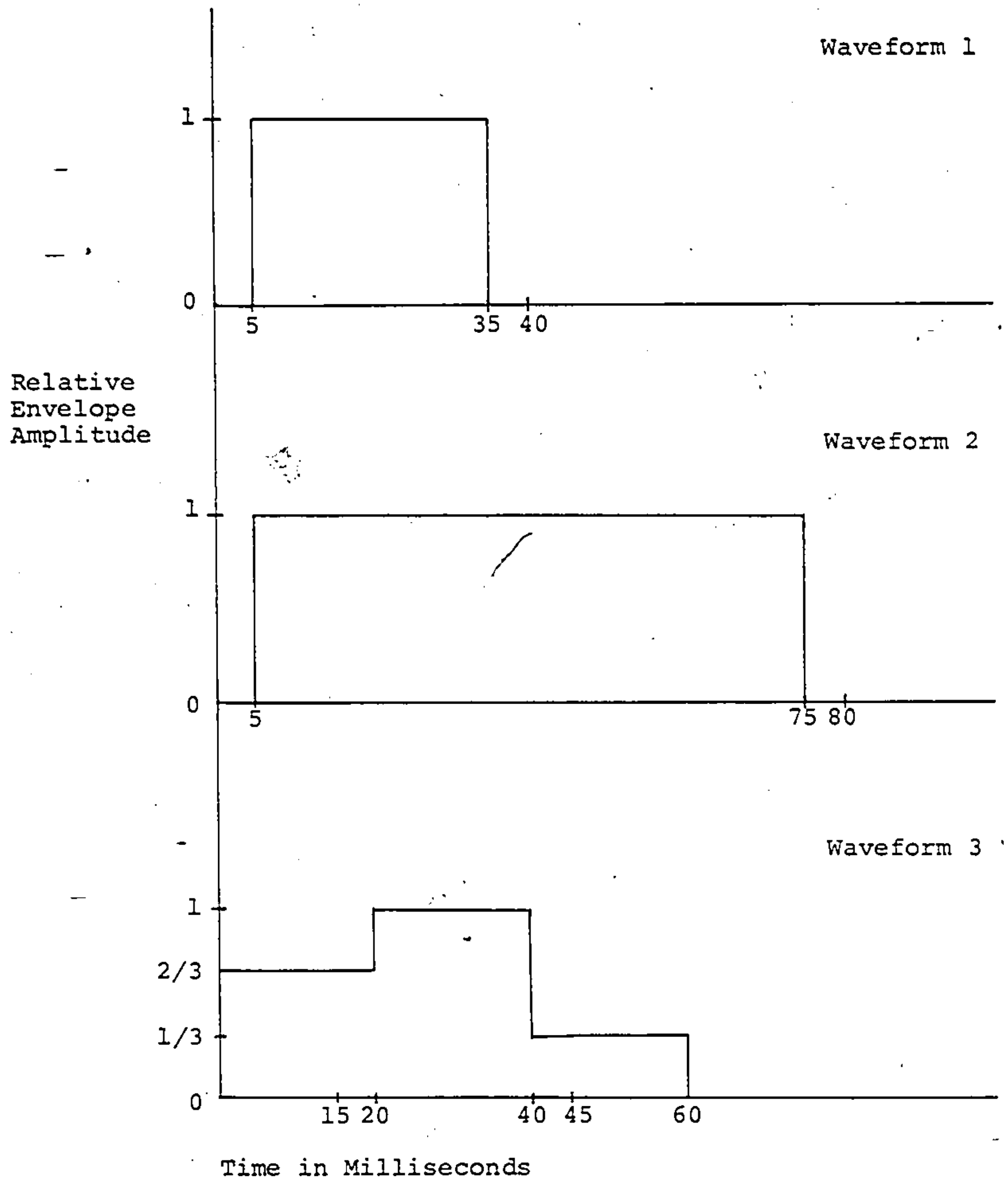


Figure 16: Envelope Shape of Injected Signal

3.3.6 Interrupts

There are three interrupt routines used - RST 5.5 which is dedicated to keyboard interrupt, RST 6.5 which is used for the TSYNC interrupt, and RST 7.5 which is used for the timer interrupt. The RST 5.5, RST 6.5, and RST 7.5 hardware interrupts are different in function in that they are maskable through the use of the SIM (Set Interrupt Masks) instruction, which enables or disables these interrupts by clearing or setting corresponding mask flags based on data in the accumulator. The status of the interrupt mask previously set may be read by performing a RIM (Read Interrupt Masks) instruction.

The input interrupt routine (ININT) is entered when the RDKBD routine is waiting for a character and the user has pressed a key on the keyboard. ININT stores the input character in the input buffer and returns control to the RDKBD routine.

Figure 17 describes the timing control of the TSYNC (SYNINT) and timer interrupts (TIMINT). After COMP1 is finished, the timer is set and interrupts are enabled. The TSYNC interrupt can occur anytime after this. SYNINT starts the timer and takes care of the order of the interrupt sequence. It checks the TSYNC flag. If the flag is not cleared, ERR 1 is generated, which means that the repetition time of synchronization pulses is too short. This is because two TSYNC interrupts come in consecutively without an intervening timer interrupt. If the flag was found to be cleared, ERR 1 would not be created, and the interrupts are in the right sequence. Finally, the TSYNC flag is sét by SYNINT.

After the completion of COMP2, interrupts are enabled to allow the timer interrupt to occur. The wait time is found to be longer than the runtime of COMP2 in all circumstances. Therefore the timer interrupt must occur after the completion of COMP2. TIMINT stops the timer and generates a start pulse to the hardware. ECHO is then called to load all the output parameters to the output ports. The start pulse is now reset and the TSYNC flag is also cleared. The system will then go to the computations of COMP1 again.

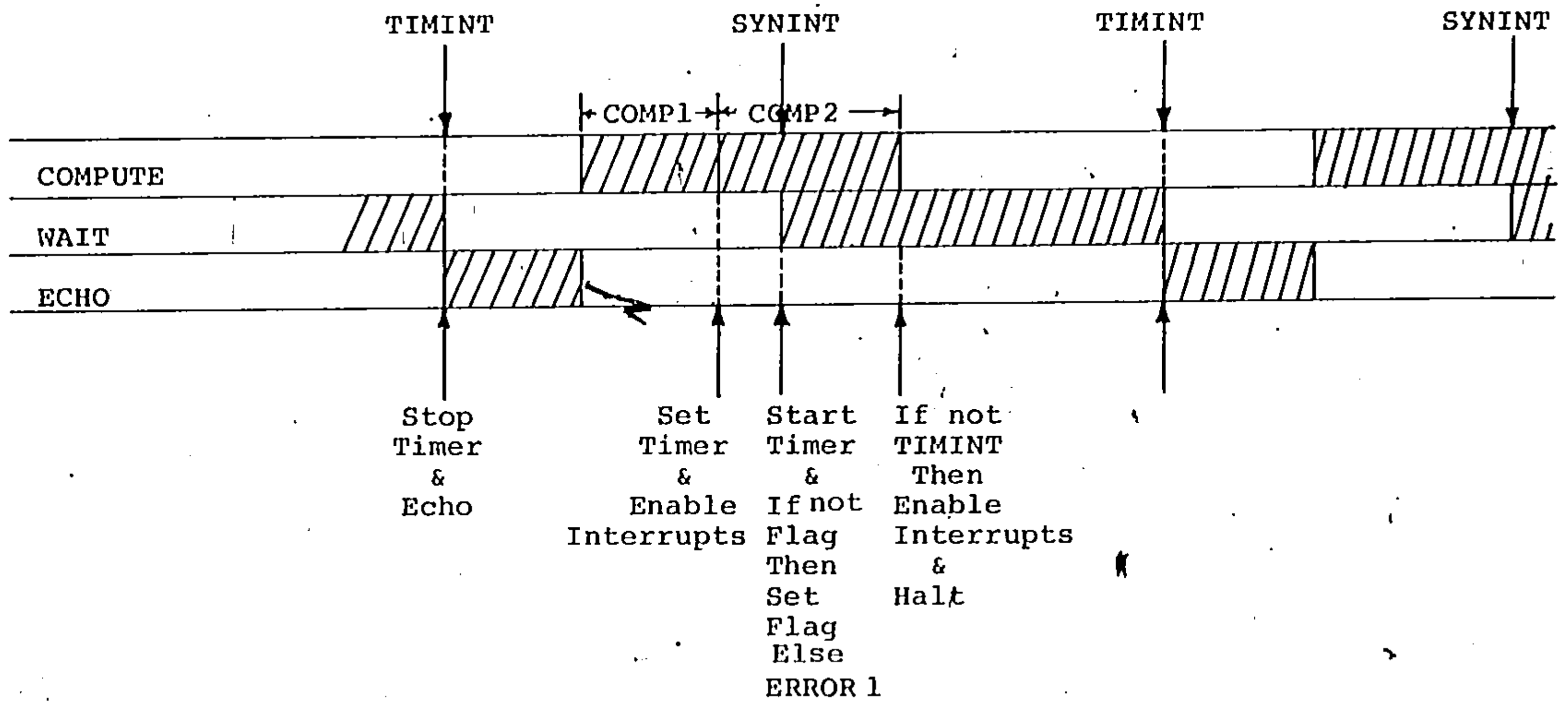

TIMINT
SYNINT
TIMINT
SYNINT
COMP1
COMP2
COMPUTE
WAIT
ECHO
Stop Timer & Echo
Set Timer & Enable Interrupts
Start Timer & If not Flag Then Set Flag Else ERROR 1
If not TIMINT Then Enable Interrupts & Halt

Figure 17: Timing Control of TSYNC and Timer Interrupts

3.3.7 Output, ECHO

ECHO transfers all available output data stored in the memory to the output ports. The port assignments are shown in Figure 18. ECHO has to account for the fact that NELDEL and EVAMP are transferred to the high order bits, and NOLDEL and ODAMP are transferred to the low order bits of the respective ports. The consecutive points, representing the envelope, are to be transferred to the output ports every five milliseconds until the last value has been transferred.

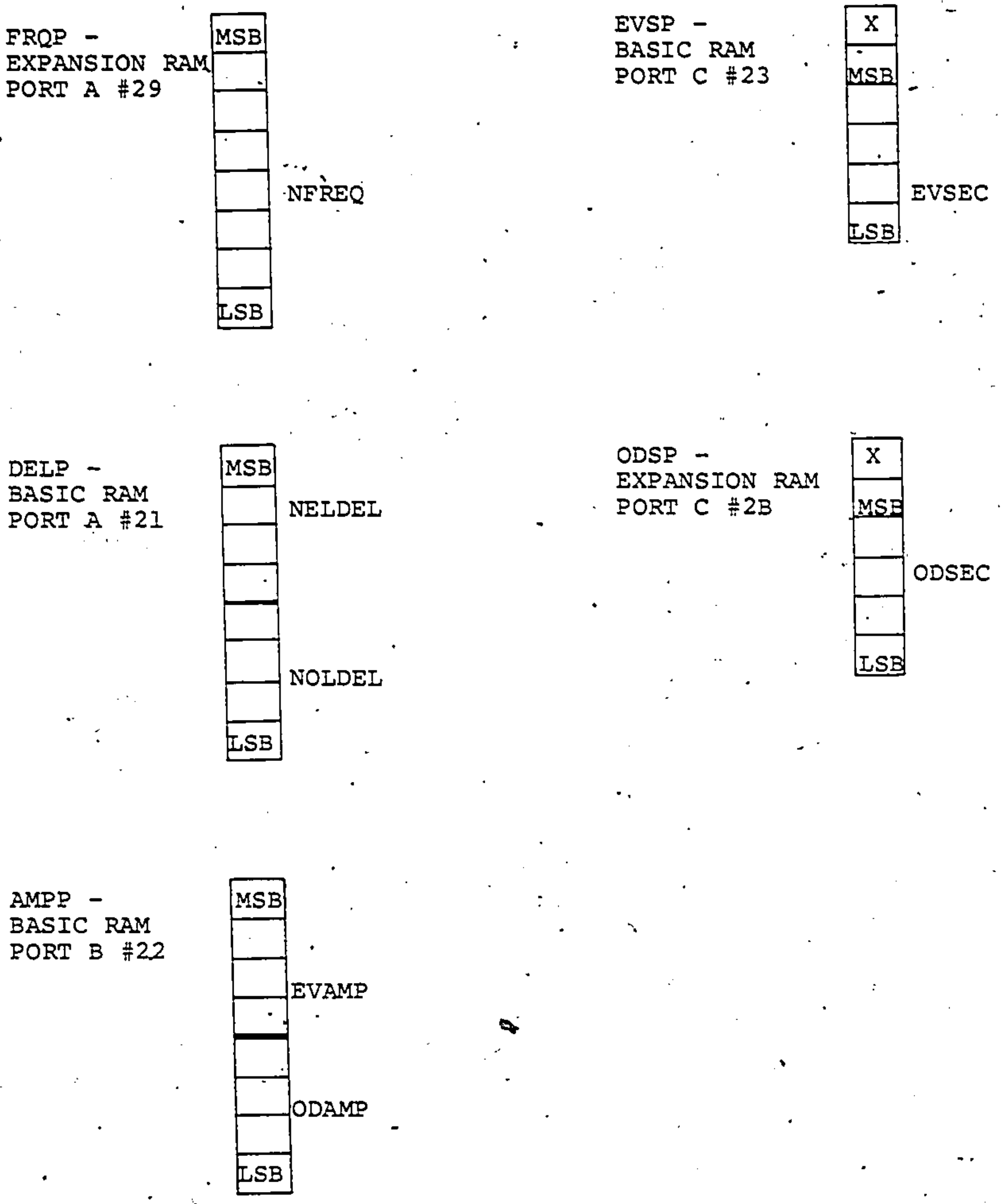


Figure 18: Port Assignments for Output Data

Chapter IV
HARDWARE DESIGN

4.1 INTRODUCTION

Referring to Chapter III 'Software Design', the digital outputs generated by the computer are the even sector number, odd sector number, even sector amplitude, odd sector amplitude, frequency of return signal, even left sector delay, odd left sector delay and a start pulse. A start pulse has to be initiated before the delivery of the above digital data to the hardware.

To provide analog signal inputs to the sonar system, the hardware is required to generate a signal, the frequency of which is the transmitted frequency modified by the doppler shift. (Refer to Sections 2.2.4 and 3.2.2.3.) This signal should appear on two adjacent sectors to indicate the correct position of the target. (Refer to Sections 2.2.5 and 3.2.2.1.) For the fine tracking in the sonar system, two signals, representing the right and left half beams for each sector, account for the phase difference in the half beams. (Refer to Sections 2.2.6 and 3.2.2.2.) The amplitudes of each of the signals are determined by the calculated normalized values and by the AGC (Automatic Gain Control) feedback from the sonar system. (Refer to Sections 2.2.7, 3.2.2.4 and 3.3.8.)

The overall design is shown in block diagram form in Figure 19. The number of bits has been defined for every output parameter of the software. The right and left half beams of a sector are addressed by the same number since they are in the same sector. As the sectors are grouped in sets of two, there are four groups of eighteen half beams. Thus, for the main and adjacent sectors excited, there are altogether four 18-output multiplexing circuits. Two of these indicate the right and left half beams of an even sector, and the other two indicate those of an odd sector.

The digital data representing the amplitude envelope is clocked into the multiplying D/A converters by a number of frequency signals coming out of the clock circuit. Only three clocks are required because the right half beam of each sector is taken as a reference delay of 30 microseconds and the other two left half beams are clocked at the required times corresponding to the computed delays. Both half beams in a particular sector have the same amplitude. For the clock circuit, the frequency input data is used to count the 10 MHz down to the required doppler shifted frequency. The signal outputs of this frequency are delayed to account for the relative delay between right and left half beams. The input data for the left beam delays of both sectors are set to give the computed delay times.

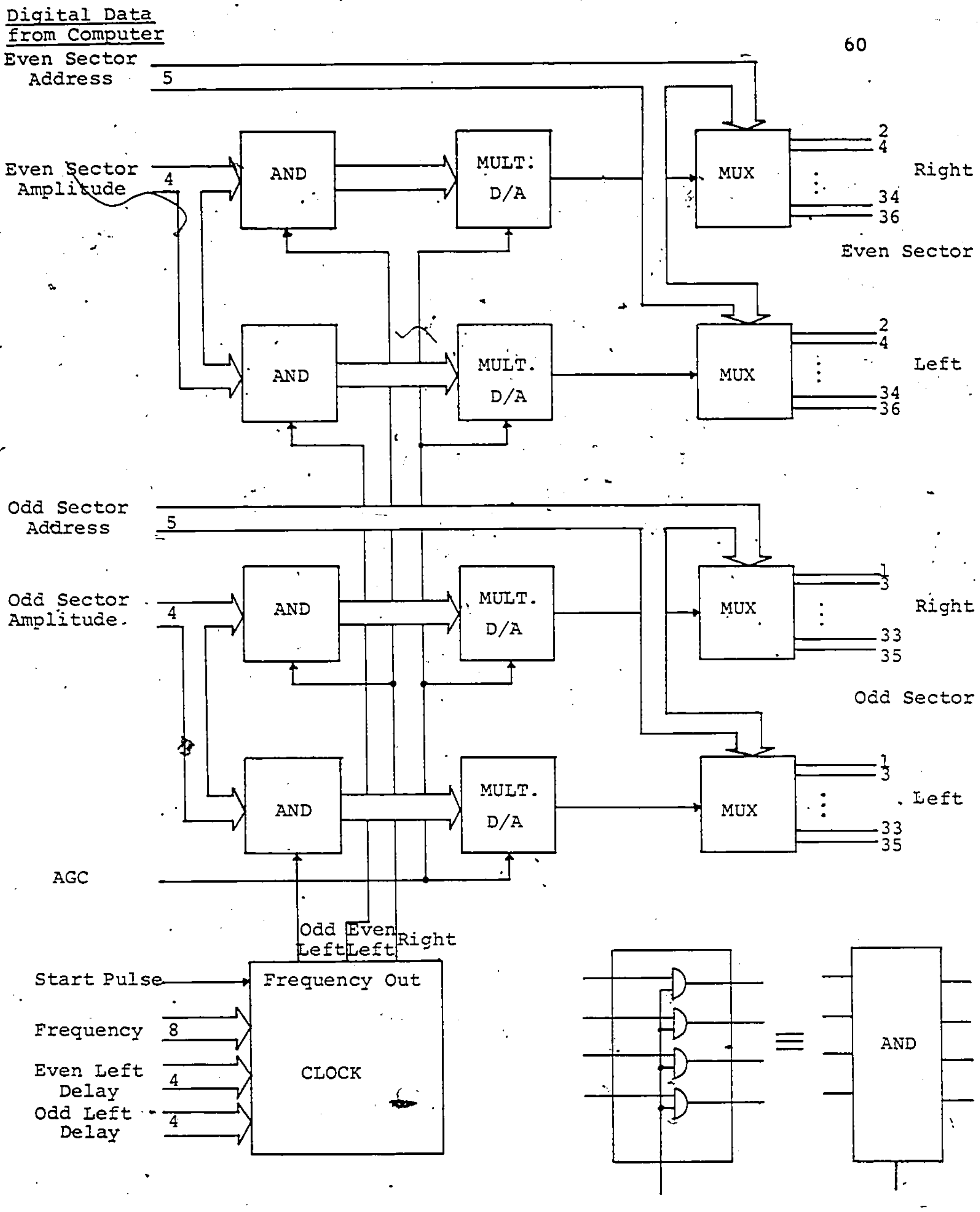


Figure 19: Block Diagram of Hardware Design

4.2 CLOCK CIRCUIT

To the clock circuit the digital input data representing frequency; even left sector delay and odd left sector delay are applied. The start pulse initiates the simulation of the return signal; in other words, it starts the generation of the output frequency signals of the clock circuit. The block diagram of the clock circuit is shown in Figure 20.

The resolution required to display the correct doppler frequency is approximately 100 nanoseconds. Subsequently, a 10 megahertz crystal is needed in the clock circuit. The digital frequency data from the computer is calculated to generate the signal return frequency modified by the doppler shift. The hardware is designed in a way that this digital data is used to count the 10 megahertz crystal down to the required return frequency of the signal. This frequency is the same for all right and left half beams, but the signal representing this frequency should be delayed appropriately to generate different frequency signals representing the right and left half beams. The clock circuit uses the synchronous digital counters [12]. The frequency data obtained from the computer is the same for all counters (COUNTER 1, COUNTER 2 and COUNTER 3).

The phase difference of the right and left half beams is represented by the relative time delay of the two half beams. The signal representing the frequency is therefore delayed to generate two signals of the same frequency to account for the relative time delay of the two beams. A constant 30 microseconds delay is determined for the time delay of both the right half beams. This is done to avoid a possible negative time delay. Thus, variable time delays of the left half beams are adjusted to indicate the relative time delays between the right and left half beams. Now the other digital counters (COUNTER 4, COUNTER 5 and COUNTER 6) are responsible to generate these time delays. The 10 megahertz crystal is also designed to be the clock for these digital counters. Taking the right half beam as an example, the counters (COUNTER 4) are preset to provide a 30 microseconds count from the 10 megahertz clock. All the counters (COUNTER 1 and COUNTER 4) associated with the right half beam have to be interconnected to provide the proper timing of the output signal of the frequency. Line 'a' indicates the enable line from COUNTER 4 to COUNTER 1 after a 30 microseconds delay is created. The counters for the other frequency outputs are similarly connected. Instead, the counters receive the variable time delay digital data of the two left half beams from the computer. The individual parts of the clock circuit will be further explained in the following sections.

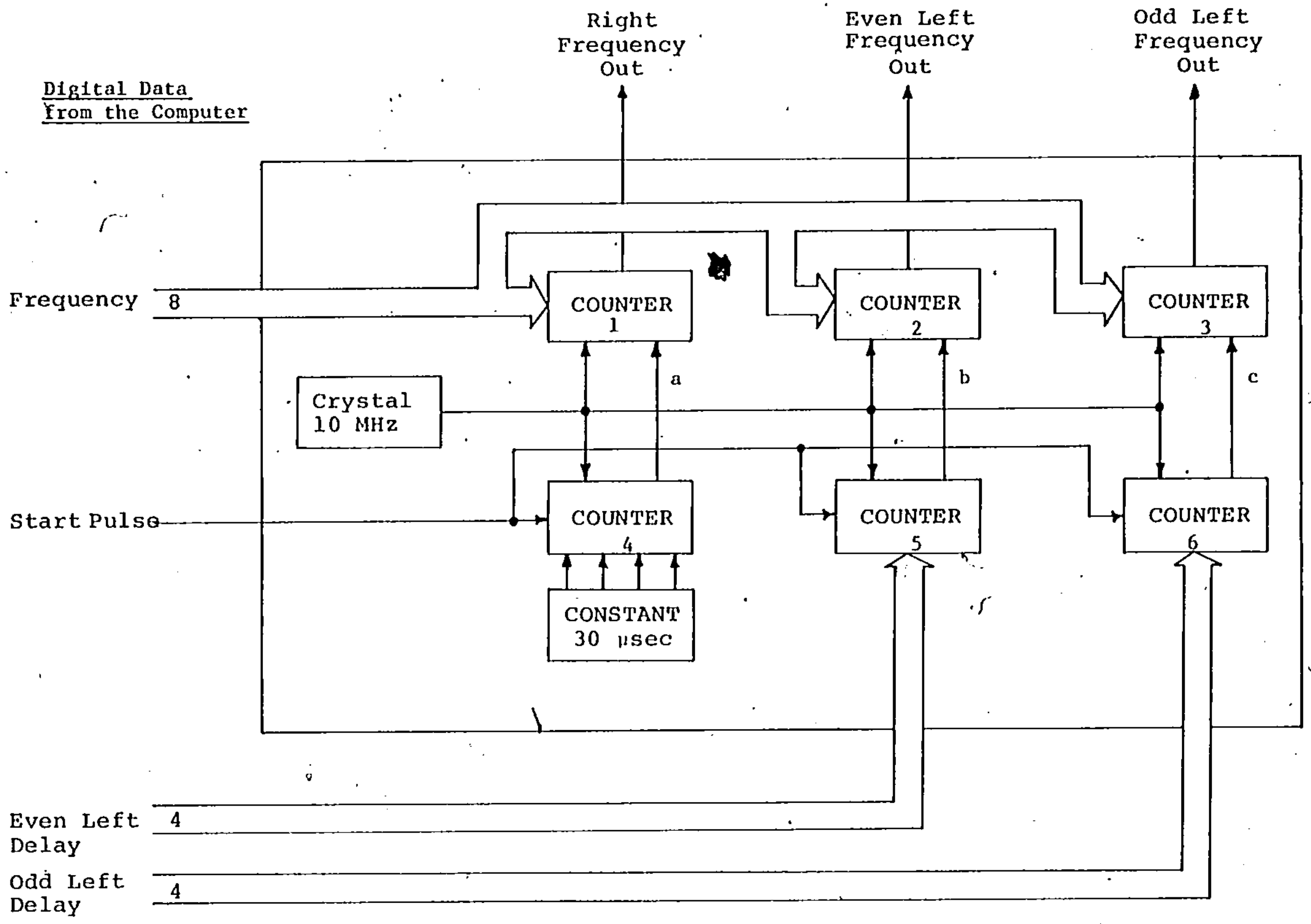


Figure 20: Block Diagram of Clock Circuit

4.2.1 Implementation of 10 Megahertz Frequency

The '10 MHz Crystal' in Figure 20 is to generate a 10 megahertz frequency in synchronization with the enable pulse from the computer. This enable pulse is the start signal to simulate the return signal.

