TRANSIENT IMPEDANCE DISPLAY UNIT

by

JIHAD M. A. RASHID

A thesis
presented to the University of Manitoba
in partial fulfillment of the
requirements for the degree of
Master of Science
in
Electrical Engineering

Winnipeg, Manitoba, 1983

June 29, 1983

TRANSIENT IMPEDANCE DISPLAY UNIT

BY

JIHAD M.A. RASHID

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

To:

Miss Layalee K. Rashid with LOVE

ACKNOWLEDGMENTS

The author wishes to acknowledge his true indebtedness to professor GLEN SWIFT for suggesting the topic and never failing, father style, guidance and support during work on this thesis. Many thanks to professor MAURICE YUNIK, his many suggestions and assistance were of great importance.

The author would like to thank Dr. DOUG CHAPMAN of Manitoba Hydro and professor FRED ZEILER of department of applied mathematics for examining this thesis.

In addition the author would like to express his gratitude to all technical staff in the electronics shop for their help and in particular for Mr. Ken Biegun for building the final design version and the continuous assistance.

Manitoba Hydro and NSERC financial aid is appreciated.

ABSTRACT

In this thesis it is attempted to design an Impedance display device capable of providing a visual display, on an oscilloscope screen, the trajectories of the 60-Hz transient impedance seen at a relaying point during a power system swing. A digital approach is adopted using the Motorola Inc. MC6802 microprocessor. The design is built and tested on a simulated one machine versus infinite bus power system.

CONTENTS

Chapter I

INTRODUCTION

Since many protective relays work on the principle of sensing the current and voltage at a relaying point and calculating the 60-Hz impedance seen in a particular direction, it is useful to plot system apparent impedance as seen by the relay on an impedance plane. A visual display of this instantaneous impedance trajectory on an oscilloscope screen, along with relay characteristics, gives a better understanding of system performance. From these trajectories the difference between normal, fault, and swing conditions can be detected.

When a power system "swings", the apparent impedance seen by a distance relay can be mis-interpretted as a fault condition and cause an undesired trip. A distance relay should be capable of discriminating between fault and power swing conditions. It is necessary too that the relay is set for the maximum expected steady-state loading. A power system apparent impedance along with an ohmic type of protective relay characteristic gives a complete picture of system behaviour under swing conditions.

The impedance diagram in Figure 1.1 shows two generators at the ends of a transmission line swinging with respect to each other. The locus of the apparent impedance as seen by a relay at point A is shown in dotted line. Point L on this locus represent the normal steady-state loading of the system. As instability develops and one generator (A) swings ahead of the other generator (D), this point will drift along a trajectory defined by the ratio of generators voltages, say $E_A/E_D = 1.1$. A distance relay located at A may interpret the swing as a fault, and to avoid this, out-of-step blocking can be employed. Conversely, a relay may be used to detect an unstable condition. This is called an out-of-step tripping relay shown in Fig. 1.1.

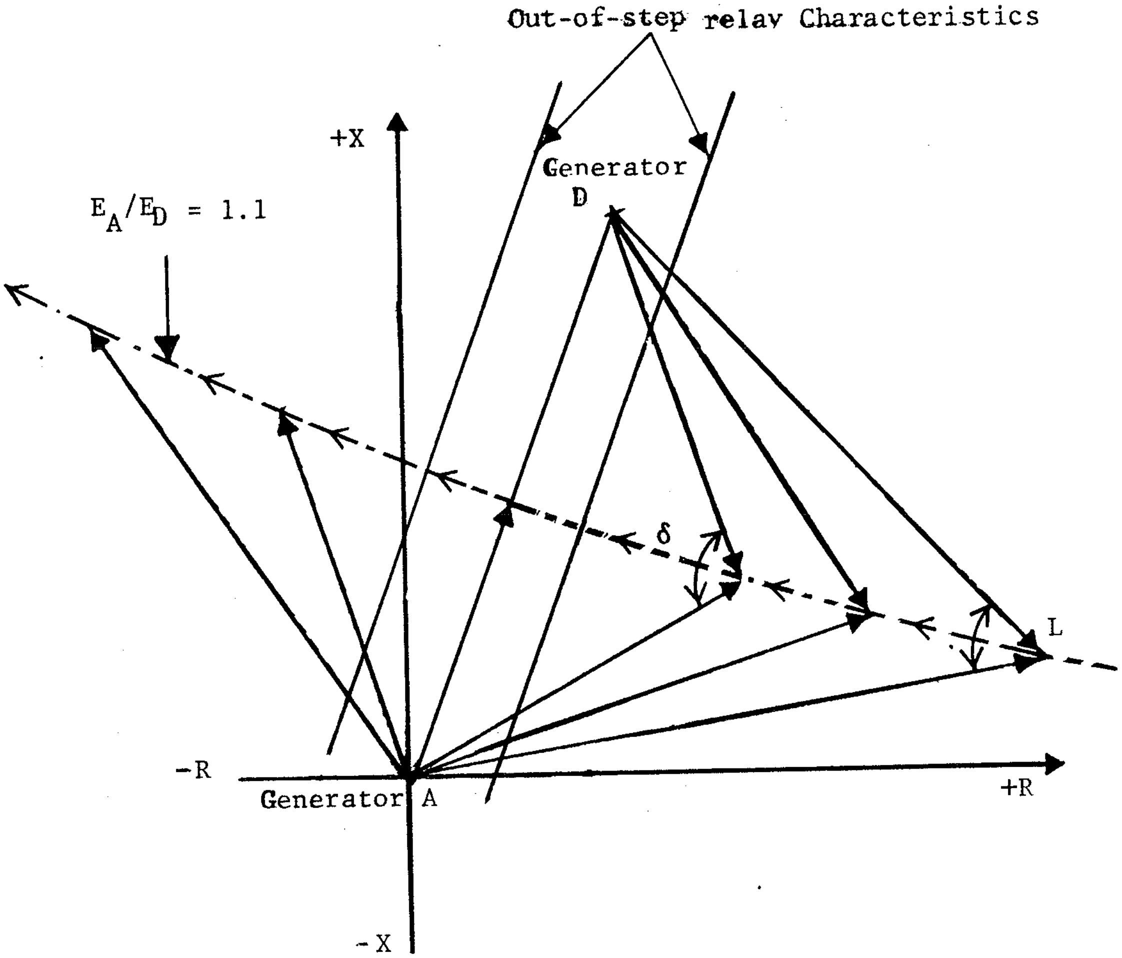


Fig. 1.1 Swing ohm trajectory on impedance plane

Chapter II

POWER SWINGS AND PROTECTIVE RELAYING

2.1 INTRODUCTION TO POWER SYSTEM STABILITY:

Instability of a power system comes about as a result of attempting to transfer power from one location to another in excess of the system capability. This system capability is known as stability limit and is dependent on the source voltage levels, the impedance between the sources and sine of the electrical angle between these voltages assuming all resistances and saliency can be ignored. The steady-state limit can be expressed in mathematical form for a two machine system as:

$$P_{max.} = \frac{E_s \cdot E_r}{X} \quad \text{Watt} \quad \ldots\ldots\ldots\ldots\ldots 2.1$$

where the symbols have the following meanings:

$P_{max.}$ is the maximum power transfered

E_s is the sending end generator internal voltage (line to line volts)

E_r is the receiving end generator internal voltage (line to line volts)

X is the total reactance between the generators internal voltages in ohms per phase

Equation 2.1 gives the maximum power transfer for the basic system shown in Figure 2.1 and this is for the condition where the angle between sending and receiving end voltages is ninety degrees. This equation can be generalised to give the system power transfer capability

$$P = \frac{E_s \cdot E_r}{X} \cdot \sin(\delta) \qquad \text{..............-....2.2}$$

where δ is the angle of separation between E_s and E_r .

Plotting this power versus the angle of separation gives us a very useful power-angle curve as shown in Figure 2.2. It is often used in preliminary power system stability studies.

Stability studies are usually classified into three types depending upon the magnitude and nature of disturbances. These three types are transient, dynamic, and steady-state stability studies.

Transient stability studies are aimed at determining if the system will remain in synchronism and survive a major disturbance. Major disturbances can be classified as transmission system faults, line switching, loss of generating units, or sudden load changes.

Dynamic and steady-state stability studies are aimed at the stability of the locus of essentially steady-state operating points of the system. The distinction between these three types comes in the degree of detail used to model the system machines. Dynamic and steady-state stabilities are essentially the same. Steady state problems use a simple generator model which treats the generator as a constant voltage source. But in dynamic stability problems the turbine-governing system and the excitation system along with a synchronous machine model which provides for variation in flux- linkages in machine air-gap is used. The nonlinear differential algebraic equations for the system can be replaced by a set of linear equations which are then solved to determine whether the set of machines will remain in synchronism following small disturbances from the operating point.

Transient stability studies are undertaken more often reflecting their greater importance in practice. They involve, in detail, large disturbances which do not permit the linearization of system non-linear differential and algebraic equations which must be solved by a step-by-step procedure or direct methods. Transient stability problems can be categorized into two categories:

1. First-swing category which is based on a simple machine model without representation of control systems. The time period under study is usually the first few seconds following system faults.
2. Multi-swing category which takes into account a representation of machine control systems and more complex machine models to reflect its proper behaviour. The time period under study extends usually to a longer period than that for the other two types.

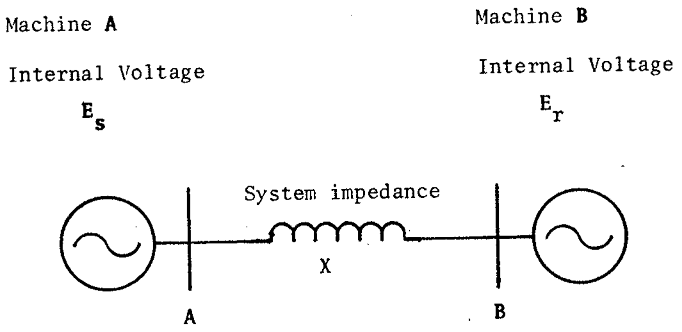


Fig. 2.1 Basic two machine system

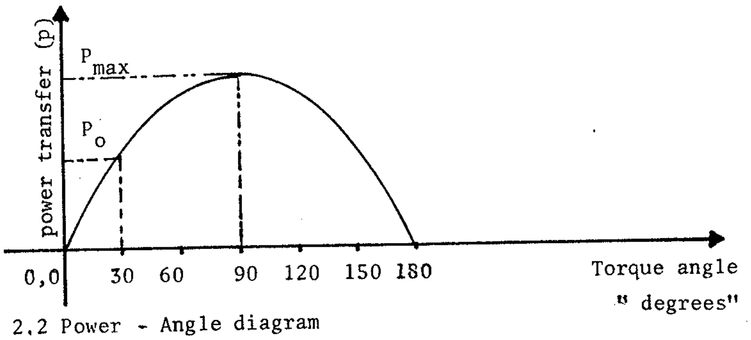


Fig. 2.2 Power - Angle diagram

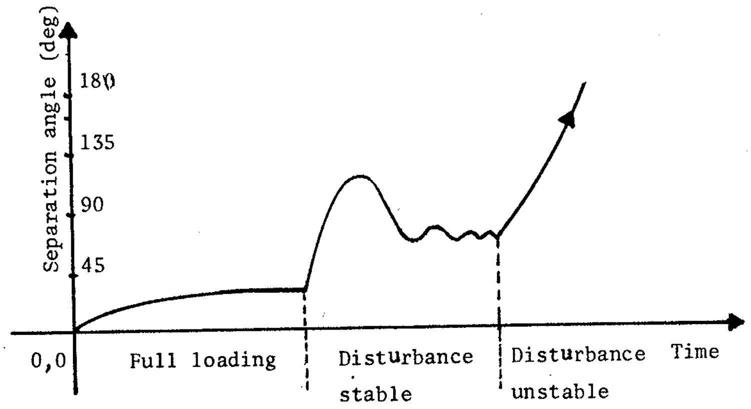


Fig. 2.3 Separation angle - time curve

As was stated before, to show whether a set of machines will remain in synchronizm after a disturbance or not, a solution of the differential equations has to be investigated.

The fundamental equation which governs the rotational dynamics of these synchronous machines in stability studies, called the swing equation, is of second order and can be written in the form [9]

$$\frac{2\,H}{\omega_s} \cdot \frac{d^2\delta}{dt^2} = P_m - P_e = P_a \qquad \text{................2.3}$$

where the symbols have the following meanings:

H = stored kinetic energy in megajoules at sync. speed divided by machine rating in MVA

$$H = \frac{J}{2} \quad \frac{-\omega_s^{\,2}}{S_m} \qquad \text{................2.4}$$

J is the total moment of inertia of the rotor masses in Kg-m

ω_S is the machine synchronous speed in mechanical radians per second

S_m is machine three phase rating MVA

δ is the rotor angular displacement in units consistent with ω_s , e.g. radians.

P_m is the mechanical input power to the machine shaft less rotational losses.

P_e is the electrical power crossing machine air-gap

P_a is the accelerating power, that is, the unbalance between electrical and mechanical power.

The aforementioned second-order differential equation can be written as two first order differential equations

$$\frac{2H}{\omega_s} \frac{d\omega}{dt} = P_m - P_e \qquad \text{..................2.5}$$

$$\frac{d\,\delta}{dt} = \omega - \omega_s \qquad \text{..................2.6}$$

When the swing equation is solved we obtain the expression for δ as a function of time . A graph of this solution is called the swing curve of the machine. Inspection of these swing curves for all machines in the system will yield whether the machines will remain in synchronism after a disturbance or not. A typical swing curve is shown in Figure 2.3

2.1.1 Power System Relaying on R-X Diagram:

Practical system economics and reliability demands that more and more dependance be placed on system relaying. Because of this increased dependence on relay performance, relays using multiple operating quantities were developed. These operating quantities are voltage, current and phase angle. One advantage of these three operating quantities is that relay characteristics as well as power system conditions can be plotted on the same R-X diagram from which system and relay behaviour can be predicted and analyzed.