The implementation is shown in Figure 21. The IC 74624 is a voltage-controlled oscillator, to which the 10 megahertz crstal is clamped. Pin 6 is the output providing a 10 megahertz frequency and pin 8 is the complementary output. The 10 megahertz is the clock frequency of the flip-flop 7474 which is used to synchronize the frequency with the 'Enable Pulse'. Figure 22 shows the timing diagram. The 'Enable Pulse' can appear at any time instant from the computer. The outputs now go to the other circuits to generate the frequency and the time delays.

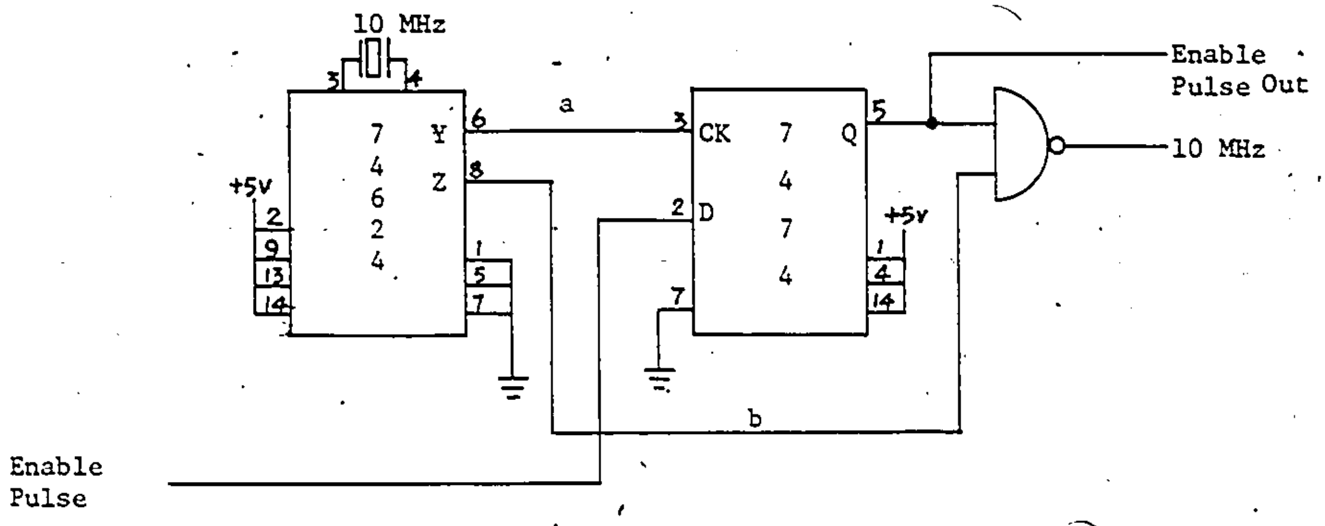


Figure 21: Implementation of the 10 MHz Frequency Source

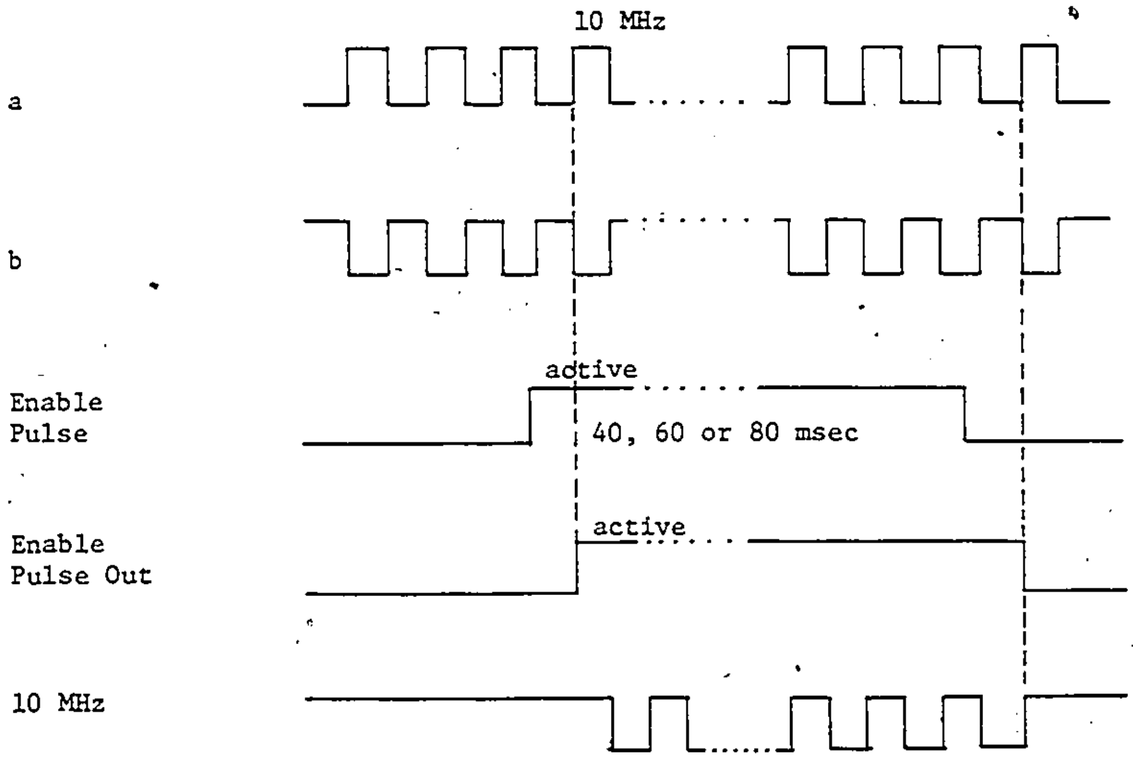


Figure 22: Timing Diagram for the 10 MHz Frequency Source

4.2.2 Implementation of the Time Delays

As was mentioned in Section 3.2.2.3, the delays can appear anywhere between 0 and 75 microseconds, with a 5 microseconds resolution. The block 'COUNTER 5' in Figure 20 is taken as an example, since 'COUNTER 4' is just a particular case with a constant 30 microseconds delay, and 'COUNTER 6' is an equivalent block with different input data.

The implementation to obtain a particular time delay is shown in Figure 23. The digital data generated from the computer is to produce a time delay between 0 and 75 microseconds with a 5 microseconds resolution. Therefore, the circuit is implemented in such a way as to generate a 5 microseconds delay in the first place, then the 200 kilohertz, which corresponds to 5 microseconds, is in turn the clock frequency of the other counter to generate the required time delay. The synchronous 4-bit up/down counters 74193 are used to count down the 10 megahertz frequency. The first two counters are used to generate a 5 microseconds time delay. Thus, these two counters are preset at a decimal value of 50, which is used to count the 10 megahertz down to 200 kilohertz, which corresponds to 5 microseconds. This configuration is only constructed once, since the output indicated by '200 KHz' can be shared by the blocks 'COUNTER 4' and 'COUNTER 6' in Figure 20. The 'borrow' output from the first counter, which is associated with the preset least significant bits, is connected to the 'count down' input of the second counter. This connection may provide a proper division for the 10 megahertz, since only a single inverted pulse for the 'borrow' is generated, whenever the first counter counts down to zero. The 'borrow' in turn is the clock pulse to count down the second counter. The ultimate inverted pulse generated by the 'borrow' output of the second counter is fed back for the loading mechanism of the preset inputs to generate a 5 microseconds time interval repetitively. Now the 200 kilohertz is the frequency of the third 74193 counter. This counter will receive a 4-bit datum representing the delay of the left half beam of the even sector. The counter will generate the required time delay between 0 and 75 microseconds through the 'borrow' output. Again, an inverted pulse is produced. Now, a set-reset flip-flop is used to produce the enable signals to the other circuit for the generation of the frequency.

The timing diagram is shown in Figure 24. A digital input is taken, for example, to be 0010. Therefore, a 10 microseconds delay is created, as shown in 'Borrow 2'. The flip-flop is thus set to provide enable signals to Figure 25, which generates the required frequency.

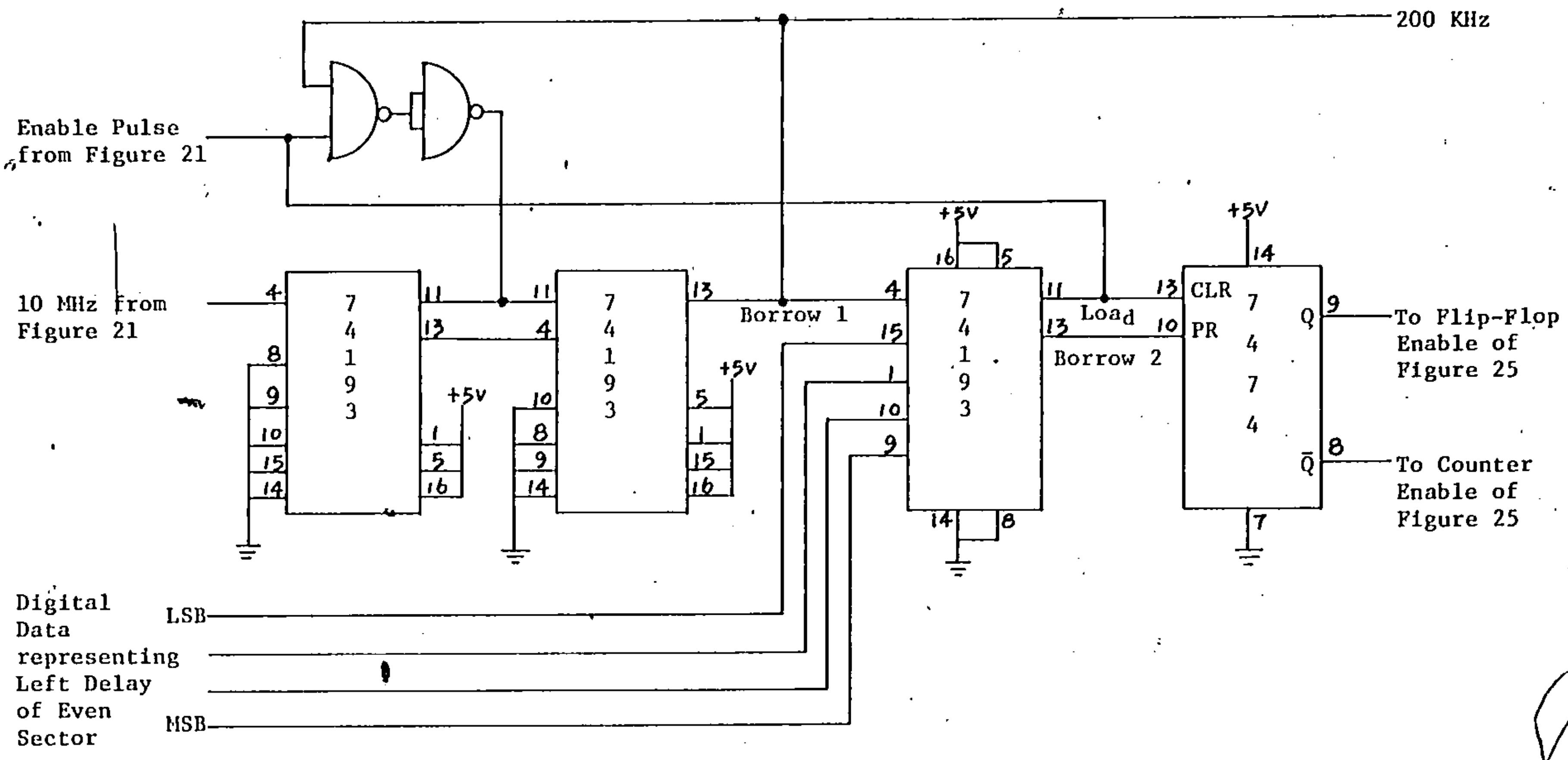

200 KHz
Enable Pulse from Figure 21
10 MHz from Figure 21
7 4 1 9 3
7 4 1 9 3
7 4 1 9 3
7 4 7 4
+5V
Borrow 1
Load
Borrow 2
CLR
PR
Q
Q̄
To Flip-Flop Enable of Figure 25
To Counter Enable of Figure 25
Digital Data representing Left Delay of Even Sector
LSB
MSB

Figure 23: Implementation of the Time Delay of the Left Half Beam of the Even Sector

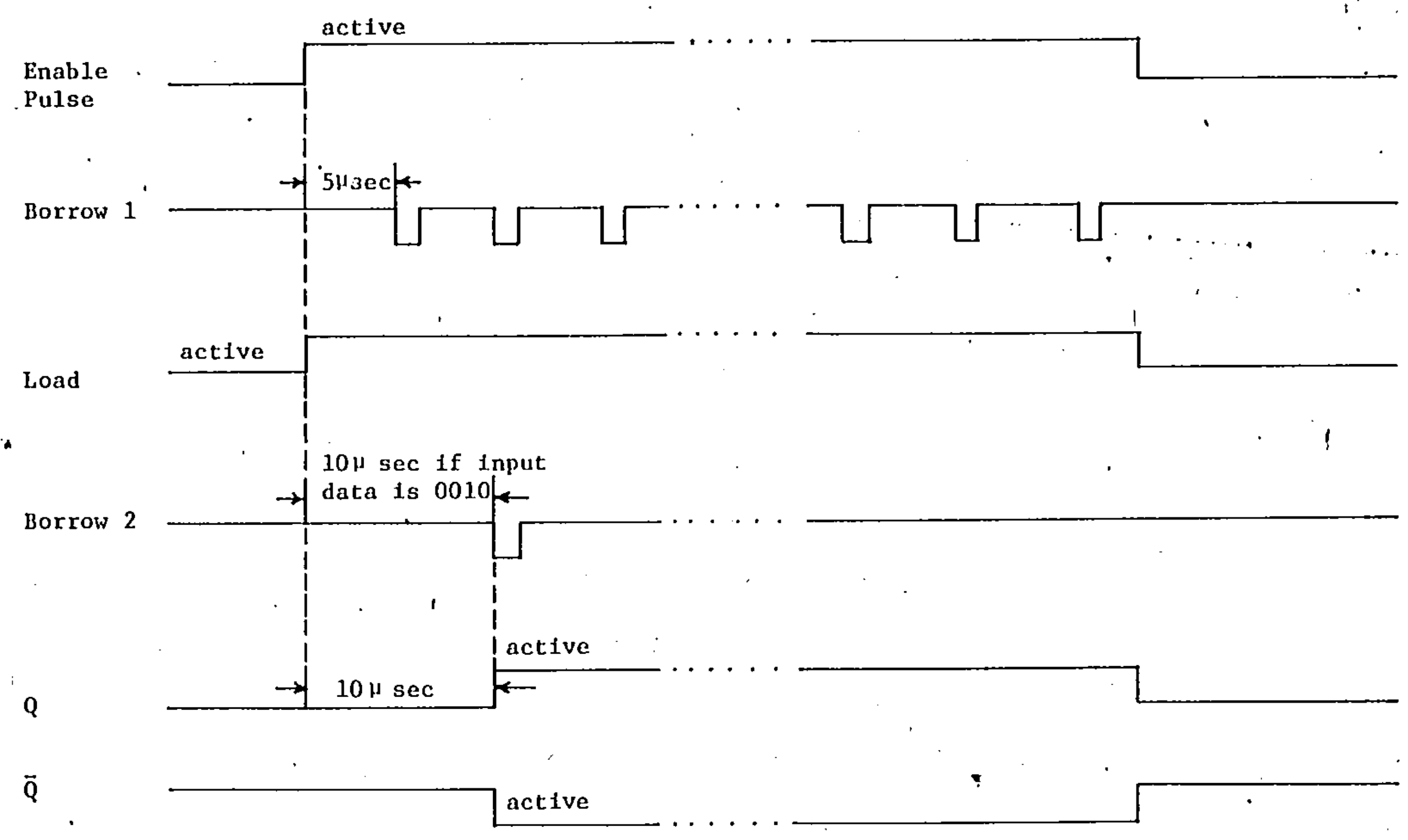


Figure 24: Timing Diagram for the Time Delay

4.2.3 Implementation of the Return Signal Frequency

In Section 3.2.2.3, the data computed representing the frequency is to count the 10 megahertz down to twice the required frequency. This frequency is then divided by two to obtain the required frequency. This is done to obtain even duty cycles of the required frequency since the borrow output of the counter only gives one inverted pulse for every time it counts down to zero. The return signal frequency range is 7.2 KHz + 250 Hz. Twice this frequency range is from 13.9 KHz to 14.9 KHz. The number to divide the 10 MHz to the above double frequency range is from 671 to 719. Twelve bits are needed to code these numbers. However, the four most significant bits are always the same. Thus eight bits are sufficient.

The circuit to generate the return signal frequency is shown in Figure 25, which is equivalent to the block 'COUNTER 1' in Figure 20. In fact, COUNTER 1, COUNTER 2 and COUNTER 3 are all equivalent since the same frequency output is to be provided. 74193 counters are used to count down the 10 megahertz frequency. The 'borrow' output from the first counter, which is associated with the least significant bits, is connected to the 'count down' input of the second counter. The second counter is similarly connected to the third counter, which is associated with the four most significant bits. This connection may provide a proper division for the 10 megahertz. In addition, the input data should be loaded into ~~the~~ counters at the same time. The ultimate inverted pulse generated is from the counter which is associated with the most significant bits. This pulse is fed back for the loading mechanism and the inverted pulse is repeatedly generated to create the required frequency. This will be further explained in the timing diagram. At this point, the flip-flop 7474 is used to divide down the frequency obtained to one-half. The flip-flop is enabled when the count down for the corresponding time delay (from Figure 23) is achieved. Also, the 'Counters enable' is active at the same time. This is to ensure that the required frequency is produced only after a certain time delay, which is generated by the time delay circuit.

The timing diagram is shown in Figure 26. The counters and the flip-flop are enabled as soon as the time delay is achieved. The digital input data are repetitively loaded into the counters whenever a ultimate 'borrow' is created. This continuous 'borrow' signal represents a frequency which is twice the required frequency. The double frequency is then divided by two to obtain the required frequency.

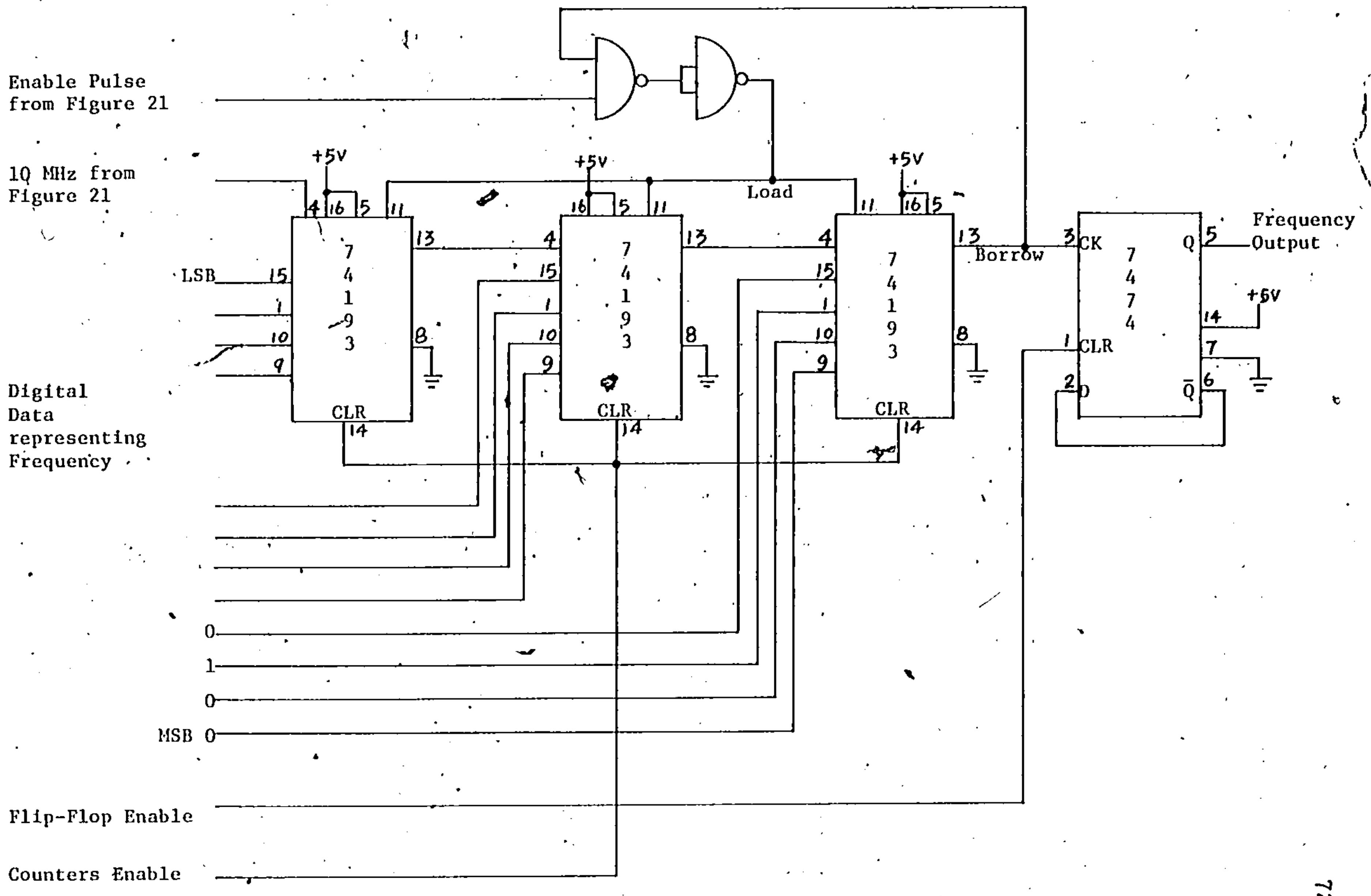


Figure 25: Implementation of the Return Signal Frequency

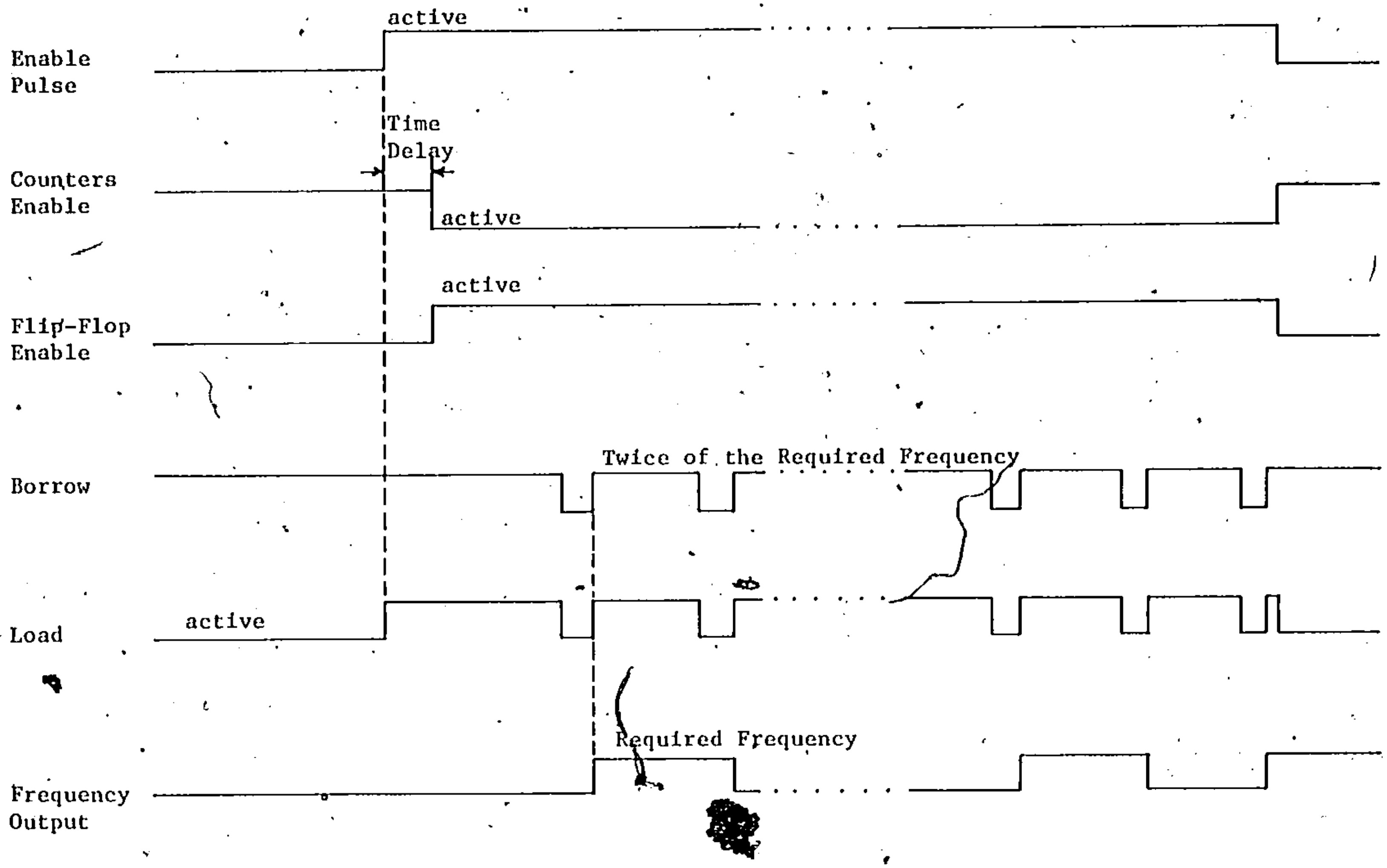


Figure 26: Timing Diagram for the Return Signal Frequency

4.3 MULTIPLYING DIGITAL-TO-ANALOG CONVERTER (MDAC) AND MULTIPLEXING CIRCUIT

Referring to Figure 19, the MDAC and the multiplexing circuits have been shown in block diagram form. There are four similar circuits constructed to interface with the sonar system. Thus, only one of the four parallel circuits is to be discussed here. This is shown in Figure 27.