These relays are known as ohmic relays. In order to understand the operating characteristics of these relays we must look into the torque-producing components, namely:

1. Current component: Torque is proportional to square of current.
2. Voltage component: Torque is proportional to square of voltage.
3. Voltage and current product component: Torque is proportional to current-voltage product times a function of the angle between them.
4. Control spring torque.

The general torque equation for an ohmic relay can be written in the form[2]:

$$\text{Torque(T)} = \pm K_1.E^2 \pm K_2.I^2 \pm K_3.E.If(\gamma,\theta) \pm K_4 \qquad \ldots\ldots\ldots\ldots 2.7$$

The conventions adopted for this equation are

1. Positive torque for contact closing.
2. Constants K_1. K_2, K_3 and K_4 are independent relay constants.
3. Spring torque, K_4, is assumed to be constant.
4. Operating quantities voltage (E), current (I) and θ the phase angle between them, are supplied to the relay.

With proper choice of relay constants and associated signs, different types of relay elements can be developed. Some of these are tabulated in Table 2.1

Table 2.1 Ohmic relay elements and their torque equations

Relay Type	Choice of Constants	Torque Equation
Directional element	$K_1 = K_2 = K_4 = 0$ $f(\gamma,\theta) = \sin(90+\gamma-\theta)$	$T = K_3.E.I.\sin(90+\gamma-\theta)$
Reactance relay	$K_1 = K_4 = 0$, K_2 positive $f(\gamma,\theta) = \sin\theta$ K_3 negative	$T = K_2.I^2 + K_3.I^2.X$
Impedance element	$K_3 = K_4 = 0$; $K_1>0$; $K_2>0$	$T = K_2.I^2 - K_1 . E^2$
Directional element with volt. restraint	$K_2 = K_4 =0$; $K_1>0$; $K_3>0$ $f(\gamma,\theta) = \sin(90+\gamma-\theta)$	$T = K_3.E.I.\sin(90+\gamma-\theta)$ $-K_1.E^2$

The characteristics of these relays are shown on the R-X diagram in Figure 2.4. It should be noted that the directional element is not an ohmic element in the usual sense, although its characteristics may be conveniently shown on an R-X diagram. one advantage of the R-X diagram is that it displays the same characteristics inspite of voltage variations due to different system faults or for the same fault under different system conditions.

The last item in table 2.1, the directional element with voltage restraint, uses the torque equation

$$T = K_3.E.I.\sin(90+\gamma-\theta) - K_1.E^2 \qquad \text{........ 2.8}$$

Where γ is the angle of maximum torque

Positive torque will be developed i.e. relay operation, under the condition

$$K_3.E.I.\sin(90+\gamma-\theta) > K_1.E^2 \qquad \text{.........} \; 2.8a$$

If we set the equality condition, i.e. operate-restraint condition, we get

$$K_3.E.I.\sin(90+\gamma-\theta) = K_1.E^2 \qquad \text{..........} \; 2.8b$$

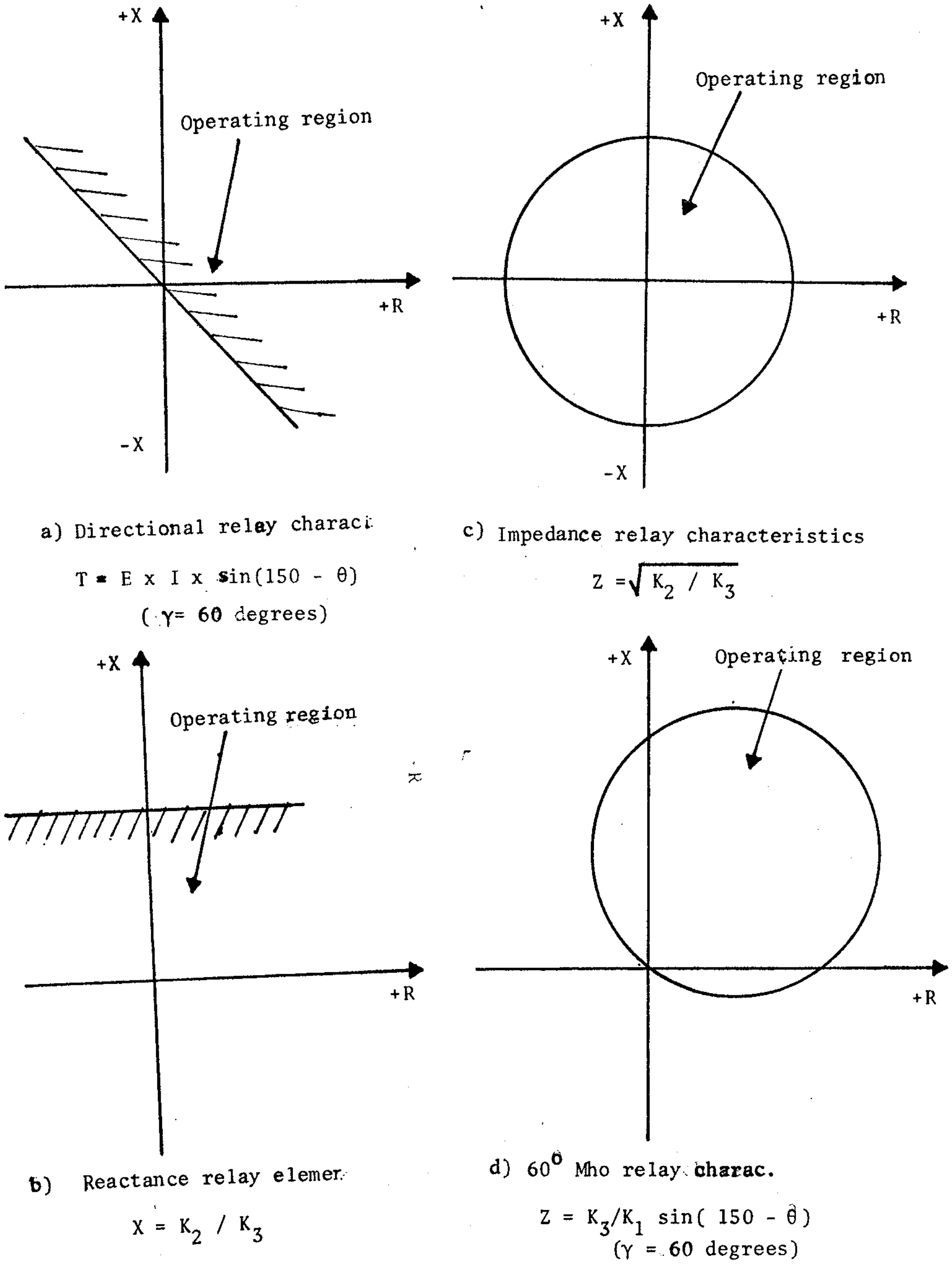
$$I.\sin(90+\gamma-\theta) = \frac{K_1}{K_3}.E \qquad \text{..........} \; 2.8c$$

Using the fact that the impedance between a fault location and the relaying point as seen by the relay is

$$Z = \frac{E}{I}$$

$$Z = \frac{K_3}{K_1}\sin(90+\gamma-\theta) \qquad \text{..........} \; 2.8d$$

Inspecting this last equation we see that a positive contact closing torque will be realized whenever the term on the right hand side is greater than the left hand side. This equation, when plotted on an R-X diagram, is shown in Figure 2.4d. This simple plot does not depend on any parameters of operating quantities but defines the operating characteristics for all values of E, I and phase angle θ.



a) Directional relay charact

$T = E \times I \times \sin(150 - \theta)$

($\gamma = 60$ degrees)

c) Impedance relay characteristics

$Z = \sqrt{K_2 / K_3}$

b) Reactance relay elemen

$X = K_2 / K_3$

d) 60° Mho relay charac.

$Z = K_3/K_1 \sin(150 - \theta)$

($\gamma = 60$ degrees)

Fig 2.4 Ohmic elements characteristics on R-X diagram

2.2 POWER SWING LOCUS ON R-X DIAGRAM:

An analysis of the characteristics of different types of relay elements has been explained on the R-X diagram. A better understanding of the whole power system and relaying system will be accomplished if both systems are plotted on one diagram. From this plot the locus of the ohmic value seen by a distance relay during power swings will give an accurate insight into the performance of the relaying system. To show the swing locus on the R-X diagram the two machine system and transmission line representation shown in Figure 2.5 will be used.

For power flow from point A to point D, the voltage at point A will lead the voltage at point D. The total current will lag E_{AD} by the angle of the total impedance between these two points. For a fixed torque angle δ (the angle between equivalent source voltages),the apparent impedance seen from point A will fall on a circle which is defined by this given torque angle. A particular impedance point on this circle is entirly dependent on the ratio of voltage magnitudes at points A and D. Figure 2.5b shows the swing locus for the system and can be determined geometrically as shown in reference 3 Appendix III. The centre of this circle is the vector AO with magnitude

$$\overline{AO} = \frac{\overline{AD}}{2\sin\delta} \qquad \text{.............2.9}$$

and

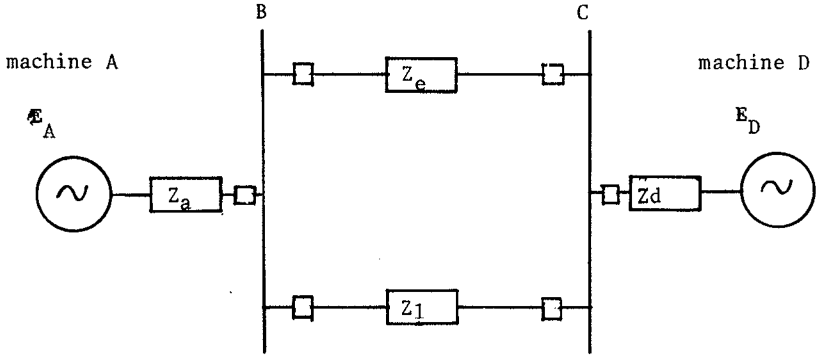


Fig 2.5a example system

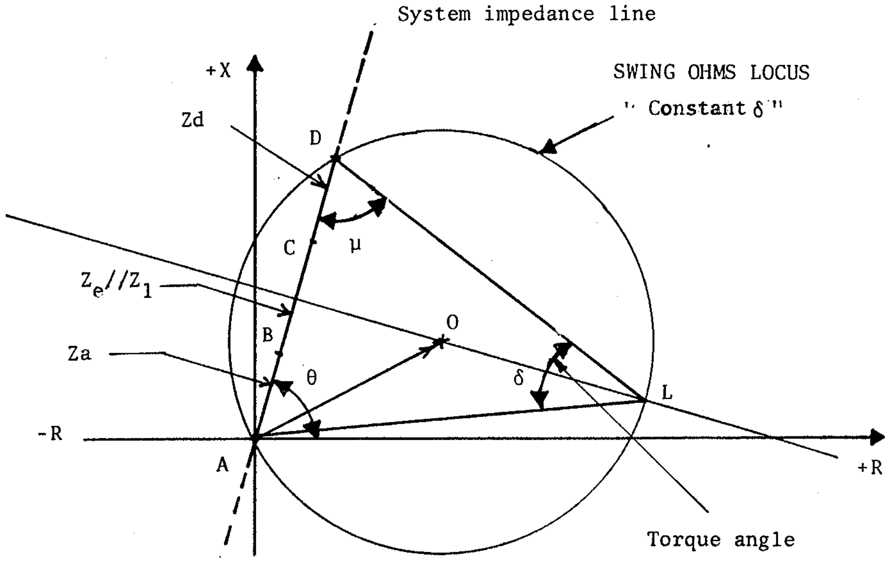


Fig 2.5b

Fig 2.5 Swing ohms locus on R-X diagram

$$\mathrm{Arg}(\overline{AO}) = \underline{\left|\theta - 90 + \delta\right.} \qquad \ldots\ldots\ldots\ldots\ldots 2.10$$

where

θ = is the phase angle between E & I

δ = is the torque angle

The vector $\overline{AL}$, which represents the load ohms as seen from point A, has the magnitude and angle as a function of the torque angle, to be:

$$\overline{AL} = \frac{\overline{AD}.\sin\mu}{\sin\delta} \qquad \ldots\ldots\ldots\ldots 2.11$$

and

$$\mathrm{Arg}(\overline{AL}) = \underline{\left|\theta+\delta+\mu - 180\right.} \qquad \ldots\ldots\ldots\ldots 2.12$$

where μ is given by

$$\frac{\cos\mu}{\sin\mu} = \frac{E_D/E_A - \cos\delta}{\sin\delta} \qquad \ldots\ldots\ldots\ldots 2.13$$

Equation 2.13 shows us how point L on the circle diagram will behave for different source voltage ratios. Assuming that torque angle remains constant, the left hand term in equation 2.13 will increase for increasing E_D/E_A ratio and, therefore, the angle will be decreasing which results in point L moving clockwise along the Swing Circle.