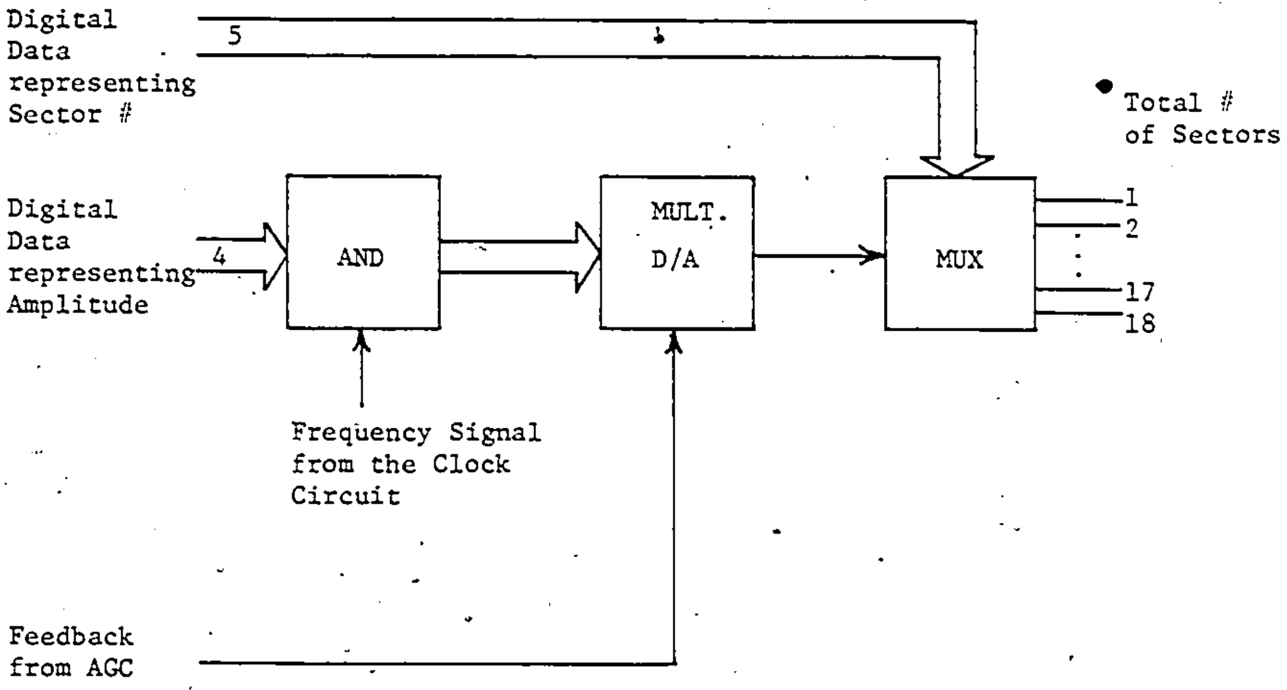

Digital Data representing Sector #
5
Total # of Sectors
Digital Data representing Amplitude
4
AND
MULT. D/A
MUX
1
2
17
18
Frequency Signal from the Clock Circuit
Feedback from AGC

Figure 27: Block Diagram of Multiplying Digital-to-Analog Converter and Multiplexing Circuit

Four bits of amplitude data from the computer are clocked into the MDAC according to the frequency generated by the clock circuit. The amplitude generated by the MDAC is determined from the digital input data and the AGC (Automatic Gain Control) feedback from the sonar system. The signal determined is then injected into one of the thirty-six circuits of the sonar system, by means of a multiplexing circuit. Referring to Section 4.1, since the sectors are grouped into even and odd numbers, eighteen channels require addressing by 5 bits of data from the computer.

4.3.1 Implementation of the Multiplying Digital-to-Analog Converter (MDAC)

The implementation of the MDAC [12], [13], [14] is shown in Figure 28. The 4-bit amplitude data is clocked into the MDAC according to the signal generated by the clock circuit. The AND Gate 7408 replaces the block indicated by 'AND'. The MC1508 MDAC output current is the linear product of the 4-bit digital word and the analog reference voltage. Voltage outputs are obtained by using an external operational amplifier (A741) as a current to voltage converter. The operational amplifier generates a positive voltage limited only by its positive supply voltage. The unity gain operational amplifier (TL072) serves as a buffer for the reference voltage of the D/A converter. The variable resistor is adjusted so that the required maximum input signal level of the AGC is 2.5 volts.

The signal that appears at the output of the circuit is shown in Figure 29. Either one of the three envelope shapes will be injected into the sonar system.

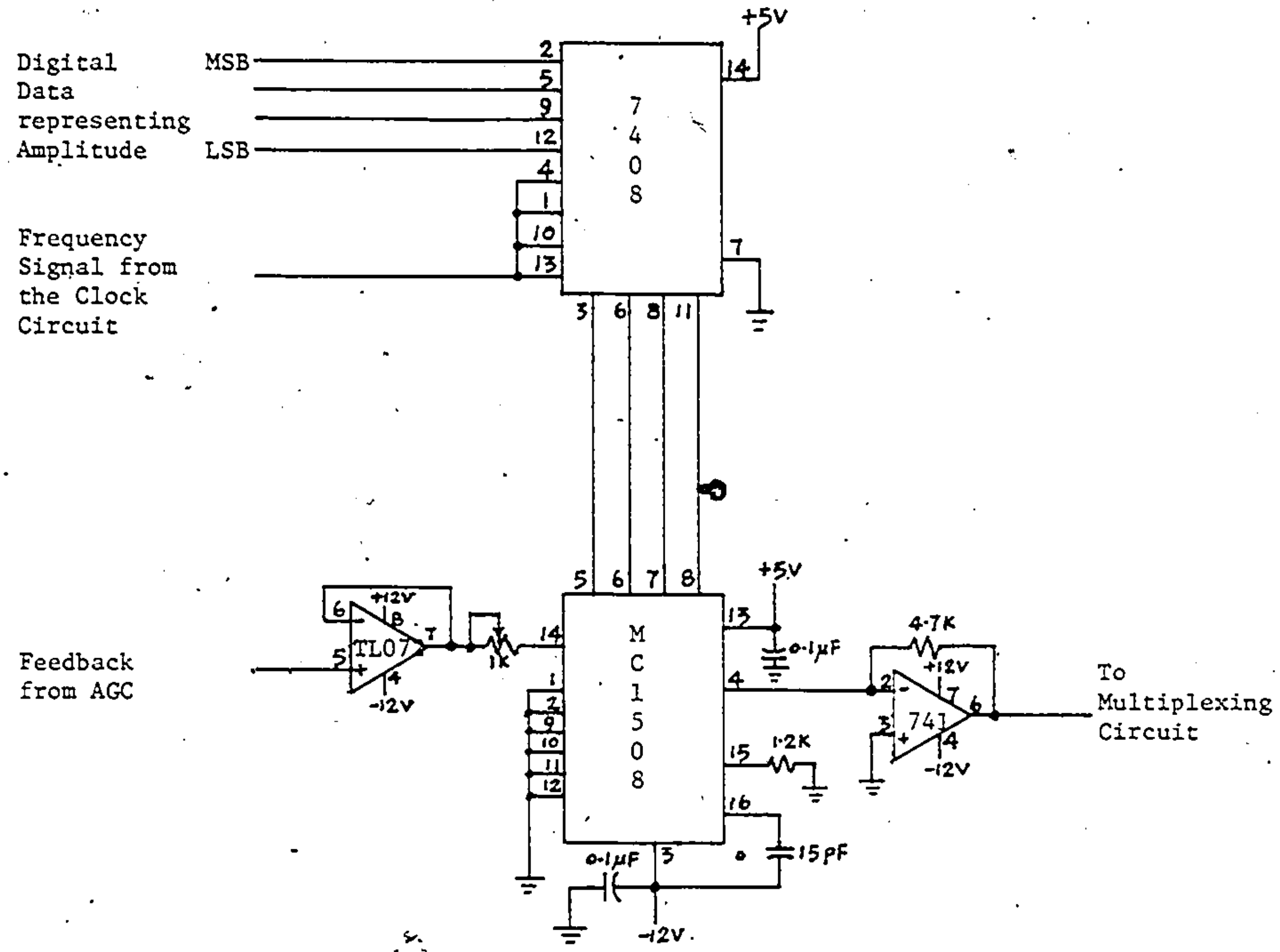
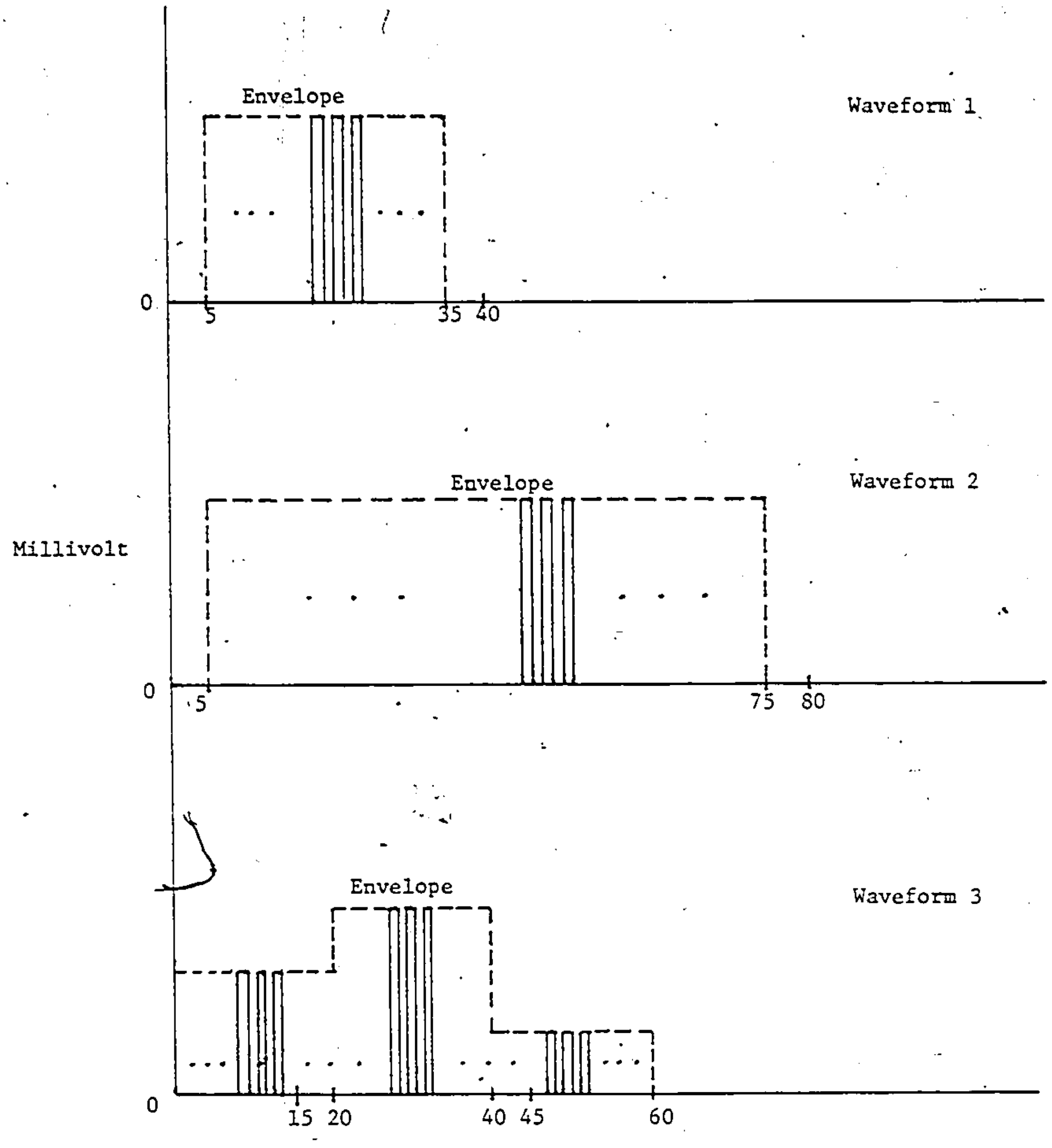


Figure 28: Implementation of the Multiplying D/A Converter (MDAC)



Time in Milliseconds, Period corresponding to the Return Frequency

Figure 29: Implementation of the Injected Signal into the Sonar System

4.3.2 Implementation of the Multiplexing Circuit

The block diagram of the multiplexing circuit is shown in Figure 30. The three least significant bits c_3, c_4, c_5 of the control inputs is to select one of the outputs from each of the multiplexers. The decoder, which is controlled by c_1 and c_2, has three output lines to enable one of the three multiplexers. Therefore, only one of the eighteen outputs is selected at one time.

The implementation of the multiplexing circuit [12], [13], [14] is shown in Figure 31. The two most significant bits of the address go to the decoder 74138 to enable one of the three multiplexers (MC14051), while one of the eight outputs of a multiplexer is selected using the remaining three least significant bits of the sector address. Therefore one of the eighteen sectors is selected at one time depending on the sector address. The level shifter (MC14504B) will shift a TTL signal to CMOS logic levels for any CMOS supply voltage between 5 and 15 volts.

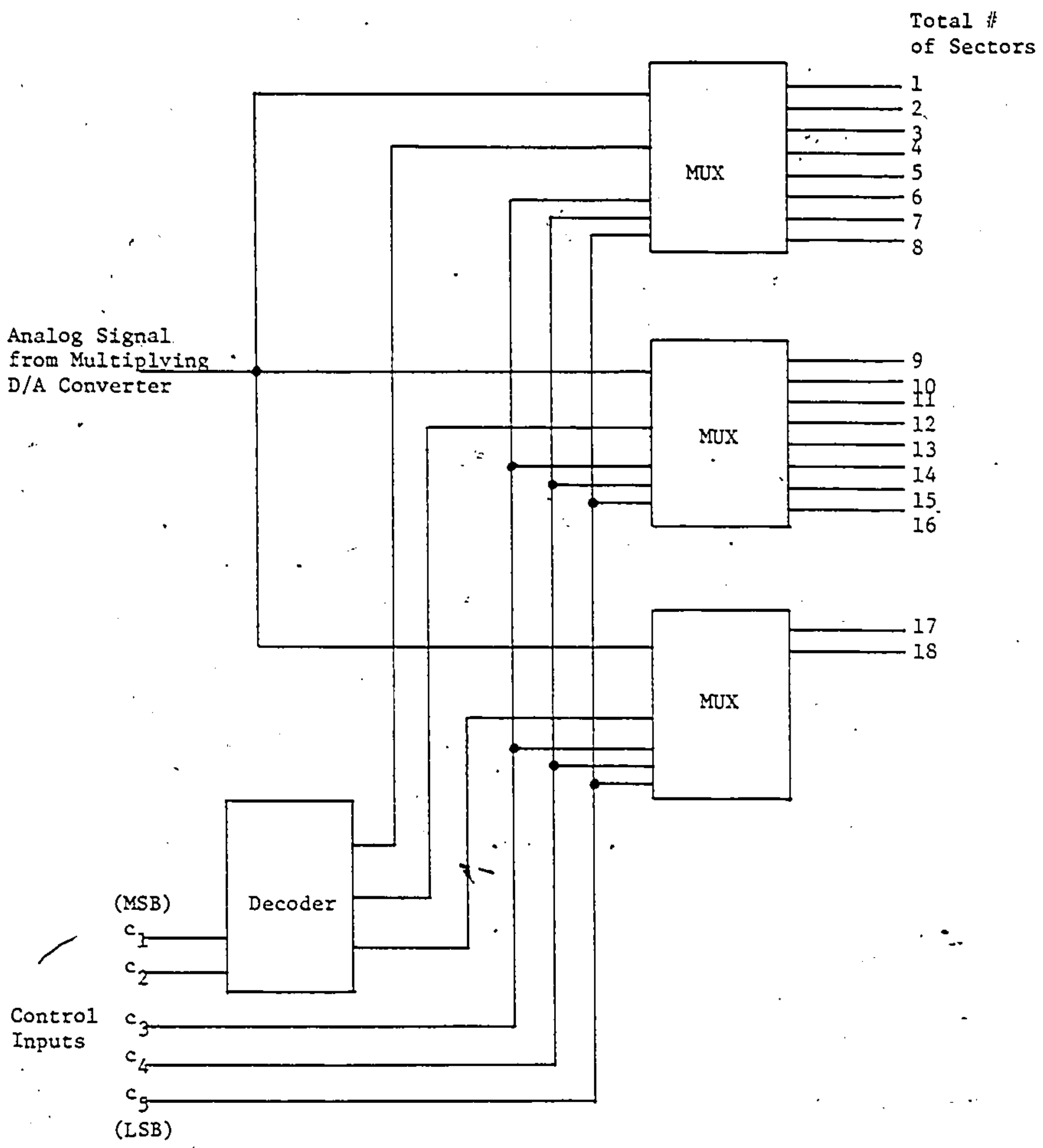


Figure 30: Block Diagram of the Multiplexing Circuit

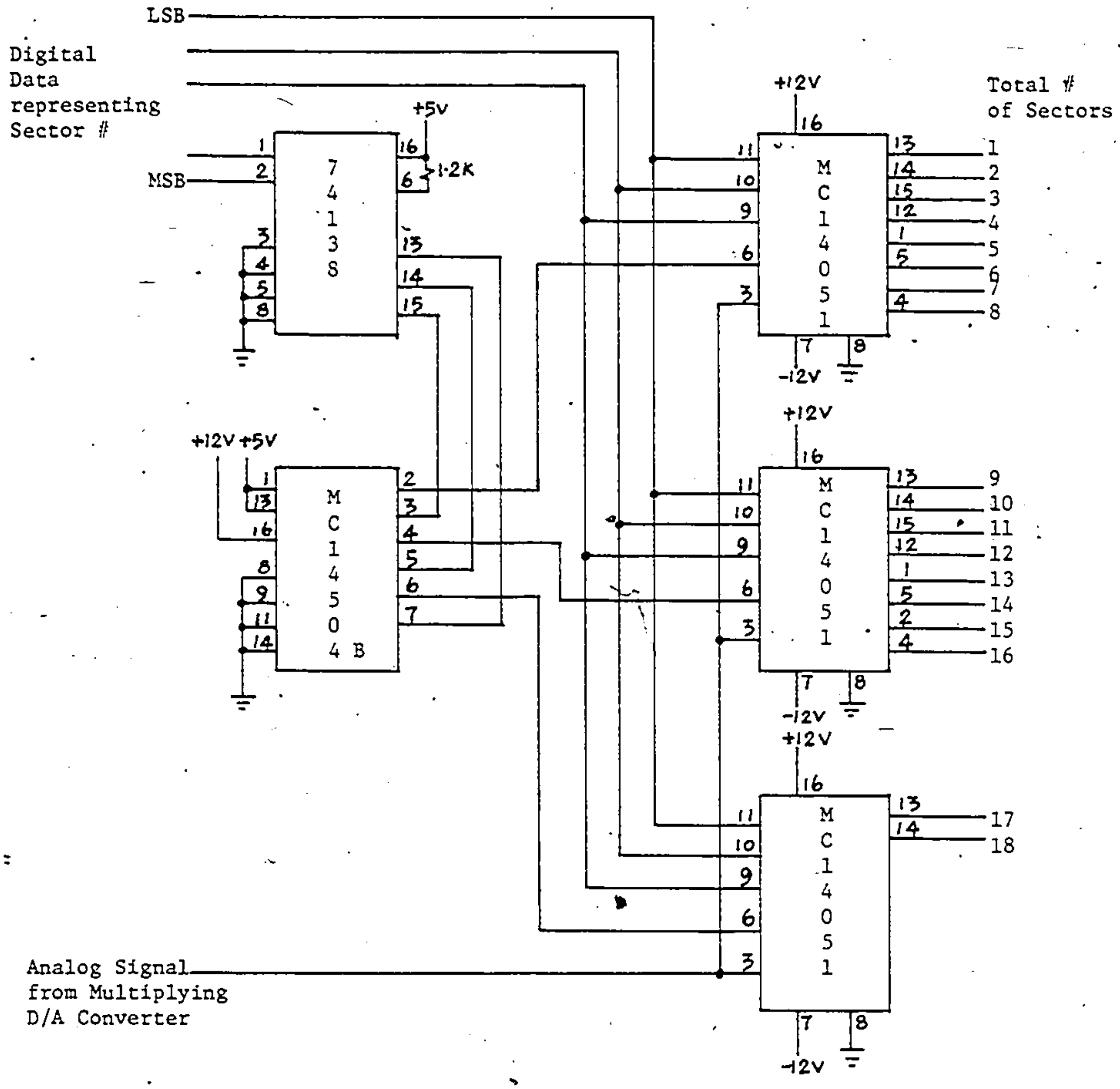


Figure 31: Implementation of the Multiplexing Circuit

4.4 IMPLEMENTATION OF SYNCHRONIZATION PULSES

The synchronization pulses serve two purposes in the system design. Firstly, they are the input to the SID (Serial Input Data) of the CPU (Central Processor Unit) for the subroutine REPTIM. Secondly, they are the interrupt signal to the input RST 6.5 of the CPU. Therefore, they must be implemented and hardwired to the above mentioned two inputs of the CPU. A monostable multivibrator (one-shot) is used to generate the required synchronization pulses of width approximately 1.2 milliseconds. Figure 32 illustrates this.

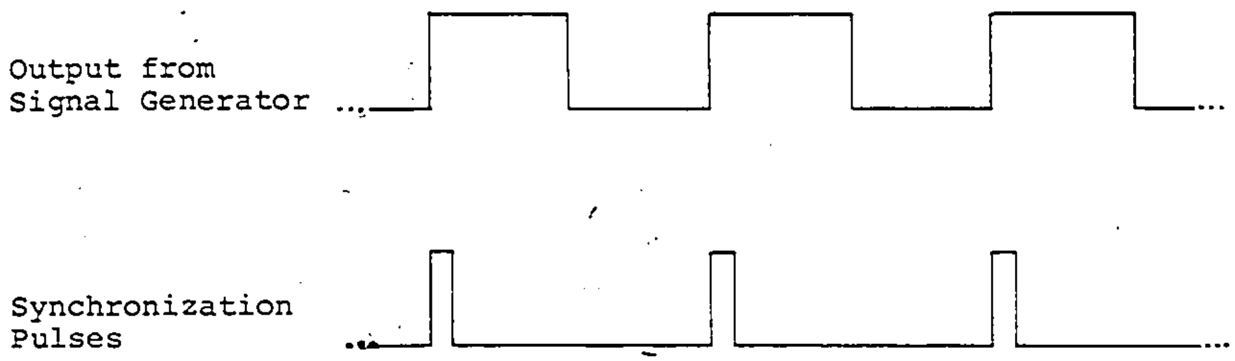


Figure 32: Implementation of Synchronization Pulses

Chapter V
CONCLUSIONS

Experiments have been conducted on the system at the Fleet School Halifax using the constructed package. The simulation concept was successfully demonstrated.

Seventy-two cables, which carry the simulated half beam signals, were connected from the simulator outputs to the injection points of the sonar system. The suitability of the proposed injection points was confirmed. The simulator provides the required signal to any one of the thirty-six Preformed Beam (PFB) cards at the correct times and at a frequency modified by the doppler effect.

The programme, stored in the SDK-85 microcomputer memory, reacts to the input data which results in the generation of the digital signals, and is followed by the D/A conversion and the display of the target on the display screen.