To show how the apparent impedance moves on the R-X diagram due to line switching, the system shown in figure 2.5a will be used. Assume this system was initially running at synchronous speed with $E_A/E_D = 1.1$, $P = P_o$ and $\delta = 30$ degrees. From the $P - \delta$ curve of Figure 2.6b the maximum

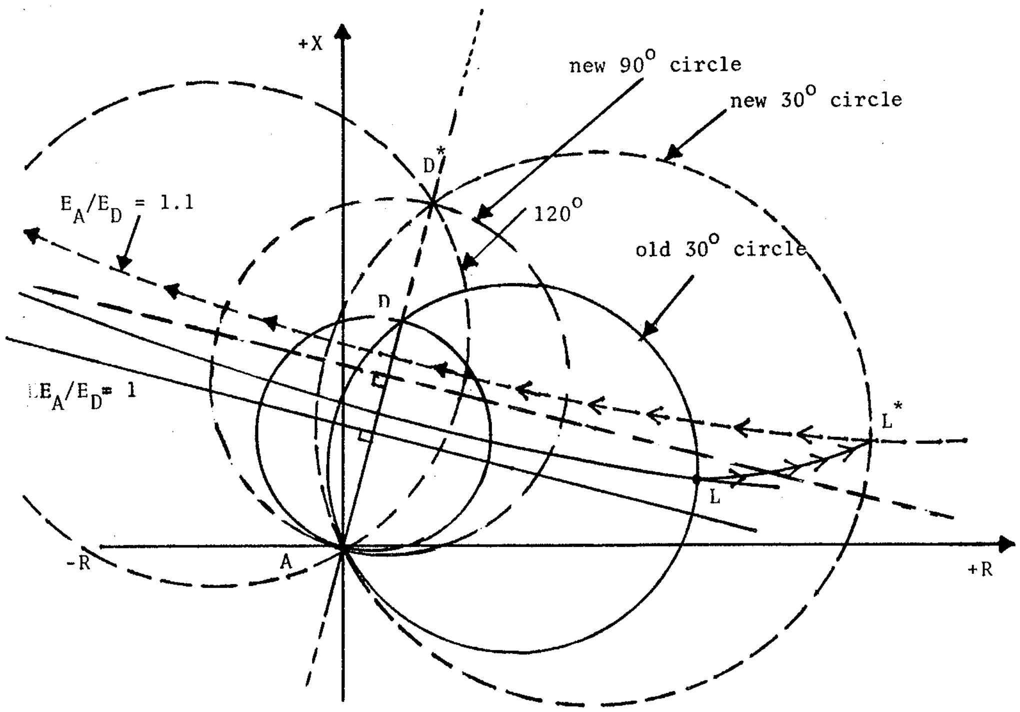


Fig. 2.6a Apparent impedance locus on R-X diagram

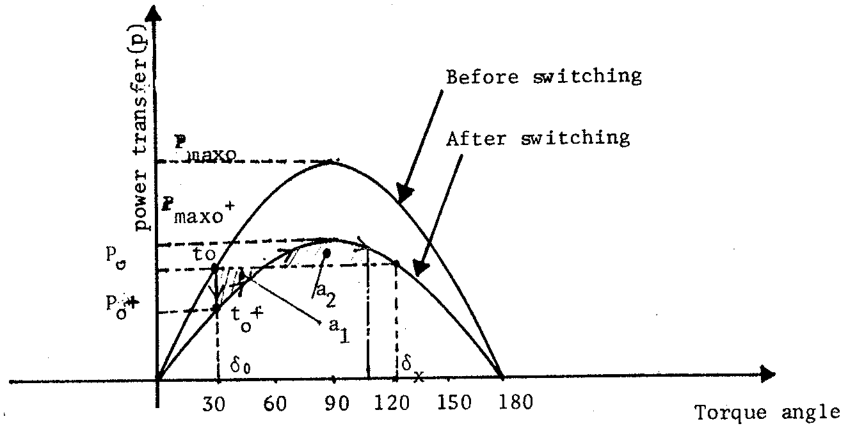


Fig. 2.6b Power transfer – torque angle curve

steady-state power transfer before switching line z_L ($t<0$) is equal to P_{maxo}.

At the first instant of switching, the impedance between points A and D has increased to AD*. This means that the power transfer capability is reduced and is shown as P_o+. The difference in power, $P_o+ - P_o$, is the accelerating power which accelerates the rotor and the rotor angle advances to say δ_x . For the machine to stay in synchronism, area's a_1 and a_2 have to be equal according to the equal are criterion:

$$\int_{\delta_0}^{\delta_x} (P_m - P_e) \, d\delta = 0$$

where $P_m - P_e$ is the accelerating power.

If the equal-area criterion cannot be satisfied, this means that δ_x goes beyond about 120 (typically) and the machine goes unstable. As is seen on the R-X diagram, the original operating condition is the point L corresponding to t=o or (Power = P_0). After line switching the new operating point is L* which corresponds to the same voltage ratio (1.1) and torque angle (30 degrees), but the electrical power output is P_0+ . As the torque angle advances towards δ_x, point L* moves along the circle E_A/E_D = 1.1 as shown by the arrows on Figure 2.6a. If the system goes unstable, this point crosses the approximate transient stability limit circle, then it continues advancing till it crosses the system impedance line at δ_x = 180 degrees where machine A will be 180 degrees out of phase with machine D.

In a general case, if the angular separation of the two machines is maintained constant while varying the voltage ratio, the apparent impedance will follow a circle which passes through both points A and D with their centers on the perpendicular bisector of the system impedance line with radii and offsets determined by the various values of this voltage ratio. This family of circles is shown in dashed lines in Figure 2.7. Another set of circles can be constructed by holding the voltage ratio constant while varying the angular separation. This new family of circles is all centred on the system impedance line with radii and offsets determined by the voltage ratio and shown in solid line in Figure 2.7. It should be noted that these two families of curves are orthogonal. For full details and mathematical derivation of these curves, the reader is advised to see reference 4. Note too an interesting circle which is centred at the system impedance centre and passes through both points A and D. For any value of voltage ratio, the apparent impedance of the system must pass through this circle as the separation angle reaches and passes 90 degrees which is the approximate steady-state stability limit.

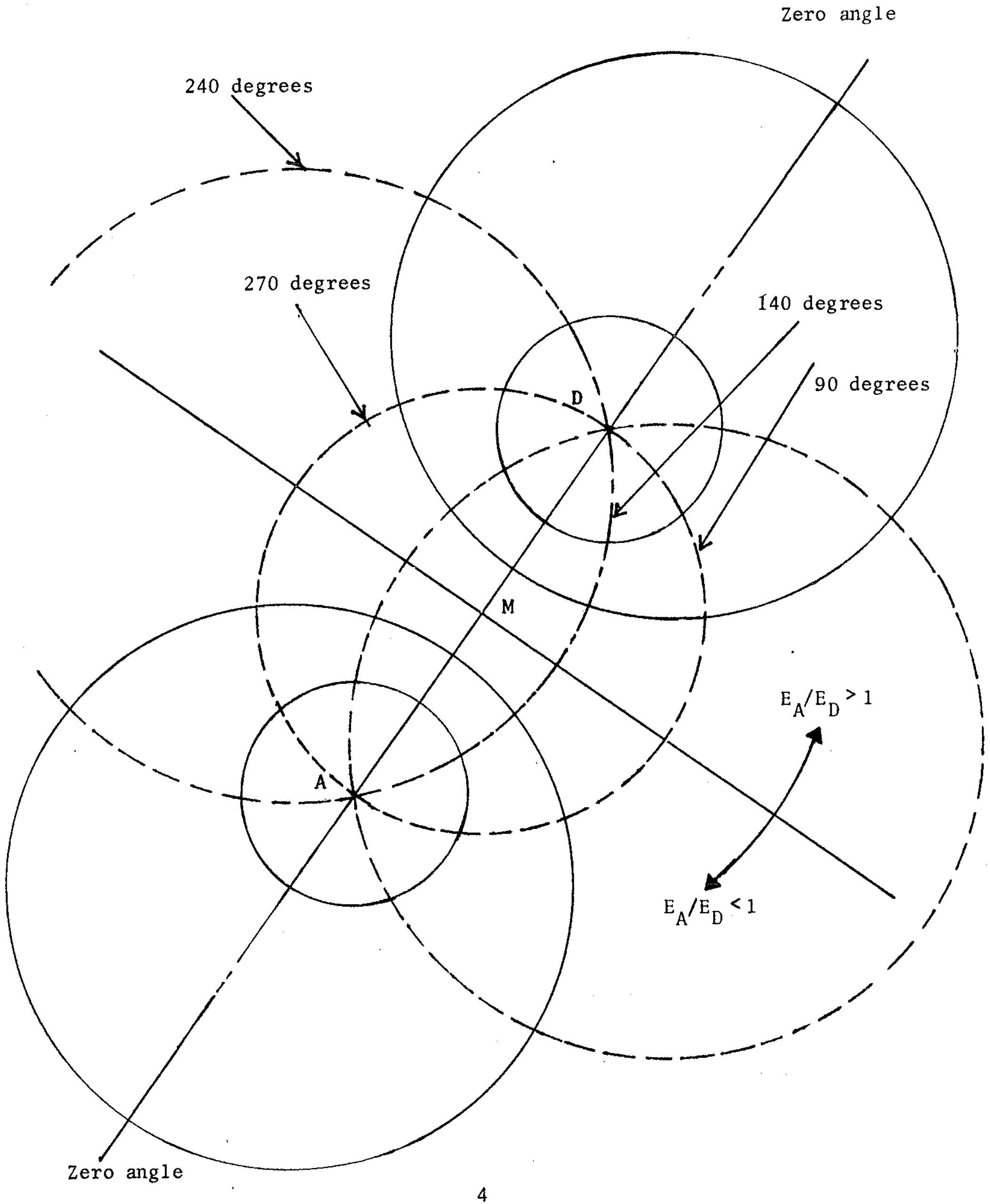


Fig. 2.7 General impedance diagram[4]

2.3 COMBINED SYSTEM AND RELAY CHARACTERISTICS ON R-X:

In order to use the R-X diagram effectively and for better understanding of total system behavior, both relay characteristics and how the system apparent impedance will vary as system conditions vary, i.e. as system machines go out of step, must be shown on the same diagram. In order to accomplish this, Figure 2.8 will be used. On this figure, only the approximate steady-state stability limit circle, i.e. 90 degree separation circle and the approximate transient stability limit, i.e. 120 degree separation circle, and the swing line for the special case where voltage ratio $E_A/E_D=1$ are shown. On the same figure, a portion of the system impedance line between stations 1 and 2 is to be protected. First zone instantaneous trip protection for 80 per cent of the line is provided by a mho element.

We have seen that the system apparent impedance follows a definite curve during an out-of-step condition or power swing. This particular curve is dependent on the voltage ratio and intersects the system impedance line at 180 degrees. Let us consider at one instant the system was carrying an interchange of loading equivallent to point L shown on Figure 2.8. A fault on the line portion 1-2 will result in a change of impedance from point L to point F on the line in practically zero time. On the other hand, during the first few swing cycles for an out- of-step condition, the apparent impedance drifts along the curve $E_A/E_D = 1$

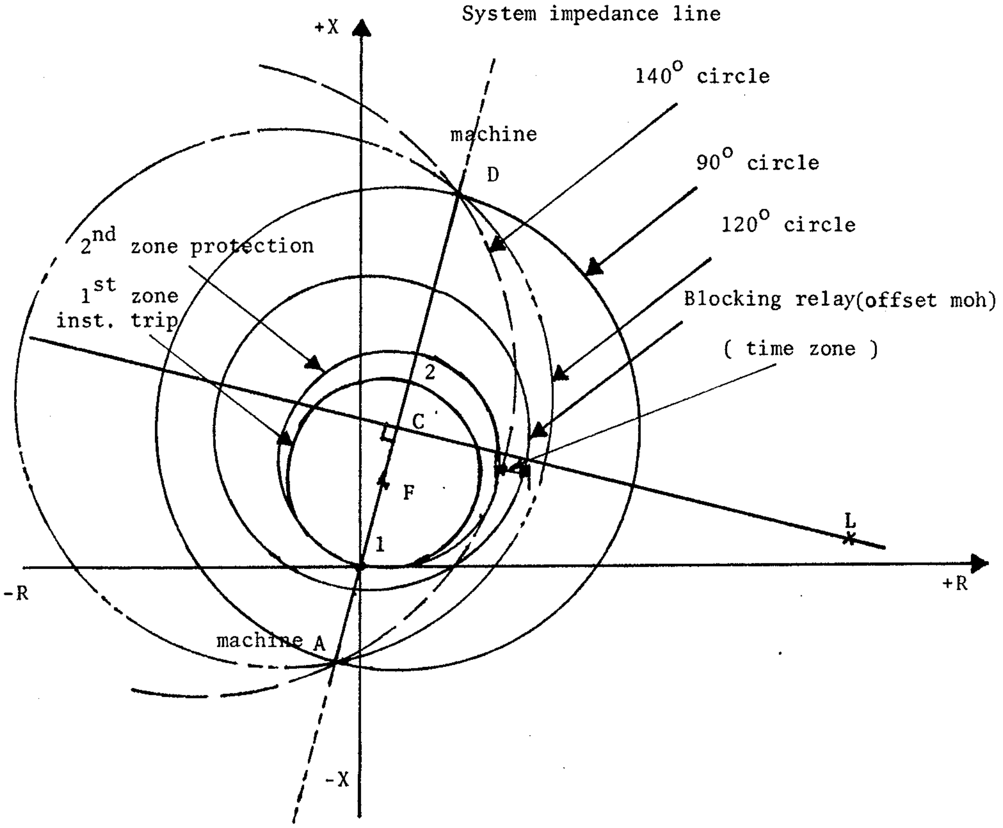


Fig. .2.8 Combined system and relay characteristics on R-X diagram

relatively slowly as compared to the fault case. If the offset mho element was set up to block the first and second zones of the mho relay, these elements would not trip as long as the out-of- step condition progressed from the initial separation condition to a 90 degree to 120 degree separation to any point inside the blocking element in a time which exceeds a predetermined minimum time. This shows that the blocking element has to pick up before the first and second zone trip element with a sufficient time for the auxiliary relays to operate. But if this time is not enough or if both the blocking and tripping elements pick up at the same time, blocking will not be realized. This time delay has to be sufficiently large to allow for the fastest swings anticipated and yet be practical to avoid unnecessary tripping.

Since practical system operation and system conditions do occasionally force the system to go out-of-step, tripping relays must detect the out-of-step condition and take the necessary measures. The basic distinction of an out-of-step condition can be realized by the fact that the apparent impedance, as seen from any one point in the system, changes from a point to the right of the system impedance line to a point on this line to a point to the left of the system impedance line. This sequence applies for power transfer from point A to point D as the machine at point A advances ahead of machine D. Another characteristic of an out-of-step condition is that the aforementioned apparent impedance change takes place over a finite period of time which is long as compared to impedance change associated with a fault condition. An ohmic relay which will recognize these distinctive characteristics can be a relay with two reactance elements. Each element has an angle of maximum torque perpendicular to the system impedance line, and an operating characteristic parallel to the system impedance line with an adjustable offset for pick-up setting selection. The first requirement for these settings is that the relay should operate for the fastest slip cycle expected. To know this maximum slip would involve various system parameters and should be based on a transient stability study of the system under consideration. A second requirement for these settings is that the relay characteristics should cover the system impedance plot on the R-X diagram. This means that any fault on the system should fall between the relay characteristics.

Chapter III

TRANSIENT IMPEDANCE DISPLAY UNIT DESIGN

3.1 DISPLAY UNIT HARDWARE DESIGN:

The transient impedance display unit is a microprocessor based device which is capable of calculating the apparent impedance seen at a relaying point using readily available input signals. These signals are a voltage signal proportional to the line voltage and a second voltage proportional to the current in that phase. The apparent impedance is outputed in the form of a voltage signal proportional to line resistance and another voltage signal proportional to the line reactance. Applying the resistance equivalent voltage to the horizontal channel of an oscilloscope and the reactance equivalent voltage signal to the vertical channel, the apparent impedance seen at the relaying point can be displayed on the oscilloscope screen

The central processing unit is the Motorola Inc. type MC6802 microprocessor. Most of the other supporting chips such as the peripheral interfacing adapters ,decoder, digital to analog and analog to digital converters are either manufactured by Motorola or National Semiconductors Inc. The main reason behind this choice is the availability of the supporting equipment and facilities such as a main computer system crossassembler and a diagnostic emulator used for system software development and system debugging.

Fig. 3.1 shows the device hardware function flow chart. It shows two almost identical circuitry used for both voltage and current input signal channels with differences in the design of signal conditioning circuits. The voltage signal is derived from the BICEPS machine line voltage through a potential transformer that provides a 6.3 V secondary voltage. The current signal is derived from the same phase through a 33 ohms resistor connected across the secondary terminals of a current transformer wired to the same line at the same point.

3.1.1 Input/Output Signal Conditioning Circuits:

The analog to digital converter chosen in this design is the ADC0804 manufactured by National Semiconductor Inc. The electrical specifications state that for a +5V DC power supply to this chip, the input range will be 0V to +5V. This comes about as a result of two on-chip diodes which are tied to each analog input terminal, $V_{in}(+)$ and $V_{in}(-)$. For a zero differential input signal at $V_{in}(^-)$ and $V_{in}(_+)$ the output code will be

$$[\,0\,0\,0\,0\quad 0\,0\,0\,0\,]$$

and for diferential input $=V_{cc}$ digital output code will be

$$[\,1\,1\,1\,1\quad 1\,1\,1\,1\,]$$

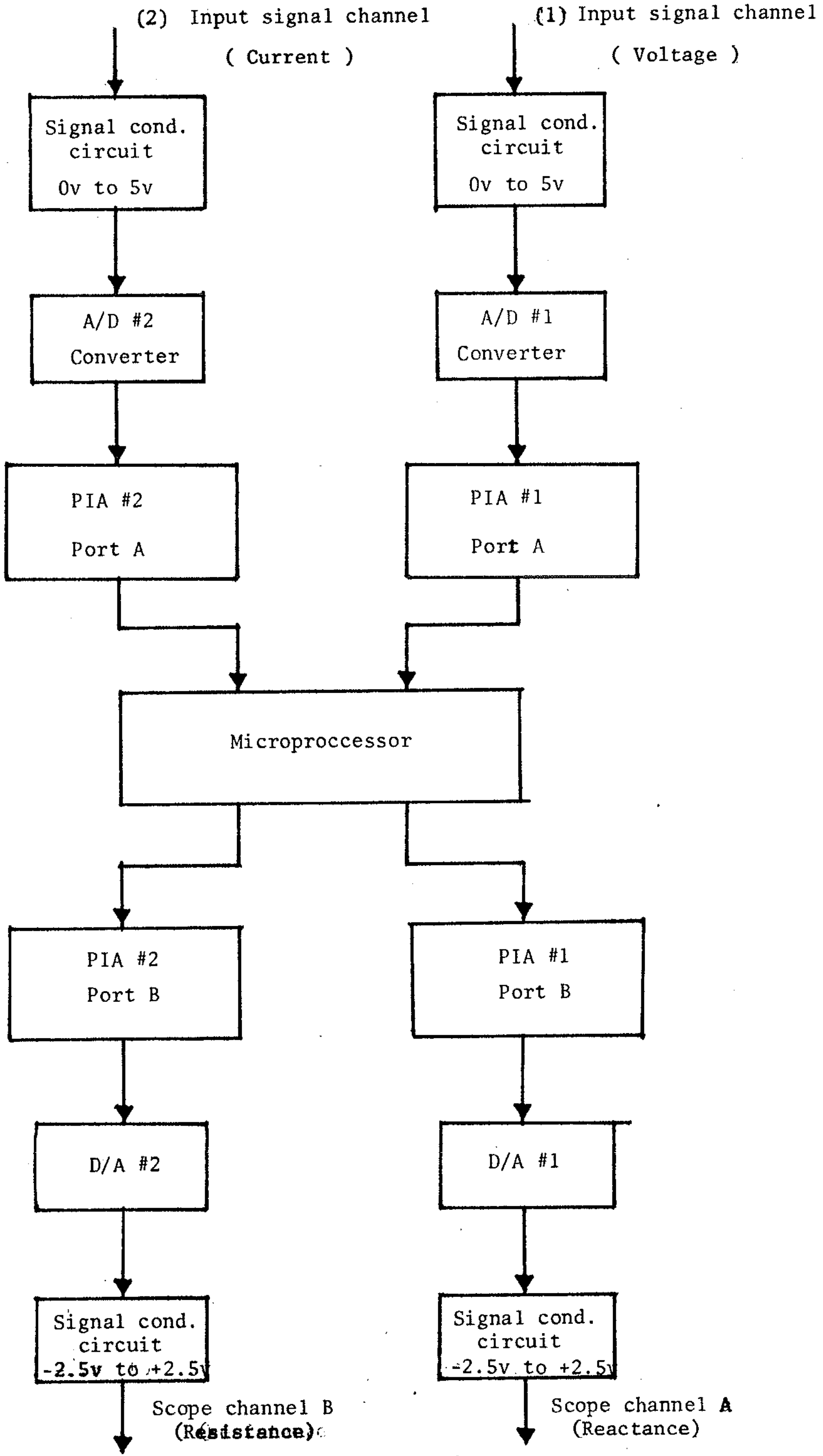

(2) Input signal channel
(Current)
(1) Input signal channel
(Voltage)
Signal cond. circuit 0v to 5v
Signal cond. circuit 0v to 5v
A/D #2 Converter
A/D #1 Converter
PIA #2 Port A
PIA #1 Port A
Microproccessor
PIA #2 Port B
PIA #1 Port B
D/A #2
D/A #1
Signal cond. circuit -2.5v to +2.5v
Signal cond. circuit -2.5v to +2.5v
Scope channel B (Resistance)
Scope channel A (Reactance)

Fig. 3.1 System hardware function flow chart

To achieve this 0V to +5V analog input range the input signal conditioning circuit shown in figure 3.2a is designed. As was mentioned earlier, the available signal from the BICEPS line potential transformer is 6.3 V_{rms} or -8.9V to +8.9V peak-to-peak. For the output of the operational amplifier (op-amp) to be 0V to +5V, the feedback resistance to the input resistance ratio will be

$$\frac{R_2}{R_1} = - \frac{V_o}{V_{in}} = \frac{5}{17.82} \qquad \text{..........3.1}$$

or

$$R_2 = 0.28\ R_1 \qquad \text{.........3.2}$$

and to bias the signal

$$R_3 = 2.0\ R_2 \qquad \text{...........3.3}$$

Selecting the input resistance R_1 to be equal to

$$R_1 = 5.6\ K\ \Omega$$

will yield

$$R_2 = 1.57\ K\Omega$$

$$R_3 = 3.14\ K\Omega$$

Similar arguments apply to the current signal derived from the current transformer

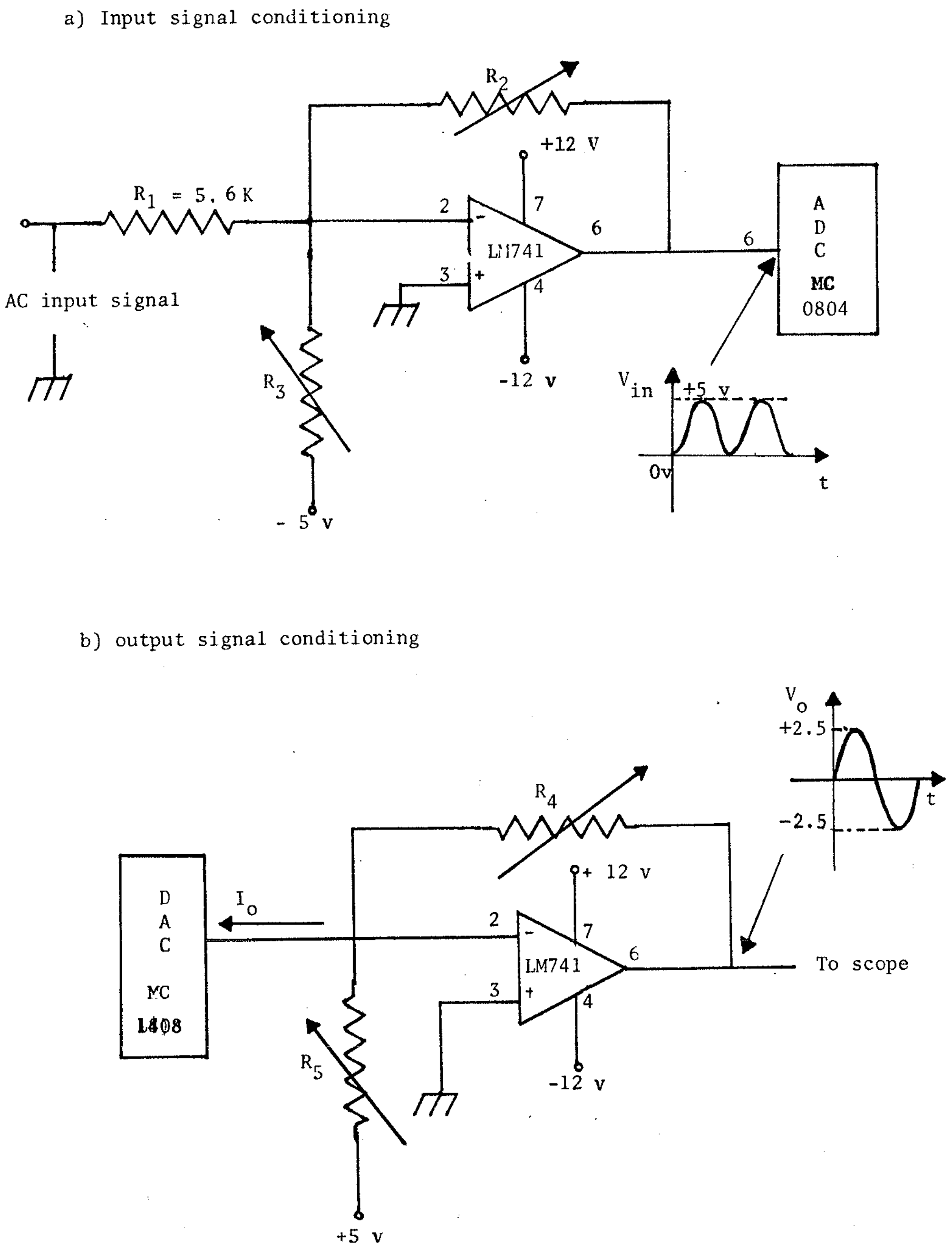
$$\frac{R_2}{R_1} = - \frac{V_o}{V_{in}} = \frac{5}{8}$$

$$R_2 = 0.625\ R_1 \qquad \text{.........3.4}$$

and setting

$$R_3 = 2\ R_2$$

Selecting the input resistance to be


a) Input signal conditioning

b) output signal conditioning

Fig. 3.2 Input/Output signal conditioning circuit

$$R_1 = 5.6 \text{ k}\Omega$$

yields

$$R_1 = 3.5 \text{ k}\Omega$$

$$R_3 = 7.0 \text{ k}\Omega$$

3.1.2 System Decoding and Memory Map

The Decoder/Demultiplexer chosen for system decoding is Texas Instrument type SN74S138. This decoder chip is shown in Figure 3.3 and provides partial address decoding using only three of the high order address lines A13, A14, and A15. The decoder accepts these three binary inputs, which are wired to chip select terminals A, B, and C. When enabled by proper signals on the input enable terminals, it provides one of eight mutually exclusive active low outputs (Y0-Y7) dependent on the conditons at the three select inputs as shown in Table 3.1. The chip enable terminals are connected as follows: Terminal G1 is always high (+5 V), G2B terminal is connected to ground and terminal G2A is wired to the processor VMA and E terminals through a NAND gate type DM74LSOON. As a result of this partial decoding scheme, table 3.2 and memory map Table 3.3 are constructed to show all on board chips and the address to which each one will respond.