The package was tested both at the University of Ottawa and at the Fleet School, and possesses the following capabilities:

(1) The ability to calculate the return signal, given the envelope required to represent the multipath effect.

(2) The ability to present the signal at ranges of 800 to 32000 yards.

(3) The ability to follow a moving target from one beam to the adjacent beam with appropriate gradual diminuation in one beam and increase in the adjacent beam.

At the Fleet School Halifax, the target was displayed on the display screen with various sets of input data to the computer. Signal parameters were measured in the sonar system in terms of the doppler frequency and the range. However, photographs were not allowed to be taken as objective measures.

In the future. the software package is to be redeveloped for use with the Sperry Univac computer (AN-UYK 502) which is commonly used by the Department of National Defense, Government of Canada. This computer was supposed to be used at the beginning of this project. However, due to some access problems, it was not functioning as required. For demonstration purposes, the Intel SDK-85 microcomputer was used instead to provide the return signals to the sonar system.

Appendix A

OVERALL SCHEMATIC DIAGRAMS

The overall schematic diagrams are shown in Figures 33, 34, and 35. The schematic diagrams of the clock circuit, which are shown in Figures 33 and 34, were built on two boards (A and C).

The schematic diagram of the multiplying digital-to-analog converter and multiplexing circuit is given in Figure 35. Four identical boards were built based on this configuration. Boards E, G, J and L describe the even right sectors, even left sectors, odd right sectors and odd left sectors respectively.

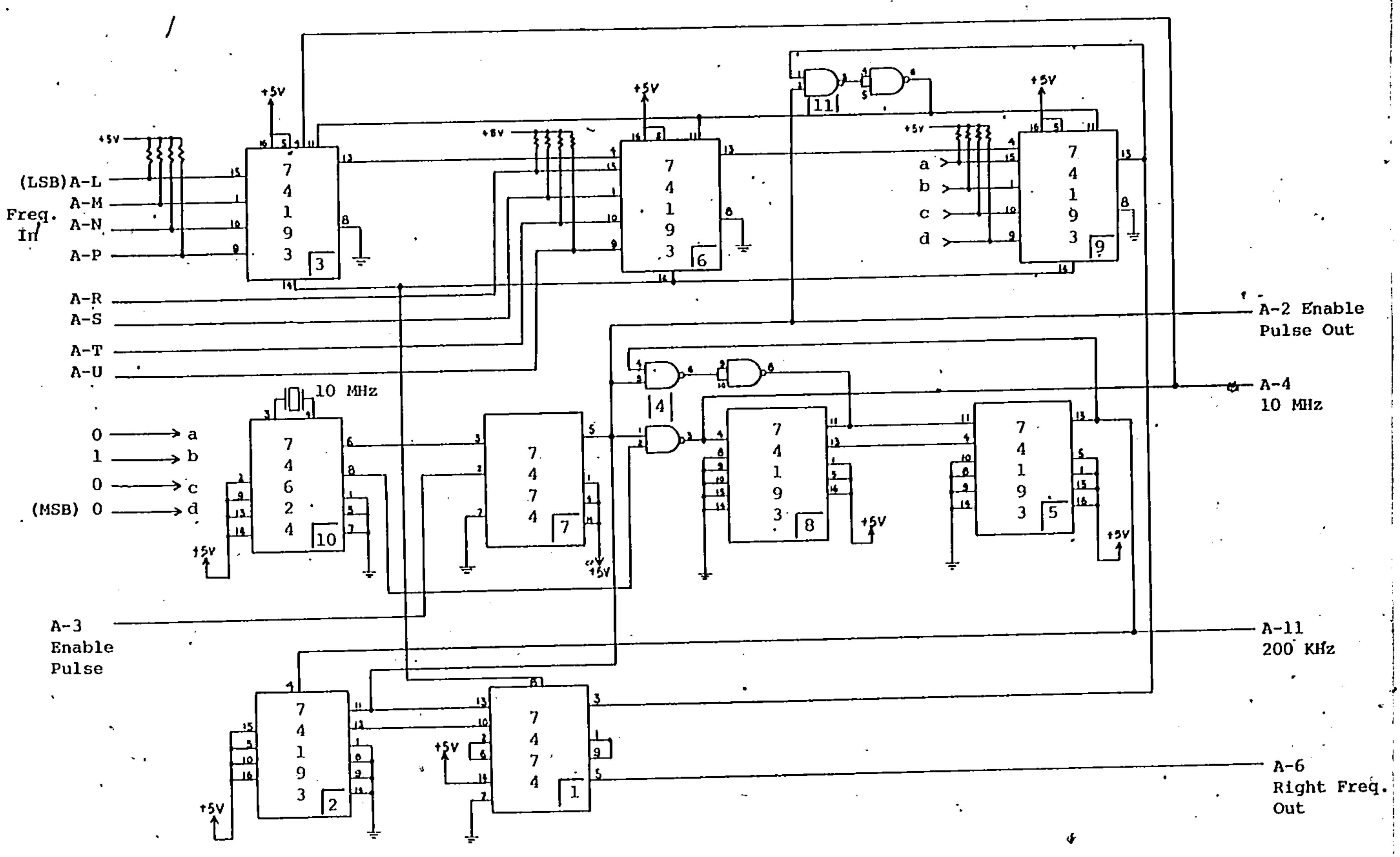
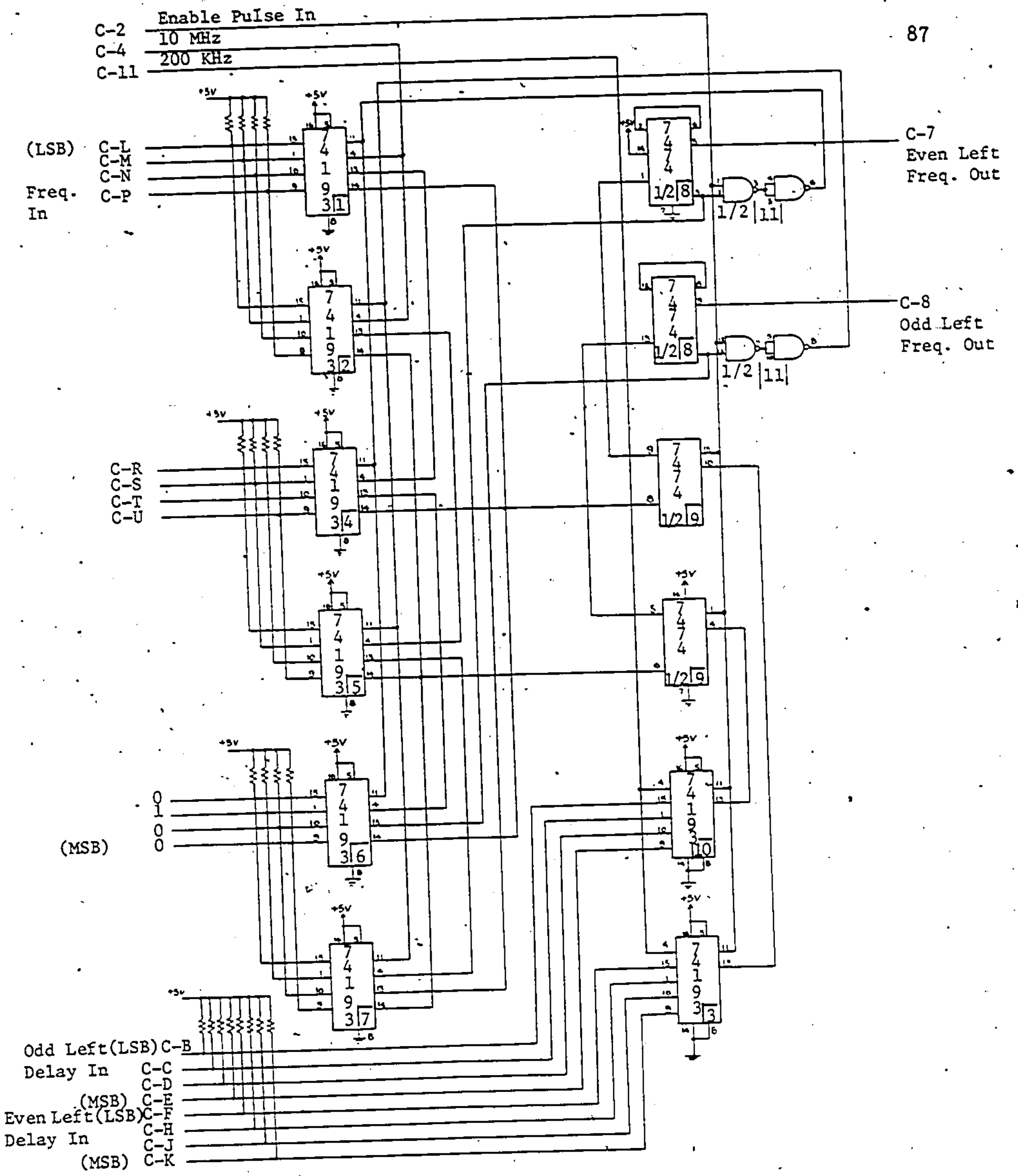
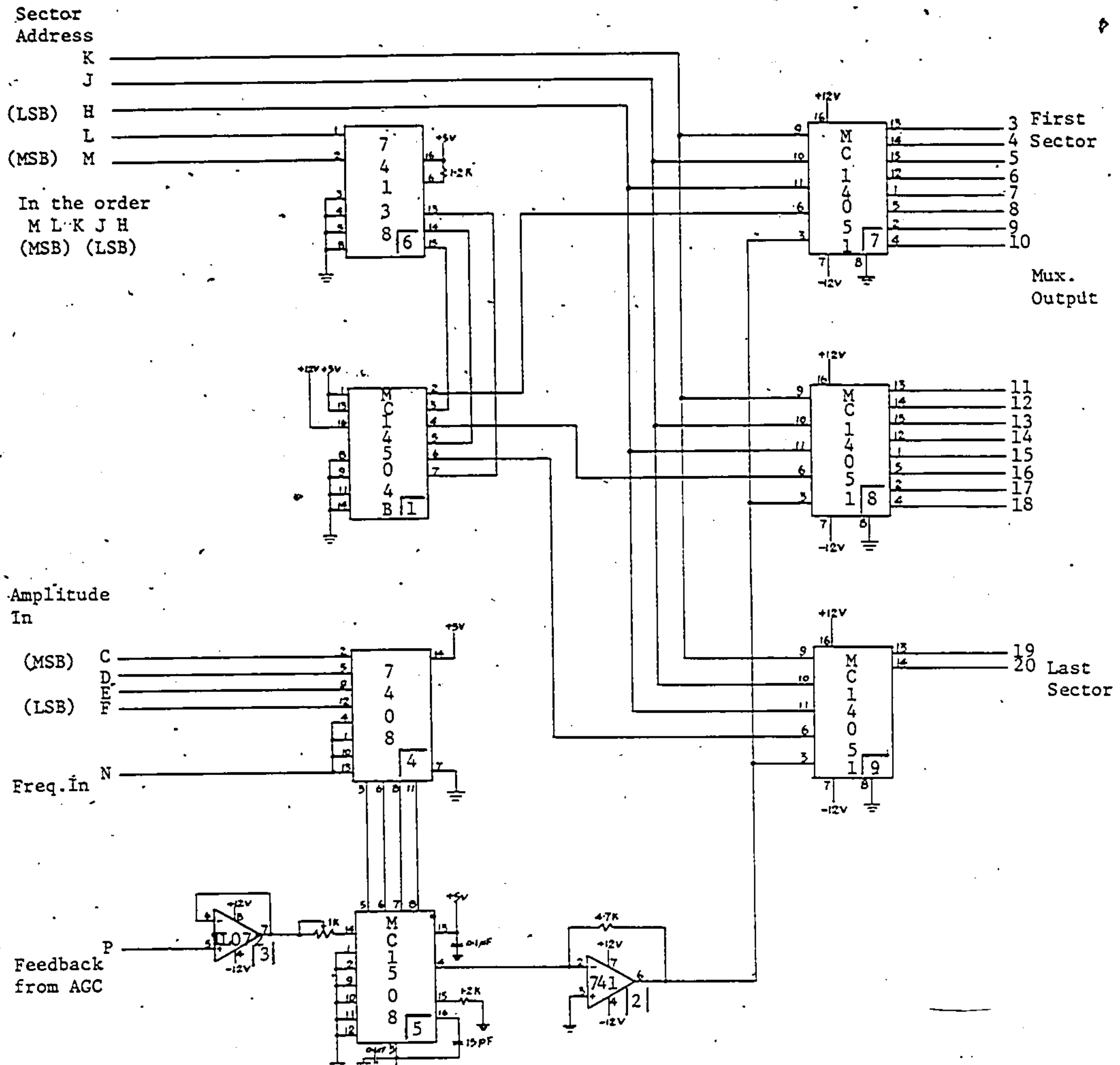


Figure 33: Schematic Diagram of Clock Circuit (Sheet 1 of 2)



BOARD C

Figure 34: Schematic Diagram of Clock Circuit (Sheet 2 of 2)



BOARDS E, G, J, L

Figure 35:Schematic Diagram of Multiplying D/A Converter and Multiplexer Circuit

Appendix B

HARDWARE INTERCONNECTIONS

The hardware was built on several boards in an enclosure. The SDK-85 microcomputer is mounted on the top; two ribbon cables are connected from the output ports to transfer data to the hardware. Two other cables from the hardware are used to interface with the sonar system.

Integrated Circuit (IC) chips layout of boards A, C, E, G, J and L are shown in Figures 36, 37 and 38. The number at the corner of each IC corresponds to the number used in the schematic diagrams.

Figures 39 and 40 show the interconnections between the ribbon cables and the boards. Sheet #1 shows the connections between the computer and the simulator hardware, while sheet #2 shows the connections between the hardware and the sonar system.

Figures 41 and 42 show the interconnections between the boards inside the enclosure. The PFB (Preformed Beam) and AGC (Automatic Gain Control) boards are also installed in the enclosure for testing purposes. However, when testing is completed, these will be removed out of the simulator.

The front view of the sonar signal simulator built is shown in Figure 43. The SDK-85 microcomputer is mounted on the top of the enclosure, in which all the constructed boards are installed.

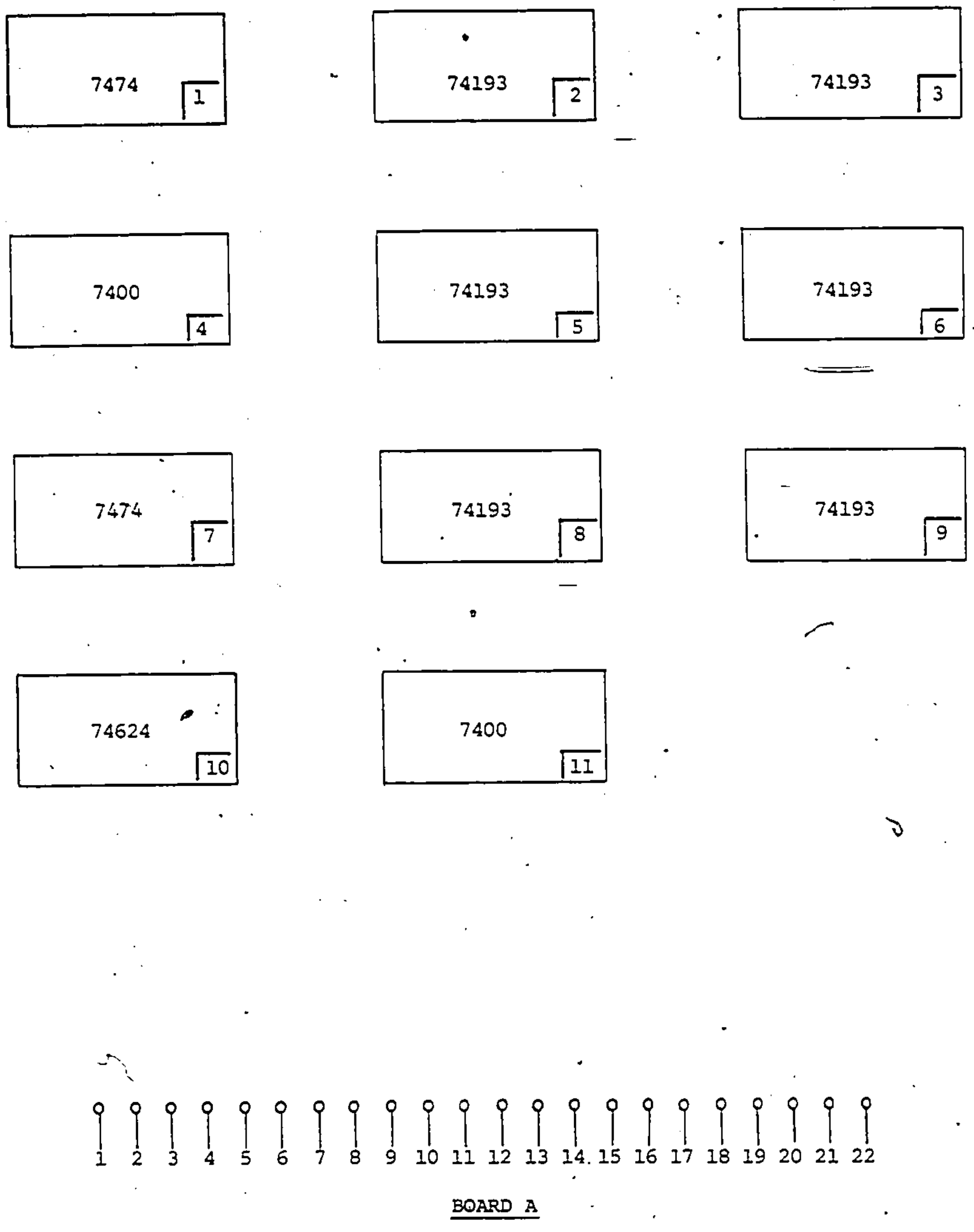
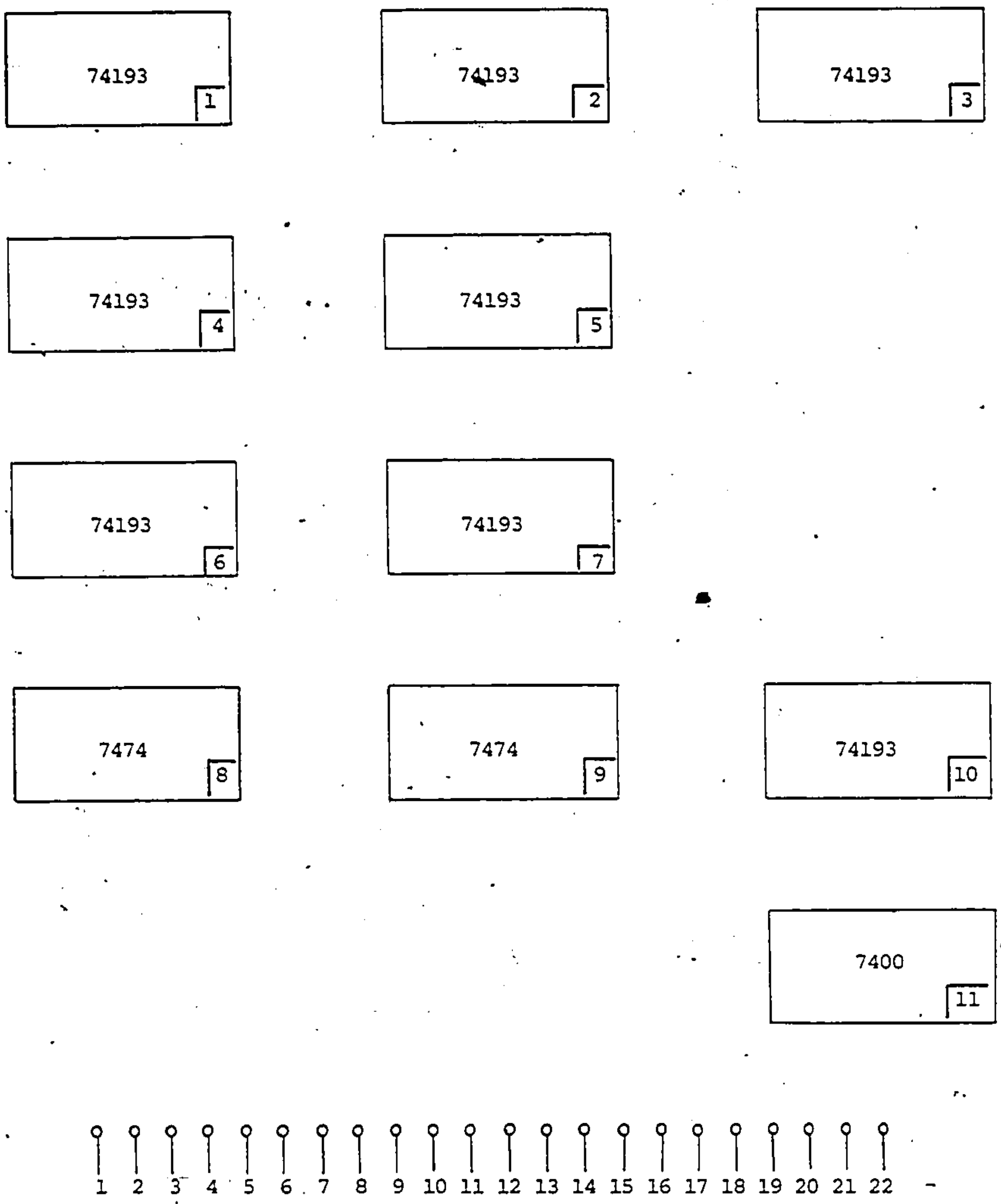
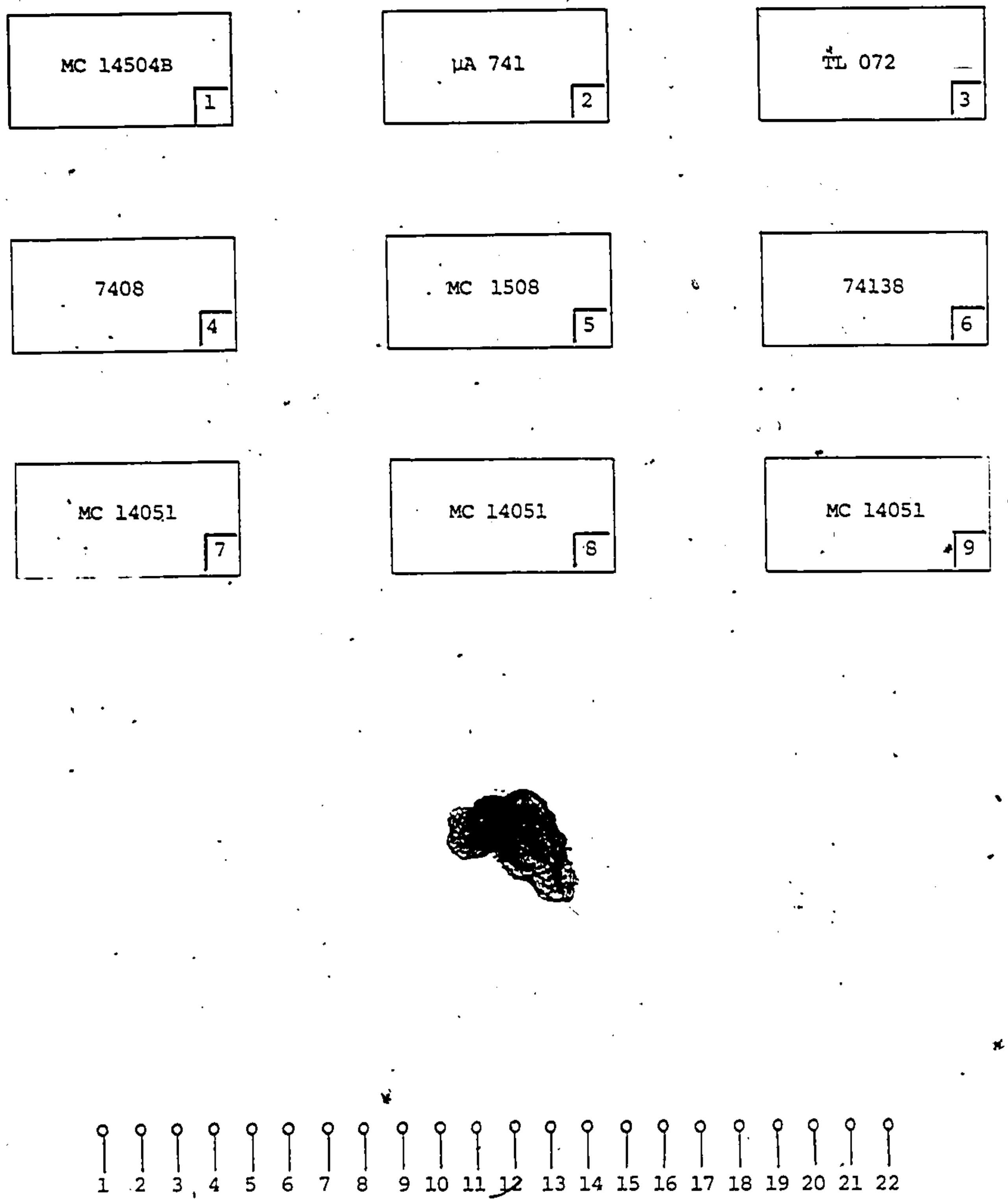