Table 3.1 Partial decoding function table

Inputs					Outputs							
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

G2* = G2A + G2B

H = high level

L = low level

Table 3.2 On board chips addresses

Chip Name	Chip Address
PIA #1	$ D000-D003
PIA #2	$ E000-E003
EPROM #1	$ F000-F7FF
EPROM #2	$ F800-FFFF
RAM	$ C000-C7FF

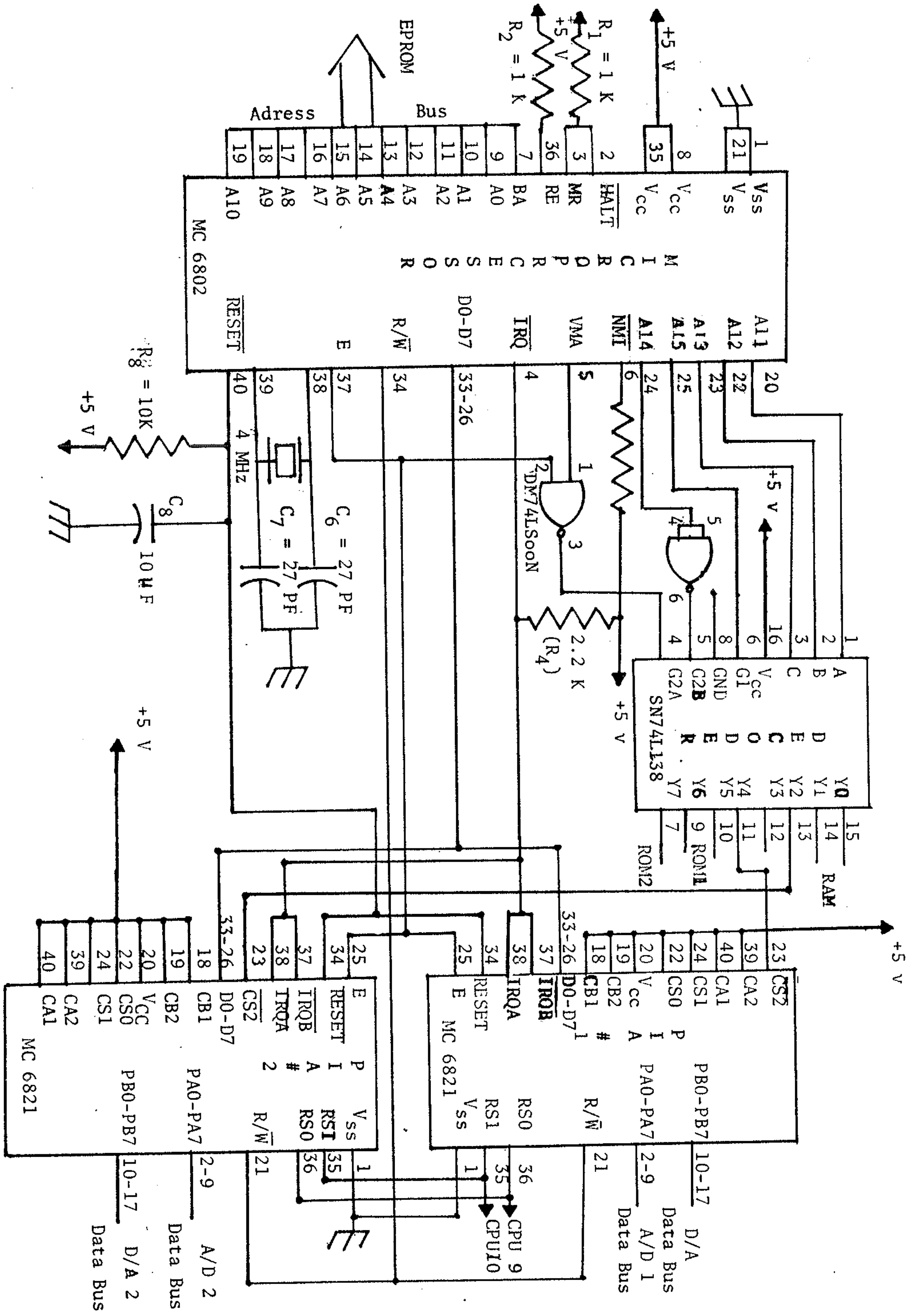


Fig. 3.3 System wiring diagram

Table 3.3 Memory map for transient impedance display unit

	Memory Location
Reset Vector	$FFFF,$FFFE
EPROM #1	$F000 - $F7FF
EPROM 2	$F800 - $FFFF
PIA #2	$E000 - $E003
PIA #1	$D000 - $D003
RAM	$C000 - $C7FF
Not used	All others

3.1.3 MC6802 Microprocessor and Peripheral Connection:

The MC6802 is an eight bit microprocessor. A detailed block diagram, programming model of the processing unit and pin assignment diagrams are given by the (manufacturer Motorola Inc.) in the appropriate data manual. Some of the features in this microprocessor chip are:

1. an on-chip internal clock oscillator and driver,
2. a 128x8 on-chip RAM located at hex addresses $0000 to $007F with the top 32 bytes having memory retention capability by utilizing the standby power supply,
3. sixteen lines unidirectional address bus, and
4. an eight line bidirectional data bus.

The programming model for this processor shows that the MPU has three 8-bit registers, two accumulators, one condition code register, and three 16-bit registers called 1) index register, 2) programme counter, and 3) stack pointer. All are vailable for use by the programmer. The two 8-bit accumulators are used to hold operands and the results of the arithmetic logic unit, ALU. The condition code register contains eight bits of information and only the two highest order bits are not used and set to ones, the other six bits are used as testable condition codes for the conditional branch instructions. Bit number four is the interrupt mask in this condition code register.

The 16-bit index register stores sixteen bits of memory address for the index mode of memory addressing. The programme counter is a 2-byte (eight bit each) register that points to the current program instruction. The stack pointer is a 2-byte (eight bit each) register that contains the address of the next available location in the stack for use.

Figure 3.3 shows the pin connection and wiring diagram of this microprocessor interfaced to two peripheral devices and a decoder. This figure shows that the $\overline{\text{HALT}}$ line is not used and, therefore, set in the high state for the interrupts to be serviced. Memory ready is also set high indicating that the clock seen at the ENABLE terminal E operates in the normal condition or unstretched. A 4MHz crystal is connected to pins 38 and 39 and because of the divide-by-four circuitry inside the chip, gives an operating clock frequency of 1MHz. The $\overline{\text{IRQ}}$ input is wired to terminals 37 and 38 on both PIAs. A low level on this input requests that an interrupt sequence be generated in the chip. The processor will first complete the current instruction before it responds to the interrupt request. At this time,if the interrupt mask bit is not set, the processor will start the interrupt sequence.

The $\overline{\text{RESET}}$ terminal is wired to the $\overline{\text{RESET}}$ terminal on each PIA. When this line is low, the MPU is inactive and the information in all the registers will be lost. When this terminal goes high, the MPU begins the restart sequence by executing a routine to initiate the processor from its reset state. For the restart, the last two memory locations (hex addresses $FFFE and $FFFF) will be used to load the program that is addressed by the program counter. During the restart routine, the interrupt mask bit is set and must be reset for the MPU to be interrupted by IRQ line.

Two peripheral interfacing adapters manufactured by Motorola Inc. type MC6821 are shown in fig.3.3. These adapters will appear to the microprocessor as memory locations. The processor will read a digital word from the input register and write a digital word into the output register under program control. The input digital word is equivalent to the analog input into the A/D converter and the output digital word is equivalent to the analog output from the D/A converter. For a complete discussion of this PIA diagram, initialization and control lines discription, the reader is advised to see reference 8 pages 212-231.

3.1.4 A/D and D/A Characteristics and Wiring:

Figure 3.4 shows the wiring diagram for the ADC0804 chosen for this application. This A/D can appear like a memory location or an I/0 port to the microprocessor and no interfacing logic is needed other than the chip select $\overline{CS}$ terminal. Block diagram and electrical specifications with the maximum ratings are given by the manufacturer (see reference 6).

This converter works on the successive approximation principle. It accepts an analog input signal, V_{in} at the input terminal and provides a corresponding digital output signal according to the equation

$$M \equiv \left[\frac{V_{in}}{V_{ref}} \right]$$

where V_{ref} is the reference voltage applied to $V_{ref}/2$ terminal. The brackets indicate that M is the closest approximation to within the resolution of M. In binary approximation form

$$M \approx \frac{V_{in}}{V_{ref}} \approx \left[a_1 \, 2^{-1} + a_2 \, 2^{-2} + \ldots\ldots\ldots + a_n \, 2^{-n} \right]$$

This equation shows that the output signal M is a binary aproximation of the voltage ratio

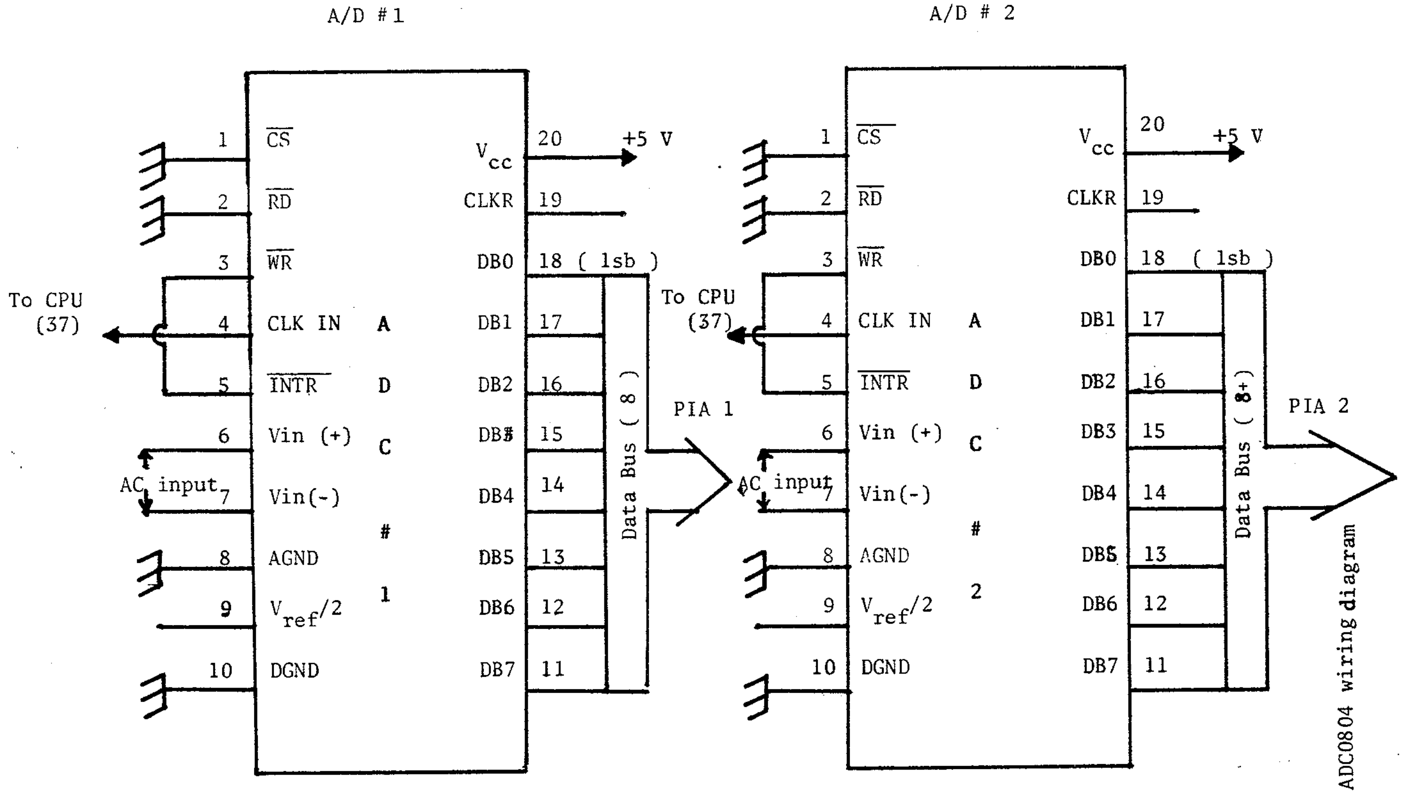


Fig. 3.4 ADC0804 wiring diagram

The A/D data bus (DBO-DB7) is wired to the PIA, port A, data bus. Because the PIA is already memory mapped by the processor and the data lines are connected to the processor data bus under program control, both terminals $\overline{CS}$ and $\overline{RD}$ are connected to ground.

The A/D Converter is clocked using the clock in the MC6802 microporcessor by connecting the CLK IN terminal directly to the E terminal on the processor. By virtue of this connection the CLKR terminal is left floating since the internal circuitry will generate the required clock by using the built-in clock oscillator. Terminal $\overline{WR}$ is connected to INTR terminal. This connection with $\overline{CS}$ connected to ground makes the A/D run in the free-running mode. To insure start- up under all possible conditions an external $\overline{WR}$ write pulse is required during the first power-up cycle. This starting process is accomplished by having $\overline{CS}$ and $\overline{WR}$ terminals simmultanously low which gives the internal clock signal combined with the internal starting flip-flop circuit,START F/F, the responsibility of the start-up process.

The digital-to-analog converter used is the Motorola DAC1408. Figure 3.5 shows the wiring diagram for this converter. The input is an 8-bit digital word via the (A1-A8) input lines connected to the PIA port B data bus. This digital word is the result of the CPU mathematical calculations representing the resistance and reactance seen by the impedance display device. Specifications and

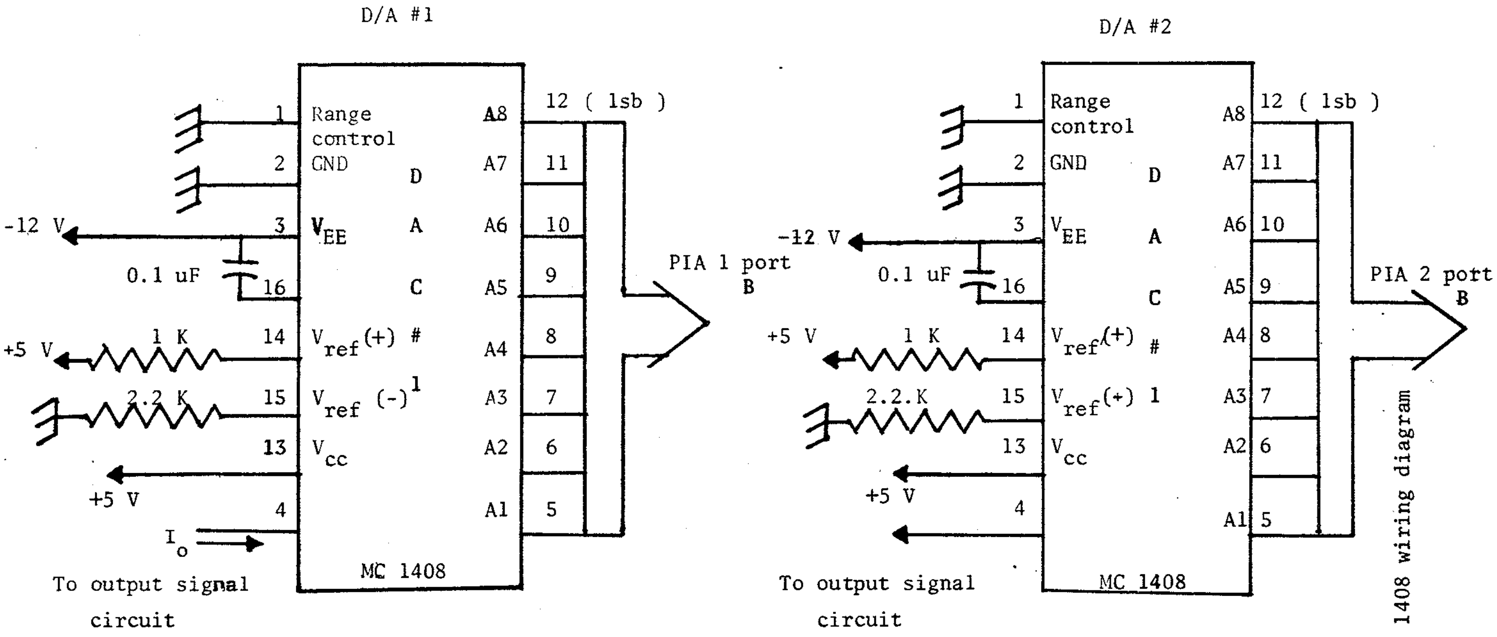


Fig. 3.5 DAC 1408 wiring diagram

characteristics issued by the manufacturer, reference 6, state that the reference amplifier input current must always flow into V_{ref}(+) regardless of the reference voltage polarity. This terminal,14, is therefore tied to the reference voltage +5V DC. The reference terminal, $V_{ref}(-)$, is tied to ground through 2.2kΩ resistor.

Output voltage is obtained by connecting the output terminal pin 4, which provides a current signal going into the converter, to the output signal conditioning circuit shown in Figure 3.2b. This circuit uses an operational amplifier with a feedback resistor R_4 . This configuration keeps the output of the MC1408 at ground potential and the operational amplifier can generate a positive voltage limited only by its power supply. The magnitude of the output voltage is dependent on the digital input and given by the equation:

$$V_o = \frac{V_{ref}}{R_{14}} \cdot R_4 \cdot \left[\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots\ldots\ldots + \frac{1}{256} \right] \text{volts}$$

where:

[A1 A2 ... A8]	is the digital input
V_{ref}	is the reference voltage at Pin 14
R_{14}	is the resistance connectedd to Pin 14
R_4	is the feedback resistor accross the op-amp.

A detailed analysis of the output signal and the converter accuracy is given in Chapter 4.

3.2 DISPLAY UNIT SOFTWARE DESIGN:

The analog input to the A/D converter is a voltage signal 0V to +5V. This A/D converter puts out a digital word $00 to $FF. The zero-crossing of the input signal to the signal conditioning circuit is the same as the +2.5V value of the input signal to the A/D converter which is equivalent to digital word $80 at the output of A/D converter. A software program was developed to calculate the resistance, reactance and their locations on the R-X diagram. This programme is stored in a two (2K-byte each) EPROM memory chips wired and decoded as shown in fig. 3.6

3.2.1 Peak-Picking and Phase Shift Calculation Subroutine:

Figure 3.7 is the flow chart for this subroutine. It starts with initializing the peripheral interphace adapters at the starting address $F000. Port A in each PIA is initialized as an input register and port B as an output register. The input register contains the instantaneous digital word equivalent to the analog input signal. Whenever the processor detects a change from $V_{in} <$ \$80 to $V_{in} \geq$ \$80 it will be interpretted as zero-crossing and continue to check for current signal zero-crossing by detecting the change in $I_{in} <$ \$80 to $I_{in} \geq$ \$ 80 During the time of looking for a current-zero, a counter is incremented by 1 for each time the processor branches back to check for current-zero. The number in this counter, N, is a Hex number equal to the phase

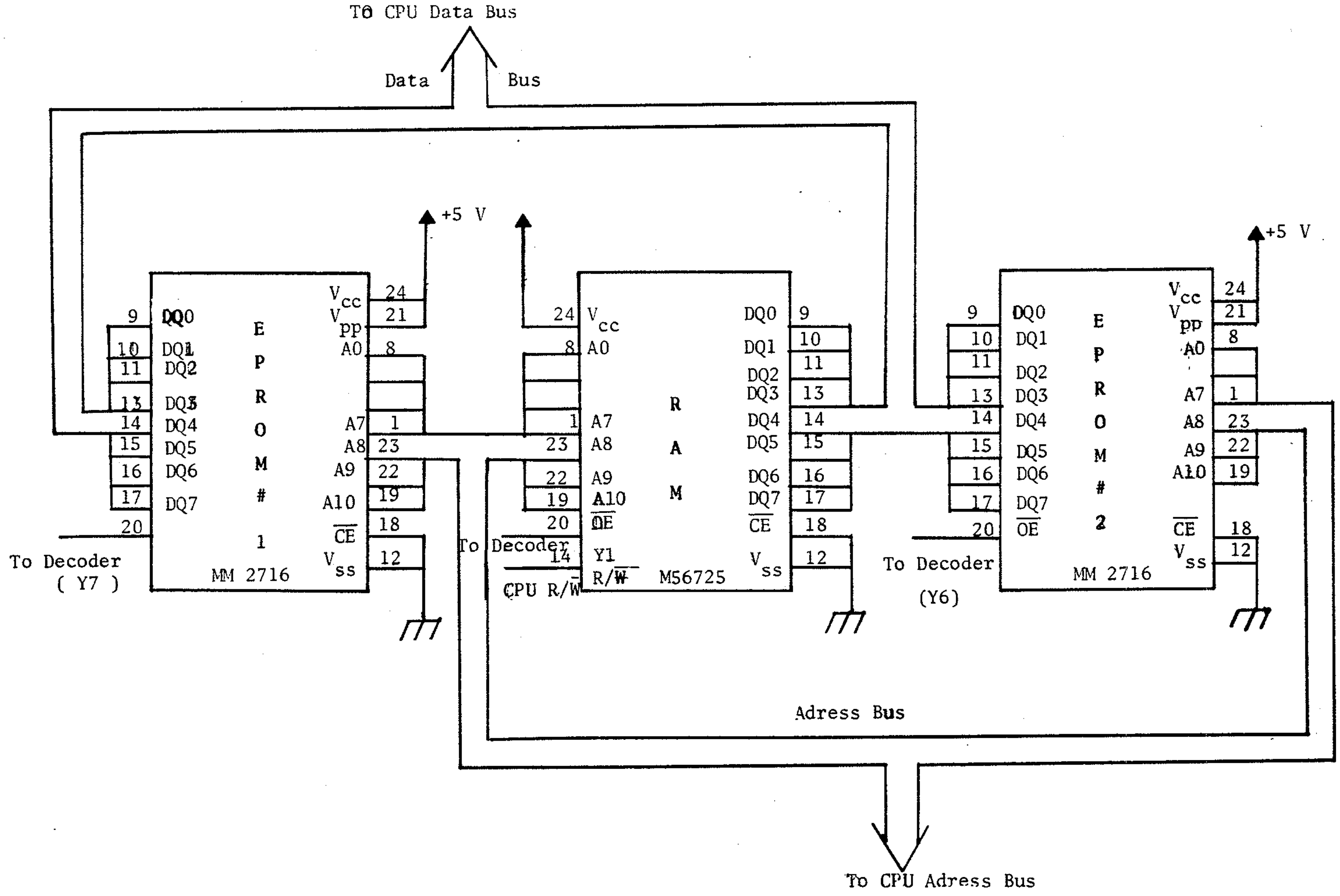


Fig. 3.6 EPROM and RAM wiring diagram

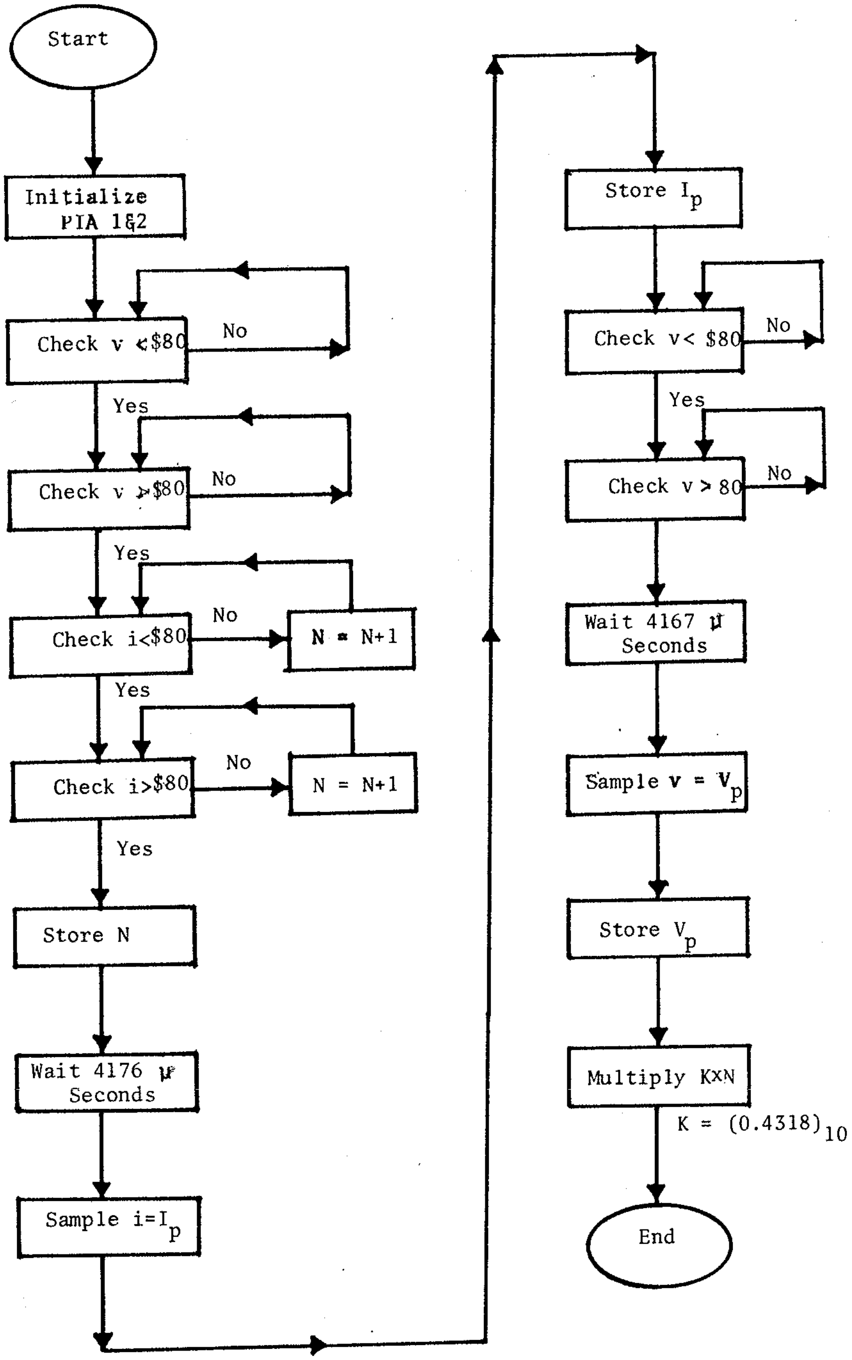


Fig. 3.7 Peak-picking and phase shift calculation routine

shift in number of laps. After the current-zero is detected the phase shift is stored and then the processor waits for 4167 μ seconds (equivalent to 90 degrees) by multiple looping in a specified loop then samples and stores the PIA #2 input register to give the current peak.

The processor then follows similar steps to find the voltage signal peak. The phase shift measurment is then multiplied by a constant to give the equivalent value in degrees. A list of important addresses used by this subroutine follows:

PIA #1 input register	$D000
PIA #1 output register	$D002
PIA #2 input register	$E000
PIA #2 output register	$E002
phase shift in degrees	$CA02, $CA03
voltage peak value	$CA10
current peak value	$CA09

After the processor calculates the three variables (current, voltage,and phase shift) it executes the sine-cosine calculation subroutine.