Figure 36: IC Chips Layout (Sheet 1 of 3)



BOARD C

Figure 37: IC Chips Layout (Sheet 2 of 3)



BOARDS E, G, J, L

Figure 38:IC Chips Layout (Sheet 3 of 3)

Ribbon Cable 1	Boards E,G	
1	L	Even Sector
4	M	
3	K	
6	J	
5	H	
8	C (MSB)	Even Amplitude
7	D	
10	E	
9	F (LSB)	
	Boards J,L	
12	C (MSB)	Odd Amplitude
11	D	
14	E	
13	F (LSB)	
	Board C	
16	K (MSB)	Even Left Delay
15	J	
18	H	
17	F (LSB)	
20	E (MSB)	Odd Left Delay
19	D	
22	C	
21	B (LSB)	
25	GND	
26	GND	

In the order
M L K J H
(MSB) (LSB)

Ribbon Cable 2	Boards J,L	
1	L	Odd Sector
4	M	
3	K	
6	J	
5	H	
	Boards A,C	
8	U	
7	T	
10	S	
9	R	
12	P	Frequency
11	N	
14	M	
13	L (LSB)	
25	GND	
26	GND	

Figure 39: Interconnection between the Ribbon Cables and the Boards (Sheet 1 of 2)

Ribbon Cable 3	Board E	
1	3	
2	4	
3	5	
4	6	Even
5	7	Right
6	8	Sectors
7	9	
8	10	
9	11	
10	12	
11	13	
12	14	
13	15	
14	16	
15	17	
16	18	
17	19	
18	20	

3 : First Sector

20: Last Sector

Ribbon Cable 4	Board J	
1	3	
2	4	
3	5	
4	6	Odd
5	7	Right
6	8	Sectors
7	9	
8	10	
9	11	
10	12	
11	13	
12	14	
13	15	
14	16	
15	17	
16	18	
17	19	
18	20	

	Board G	
19	3	
20	4	
21	5	
22	6	
23	7	Even
24	8	Left
25	9	Sectors
26	10	
27	11	
28	12	
29	13	
30	14	
31	15	
32	16	
33	17	
34	18	
35	19	
36	20	

	Board L	
19	3	
20	4	
21	5	
22	6	
23	7	Odd
24	8	Left
25	9	Sectors
26	10	
27	11	
28	12	
29	13	
30	14	
31	15	
32	16	
33	17	
34	18	
35	19	
36	20	

Figure 40: Interconnection between the Ribbon Cables and the Boards (Sheet 2 of 2)

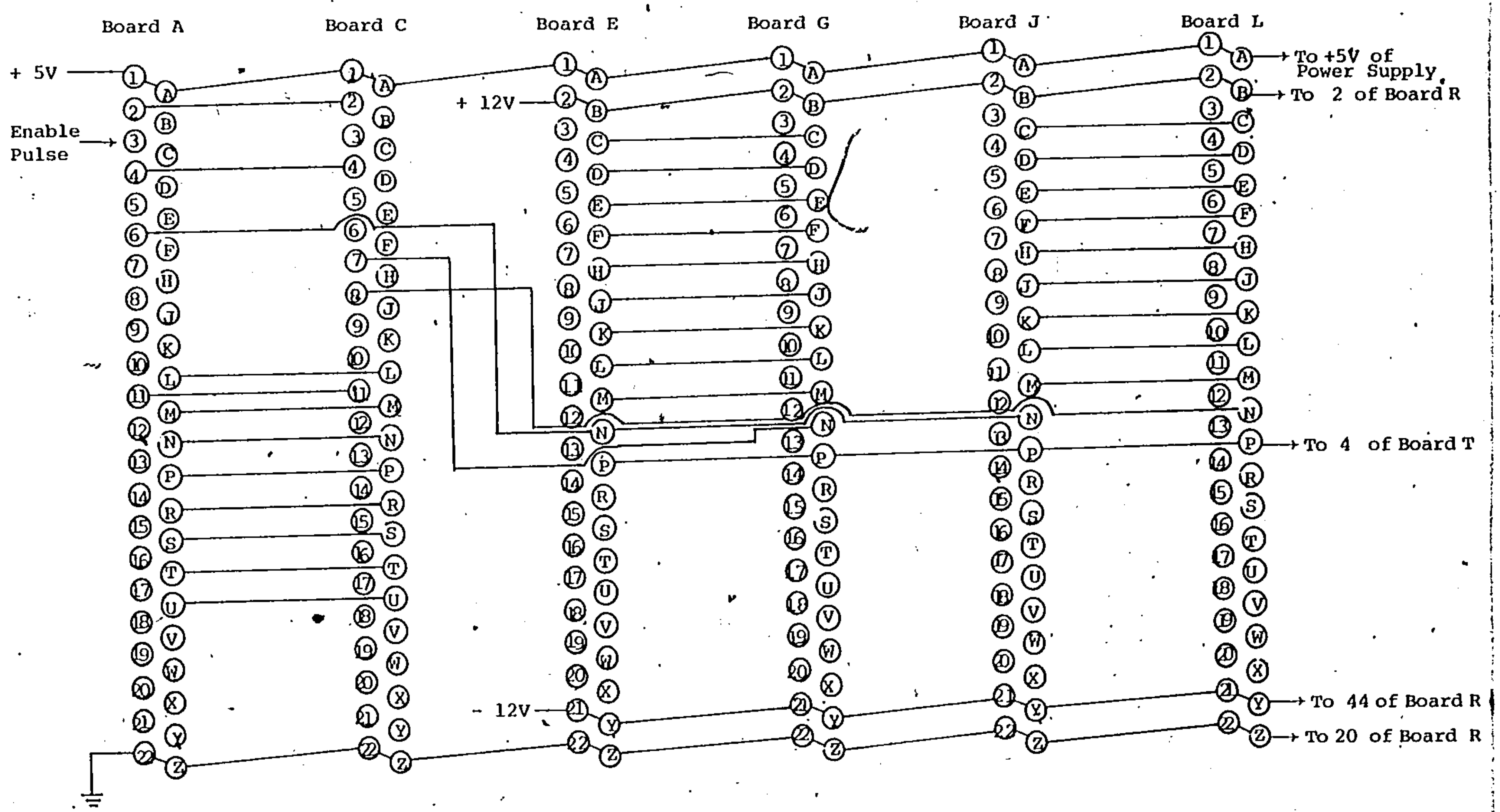


Figure 41: Interconnection between the Boards (Sheet 1 of 2)

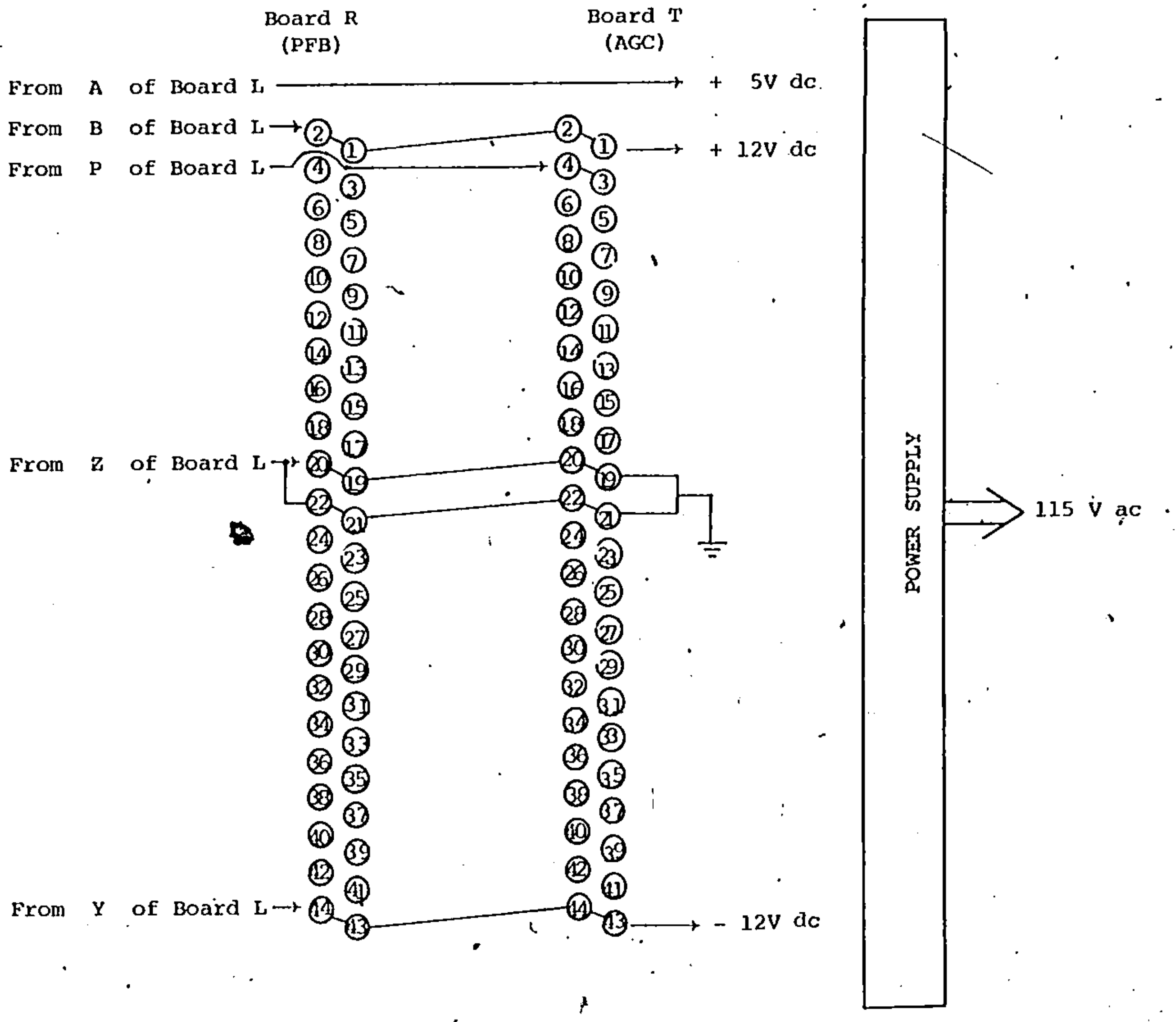


Figure 42:Interconnection between the Boards (Sheet 2 of 2)

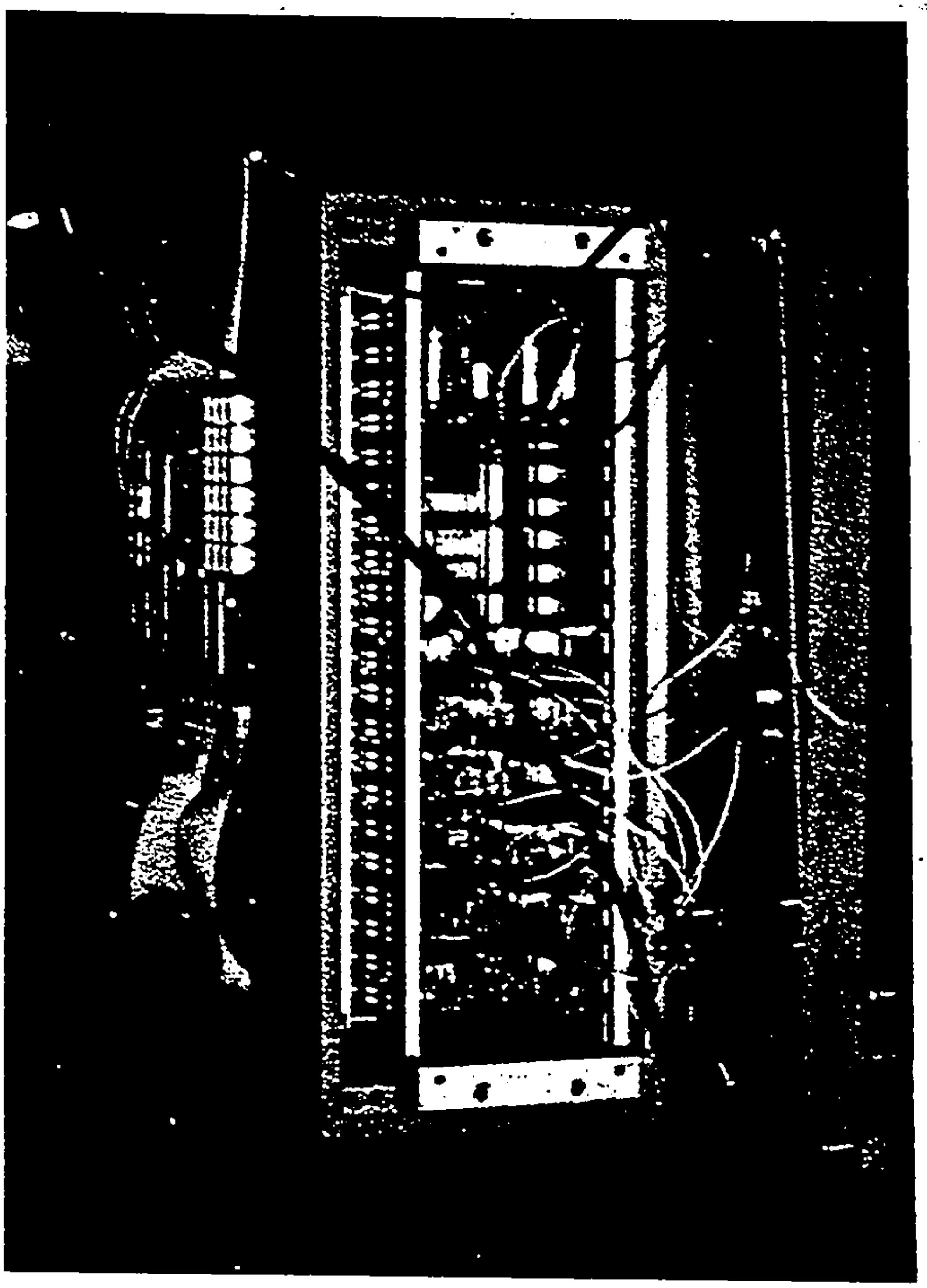


Figure 43: Front View of the Sonar Signal Simulator Package

Appendix C

LIST OF PARTS

Chips are shown in the layout diagrams. The parts used are listed as follows:

BOARD A

Number	Type	Description
IC1	7474	Dual D-Type Positive Edge-Triggered Flip-Flop
IC2	74193	Synchronous 4-Bit Binary Up/Down Counter
IC3	74193	Above
IC4	7400	Quad 2-Input NAND Gate
IC5	74193	Above
IC6	74193	Above
IC7	7474	Above
IC8	74193	Above
IC9	74193	Above
IC10	74624	Voltage-Controlled Oscillator
IC11	7400	Above

BOARD C

Number	Type	Description
IC1	74193	Above
IC2	74193	Above
IC3	74193	Above
IC4	74193	Above
IC5	74193	Above
IC6	74193	Above
IC7	74193	Above
IC8	7474	Above
IC9	7474	Above
IC10	74193	Above
IC11	7400	Above

BOARDS E, G, J, L

Number	Type	Description
IC1	MC14504B	TTL or CMOS to CMOS Hex Level Shifter
IC2	A741	Frequency-Compensated Operational Amplifier
IC3	TL072	Low-Noise JFET-Input Operational Amplifier
IC4	7408	Quad 2-Input AND Gate
IC5	MC1508	8-Bit Multiplying D/A Converter
IC6	74138	3-to-8 Line Decoder/Demultiplexer
IC7	MC14051B	8-Channel Analog Multiplexer
IC8	MC14051B	Above
IC9	MC14051B	Above

MEMORY

Quantity	Type	Description
1	MK4118-3	1K x 8 Static RAM
7	MK2716-6	2K x 8 UV Erasable PROM

Appendix D

OPERATION INSTRUCTIONS

In order to operate the simulator, the system requires that various input parameters be entered by the operator via the SDK-85 keyboard. The input procedures are:

RESET	resets the system, terminates exexcution.
GO 8000	load program counter with 8000H; this is the starting address of the simulation programme.
EXEC	executes the programme.

This EXEC command starts the programme; at this point a '0' will appear in the data field.

NEXT	a '1' will appear in the data field; this indicates that the first of a list of seven system parameters is now available for inspection and possible change.

These seven parameters, as identified by the display of an integer between 1 and 7 in the data field, are:

VELREC	(knots)	1
VELTAR	(knots)	2
RANGE	(yards x 100)	3
ANGLE	(degrees)	4
ALPHA	(degrees)	5
ENLOP		6

VSOUND (feet/second) 7

A more detailed description of these parameters may be found in Section 3.3.3.

As stated above, a '1' will appear in the data field after the EXEC key is pressed. This indicates that VELREC is available for inspection and change. The address field will contain a previous value for VELREC or, if the system has just been turned on, a random value. If it is desired to retain the value displayed for VELREC, proceed to parameter 2 (VELTAR) by, pressing EXEC, a '2' should appear in data field. If a new value for VELREC is required, press NEXT. This will clear the address field and a new value may be typed in. Then proceed to parameter 2 by pressing EXEC - a '2' should then appear in data field. The exact same procedure is used for the six remaining pararmeters. Bear in mind the admissable ranges of the various parameters as described in Section 3.3.3. If a particular value is out of range, the display will blank out when it is attempted to enter the value by pressing EXEC. In addition, enter the RANGE (parameter 3) in units of 100 of yards. Thus 800 yards is entered as 8 and 32000 yards is entered as 320. After entering VSOUND and pressing EXEC, the data entry has been completed. At this point all LED's will be activated until the subroutines REPTIM and UNITS have finished - then GO is displayed in the data field to indicate that the simulation has begun.

If a simulation is rerun with the same data as entered previously, it is not necessary to enter all the same parameters again. Instead, enter the following

```
RESET
GO 8000
EXEC          '0' displayed in data field
GO
```

However, in order to do this successfully, ensure that the frequency of the synchronization pulses has not been changed by the operator.

Appendix E

COMPUTER PROGRAMMES

Polar to Rectangular Coordinates Conversion

```
      SUBROUTINE PECPOL
      COMMON /INDATA/ TIME,TSYNC,VELREC,VELTAR,RANGE,ANGLE,ALPHA,VSOUND,
     *                XVELTA,YVELTA,XINTAR,YINTAR

      XVELTA=VELTAR*COS(ALPHA)
      YVELTA=VELTAR*SIN(ALPHA)
      XINTAR=RANGE*COS(ANGLE)
      YINTAR=RANGE*SIN(ANGLE)
      RETURN
      END
```

Provide Information of Time Delay of Return Signal

```
          SUBROUTINE COMP1
          INTEGER*1  FLG.
          COMMON  /INDATA/ TIME,TSYNC,VELREC,VELTAR,RANGE,ANGLE,ALPHA,VSOUND,
         *                 XVELTA,YVELTA,XINTAR,YINTAR
          COMMON /TEMP/ TA,TB,TC,TD,TE,XTAR2,YTAR2,FLG,XREC1
    C
    C     TO COMPUTE THE POSITION OF RECEIVER AT THE TIME OF TRANSMISSION
    C
          XREC1=VELREC*TIME
    C
    C     TO COMPUTE THE REFLECTION TIME
    C
          TA=XVELTA**2+YVELTA**2-VSOUND**2
          TB=2.*((XINTAR-XREC1)*XVELTA+YINTAR*YVELTA+TIME*VSOUND**2)
          TC=(XINTAR-XREC1)**2+YINTAR**2-TIME*TIME*VSOUND**2
          TD=TB*TB-4.*TA*TC
          TE=-(TB+SQRT(TD))/(2.*TA)
    C
    C     TO COMPUTE THE POSITION OF THE TARGET
    C
          XTAR2=XINTAR+XVELTA*TE
          YTAR2=YINTAR+YVELTA*TE
    C
    C     TO COMPUTE THE RECEIVE TIME
    C
          TA=(VELREC+VSOUND)*(VELREC-VSOUND)
          TB=2.*(TE*VSOUND*VSOUND-VELREC*XTAR2)
          TC=XTAR2**2+YTAR2**2-TE*TE*VSOUND**2
          TD=TB*TB-4.*TA*TC
          TE=-(TB+SQRT(TD))/(2.*TA)
          RETURN
          END
```

The Rest of Computations in Signal Generation

```
            SUBROUTINE COMP2
            REAL L1,L2,L3,L4,L5,L6,L7,L8
            INTEGER SEC1,SEC2
            INTEGER*1 EVSEC,ODSEC,NELDEL,NOLDEL,FLG
            COMMON  C1,COEF,PI,PI05,PI15,PI20,FREQ1
            COMMON  /INDATA/ TIME,TSYNC,VELREC,VELTAR,RANGE,ANGLE,ALPHA
     *                   ,VSOUND,XVELTA,YVELTA,XINTAR,YINTAR
            COMMON /RESULT/ AMPEV,AMPOD,EVSEC,ODSEC,NELDEL,NOLDEL,XFREQ
            COMMON /TEMP/ TA,TB,TC,TD,TE,XTAR2,YTAR2,FLG,XREC1
      C
      C     TO COMPUTE THE POSITION OF RECEIVER AT THE RECEIVE TIME
      C
            XREC3=VELREC*TE
      C
      C     TO COMPUTE THE RANGE
      C
            RANGE=(XREC3-XTAR2)**2+YTAR2**2
            RANGE=SQRT(RANGE)
      C
            IF (RANGE .LT. 300. ) GO TO 450
            IF (RANGE .GT. 12000.) GO TO 451
      C
      C     TO COMPUTE THETA1, THETA2, THETA3 AND THETA4
      C
            TA=YTAR2*YTAR2
            L1=(XTAR2-XREC1)**2+TA
            L2=XREC3-XREC1
            L3=(XREC3-XTAR2)**2+TA
            L4=(XTAR2-XINTAR)**2+(YTAR2-YINTAR)**2
            L5=(XREC1-XINTAR)**2+YINTAR**2
            TA=SQRT(L1)
            TB=ABS(L2)
            TC=SQRT(L3)
            TD=SQRT(L4)
            TE=SQRT(L5)
            L6=(L1+TB*TB-L3)/(2.*TA*TB)
            L7=(L4+L1-L5)/(2.*TD*TA)
            L8=(TB*TB+L3-L1)/(2.*TB*TC)
            THETA1=ACOS(L6)
            THETA2=ACOS(L7)
            THETA4=ACOS(L8)
            THETA3=THETA1+THETA4-THETA2
      C
      C     TO COMPUTE THETA
      C
            IF (YTAR2) 50,50,40
         40 THETA=PI+THETA4
            GO TO 60
         50 THETA=PI-THETA4
         60 THETA=THETA/COEF
            ITHETA=IFIX(THETA)
```

```
              N=ITHETA/10
      C
      C       TO COMPUTE THE SECTORS
      C
              IF (MOD(ITHETA,10)) 70,80,70
      80      IF (ITHETA) 90,100,90
      100     SEC1=1
              SEC2=36
              GO TO 140
      90      SEC1=N
              SEC2=N+1
              GO TO 140
      C
      70      IF (MOD(ITHETA,10)-5) 110,120,120
      120     SEC1=N-1
              SEC2=N+2
              GO TO 140
      110     SEC1=N+1
              SEC2=N
      C
              IF (N .EQ. 0) SEC2=36
              GO TO 140
      120     SEC1=N-1
              SEC2=N+2
      C
              IF (N .EQ. 35) SEC2=1
      140     IF (MOD(SEC1,2)) 144,145,144
      145     EVSEC=SEC1
              ODSEC=SEC2
              GO TO 148
      144     ODSEC=SEC1
              EVSEC=SEC2
      148     EVSEC=(EVSEC-2)/2
              ODSEC=(ODSEC-1)/2
      C
      C       TO COMPUTE THE ANGLE DEVIATED FROM THE CENTER OF THE BEAM
      C
              GAMMA1=(SEC1*10-5)-ITHETA
              GAMMA2=(SEC2*10-5)-ITHETA
              GAMMA1=ABS(GAMMA1)
              GAMMA2=ABS(GAMMA2)
      C-
      C       TO COMPUTE THE TIME DELAYS
      C
              IF (GAMMA2 .GT. 180.) GAMMA2=360.-GAMMA2
              GAMMA1=GAMMA1*COEF
              GAMMA2=GAMMA2*COEF
              TA=6.E5/VSOUND
              DEL1=TA*SIN(GAMMA1)
              DEL2=TA*SIN(GAMMA2)
      C
              IF ((SEC1 .EQ. 36) .AND. (SEC2 .EQ. 1)) GO TO 270
              IF ((SEC1 .EQ. 1) .AND. (SEC2 .EQ. 36)) GO TO 280
              IF (SEC1-SEC2) 290,290,310
      290     IF (MOD(SEC1,2)) 330,330,340
      330     ELDEL=DEL1
              ORDEL=DEL2
              GO TO 350
        340   OLDEL=DEL1
              ERDEL=DEL2
              GO TO 351
      C
        310   IF (MOD(SEC1,2)) 370,370,380
        370   ERDEL=DEL1
              OLDEL=DEL2
              GO TO 351
        380   ORDEL=DEL1
              ELDEL=DEL2
              GO TO 350
        270   ELDEL=DEL1
              ORDEL=DEL2
              GO TO 350
        280   ORDEL=DEL1
              ELDEL=DEL2
        350   ELDEL=30.+ELDEL
              OLDEL=30.-ORDEL
              GO TO 352
        351   OLDEL=30.+OLDEL
              ELDEL=30.-ERDEL
      C
        352   IF (ELDEL .LT. 0.) ELDEL=0
              IF (ELDEL .GT. 75.) ELDEL=75.
              ELDEL=(ELDEL/5.)+0.5
              NELDEL=IFIX(ELDEL)
      C
              IF (OLDEL .LT. 0.) OLDEL=0
              IF (OLDEL .GT. 75.) OLDEL=75.
              OLDEL=(OLDEL/5.)+0.5
              NOLDEL=IFIX(OLDEL)
      C
      C       TO COMPUTE THE FREQUENCY
      C
              FREQ=FREQ1*VSOUND/(VSOUND-VELPEC*L6)
              FREQ=FREQ*(VSOUND-VELTAR*L7)/VSOUND
              FREQ=FREQ*VSOUND/(VSOUND-VELTAR*COS(THETA2))
              FREQ=FREQ*(VSOUND-VELPEC*L9)/VSOUND
              FREQ=10000./(2.*FREQ)+0.5
              NFREQ=IFIX(FREQ)
      C
              GAMMA1=9.*GAMMA1
              GAMMA2=9.*GAMMA2
              GAIN1=COS(GAMMA1)
              GAIN2=COS(GAMMA2)
      C
      C       TO COMPUTE THE AMPLITUDES
      C
              IF ((ALPHA .GE. 0.) .AND. (ALPHA .LE. PI05)) GO TO 190
              IF ((ALPHA .GE. PI) .AND. (ALPHA .LE. PI15)) GO TO 180
              GO TO 200
        180   ALPHA=ALPHA-PI
        190   IF (YTAR2) 210,220,220
        210   Z1=THETA4-ALPHA
              IF ((Z1 .GE. -PI05) .AND. (Z1 .LE. 0.)) BETA=PI05+Z1
              IF ((Z1 .GE. 0.) .AND. (Z1 .LE. PI)) BETA=ABS(Z1-PI05)
              GO TO 230
      220     Z2=THETA4+ALPHA
      C
              IF ((Z2 .GE. 0.) .AND. (Z2 .LE. PI)) BETA=ABS(Z2-PI05)
              IF ((Z2 .GE. PI) .AND. (Z2 .LE. PI15)) BETA=PI15-Z2
              GO TO 230
      200     IF ((ALPHA .GE. PI05) .AND. (ALPHA .LE. PI)) ALPHA=PI-ALPHA
              IF ((ALPHA .GE. PI15) .AND. (ALPHA .LE. PI20)) ALPHA=PI20-ALPHA
              IF (YTAR2) 220,210,210
      230     SIGMA=COS(BETA)
      C
              IF (BETA .EQ. PI05) SIGMA=0.1
              TB=SQRT(SIGMA)
              AMP1=GAIN1*TB
              AMP2=GAIN2*TB
      C
              IF (MOD(SEC1,2)) 601,602,601
      602     AMPEV=AMP1
              AMPOD=AMP2
              GO TO 605
      601     AMPOD=AMP1
              AMPEV=AMP2
      C
      C       TO INCREMENT THE TRANSMISSION TIME BY A SINGLE PERIOD OF
      C       SYNCHRONIZATION PULSES
      C
      605     TIME=TIME+TSYNC
      C
              FLG=0
              RETURN
      C
      450     FLG=1
              RETURN
      C
      451     FLG=2
              RETURN
              END
```

```
MODULE INFORMATION:

     CODE AREA SIZE     = 0AE0H   2784D
     VARIABLE AREA SIZE = 0080H    128D
     MAXIMUM STACK SIZE = 000EH     14D
     202 LINES READ

     0 PROGRAM ERROR(S) IN PROGRAM UNIT COMP2

     0 TOTAL PROGRAM ERROR(S)
     END OF FORTRAN COMPILATION
```