3.2.2 Sine-cosine the Phase Shift Angle Subroutine

The algorithms of these subroutines are shown in Figures 3.8 and 3.9. The main idea behind the sine-cosine routine is that the processor keeps subtracting 90 degrees from the phase shift angle and continously checking for carry bit set. When the carry bit set is detected it branches to

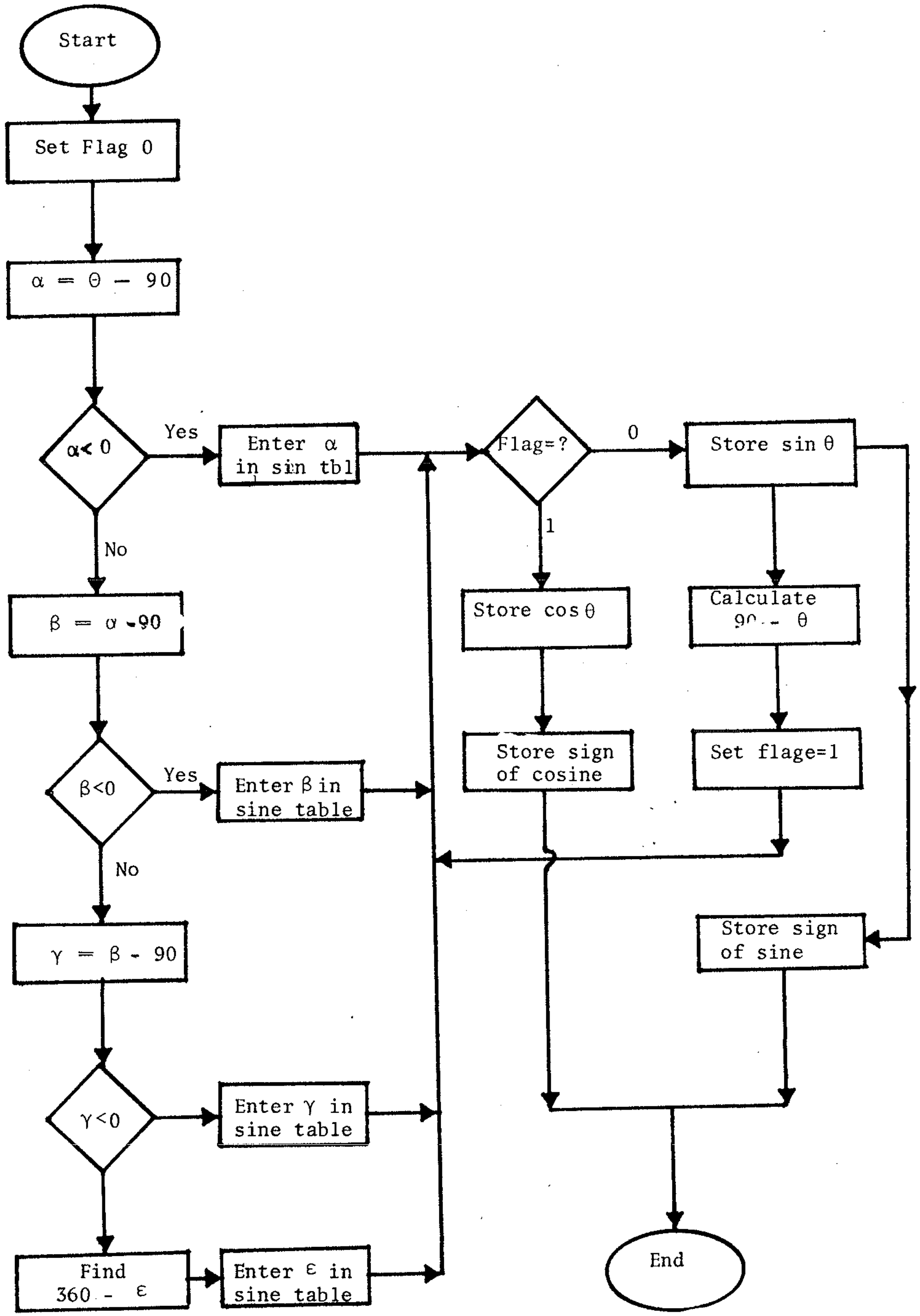


Fig. 3.8 Sin/Cos phase shift angle calculation

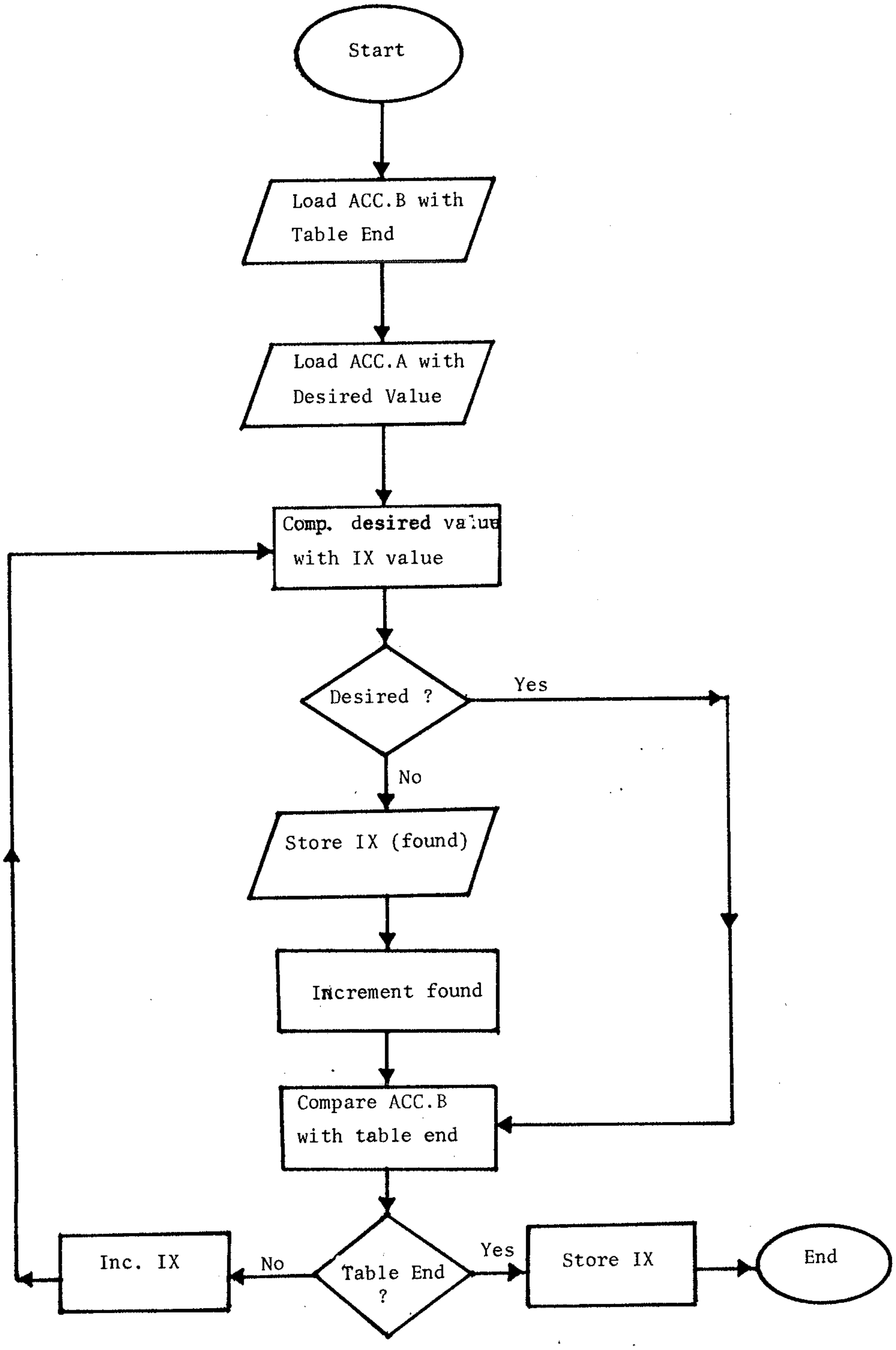

Start
Load ACC.B with Table End
Load ACC.A with Desired Value
Comp. desired value with IX value
Desired ?
Yes
No
Store IX (found)
Increment found
Compare ACC.B with table end
Table End ?
No
Yes
Inc. IX
Store IX
End

Fig 3.9 Table search flow chart

execute the table search routine where it calculates the sine and cosine of the phase angle. The signs are identified by setting the first bit in the corresponding flag registers as shown below:

Sign of Sine/cosine	Angle location	R - X Flag
cosine negative	second quad.	$CA3B
cosine negative	third quad.	$CA3C
sine negative	third quad.	$CA3D
sine negative	fourth quad.	$CA3E

3.2.3 Resistance-Reactance Calculation Routine

Figure 3.10 is a flow chart for the impedance calculation RTN. The program starts by reading the current I_p and imposing a minimum value of $20, then multiply this Ip by 4. The processor then reads the voltage V_p . If V_p is equal to zero, the resistance and reactance values are forced to $00 and outputed in PIA #1 and PIA #2 output registers otherwise it tests R-X flag to do either resistance or reactance calculations:

$$R = \frac{V_p \;.\; \cos\theta}{I_p \;.\; 4}$$

$$X = \frac{V_p \;.\; \sin\theta}{I \;.\; 4}$$

The sign of the R and X is determined by the subroutine shown in figure 3.11. The processor starts by putting a limit on maximum R and maximum X to be equal $7F. Then it checks the sign of sine/cosine and implements the logic given in the following table.

Logic table for sign of R & X in four qudrants

1st quad.	2nd quad.	3rd quad	4th quad	logic
yes	----	----	----	R+ = $80 + R
				X+ = $80 + X
----	yes	----	----	R- = $80 - R
				X+ = $80 + X
----	----	yes	----	R- = $80 - R
				X- = $80 - X
----	----	----	yes	R+ = $80 + R
				X- = $80 - X

Where R and X are the previously calculated values and R+, X+, R-, X- are the adjusted values with corresponding signs.