Input/Output and Timing Control

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                    1         NAME    SONAR
                    2         EXTRN   FADD,FDIV,FIXSD,FLOAD,FLTDS,FMUL,FSET,FSTOR,FSUB
                    3         EXTRN   COMP1,COMP2,FPAGO,RECPOL
                    4         STKLN   10H
                    5 ;------------------------------------------------------------------
                    6 ;               PROGRAM CONSTANTS
                    7 ;------------------------------------------------------------------
 0090               8 ADISP   EQU     90H     ;CONTROL CHARACTER TO INDICATE OUTPUT TO
                    9                         ;ADDRESS FIELD OF DISPLAY
 00DC              10 ALLOFF  EQU     0DCH    ;CONTROL CHARACTER TO CLEAR DISPLAY DURING
                   11                         ;BLANKING PERIOD
 00D0              12 ALLON   EQU     0D0H    ;CONTROL CHARACTER TO TURN ON DISPLAY DURING
                   13                         ;BLANKING PERIOD
 0022              14 AMPP    EQU     22H     ;OUTPUT PORT FOR AMPLITUDE
 0020              15 BCSR    EQU     20H     ;OUTPUT PORT FOR CSR IN BASIC RAM
 1E00              16 BTIMC   EQU     1E00H   ;BASIC RAM TIMER COUNT FOR 400 HZ OUTPUT
 0025              17 BTIMH   EQU     25H     ; -"-  -"- HIGH ORDER BYTE OF TIMER COUNT
 0024              18 BTIML   EQU     24H     ; -"-  -"- LOW  -"-  -"-  -"-  -"-
 0040              19 BTIMM   EQU     40H     ; -"-  -"- TIMER MODE - CONTINUOUS SQUARE WAVE
 1900              20 CNTRL   EQU     1900H   ;ADDRESS FOR SENDING CONTROL CHARACTERS TO
                   21                         ;DISPLAY CHIP (8279)
 0094              22 DDISP   EQU     94H     ;CONTROL CHARACTER TO INDICATE OUTPUT TO
                   23                         ;DATA FIELD OF DISPLAY
 0021              24 DELP    EQU     21H     ;OUTPUT PORT FOR DELAY
 1800              25 DSPLY   EQU     1800H   ;ADDRESS FOR SENDING CHARACTERS TO DISPLAY
 0028              26 ECSR    EQU     28H     ;OUTPUT PORT FOR CSR IN EXPANSION RAM
 0080              27 EMPTY   EQU     80H     ;MSB=1 INDICATES EMPTY INPUT BUFFER
 002D              28 ETIMH   EQU     2DH     ;EXPANSION RAM HIGH ORDER BYTE OF TIMER COUNT
 002C              29 ETIML   EQU     2CH     ;   -"-    -"- LOW  -"-   -"-   -"-   -"-
 0023              30 EVSP    EQU     23H     ;OUTPUT PORT FOR EVEN SECTOR ADDRESS
 0010              31 EXEC    EQU     10H     ;CHARACTER GENERATED BY 'EXEC' KEY
 0029              32 FRQP    EQU     29H     ;OUTPUT PORT FOR FREQUENCY
 0012              33 GO      EQU     12H     ;CHARACTER GENERATED BY 'GO' KEY
 00C0              34 HOUT    EQU     0C0H    ;CHARACTER TO SET SOD=1
 0040              35 LOUT    EQU     40H     ;   -"-      -"-   SOD=0
 0011              36 NEXT    EQU     11H     ;CHARACTER GENERATED BY 'NEXT' KEY
 002B              37 ODSP    EQU     2BH     ;OUTPUT PORT FOR ODD SECTOR ADDRESS
 0040              38 READ    EQU     40H     ;CONTROL CHARACTER TO INDICATE INPUT FROM KEYBOARD
 004F              39 TSTOP   EQU     4FH     ;CONTROL CHARACTER TO STOP TIMER AND INITIALIZE
                   40                         ;PORTS AS THE OUTPUT ONES
 00CF              41 TSTRT   EQU     0CFH    ;CONTROL CHARACTER TO START TIMER
                   42
                   43 ;-------------------------------------------------------------------
                   44 ;               RESET ENTRY POINT
                   45 ;-------------------------------------------------------------------
                   46         ASEG
 8000              47         ORG     8000H
 8000 AF           48 SONAR:  XRA     A       ;INITIALIZE 8279 FOR 8 CHARACTER DISPLAY, LEFT
                   49                         ;ENTRY, 2 KEY LOCKOUT
 8001 320019       50         STA     CNTRL
 8004 3EDC         51         MVI     A,ALLOFF;CLEAR DISPLAY
 8006 320019       52         STA     CNTRL
 8009 3E20         53         MVI     A,20H   ;WAIT OUT BLANKING TIME
 800B 3D           54         DCR     A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0        SONAR

  LOC  OBJ          LINE        SOURCE STATEMENT

  800C C20B80        55             JNZ     $-1
  800F 3E4F          56             MVI     A,TSTOP ;INITIALIZE ALL PORTS AS OUTPUTS
  8011 D320          57             OUT     BCSR
  8013 D328          58             OUT     ECSR
  8015 3E5E          59             MVI     A,(BTIMC SHR 8) OR BTIMM ;SET BASIC TIMER
  8017 D325          60             OUT     BTIMH
  8019 3E00          61             MVI     A,BTIMC AND 0FFH
  801B D324          62             OUT     BTIML
  801D 3ECF          63             MVI     A,TSTRT ;START BASIC TIMER
  801F D320          64             OUT     BCSR
  8021 310000   S    65             LXI     SP,STACK;INITIALIZE STACK
  8024 21AF85        66             LXI     H,ZERO  ;DISPLAY ZERO IN DATA FIELD
  8027 CD6F83        67             CALL    DDF
  802A 3E0E          68             MVI     A,0EH   ;UNMASK RST5.5 INTERRUPT
  802C 30            69             SIM
  802D 3E80          70             MVI     A,EMPTY ;SET BUFFER EMPTY FLAG
  802F 32FE27        71             STA     IBUFF
                     72 ;-----------------------------------------------------------------
                     73 ;                  GET PARAMETERS FOR SIMULATION
                     74 ;-----------------------------------------------------------------
  8032 CD7684        75 INSTR:      CALL    RDKBD   ;GET CHARACTER FROM KEYBOARD
  8035 FE12          76             CPI     GO      ;WAS THIS 'GO' KEY ?
  8037 CA4D80        77             JZ      SIMUL   ;IF SO - GO SIMULATE
  803A FE11          78             CPI     NEXT    ;WAS THIS 'NEXT' KEY ?
  803C C23280        79             JNZ     INSTR   ;NO - WAIT FOR VALID INSTRUCTION
  803F CD0182        80             CALL    DATAIN  ;GET STARTING DATA FROM KEYBOARD
  8042 3ED0          81             MVI     A,ALLON ;TURN ON ENTIRE DISPLAY
  8044 320019        82             STA     CNTRL
  8047 CD8684        83             CALL    REPTIM  ;COMPUTE REPETITION TIME FOR TSYNC PULSES
  804A CD1C85        84             CALL    UNITS   ;STANDARDIZE UNITS
  804D 210000        85 SIMUL:      LXI     H,0     ;SET TIME=0
  8050 224727        86             SHLD    TIME
  8053 224927        87             SHLD    TIME+2
  8056 3EDC          88             MVI     A,ALLOFF;CLEAR ENTIRE DISPLAY
  8058 320019        89             STA     CNTRL
  805B 3E20          90             MVI     A,20H   ;WAIT OUT BLANKING TIME
  805D 3D            91             DCR     A
  805E C25D80        92             JNZ     $-1
  8061 21AF85        93             LXI     H,SIXTY ;DISPLAY '60' IN DATA FIELD
  8064 CD6F83        94             CALL    DDF
  8067 3E19          95             MVI     A,19H   ;MASK-OUT KEYBOARD INTERRUPTS AND SONAR
  8069 30            96             SIM             ;TSYNC AND TIMER INTERRUPTS
  806A AF            97             XRA     A       ;CLEAR TSYNC FLAG
  806B 321227        98             STA     FLAG
  806E CD0000   E    99 COMPUT:     CALL    COMP1   ;START COMPUTATIONS
  8071 CDCF84       100             CALL    SETTIM  ;SET TIMER
  8074 FB           101             EI
  8075 CD0000   E   102             CALL    COMP2   ;COMPLETE COMPUTATIONS
  8078 3AA127       103             LDA     FLG     ;TEST FLG
  807B A7           104             ANA     A
  807C CA8680       105             JZ      SKIP    ;OK
  807F 1F           106             RAR             ;FLG=1 ?
  8080 DABA83       107             JC      ERR2    ;TARGET TOO CLOSE
  8083 C3C083       108             JMP     ERR3    ;TARGET TOO FAR
  8086 CD9480       109 SKIP:       CALL    AMP
```

```
 LOC  OBJ         LINE        SOURCE STATEMENT

 8089 20           110          RIM
 808A E508         111          ANI     8        ;INTERRUPTS ENABLED ?
 808C C28980       112          JNZ     $-3      ;YES: WAIT FOR TSYNC INTERRUPT
 808F FB           113          EI               ;NO, WAIT FOR TIMER INTERRUPT
 8090 76           114          HLT
 8091 C36E80       115          JMP     COMPUT
                   116 ;------------------------------------------------------------
                   117 ;                           SUBROUTINES
                   118 ;------------------------------------------------------------
                   119 ;NAME:      AMP - AMPLITUDE OF ECHOIC SIGNAL
                   120 ;INPUTS:    NONE
                   121 ;OUTPUTS:   NONE
                   122 ;CALLS:     FADD,FIXSD,FLOAD,FMUL
                   123 ;DESTROYS:  A,B,C,D,E,H,L
                   124 ;DESCRIPTION: AMP COMPUTES AND STORES CONSECUTIVE SAMPLES OF AMPLITUDE OF
                   125 ;             THE SIMULATED ECHOIC SIGNAL
                   126
 8094 010027       127 AMP:     LXI     B,FPR    ;REGS B,C POINT TO FPR
 8097 117727       128          LXI     D,AMPEV  ;REGS D,E POINT TO (AMPEV)
 809A 3A2127       129          LDA     ENLOP    ;GET ENLOP DATA
 809D FE02         130          CPI     2        ;JUMP TO APPROPRIATE ENVELOPE
 809F CAE480       131          JZ      ENLOP2
 80A2 D21F81       132          JNC     ENLOP3
 80A5 AF           133 ENLOP1:  XRA     A        ;GENERATE FIRST SAMPLE(EVAMP)
 80A6 322527       134          STA     EVAMP
 80A9 CDCB81       135          CALL    EXP1     ;AMPEV*15
 80AC CDD481       136          CALL    EXP2     ;AMPEV*15 +0.5
 80AF 112627       137          LXI     D,EVAMP+1 ;REGS D,E POINT TO (EVAMP+1)
 80B2 CD0000   E   138          CALL    FIXSD    ;FP TO INTEGER CONVERSION AND STORE RES[illegible]
 80B5 3A2627       139          LDA     EVAMP+1  ;GENERATE DIFFERENT SAMPLES
 80B8 2E07         140          MVI     L,7
 80BA CDDA81       141          CALL    STORE
 80BD AF           142          XRA     A        ;LAST SAMPLE AND ZERO THE REST LOCATIONS
 80BE 2E0A         143          MVI     L,0AH
 80C0 CDDA81       144          CALL    STORE
 80C3 323627       145          STA     ODAMP    ;GENERATE FIRST SAMPLE(ODAMP)
 80C6 117B27       146          LXI     D,AMPOD  ;REGS D,E POINT TO (AMPOD)
 80C9 CDCB81       147          CALL    EXP1     ;AMPOD*15
 80CC CDD481       148          CALL    EXP2     ;AMPOD*15 +0.5
 80CF 113727       149          LXI     D,ODAMP+1 ;REGS D,E POINT TO (ODAMP+1)
 80D2 CD0000   E   150          CALL    FIXSD
 80D5 3A3727       151          LDA     ODAMP+1  ;GENERATE DIFFERENT SAMPLES
 80D8 2E07         152          MVI     L,7
 80DA CDDA81       153          CALL    STORE
 80DD AF           154          XRA     A        ;LAST SAMPLE AND ZERO THE REST LOCATIONS
 80DE 2E0A         155          MVI     L,0AH
 80E0 CDDA81       156          CALL    STORE
 80E3 C9           157          RET
 80E4 AF           158 ENLOP2:  XRA     A        ;FIRST SAMPLE
 80E5 322527       159          STA     EVAMP
 80E8 CDCB81       160          CALL    EXP1
 80EB CDD481       161          CALL    EXP2
 80EE 112627       162          LXI     D,EVAMP+1
 80F1 CD0000   E   163          CALL    FIXSD
 80F4 3A2627       164          LDA     EVAMP+1  ;GENERATE DIFFERENT SAMPLES
```

```
  LOC  OBJ         LINE        SOURCE STATEMENT

  80F7 2E0F         155          MVI    L,15
  80F9 CDDA81       156          CALL   STORE
  80FC AF           157          XRA    A        ;LAST SAMPLE
  80FD 322527       168          STA    EVAMP+15
  8100 323527       169          STA    ODAMP   ;FIRST SAMPLE
  8103 117B27       170          LXI    D,AMPOD
  8106 CDCB81       171          CALL   EXP1
  8109 CDD481       172          CALL   EXP2
  810C 113727       173          LXI    D,ODAMP+1
  810F CD0000   E   174          CALL   FIXSD
  8112 3A3727       175          LDA    ODAMP+1 ;GENERATE DIFFERENT SAMPLES
  8115 2E0F         176          MVI    L,15
  8117 CDDA81       177          CALL   STORE
  811A AF           178          XRA    A        ;LAST SAMPLE
  811B 324627       179          STA    ODAMP+15
  811E C9           180          RET
  811F CDCB81       181 ENLOP3:  CALL   EXP1
  8122 11C185       182          LXI    D,F0P6  ;REGS D,E POINT TO A [illegible]
  8125 CD0000   E   183          CALL   FMUL    ;AMPEV*15 *0.6667
  8128 CDD481       184          CALL   EXP2    ;AMPEV*15 *0.6667+0.5
  812B 112527       185          LXI    D,EVAMP ;GENERATE FIRST FOUR SAMPLES
  812E CD0000   E   186          CALL   FIXSD
  8131 3A2527       187          LDA    EVAMP
  8134 2E04         188          MVI    L,4
  8136 CDDA81       189          CALL   STORE
  8139 117727       190          LXI    D,AMPEV ;GENERATE SAMPLE 5 TO SAMPLE 9
  813C CDCB81       191          CALL   EXP1
  813F CDD481       192          CALL   EXP2
  8142 112927       193          LXI    D,EVAMP+4
  8145 CD0000   E   194          CALL   FIXSD
  8148 3A2927       195          LDA    EVAMP+4
  814B 2E05         196          MVI    L,5
  814D CDDA81       197          CALL   STORE
  8150 117727       198          LXI    D,AMPEV ;GENERATE SAMPLE 10 TO SAMPLE 13
  8153 CDCB81       199          CALL   EXP1
  8156 113985       200          LXI    D,F0P3
  8159 CD0000   E   201          CALL   FMUL    ;AMPEV*15 *0.3333
  815C CDD481       202          CALL   EXP2    ;AMPEV*15 *0.3333+0.5
  815F 112E27       203          LXI    D,EVAMP+9
  8162 CD0000   E   204          CALL   FIXSD
  8165 3A2E27       205          LDA    EVAMP+9
  8168 2E04         206          MVI    L,4
  816A CDDA81       207          CALL   STORE
  816D AF           208          XRA    A        ;GENERATE ZEROS
  816E 2E05         209          MVI    L,5
  8170 CDDA81       210          CALL   STORE
  8173 117B27       211          LXI    D,AMPOD ;GENERATE FIRST FOUR SAMPLES
  8176 CDCB81       212          CALL   EXP1
  8179 11C185       213          LXI    D,F0P6
  817C CD0000   E   214          CALL   FMUL
  817F CDD481       215          CALL   EXP2
  8182 113627       216          LXI    D,ODAMP
  8185 CD0000   E   217          CALL   FIXSD
  8188 3A3627       218          LDA    ODAMP
  818B 2E04         219          MVI    L,4
```

```
LOC  OBJ           LINE        SOURCE STATEMENT

818D CDDA81                CALL    STORE
8190 117B27                LXI     D,AMPOD ;GENERATE SAMPLE 5 TO SAMPLE 8
8193 CDCB81                CALL    EXP1
8196 CDD481                CALL    EXP2
8199 113A27                LXI     D,ODAMP+4
819C CD0000   E            CALL    FIXSD
819F 3A3A27                LDA     ODAMP+4
81A2 2E05                  MVI     L,5
81A4 CDDA81                CALL    STORE
81A7 117B27                LXI     D,AMPOD ;GENERATE SAMPLE 10 TO SAMPLE [illegible]
81AA CDCB81                CALL    EXP1
81AD 11B985                LXI     D,FAP3
81B0 CD0000   E            CALL    FMUL
81B3 CDD481                CALL    EXP2
81B6 113F27                LXI     D,ODAMP+9
81B9 CD0000   E            CALL    FIXSD
81BC 3A3F27                LDA     ODAMP+9
81BF 2E04                  MVI     L,4
81C1 CDDA81                CALL    STORE
81C4 AF                    XRA     A         ;GENERATE ZEROS
81C5 2E05                  MVI     L,5
81C7 CDDA81                CALL    STORE
81CA C9                    RET
81CB CD0000   E      EXP1: CALL    FLOAD     ;(AMPEV)OR(AMPOD) LOADED INTO FAC
81CE 11C585                LXI     D,F15     ;AMPEV (OR AMPOD)*15
81D1 C30000   E            JMP     FMUL
81D4 11BD85          EXP2: LXI     D,FAP5    ;REGS D,E POINT TO 2.5[illegible]
81D7 C30000   E            JMP     FADD      ;ADDITION
81DA 2D              STORE: DCR    L         ;STORE SAMPLES IN CONSECUTIVE MEMORY LOCATIONS
81DB C8                    RZ
81DC 13                    INX     D
81DD 12                    STAX    D
81DE C3DA81                JMP     STORE
                     ;------------------------------------------------------------
                     ;NAME        ATEST - ANGLE TEST
                     ;INPUTS:     D,E - BCD NUMBER REPRESENTING AN ANGLE IN DEGREES
                     ;OUTPUTS:    CY=1 WHEN ANGLE EXCEEDS 360 DEGREES
                     ;CALLS:      NOTHING
                     ;DESTROYS:   PSW
                     ;DESCRIPTION: ATEST CHECKS IF BCD NUMBER BELONGS TO 0-360 RANGE

81E1 3E03            ATEST: MVI    A,3
81E3 BA                    CMP     D
81E4 CAE881                JZ      AT1
81E7 C9                    RET
81E8 3E60            AT1:  MVI     A,60H
81EA BB                    CMP     E
81EB C9                    RET
                     ;------------------------------------------------------------
                     ;NAME:       BINBCD - CONVERT BCD INTO BINARY
                     ;INPUTS:     A - BCD NUMBER TO BE CONVERTED
                     ;OUTPUTS:    A - RESULTING BINARY NUMBER
                     ;CALLS:      NOTHING
                     ;DESTROYS:   H,L
                     ;DESCRIPTION: BINBCD CONVERTS TWO DIGIT BCD NUMBER IN ACCUMULATOR INTO
```

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                   275 ;           A NATURAL BINARY FORMAT
                   276
81EC 6F            277 BINBCD: MOV    L,A     ;SAVE BCD DIGIT IN L
81ED E60F          278         ANI    0FH     ;SELECT LS DIGIT AND
81EF 67            279         MOV    H,A     ;SAVE IT IN H
81F0 7D            280         MOV    A,L     ;GET BCD DIGIT BACK
81F1 E6F0          281         ANI    0F0H    ;SELECT MS DIGIT AND
81F3 0F            282         RRC            ;MULTIPLY IT BY TEN
81F4 0F            283         RRC
81F5 6F            284         MOV    L,A
81F6 0F            285         RRC
81F7 0F            286         RRC
81F8 85            287         ADD    L
81F9 07            288         RLC
81FA 84            289         ADD    H       ;ADD LS DIGIT
81FB C9            290         RET
                   291 ;------------------------------------------------------------------
                   292 ;NAME:       DAF - DISPLAY IN DATA FIELD
                   293 ;INPUTS:     H,L - ADDRESS OF CHARACTERS TO BE DISPLAYED
                   294 ;OUTPUTS:    NONE
                   295 ;CALLS:      OUTPUT
                   296 ;DESTROYS:   A,B,C,D,E,H,L
                   297 ;DESCRIPTION: DAF PREPARES PARAMETERS FOR OUTPUT SUBROUTINE TO DISPLAY
                   298 ;            CHARACTERS POINTED BY H,L IN ADDRESS FIELD
                   299
81FC AF            300 DAF:    XRA    A       ;USE ADDRESS FIELD
81FD 47            301         MOV    B,A     ;NO DECIMAL INDICATOR
81FE C34094        302         JMP    OUTPUT  ;OUTPUT FOR DISPLAY
                   303 ;------------------------------------------------------------------
                   304 ;NAME:       DATAIN - INPUT DATA
                   305 ;INPUTS      NONE
                   306 ;OUTPUTS:    NONE
                   307 ;CALLS:      ATEST,DDF,ETEST,FLTBCD,FMUL,FSET,FSTOR,UPDATE
                   308 ;DESTROYS:   A,B,C,D,E,H,L
                   309 ;DESCRIPTION: DATAIN READS STARTING VALUES OF DIFFERENT VARIABLES FROM THE
                   310 ;            KEYBOARD. EACH VALUE IS TESTED FOR ADMISSIBLE RANGE, DISPLAYED
                   311 ;            AND STORED IN MEMORY IN FLOATING POINT AND/OR BCD FORMAT
                   312
8201 010027        313 DATAIN: LXI    B,FPR   ;INITILIZE FAC
8204 C5            314         PUSH   B
8205 010000        315         LXI    B,0
8208 CD0000   E    316         CALL   FSET
820B 21A185        317 DATA1:  LXI    H,ONE   ;DISPLAY ONE IN DATA FIELD
820E CD6F83        318         CALL   DDF
8211 2A1727        319         LHLD   VELRB   ;GET VELREC
8214 CD7085        320         CALL   UPDATE  ;UPDATE VELREC
8217 C23382        321         JNZ    REP1    ;INVALID DATA - REPEAT
821A AF            322         XRA    A       ;IF VELREC>20 , REPEAT
821B BA            323         CMP    D
821C C23382        324         JNZ    REP1
821F 3E30          325         MVI    A,30H
8221 BB            326         CMP    E
8222 DA3382        327         JC     REP1
8225 EB            328         XCHG           ;STORE UPDATED VALUE
8226 221727        329         SHLD   VELRB
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

```
 LOC  OBJ          LINE        SOURCE STATEMENT

 8229 EB            330            XCHG
 822A 214F27        331            LXI     H,VELREC;POINTER TO MEMORY
 822D CDF983        332            CALL    FLTBCD  ;CONVERT INTO FPF AND STORE
 8230 C33C82        333            JMP     DATA2
 8233 210000        334 REP1:      LXI     H,0     ;SET VELREC=0
 8236 221727        335            SHLD    VELRS
 8239 C30B82        336            JMP     DATA1   ;TRY AGAIN
 823C 21AD85        337 DATA2:     LXI     H,TWO   ;DISPLAY TWO IN DATA FIELD
 823F CD6F83        338            CALL    DDF
 8242 2A1927        339            LHLD    VELTR   ;GET VELTAR
 8245 CD7085        340            CALL    UPDATE
 8248 C26482        341            JNZ     REP2
 824B AF            342            XRA     A       ;IF VELTAR>30 REPEAT
 824C BA            343            CMP     D
 824D C26482        344            JNZ     REP2
 8250 3E20          345            MVI     A,20H
 8252 BB            346            CMP     E
 8253 DA6482        347            JC      REP2
 8256 EB            348            XCHG
 8257 221927        349            SHLD    VELTR
 825A EB            350            XCHG
 825B 215327        351            LXI     H,VELTAR
 825E CDF983        352            CALL    FLTBCD
 8261 C36D82        353            JMP     DATA3
 8264 210000        354 REP2:      LXI     H,0     ;SET VELTAR=0
 8267 221927        355            SHLD    VELTR
 826A C33C82        356            JMP     DATA2
 826D 21A585        357 DATA3:     LXI     H,THREE ;DISPLAY THREE IN DATA FIELD
 8270 CD6F83        358            CALL    DDF
 8273 2A1B27        359            LHLD    RANGEB  ;GET RANGE
 8276 CD7085        360            CALL    UPDATE  ;UPDATE RANGE IN HUNDREDS OF YARDS
 8279 C2B382        361            JNZ     REP3
 827C AF            362            XRA     A       ;IF RANGE<7 OR RANGE>320 , REPEAT
 827D BA            363            CMP     D
 827E C28A82        364            JNZ     L1
 8281 3E07          365            MVI     A,7
 8283 BB            366            CMP     E
 8284 D2B382        367            JNC     REP3
 8287 C39982        368            JMP     OK
 828A 3E03          369 L1:        MVI     A,3
 828C BA            370            CMP     D
 828D DAB382        371            JC      REP3
 8290 C29982        372            JNZ     OK
 8293 3E20          373            MVI     A,20H
 8295 BB            374            CMP     E
 8296 DAB382        375            JC      REP3
 8299 EB            376 OK:        XCHG
 829A 221B27        377            SHLD    RANGEB
 829D EB            378            XCHG
 829E 215727        379            LXI     H,RANGE
 82A1 CDF983        380            CALL    FLTBCD  ;CONVERT INTO FPF
 82A4 11C985        381            LXI     D,F100  ;MULTIPLY BY HUNDRED
 82A7 CD0000     E  382            CALL    FMUL
 82AA 115727        383            LXI     D,RANGE
 82AD CD0000     E  384            CALL    FSTOR   ;STORE
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

```
 LOC  OBJ         LINE        SOURCE STATEMENT

 82B0 C3BC82                    JMP     DATA4
 82B3 210000             REP3:  LXI     H,0      ;SET RANGE=0
 82B6 221B27                    SHLD    RANGEB
 82B9 C3B082                    JMP     DATA3
 82BC 21A785             DATA4: LXI     H,FOUR   ;DISPLAY FOUR IN DATA FIELD
 82BF CD6F83                    CALL    DDF
 82C2 2A1D27                    LHLD    ANGLEB   ;GET ANGLE
 82C5 CD7085                    CALL    UPDATE
 82C8 C2DF82                    JNZ     REP4
 82CB CDE181                    CALL    ATEST
 82CE DADF82                    JC      REP4     ;INVALID DATA - REPEAT
 82D1 EB                        XCHG
 82D2 221D27                    SHLD    ANGLEB
 82D5 EB                        XCHG
 82D6 215B27                    LXI     H,ANGLE
 82D9 CDF983                    CALL    FLTBCD
 82DC C3E882                    JMP     DATA5
 82DF 210000             REP4:  LXI     H,0      ;SET ANGLE=0
 82E2 221D27                    SHLD    ANGLEB
 82E5 C3BC82                    JMP     DATA4
 82E8 21A985             DATA5: LXI     H,FIVE   ;DISPLAY FIVE IN DATA FIELD
 82EB CD6F83                    CALL    DDF
 82EE 2A1F27                    LHLD    ALPHAB   ;GET ALPHA
 82F1 CD7085                    CALL    UPDATE
 82F4 C20B83                    JNZ     REP5
 82F7 CDE181                    CALL    ATEST
 82FA DA0B83                    JC      REP5
 82FD EB                        XCHG
 82FE 221F27                    SHLD    ALPHAB
 8301 EB                        XCHG
 8302 215F27                    LXI     H,ALPHA
 8305 CDF983                    CALL    FLTBCD
 8308 C31483                    JMP     DATA6
 830B 210000             REP5:  LXI     H,0      ;SET ALPHA=0
 830E 221F27                    SHLD    ALPHAB
 8311 C3E882                    JMP     DATA5
 8314 21AB85             DATA6: LXI     H,SIX    ;DISPLAY SIX IN DATA FIELD
 8317 CD6F83                    CALL    DDF
 831A 2A2127                    LHLD    ENLOP    ;GET ENLOP
 831D CD7085                    CALL    UPDATE
 8320 C23983                    JNZ     REP6
 8323 AF                        XRA     A        ;SET ACCUMULATOR TO ZERO
 8324 BA                        CMP     D        ;HIGH ORDER BYTE CONTAINS ZERO ?
 8325 C23983                    JNZ     REP6     ;NO - ERROR
 8328 BB                        CMP     E        ;LOW ORDER BYTE CONTAINS ZERO ?
 8329 CA3983                    JZ      REP6     ;YES - ERROR
 832C 3E03                      MVI     A,3
 832E BB                        CMP     E        ;LOW ORDER BYTE GREATER THAN THREE ?
 832F DA3983                    JC      REP6     ;YES - ERROR
 8332 EB                        XCHG             ;OK - STORE
 8333 222127                    SHLD    ENLOP
 8336 C34283                    JMP     DATA7
 8339 210000             REP6:  LXI     H,0      ;SET ENLOP=0
 833C 222127                    SHLD    ENLOP
 833F C31483                    JMP     DATA6
```

```
  LOC  OBJ         LINE        SOURCE STATEMENT

  8342 21A085       440 DATA7:  LXI    H,SEVEN ;DISPLAY SEVEN IN DATA FIELD
  8345 CD6F83       441         CALL   DDF
  8348 2A2327       442         LHLD   VSNDB  ;GET VSOUND
  834B CD7085       443         CALL   UPDATE
  834E C26683       444         JNZ    REP7
  8351 3E99         445         MVI    A,99H  ;TEST FOR DECIMAL NUMBER
  8353 BA           446         CMP    D
  8354 DA6683       447         JC     REP7
  8357 BB           448         CMP    E
  8358 DA6683       449         JC     REP7
  835B EB           450         XCHG
  835C 222327       451         SHLD   VSNDB
  835F EB           452         XCHG
  8360 215327       453         LXI    H,VSOUND
  8363 C3F983       454         JMP    FLTBCD  ;CONVERT INTO BCD
  8366 210000       455 REP7:   LXI    H,0     ;SET VSOUND=0
  8369 222327       456         SHLD   VSNDB
  836C C34283       457         JMP    DATA7
                    458 ;------------------------------------------------------------------
                    459 ;NAME:      DDF - DISPLAY IN DATA FIELD
                    460 ;INPUTS:    H,L - ADDRESS OF CHARACTERS TO BE DISPLAYED
                    461 ;OUTPUTS:   NONE
                    462 ;CALLS:     OUTPUT
                    463 ;DESTROYS:  A,B,C,D,E,H,L
                    464 ;DESCRIPTION: DDF PREPARES PARAMETERS FOR OUTPUT SUBROUTINE TO DISPLAY
                    465 ;           CHARACTERS POINTED BY H,L IN DATA FIELD
                    466
  836F AF           467 DDF:    XRA    A
  8370 47           468         MOV    B,A     ;NO DECIMAL INDICATOR
  8371 3C           469         INR    A       ;USE DATA FIELD
  8372 C34D84       470         JMP    OUTPUT  ;OUTPUT FOR DISPLAY
                    471 ;------------------------------------------------------------------
                    472 ;NAME:      ECHO - RETURN SIGNAL
                    473 ;INPUTS:    NONE
                    474 ;OUTPUTS:   NONE
                    475 ;CALLS:     NOTHING
                    476 ;DESTROYS:  A,B,C,D,E,H,L
                    477 ;DESCRIPTION: ECHO OUTPUTS EVSEC,ODSEC,NFREQ,NELDEL,NOLDEL,EVAMP AND
                    478 ;           ODAMP TO THE OUTPUT PORTS
                    479
  8375 3A7F27       480 ECHO:   LDA    EVSEC   ;LOAD EVEN SECTOR ADDRESS TO OUTPUT PORT
  8378 D323         481         OUT    EVSP
  837A 3A8027       482         LDA    ODSEC   ;LOAD ODD SECTOR ADDRESS TO OUTPUT PORT
  837D D32B         483         OUT    ODSP
  837F 3A8327       484         LDA    NFREQ   ;LOAD FREQUENCY TO OUTPUT PORT
  8382 D329         485         OUT    FROP
  8384 218227       486         LXI    H,NOLDEL;LOAD LEFT ODD DELAY AND LEFT EVEN DELAY
  8387 3A8127       487         LDA    NELDEL  ;TO OUTPUT PORT
  838A 07           488         RLC
  838B 07           489         RLC
  838C 07           490         RLC
  838D 07           491         RLC
  838E 86           492         ADD    M
  838F D321         493         OUT    DELP
  8391 1611         494         MVI    D,17    ;LOAD ODD AND EVEN AMPLITUDES TO OUTPUT
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

```
LOC  OBJ         LINE        SOURCE STATEMENT

8393 213627       495          LXI    H,ODAMP  ;PORT FOR EVERY SWEEP
8396 012527       496          LXI    B,EVAMP
8399 0A           497 SAMPLE:  LDAX   B
839A 07           498          RLC
839B 07           499          RLC
839C 07           500          RLC
839D 07           501          RLC
839E 86           502          ADD    M
839F D322         503          OUT    AMPP
83A1 15           504          DCR    D
83A2 C8           505          RZ
83A3 1E05         506          MVI    E,5
83A5 3ED9         507 DLY      MVI    A,0D9H
83A7 3D           508          DCR    A
83A8 C2A783       509          JNZ    $-1
83AB 1D           510          DCR    E
83AC C2A583       511          JNZ    DLY
83AF 23           512          INX    H
83B0 03           513          INX    B
83B1 C39983       514          JMP    SAMPLE
                  515 ;-----------------------------------------------------------------------
                  516 ;NAME:      ERR1 & ERR2 & ERR3 - ERRORS
                  517 ;INPUTS:    NONE
                  518 ;OUTPUTS:   NONE
                  519 ;CALLS:     DAF,DDF
                  520 ;DESTROYS: A,B,C,D,E,H,L
                  521 ;DESCRIPTION: THIS TWO ENTRY SUBROUTINE DISPLAYS ERROR MESSAGE AND HALTS
                  522 ;             COMPUTER TO WAIT FOR RESET SIGNAL
                  523
83B4 21A185       524 ERR1:    LXI    H,ONE    ;DISPLAY ONE IN DATA FIELD
83B7 C3C383       525          JMP    $+12
83BA 21A385       526 ERR2:    LXI    H,TWO    ;DISPLAY TWO IN DATA FIELD
83BD C3C383       527          JMP    $+6
83C0 21A585       528 ERR3:    LXI    H,THREE  ;DISPLAY THREE IN DATA FIELD
83C3 CD6F83       529          CALL   DDF
83C6 21B585       530          LXI    H,ERROR  ;DISPLAY ERR IN ADDRESS FIELD
83C9 CDFC81       531          CALL   DAF
83CC F3           532          DI
83CD 76           533          HLT
                  534 ;---------------------------------------------------------------
                  535 ;NAME:      EXPAND - EXPAND BCD NUMBER FOR DISPLAY
                  536 ;INPUT:     D,E - 4 DIGIT BCD NUMBER
                  537 ;OUTPUT:    H,L - ADDRESS OF OUTPUT BUFFER
                  538 ;CALLS:     NOTHING
                  539 ;DESTROYS: A,H,L
                  540 ;DESCRIPTION: EXPAND EXPANDS 4-DIGIT BCD NUMBER INTO 4 BYTES. EACH
                  541 ;             BCD DIGIT IS PLACED IN THE LOW ORDER NIBBLE OF A BYTE
                  542 ;             WHOSE HIGH ORDER NIBBLE IS SET TO ZERO. WHEN DIGIT IS
                  543 ;             A LEADING ZERO, IT IS REPLACED WITH BLANK. THE RESULTING
                  544 ;             BYTE IS STORED IN THE OUTPUT BUFFER
                  545
83CE C5           546 EXPAND:  PUSH   B
83CF 060A         547          MVI    B,10     ;INITIALIZE B WITH BLANK FOR LEADING ZEROS
83D1 21F927       548          LXI    H,OBUFF  ;POINTER TO OUTPUT BUFFER
83D4 4A           549          MOV    C,D      ;CONVERT D3 & D2 INTO SINGLE CHARACTERS
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

```
LOC  OBJ         LINE       SOURCE STATEMENT

83D5 CDE283                     CALL   CNVRT
83D8 23                         INX    H        ;UPDATE POINTER
83D9 4B                         MOV    C,E      ;CONVERT D1 & D0 INTO SINGLE CHARACTERS
83DA CDE283                     CALL   CNVRT
83DD 21F927                     LXI    H,OBUFF  ;RETURN ADDRESS OF OUTPUT BUFFER,D, + 
83E0 C1                         POP    B
83E1 C9                         RET
83E2 79                   CNVRT: MOV   A,C      ;CONVERT 4 HIGH ORDER BITS
83E3 0F                         RRC
83E4 0F                         RRC
83E5 0F                         RRC
83E6 0F                         RRC
83E7 CDEC83                     CALL   CNV0
83EA 23                         INX    H        ;UPDATE POINTER
83EB 79                         MOV    A,C      ;CONVERT 4 LOW ORDER BITS
83EC E60F                 CNV0: ANI    0FH      ;MASK OUT 4 HIGH ORDER BITS
83EE C2F583                     JNZ    CNV1     ;SKIP IF CHARACTER NOT EQUAL ZERO
83F1 78                         MOV    A,B      ;GET ADEQUATE REPRESENTATION OF ZERO
83F2 C3F783                     JMP    $+5      ;SKIP FOR CHARACTER WAS ZERO
83F5 0600                 CNV1: MVI    B,0      ;SET B=0 FOR NOT LEADING ZERO
83F7 77                         MOV    M,A      ;SEND CHARACTER TO OUTPUT BUFFER
83F8 C9                         RET
                             ;-----------------------------------------------------------------
                             ;NAME:      FLTBCD
                             ;INPUTS:    D,E - FOUR DIGIT BCD
                                         H,L - ADDRESS OF RESULT
                             ;OUTPUTS:   NONE
                             ;CALLS:     FLTDS,FSTOR
                             ;DESTROYS:  A,B,C,D,E,H,L
                             ;DESCRIPTION:FLTBCD CONVERT BCD DIGIT INTO FLOATING POINT FORMAT FPR
                             
83F9 E5                   FLTBCD: PUSH H        ;SAVE ADDRESS OF RESULT ON STACK
83FA 7B                         MOV    A,E      ;GET TWO LOWER DIGITS
83FB CDEC81                     CALL   BINBCD   ;CONVERT INTO BINARY
83FE 4F                         MOV    C,A      ;SAVE IN C
83FF 7A                         MOV    A,D      ;GET TWO HIGHER DIGITS
8400 CDEC81                     CALL   BINBCD   ;CONVERT INTO BINARY AND
8403 CD3D84                     CALL   MBH      ;MULTIPLY BY HUNDRED
8406 79                         MOV    A,C      ;ADD LOWER BYTE
8407 85                         ADD    L
8408 6F                         MOV    L,A
8409 7C                         MOV    A,H
840A CE00                       ACI    0
840C 67                         MOV    H,A
840D 221327                     SHLD   TEMP     ;STORE IN TEMP
8410 210000                     LXI    H,0      ;CLEAR TWO HIGHER BYTES OF A STANDARD INTEGER FORMAT
8413 221527                     SHLD   TEMP+2
8416 010027                     LXI    B,FPR    ;POINTER TO FPR
8419 111327                     LXI    D,TEMP   ;CONVERT INTO FPR
841C CD0000    E                CALL   FLTDS
841F D1                         POP    D        ;RETRIEVE ADDRESS OF RESULT
8420 C30000    E                JMP    FSTOR
                             ;-----------------------------------------------------------------
                             ;NAME:      ININT - INPUT INTERRUPT ROUTINE
                             ;INPUTS:    NONE
```

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;OUTPUTS:    NONE
                        ;CALLS       NOTHING
                        ;DESTROYS:   NOTHING
                        ;DESCRIPTION: ININT IS ENTERED WHEN POKED ROUTINE IS WAITING FOR A
                        ;             CHARACTER AND THE USER HAS PRESSED A KEY ON THE KEYBOARD
                        ;             THE INPUT CHARACTER IS STORED IN THE INPUT BUFFER

8423 E5                 ININT:  PUSH    H
8424 F5                         PUSH    PSW
8425 210019                     LXI     H,CNTRL  ;ADDRESS FOR CONTROL CHARACTER OUTPUT
8428 3640                       MVI     M,READ   ;OUTPUT CONTROL CHARACTER FOR READING FROM KEYBOARD
842A 25                         DCR     H        ;ADDRESS FOR CHARACTER INPUT
842B 7E                         MOV     A,M      ;READ A CHARACTER
842C E63F                       ANI     3FH      ;ZERO TWO HIGH ORDER BITS
842E 32FE27                     STA     IBUFF    ;STORE CHARACTER IN INPUT BUFFER
8431 F1                         POP     PSW
8432 E1                         POP     H
8433 C9                         RET
                        ;------------------------------------------------------------------
                        ;NAME:       INSDG - INSERT DIGIT
                        ;INPUTS.     A - BCD DIGIT TO BE INSERTED
                        ;            D,E - BCD NUMBER
                        ;OUTPUTS:    D,E - BCD NUMBER WITH DIGIT INSERTED
                        ;CALLS:      NOTHING
                        ;DESTROYS:   A
                        ;DESCRIPTION: INSDG SHIFTS THE CONTENTS OF D,E LEFT 4 BITS AND INSERTS
                        ;             THE BCD DIGIT IN A IN THE EMPTIED POSITION

8434 EB                 INSDG:  XCHG             ;EXCHANGE D,E WITH H,L
8435 29                         DAD     H        ;SHIFT H,L LEFT 4 BITS
8436 29                         DAD     H
8437 29                         DAD     H
8438 29                         DAD     H
8439 B5                         ORA     L
843A 6F                         MOV     L,A
843B EB                         XCHG             ;EXCHANGE BACK
843C C9                         RET
                        ;------------------------------------------------------------------
                        ;NAME:       MBH - MULTIPLY BY HUNDRED
                        ;INPUTS:     A - NUMBER TO BE MULTIPLIED
                        ;OUTPUTS:    H,L - MULTIPLIED NUMBER
                        ;CALLS:      NOTHING
                        ;DESTROYS:   D,E,H,L
                        ;DESCRIPTION: MBH MOVES A BINARY NUMBER IN ACCUMULATOR INTO H,L AND
                        ;             MULTIPLIES IT BY HUNDRED

843D 6F                 MBH:    MOV     L,A      ;MOV BINARY NUMBER (BN) TO H,L
843E 2600                       MVI     H,0
8440 29                         DAD     H
8441 29                         DAD     H        ;BN*4
8442 54                         MOV     D,H      ;SAVE (BN*4) IN D,E
8443 5D                         MOV     E,L
8444 29                         DAD     H
8445 29                         DAD     H
8446 29                         DAD     H        ;BN*32
```

```
LOC  OBJ         LINE        SOURCE STATEMENT

8447 EB                 XCHG           ;N*32 IN D,E  N*4 IN H,L
8448 19                 DAD     D      ;N*36
8449 EB                 XCHG           ;N*36 IN D,E  N*32 IN H,L
844A 29                 DAD     H      ;N*64
844B 19                 DAD     D      ;N*100
844C C9                 RET
                    ;-------------------------------------------------------------
                    ;NAME:       OUTPUT - OUTPUT CHARACTERS TO DISPLAY
                    ;INPUTS:     A - DISPLAY FLAG - 0=USE ADDRESS FIELD
                    ;                               1=USE DATA FIELD
                    ;            H,L - ADDRESS OF CHARACTERS TO BE OUTPUT
                    ;OUTPUTS:    NONE
                    ;CALLS:      NOTHING
                    ;DESTROYS:   A,C,D,E,H,L
                    ;DESCRIPTION: OUTPUT SENDS CHARACTERS TO THE DISPLAY EITHER 2 CHARACTERS
                    ;            ARE SENT TO THE DATA FIELD, OR 4 CHARACTERS ARE SENT TO THE
                    ;            ADDRESS FIELD, DEPENDING ON THE DISPLAY FLAG.

844D 0F             OUTPUT: RRC            ;USE DATA FIELD
844E DA5884                 JC      OTP1   ;YES - GO SET UP TO USE DATA FIELD
8451 0E04                   MVI     C,4    ;NO - COUNT FOR ADDRESS FIELD
8453 3E90                   MVI     A,ADISP ;CONTROL CHARACTER FOR OUTPUT TO ADDRESS FIELD
8455 C35C84                 JMP     $+7    ;SKIP
8458 0E02           OTP1:   MVI     C,2    ;COUNT FOR DATA FIELD
845A 3E94                   MVI     A,DDISP ;CONTROL CHARACTER FOR OUTPUT TO DATA FIELD
845C 320019                 STA     CNTRL
845F 7E             OTP2:   MOV     A,M    ;GET OUTPUT CHARACTER
8460 EB                     XCHG           ;SAVE CHARACTER ADDRESS IN D,E
8461 219285                 LXI     H,CHARTB ;TRANSLATING TABLE BASE ADDRESS
8464 85                     ADD     L      ;USE OUTPUT CHARACTER AS A POINTER TO CHARTB
8465 6F                     MOV     L,A
8466 7C                     MOV     A,H
8467 CE00                   ACI     0
8469 67                     MOV     H,A
846A 7E                     MOV     A,M    ;GET DISPLAY FORMAT CHARACTER FROM TABLE
846B 2F                     CMA            ;COMPLEMENT IT AND SEND TO THE DISPLAY
846C 320018                 STA     DSPLY
846F 0D                     DCR     C      ;ANY MORE OUTPUT CHARACTERS?
8470 C8                     RZ             ;NO - RETURN
8471 EB                     XCHG           ;RETRIEVE CHARACTER ADDRESS IN H,L
8472 23                     INX     H      ;NEXT OUTPUT CHARACTER
8473 C35F84                 JMP     OTP2   ;GO PROCESS ANOTHER CHARACTER
                    ;-------------------------------------------------------------
                    ;NAME:       RDKBD - READ KEYBOARD
                    ;INPUTS:     NONE
                    ;OUTPUTS:    A - CHARACTER READ FROM KEYBOARD
                    ;CALLS:      NOTHING
                    ;DESTROYS:   A,H,L
                    ;DESCRIPTION: RDKBD DETERMINES WHETHER OR NOT THERE IS A CHARACTER IN
                    ;            THE INPUT BUFFER. IF NOT, RDKBD ENABLES INTERRUPTS AND
                    ;            LOOPS UNTIL THE INPUT INTERRUPT ROUTINE STORES A CHARACTER
                    ;            IN THE BUFFER. WHEN THE BUFFER CONTAINS A CHARACTER, RDKBD
                    ;            FLAGS THE BUFFER AS EMPTY AND RETURNS THE CHARACTER IN
                    ;            ACCUMULATOR.
```

```
LOC  OBJ         LINE        SOURCE STATEMENT

8476 21FE27        715 RDKBD:  LXI     H,IBUFF ;POINTER TO INPUT BUFFER
8479 7E            716         MOV     A,M     ;GET BUFFER CONTENTS
847A A7            717         ANA     A       ;IS A CHARACTER AVAILABLE.?
847B F28284        718         JP      RXIT    ;YES - EXIT FROM LOOP
847E FB            719         EI              ;NO -,READY FOR CHARACTER FROM KEYBOARD
847F C37684        720         JMP     RDKBD
8482 3600          721 RXIT:   MVI     M,EMPTY ;SET BUFFER EMPTY FLAG
8484 F3            722         DI              ;RETURN WITH INTERRUPTS DISABLED
8485 C9            723         RET
                   724 ;-----------------------------------------------------------------
                   725 ;NAME        REPTIM - REPETITION TIME
                   726 ;INPUTS:     NONE
                   727 ;OUTPUTS:    NONE
                   728 ;CALLS:      FDIV,FLTDS,FSTOR
                   729 ;DESTROYS:   A,B,C,D,E,H,L
                   730 ;DESCRIPTION: REPTIM COMPUTES A REPETITION TIME FOR PULSES PRESENTED TO
                   731              THE SID AT THE CPU
                   732
8486 21FFFF        733 REPTIM: LXI     H,0FFFFH;SET TSYNC = 0,0,FF,FF,MSB FIRST
8489 224B27        734         SHLD    TSYNC
848C 23            735         INX     H
848D 224D27        736         SHLD    TSYNC+2
8490 20            737         RIM             ;CHECK WHETHER PULSE IS PRESENT OR NOT
8491 07            738         RLC
8492 D29084        739         JNC     $-2     ;WAIT UNTIL PULSE ARRIVES
8495 CDB684        740         CALL    MILSEC  ;COUNT MILLISECONDS STARTING FROM ZERO
8498 DA9584        741         JC      $-3     ;CONTINUE UNTIL PULSE DISAPPEARS
849B CDB684        742         CALL    MILSEC  ;CONTINUE UNTIL NEXT PULSE ARRIVES
849E D29B84        743         JNC     $-3
84A1 010027        744         LXI     B,FPR   ;CONVERT TSYNC INTO FPR
84A4 114B27        745         LXI     D,TSYNC
84A7 CD0000    E   746         CALL    FLTDS
84AA 11D185        747         LXI     D,F1000 ;EXPRESS TSYNC IN SECONDS
84AD CD0000    E   748         CALL    FDIV
84B0 114B27        749         LXI     D,TSYNC ;STORE IN MEMORY
84B3 C30000    E   750         JMP     FSTOR
84B6 2A4B27        751 MILSEC: LHLD    TSYNC   ;COUNT TIME
84B9 23            752         INX     H
84BA 224B27        753         SHLD    TSYNC
84BD CDC684        754         CALL    WAIT    ;WAIT OUT ONE MILLISECOND
84C0 20            755         RIM             ;SAMPLE SID
84C1 07            756         RLC
84C2 00            757         NOP             ;FOR TIME BALANCE
84C3 00            758         NOP
84C4 00            759         NOP
84C5 C9            760         RET
84C6 3E40          761 WAIT:   MVI     A,40H   ;LOOP FOR ONE MILLISECOND
84C8 E3            762         XTHL
84C9 E3            763         XTHL
84CA 3D            764         DCR     A
84CB C2C884        765         JNZ     WAIT+2
84CE C9            766         RET
                   767 ;-----------------------------------------------------------------
                   768 ;NAME:       SETTIM - SET TIMER
                   769 ;INPUTS:     NONE
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

```
 LOC  OBJ         LINE        SOURCE STATEMENT

                   770 ;OUTPUTS:  NONE
                   771 ;CALLS:     FIXSD.FLOAD.FMUL.FSUB
                   772 ;DESTROYS: A.B.C.D.E
                   773 ;DESCRIPTION: SETTIM COMPUTES AND OUTPUTS TIMER COUNT FOR TRANGE
                   774
 84CF 010027       775 SETTIM: LXI     B.FPR    ;POINTER TO FPR
 84D2 119527       776         LXI     D.TE     ;COMPUTE TRANGE IN SECONDS
 84D5 CD0000   E   777         CALL    FLOAD
 84D8 114727       778         LXI     D.TIME
 84DB CD0000   E   779         CALL    FSUB
 84DE 11CD85       780         LXI     D.F400   ;MULTIPLY BY 400 TO COUNT WITH 400 HZ CLOCK
 84E1 CD0000   E   781         CALL    FMUL
 84E4 111327       782         LXI     D.TEMP   ;CONVERT INTO INTEGER
 84E7 CD0000   E   783         CALL    FIXSD
 84EA 3A1427       784         LDA     TEMP+1   ;GET HIGH ORDER BYTE OF TIMER COUNT
 84ED E63F         785         ANI     3FH      ;SINGLE SQUARE WAVE MODE
 84EF D32D         786         OUT     ETIMH    ;SET HIGH ORDER BYTE OF TIMER COUNT
 84F1 3A1327       787         LDA     TEMP     ;GET LOW ORDER BYTE OF TRANGE
 84F4 D32C         788         OUT     ETIML    ;SET LOW ORDER BYTE OF TIMER COUNT
 84F6 C9           789         RET
                   790 ;------------------------------------------------------------
                   791 ;NAME:      SYNINT - TSYNC INTERRUPT ROUTINE
                   792 ;INPUTS:    NONE
                   793 ;OUTPUTS:   NONE
                   794 ;CALLS:     NOTHING
                   795 ;DESTROYS: NOTHING
                   796 ;DESCRIPTION: SYNINT STARTS THE TIMER AND TAKES CARE ABOUT THE ORDER IN
                   797 ;            THE INTERRUPT SEQUENCE
                   798
 84F7 F5           799 SYNINT: PUSH    PSW
 84F8 3ECF         800         MVI     A.TSTRT ;START TIMER
 84FA D328         801         OUT     ECSR
 84FC 3A1227       802         LDA     FLAG
 84FF 17           803         RAL              ;TSYNC FLAG CLEARED ?
 8500 DAB483       804         JC      ERR1     ;NO. ERROR 1 = REPETITION TIME OF TSYNC TOO SMALL
 8503 37           805         STC              ;SET TSYNC FLAG
 8504 1F           806         RAR
 8505 321227       807         STA     FLAG
 8508 F1           808         POP     PSW
 8509 C9           809         RET
                   810 ;------------------------------------------------------------
                   811 ;NAME:      TIMINT - TIMER INTERRUPT ROUTINE
                   812 ;INPUTS:    NONE
                   813 ;OUTPUTS:   NONE
                   814 ;CALLS:     ECHO
                   815 ;DESTROYS: A,B,C,D,E,H,L
                   816 ;DESCRIPTION: TIMINT GENERATES START PULSE FOR THE CLOCK CIRCUITRY
                   817 ;            AND SIMULATES RETURN SIGNAL
                   818
 850A 3E4F         819 TIMINT: MVI     A,TSTOP ;STOP TIMER
 850C D328         820         OUT     ECSR
 850E 3EC0         821         MVI     A,HOUT  ;SET START PULSE
 8510 30           822         SIM
 8511 CD7583       823         CALL    ECHO    ;SIMULATE RETURN SIGNAL
 8514 3E40         824         MVI     A,LOUT  ;RESET START PULSE
```

```
LOC  OBJ              SOURCE STATEMENT

8516 30                   SIM
8517 AF                   XRA     A         ;RESET TSYNC FLAG
8518 321227               STA     FLAG
851B C9                   RET
                     ;-------------------------------------------------------------------
                     ;NAME:        UNITS - EXPRESS DATA IN APPROPRIATE UNITS
                     ;INPUTS:      NONE
                     ;OUTPUTS:     NONE
                     ;CALLS:       FLOAD,FMUL,F000O,FSTOR,RECPOL
                     ;DESTROYS:    A,B,C,D,E,H,L
                     ;DESCRIPTION: UNITS CONVERTS DIFFERENT DATA INTO APPROPRIATE UNITS IN
                     ;              RECTANGULAR COORDINATES

851C 010027          UNITS:  LXI     B,FPR
851F 11D585                  LXI     D,C1      ;CONVERT VELTAR FROM KNOTS INTO YARDS/SEC
8522 CD0000     E            CALL    FLOAD
8525 115327                  LXI     D,VELTAR
8528 CD0000     E            CALL    FMUL
852B CD0000     E            CALL    FSTOR
852E 11D585                  LXI     D,C1      ;CONVERT VELREC FROM KNOTS INTO YARDS/SEC
8531 CD0000     E            CALL    FLOAD
8534 114F27                  LXI     D,VELREC
8537 CD0000     E            CALL    FMUL
853A CD0000     E            CALL    FSTOR
853D 11D985                  LXI     D,COEF    ;CONVERT ANGLE INTO RADIANS
8540 CD0000     E            CALL    FLOAD
8543 115B27                  LXI     D,ANGLE
8546 CD0000     E            CALL    FMUL
8549 CD0000     E            CALL    FSTOR
854C 11D985                  LXI     D,COEF    ;CONVERT ALPHA INTO RADIANS
854F CD0000     E            CALL    FLOAD
8552 115F27                  LXI     D,ALPHA
8555 CD0000     E            CALL    FMUL
8558 CD0000     E            CALL    FSTOR
855B 11B985                  LXI     D,F0P3    ;CONVERT VSOUND INTO YARDS/SEC
855E CD0000     E            CALL    FLOAD
8561 116327                  LXI     D,VSOUND
8564 CD0000     E            CALL    FMUL
8567 CD0000     E            CALL    FSTOR
856A CD0000     E            CALL    F00G0     ;SYSTEM CALL PROCEEDING ANY FORTRAN SUBROUTINE
856D C30000     E            JMP     RECPOL
                     ;-------------------------------------------------------------------
                     ;NAME:        UPDATE - UPDATE STARTING VALUE
                     ;INPUTS:      H,L - BCD NUMBER TO BE UPDATED
                     ;OUTPUTS:     D,E - UPDATED NUMBER
                     ;CALLS:       DAF,EXPAND,INSDG,ROKBD
                     ;DESTROYS:    A,D,E,H,L
                     ;DESCRIPTION: UPDATE DISPLAYS A FORMER STARTING VALUE IN ADDRESS FIELD
                     ;              AND UPDATES IT ACCORDING TO WHAT'S TYPED IN FROM THE KEYBOARD
                     ;              ACCEPTED ARE FOUR LAST DECIMAL NUMBERS BEFORE THE      KEY IS
                     ;              PRESSED. LEADING ZEROS ARE BLANKED OUT

8570 EB              UPDATE: XCHG              ;MOVE BCD NUMBER TO D,E
8571 CDCE83                  CALL    EXPAND    ;EXPAND THIS FOR DISPLAY
8574 D5                      PUSH    D
```

```
  LOC  OBJ        LINE        SOURCE STATEMENT

  8575 CDFC81      880         CALL    DAF
  8578 D1          881         POP     D
  8579 CD7684      882 GCHAR:  CALL    RDKBD   ;READ KEYBOARD
  857C FE0A        883         CPI     0AH     ;IS CHARACTER A DECIMAL DIGIT ?
  857E D28F85      884         JNC     NDEC    ;NO - GO CHECK FOR TERMINATOR
  8581 CD3484      885         CALL    INSDG   ;INSERT NEW DIGIT
  8584 CDCE83      886         CALL    EXPAND  ;EXPAND BCD NUMBER FOR DISPLAY
  8587 D5          887         PUSH    D       ;SAVE NUMBER
  8588 CDFC81      888         CALL    DAF     ;DISPLAY IN ADDRESS FIELD
  858B D1          889         POP     D       ;RESTORE NUMBER
  858C C37985      890         JMP     GCHAR   ;GO GET NEXT CHARACTER
  858F FE10        891 NDEC:   CPI     EXEC    ;WAS LAST CHARACTER 'EXEC' ?
  8591 C9          892         RET             ;IF SO, RETURN WITH Z
                   893 ;------------------------------------------------------------
                   894 ;       TABLE FOR TRANSLATING CHARACTERS TO DISPLAY
                   895 ;------------------------------------------------------------
  8592 F3          896 CHARTB: DB      0F3H    ;0
  8593 60          897         DB      60H     ;1
  8594 B5          898         DB      0B5H    ;2
  8595 F4          899         DB      0F4H    ;3
  8596 66          900         DB      66H     ;4
  8597 D6          901         DB      0D6H    ;5
  8598 D7          902         DB      0D7H    ;6
  8599 70          903         DB      70H     ;7
  859A F7          904         DB      0F7H    ;8
  859B 76          905         DB      76H     ;9
  859C 00          906         DB      00H     ;BLANK
  859D 97          907         DB      97H     ;E
  859E 85          908         DB      85H     ;R (LOWER CASE)
                   909 ;------------------------------------------------------------
                   910 ;               MESSAGES TO DISPLAY
                   911 ;------------------------------------------------------------
  859F 0A          912 ZERO:   DB      10,0
  85A0 00
  85A1 0A          913 ONE:    DB      10,1
  85A2 01
  85A3 0A          914 TWO:    DB      10,2
  85A4 02
  85A5 0A          915 THREE:  DB      10,3
  85A6 03
  85A7 0A          916 FOUR:   DB      10,4
  85A8 04
  85A9 0A          917 FIVE:   DB      10,5
  85AA 05
  85AB 0A          918 SIX:    DB      10,6
  85AC 06
  85AD 0A          919 SEVEN:  DB      10,7
  85AE 07
  85AF 06          920 SIXTY:  DB      6,0
  85B0 00
  85B1 0A          921 BLANKS: DB      10,10,10,10
  85B2 0A
  85B3 0A
  85B4 0A
  85B5 0B          922 ERROR:  DB      11,12,12,10
```

```
LOC  OBJ          LINE        SOURCE STATEMENT

85B6 0C
85B7 0C
85B8 0A
                   923 ;--------------------------------------------------------------
                   924 ;                 FLOATIG POINT CONSTANTS
                   925 ;--------------------------------------------------------------
85B9 AA            926 F0P3:    DB      0AAH,0AAH,0AAH,3EH
85BA AA
85BB AA
85BC 3E
85BD 00            927 F0P5:    DB      0,0,0,3FH
85BE 00
85BF 00
85C0 3F
85C1 AA            928 F0P6:    DB      0AAH,0AAH,2AH,3FH
85C2 AA
85C3 2A
85C4 3F
85C5 00            929 F15:     DB      0,0,70H,41H
85C6 00
85C7 70
85C8 41
85C9 00            930 F100:    DB      0,0,0C8H,42H
85CA 00
85CB C8
85CC 42
85CD 00            931 F400:    DB      0,0,0C8H,43H
85CE 00
85CF C8
85D0 43
85D1 00            932 F1000:   DB      0,0,7AH,44H
85D2 00
85D3 7A
85D4 44
85D5 F8            933 C1:      DB      0F8H,0DBH,14H,3FH
85D6 DB
85D7 14
85D8 3F
85D9 35            934 COEF:    DB      35H,0FAH,8EH,3CH
85DA FA
85DB 8E
85DC 3C
85DD DB            935 PI:      DB      0DBH,0FH,49H,40H
85DE 0F
85DF 49
85E0 40
85E1 DB            936 PI05:    DB      0DBH,0FH,0C9H,3FH
85E2 0F
85E3 C9
85E4 3F
85E5 E4            937 PI15:    DB      0E4H,0CBH,96H,40H
85E6 CB
85E7 96
85E8 40
85E9 DB            938 PI20:    DB      0DBH,0FH,0C9H,40H
```

```
 LOC  OBJ         LINE        SOURCE STATEMENT

 85EA 0F
 85EB C9
 85EC 40
 85ED 66          939 FREQ1:  DB      66H,66H,0E5H,40H
 85EE 66
 85EF E5
 85F0 40
                  940 ;------------------------------------------------------------------
                  941 ;                        DATA
                  942 ;------------------------------------------------------------------
                  943          ASEG
 2700             944          ORG     2700H
 2700             945 FPR:     DS      18        ;FLOATING POINT RECORD
 2712             946 FLAG:    DS      1         ;TSYNC FLAG
 2713             947 TEMP:    DS      4         ;TEMPORARY STORAGE
 2717             948 VELRB:   DS      2         ;VELREC IN BCD FORMAT
 2719             949 VELTB:   DS      2
 271B             950 RANGEB:  DS      2
 271D             951 ANGLEB:  DS      2
 271F             952 ALPHAB:  DS      2
 2721             953 ENLOB:   DS      2
 2723             954 VSNDB:   DS      2
 2725             955 EVAMP:   DS      17
 2736             956 ODAMP:   DS      17
 2747             957 TIME:    DS      4
 274B             958 TSYNC:   DS      4         ;REPETITION TIME OF TSYNC PULSES
 274F             959 VELREC:  DS      4         ;VELREC IN FLOATING POINT FORMAT
 2753             960 VELTAR:  DS      4
 2757             961 RANGE:   DS      4
 275B             962 ANGLE:   DS      4
 275F             963 ALPHA:   DS      4
 2763             964 VSOUND:  DS      4
 2767             965 XVELTA:  DS      4         ;X-PART OF VELTAR
 276B             966 YVELTA:  DS      4         ;Y-PART OF VELTAR
 276F             967 XINTAR:  DS      4         ;X COORDINATE OF TARGET
 2773             968 YINTAR:  DS      4         ;Y COORDINATE OF TARGET
 2777             969 AMPEV:   DS      4
 277B             970 AMPOD:   DS      4
 277F             971 EVSEC:   DS      1
 2780             972 ODSEC:   DS      1
 2781             973 NELDEL:  DS      1
 2782             974 NOLDEL:  DS      1
 2783             975 NFREQ:   DS      2
 2785             976 TA:      DS      4
 2789             977 TB:      DS      4
 278D             978 TC:      DS      4
 2791             979 TD:      DS      4
 2795             980 TE:      DS      4
 2799             981 XTAR2:   DS      4
 279D             982 YTAR2:   DS      4
 27A1             983 FLG:     DS      1
 27A2             984 XREC1:   DS      4
 27F9             985          ORG     27F9H
 27F9             986 OBUFF:   DS      5
 27FE             987 IBUFF:   DS      1
```

```
  LOC  OBJ         LINE        SOURCE STATEMENT

                    988
                    989        END

PUBLIC SYMBOLS


EXTERNAL SYMBOLS
COMP1  E 0000     COMP2  E 0000     FADD   E 0000     FDIV   E 0000     FIXED  E 0000     FLOAD  E 0000     FLTDS  E 0000
FMUL   E 0000     FOOGO  E 0000     FSET   E 0000     FSTOP  E 0000     [illegible] E 0000     RECPOL E 0000

USER SYMBOLS
ADISP  A 0090     ALLOFF A 000C     ALLON  A 0000     ALPHA  A 275F     ALPHA2 A [illegible]     AMP    A [illegible]     [illegible] A [illegible]
AMPOD  A 277B     AMPP   A 0022     ANGLE  A 275B     ANGLEB A 2710     AT1    A [illegible]     ATEST  A [illegible]     [illegible] A [illegible]
BINBCD A 815C     BLANKS A 9551     BTIMC  A 1500     BTIMH  A 0025     BTIML  A [illegible]     BTIMY  A [illegible]     [illegible] A [illegible]
CHARTB A 8592     CNTRL  A 1900     CNV0   A 93EC     CNV1   A 93F5     CNVRT  A [illegible]     CODE   A [illegible]     COMP1  E 0000
COMP2  E 0000     COMPUT A 806E     DAF    A 91FC     DATA1  A 9200     DATA2  A [illegible]     DATA3  A [illegible]     DATA4  A [illegible]
DATA5  A 92E8     DATA6  A 9314     DATA7  A 9342     DATAIN A 9201     [illegible] A [illegible]     [illegible] A [illegible]     DELF   A [illegible]
DLY    A 93A5     DSPLY  A 1800     ECHO   A 8375     ECSP   A 0028     EMPTY  A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
ENLOP2 A 9054     ENLOP3 A 911F     EPR1   A 93B4     EPR2   A 93B8     [illegible] A [illegible]     [illegible] A [illegible]     ETIMH  A [illegible]
ETIML  A 002C     EXAMP  A 2725     EXSEC  A 277F     EXSP   A 0027     EXEC   A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
EXPAND A 9ICE     F0P3   A 9589     F0P5   A 958D     F0P6   A 9591     F100   A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
F400   A 95CD     FADD   E 0000     FDIV   E 0000     FIVE   A 95A9     FIXED  E 0000     [illegible] [illegible]     [illegible] A [illegible]
FLOAD  E 0000     FLTBCD A 83F9     FLTDS  E 0000     FMUL   E 0000     FOUR   A [illegible]     [illegible] A [illegible]     FOOGO  E 0000
FREQ1  A 95ED     FROP   A 0029     FSET   E 0000     FSTOP  E 0000     [illegible] E [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
HOUT   A 00C0     IBUFF  A 27FE     ININT  A 8423     INSDG  A 9434     INSTR  A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
MSH    A 943D     MILSEC A 8495     NDEC   A 959F     NELDEL A 2791     NEXT   A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
OBUFF  A 27F9     ODAMP  A 2735     ODSEC  A 2790     ODSP   A 0028     OK     A [illegible]     ONE    A [illegible]     [illegible] A [illegible]
OTP2   A 845F     OUTPUT A 8440     PI     A 95D0     PI05   A 9551     PI45   A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
RANGEB A 271B     ROKED  A 8475     READ   A 0040     RECPOL E 0000     REP1   A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
REP4   A 920F     REP5   A 820B     REP6   A 8229     REP7   A 9199     RESTIM A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
SETTIM A 840F     SEVEN  A 95AD     SIMUL  A 9040     SIX    A 95A9     [illegible] A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
STORE  A 81DA     SYNINT A 84F7     TA     A 2795     TB     A 2799     TC     A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
TEMP   A 2713     THREE  A 95A5     TIME   A 2747     TIMINT A 950A     TSTOP  A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
TWO    A 95A3     UNITS  A 9510     UPDATE A 9579     VELAB  A 2717     VELSEC A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
VSNDB  A 2723     VSOUND A 276D     WAIT   A 9406     XINTAR A 275F     XREC1  A [illegible]     [illegible] A [illegible]     [illegible] A [illegible]
YINTAR A 2773     YTAR2  A 279D     YVELTA A 276B     ZERO   A 959F

ASSEMBLY COMPLETE,   NO ERRORS
```

POOR COPY
COPIE DE QUALITEE INFERIEURE

REFERENCES

1. Department of National Defence, Statement of Work, Sonar Signal Injector, May 1981.

2. Department of National Defence, AN/SQS 505 Receiver Book.

3. Department of National Defence, AN/SQS 505 Transmitter Book.

4. Blake, L. V., " Radar Range-Performance Analysis ", Lexington Books, 1980.

5. Woollard, B. G., " Microprocessors and Microcomputers ", McGraw Hill Book Company, 1981.

6. Zaks, R., " Microprocessors ", Sybex Incorporated, 1977.

7. Moore, J. B., Makela, L. J., " Structured Fortran with WATFIV ", Prentice-Hall Company.

8. 8080/8085 Floating Point Arithmatic Library User's Manual.

9. MCS-80/85 Family User's Manual, October 1979.

10. SDK-85 System Design Kit User's Manual.

11. 8080/8085 Assembly Language Programming Manual.

12. The TTL Data Book for Design Engineers, Texas Instruments Incorporated.

13. Component Data Catalog, Intel, 1981.

14. Semiconductor Data Library/CMOS, Motorola Semiconductor Products Inc.