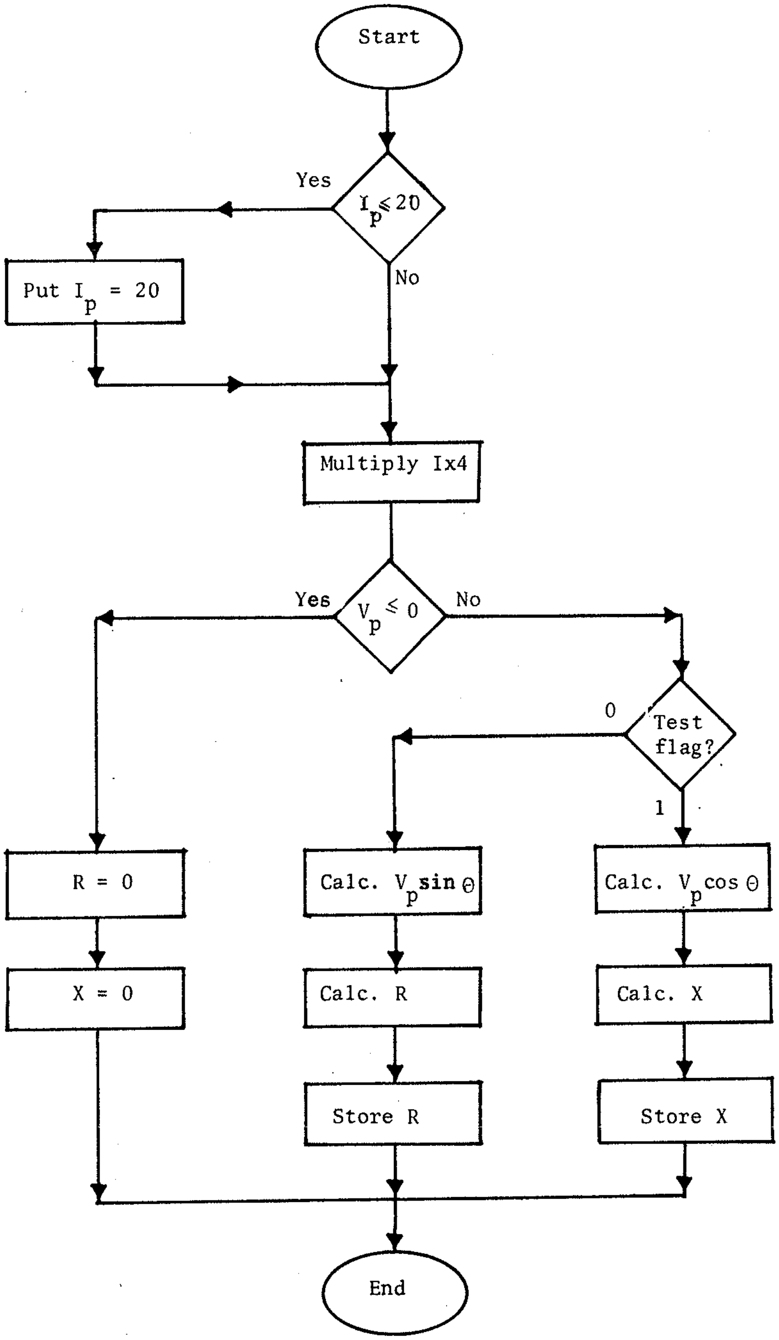


Fig. 3.10 Impedance calaulation routine flow chart

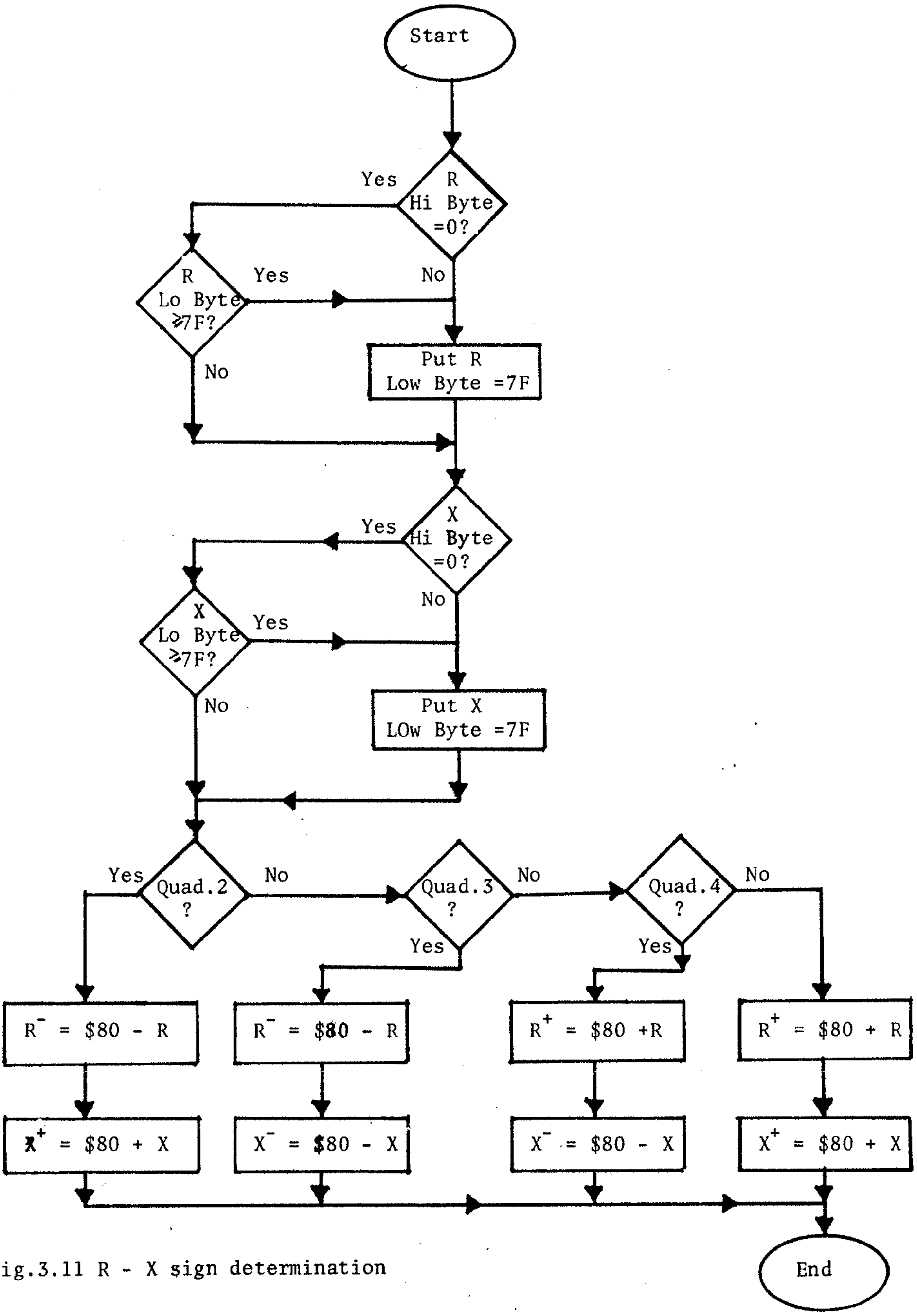


Fig.3.11 R - X sign determination

Chapter IV

COMPUTATION TECHNIQUE AND ERROR ANALYSIS

The device is designed to work on a 60 Hz system, with no harmonic content or dc offset, by accepting two voltage signals with the following ranges:

$$V_{in}(v) = -8.9v \text{ to } +8.9v$$

$$V_{in}(i) = -4v \text{ to } +4v$$

As was mentioned before, these signals are conditioned to have the ranges

$$V_{in}(v) = 0v \text{ to } +5v$$

$$V_{in}(i) = 0v \text{ to } +5v$$

The BICEPS (The Basic instrumental controllable Electrical Power system) machine rating is;

Full load line current	=	0.33	Amps
Line voltage	=	120	Volts
Full load power	=	40	Watt/phase

From these ratings we get

Impedance at full load = 364Ω

For a maximum short circuit current = $5\ I_{f.1.}$

$$Z_{min} = 72\Omega$$

under light load condition $I = \frac{1}{4} I_{f.1.}$

$$Z_{max} = 1454\ \Omega$$

The impedance range is $Z_{min} < Z > Z_{max}$ from which the resistance and reactance are calculated.

$$R = Z \cos\theta$$

$$X = Z \sin\theta$$

Sources of errors can be classified into two categories. The first category is component errors, mainly A/D and D/A converter errors. The second category is computational errors as a result of multiplication and division routines, sine table approximations and phase shift calculations.

4.1 A/D AND D/A CONVERTER ERRORS

Since the A/D terminal $V_{in}(+)$ is grounded and the reference terminal is floating, the zero digital code will need no adjustment and will be equivalent to zero analog input. Also the full digital code,$FF, will be equivalent to +5V analog input and no adjustment is needed. Manufacturers specification gives a full scale error of $\pm\frac{1}{4}$ LSB at a clock frequency of 1MHz.

Quantization errors are inherent in all digitization techniques. For A/D converters the minimum quantization error is the smallest increment of analog voltage to which the output signal can be approximated. This quantization error is given by

$$\Delta V_q = \frac{V}{r^n}$$

where

V is the analog input voltage,

r is the radix (r=2 for binary system), and

n is the number of digits (n=8 bits).

This will give the LSB bit a weight of

$$\text{LSB} = \Delta V_q = \frac{5\text{v}}{2^8} = 19.53 \text{ mv}$$

Another type of error is the sampling error. This arises from the fact that an A/D converter will look at the input signal for a short interval and leave it for a much longer time. In a general case, to minimize sampling error, a sinsoidal analog input is restricted in its maximum frequency by the relation

$$f \leqslant \frac{\Delta V}{V_p \cdot \Delta T}$$

where:

V_p is the peak input signal,

ΔT is the aperture time of the converter, and

ΔV is the resolution voltage.

Since the processor has a clock period of 1 μ sec, the time taken by this converter to count from 2^0 to 2^8 is

$$\text{conversion time} = T_c = 1\mu\text{sec} \times 2^8$$

The conversion rate is therfore

$$f_c = \frac{1}{T_c}$$

$$= 3906 \text{ conversions per second.}$$

For a 60 Hz analog input the conversion rate is thus

$$f_c = 65 \text{ conversions/cycle.}$$

From this conversion rate we can say that the sampling error for the A/D converter is negligible.

The D/A converter output is a current signal

$$I_o = K \left(\frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \ldots\ldots\ldots + \frac{A_8}{256} \right) \qquad \text{Amps}$$

$$I_0 = K \times A \qquad \text{Amps.}$$

where:

$K = \dfrac{V_{ref}}{R_{14}}$

A_1 is the most significant bit MSB

A_8 is the least significant bit LSB

A is either 1 or 0

R_{14} is the resistance connected to the converter at pin 14

In this design, these components were choosen to have values

$R_{14} = 1 \text{ K}$

$V_{ref} = +5\text{v}$

$K = 5\text{m Amp}$

The output current and voltage conversion is accomplished by the output signal conditioning circuit shown in Figure 3.2.

$$V_o = K \,.\, A.\, R_{14} \qquad \text{volts}$$

The output current for a digital word (00) is = 0 A and for a digital word (FF) is approximately 2 mA.

$$I_o = 0.005\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots\ldots\ldots + \frac{1}{256}\right) \text{ Amps}$$

Trim pots R_4 and R_5 on figure 3.2b are used for final adjustment to give a scaled output voltage to be

Digital Word (HEX)	Output Voltage (Decimal)
00	-2.5 Volt
7F	0 Volt
FF	+2.5 Volt

The accuracy test done by the manufacturer on this converter showed that the accuracy is within $\pm\frac{1}{2}$ LSB at 25 degrees C at full scale output current.

4.2 COMPUTATIONAL ERRORS:

Computational errors can be classified into the following three different types;

1) Errors in the phase angle calculation process where the phase angle in number of clock cycles has to be converted into degrees in hexadecimal number.

2) Errors due to the sine/cosine of the phase angle calculation. All entries in the look-up table are approximated to one byte long hexadecimal number. This approximation has an appreciable effect for angles between 84 to 90 degrees.

3) Errors resulted from the two-byte by two-byte divide routine.

$$\frac{\text{Dividend (2 bytes)}}{\text{Divisor (2 bytes)}} = \text{Quotient (2 byte) + Remainder (2 byte)}$$

Divide subroutine contributes largely to the total accumulated error in resistance and reactance calculation. This contribution is a function of the dividend and the divisor magnitudes. The quotient part of the division result is outputed to the D/A converter only and the remainder is discarded. The effect of discarding the remainder can be shown in an example. Assume that the input voltage is fixed at a value \$7F and in phase with the input current signal for illustrative purposes. For a current signal magnitude equal to \$30, the calculated impedance will be,

$$R = \left(\frac{\$7F}{\$30} \right)_{16} = \left(\frac{128}{48} \right)_{10}$$

$$R = (2.66)_{10}$$

$$X = 0$$

The divide routine will discard 0.666 and retain the quotient 2. The maximum R can be calculated by knowing the minimum allowed value of current signal which is equal to \$20,

$$R_{max.} = \left(\frac{\$7F}{\$20} \right)_{16} = \left(\frac{128}{32} \right)_{10}$$

$$R_{max.} = (4)_{10}$$

This value represents the maximum scale. The discarded value, 0.666, in the previous calculation gives an error of

$$\% \text{ of full scale} = \left(\frac{0.66}{4}\right)_{10} = 16.65\ \%$$

By taking different magnitudes of the divisor, I, table 4.1 is constructed to show the percentage error in R calculations. It shows too that the maximum error occurs at minimum value of divisor and is equal to 21.9%, The effect of this error can be reduced by multi-precision division calculation.

Table 4.1 Error in impedance due to division calculation Subroutine

Input I	Impedance		% error of
Hex value	Quotient	Remainder*	full scale
$20	$4	0.0	
$21	$3	0.878	21.9 %
$22	$3	0.765	19.1 %
$23	$3	0.657	16.4 %
$24	$3	0.555	13.9 %
$25	$3	0.459	11.5 %
$26	$3	0.368	9.2 %
$27	$3	0.282	7.0 %
$28	$3	0.20	5.1 %
$29	$3	0.122	3.1 %
$30	$2	0.666	16.66%
$35	$2	0.415	10.37%
$40	$2	0.462	11.50%
$50	$1	0.60	15.0 %
$60	$1	0.333	8.3 %
$70	$1	0.143	3.5 %
$80	$1	0.0	0.0 %

* Note: Remainder is listed in base 10

4.3 PHASE ANGLE, PEAK VOLTAGE/CURRENT CONSTANTS:

To calculate the peak values of both current and voltage signals, a voltage zero is detected first then the processor looks for a current zero by looping in a specified loop. During this looping process the processor keeps incrementing a counter by one count for each lap till a current zero is detected. The number stored in the counter, N, is the measure of the phase shift . The processor then waits for a period of 90 degrees by looping in a timing loop for 346 times. This constant is calculated by knowing the number of instructions, the time needed to execute each instruction in the loop and the processor clock period. The processor then will read the PIA #2 input register which will contain the peak value of the current signal. To calculate the voltage peak, the processor will look for the next voltage zero-crossing and waits a time equivalent to 90 degrees as it did for the current signal except that the constant will be 231.

Since the processor is running at 1MHz frequency, the clock period is 1 sec, and for a 60 Hz system the count (N) in number of laps can be changed into degrees as follows:

$$\theta = \text{constant} \times N.$$

To complete one lap (one increment in N) the processor needs 20 seconds: calculated by knowing the number of instructions in this loop and the number of clock cycles needed to execute each instruction.

Since

$$\frac{46.3\ \mu\ \text{sec/ degree}}{20\ \ \mu\ \text{sec /degree}} = 2.31$$

the phase angle = 0.4319 x N, in degrees

In hexadecimal form,

$$(0.4319)_{10} = (0.6E90)_{16}$$

Therefore

$$(\theta)_{10} = (0.6E90)_{16} N$$

After the phase angle is calculated, the sine and cosine are found out by the table look-up method.

A sine table for the phase angle was constructed and written into memory for angles beteen $(0)_{10}$ to $(90)_{10}$ degrees, $(00)_{16}$ to $(5A)_{16}$, in 1 degree increments which makes the table length equal to (5A) entries. This table is located at address $F100 to F15A. The sine of any angle can be obtained by going to that location. To calculate the cosine of the same angle the processor calculates

$$\cos \theta = \sin (90 - \theta)$$

by using the same table. As was stated in Chapter 3, the sign of sine/cosine is indicated by setting a corresponding " R-X flag " register for each quadrant.

4.4 ERROR AVERAGING TECHNIQUE:

Each of the sources of computational errors discussed earlier contributes to the total error in the output signal. This total error was found to be an unacceptable percentage of the total output signal. Therefore, an output signal averaging subroutine was developed. This routine performs the duty of filtering the errors resulting from the computation process before outputing the resistance and reactance values to the peripheral device. This duty is accomplished by averaging 8 samples of calculation. During the device starting process and after a reset signal a stack of 16 consecutive values, 8 for average resistance calculation and 8 for average reactance calculation, is cleared and the first sample of R and X is divided by 8 using the logic shift-right instruction three times. This first sample is stored in the stack and the PIA output register. The processor then loops back to calculate the second sample. After 8 samples have been added, the processor outputs the stack sum to the PIA to give the average value of 8 calculations.

After the processor obtains the first sample in the next cycle, it subtracts the oldest one from the total sum of the stack and adds the new one to both the stack and the stack sum for outputing.

Chapter V

DEVICE TESTING AND CALIBRATION PROCEDURE

To determine the accuracy and performance of this device two different types of tests were performed. In the first test the device was subjected to steady state-conditions and in the second test, to transient conditions.

5.1 DEVICE TESTING AND RESULTS:

In the first test two steady-state signals where applied at the input terminals V and I with amplitudes ±8.9 V_{p-p} at 60-Hz and ±4 V_{p-p} at 60-Hz respectively. When the device was reset and running, the output resistance and reactance where observed on a two channel oscilloscope as a function of time and shown in figures 5.1a and 5.1b. These figures show that when the device is reset ,t=0, the output R and X are zero and start increasing in eight steps to their full values. This is the result of the 8-samples averaging routine mentioned in Chapter 4. The length of each step is the time taken by the program to detect the V and I zero crossing and performing the calculation cycle. This time (t_s) is equal to

$$t_s = 0.2 \text{ x } 0.2 \text{ sec/div}$$
$$= 0.04 \text{ seconds}$$

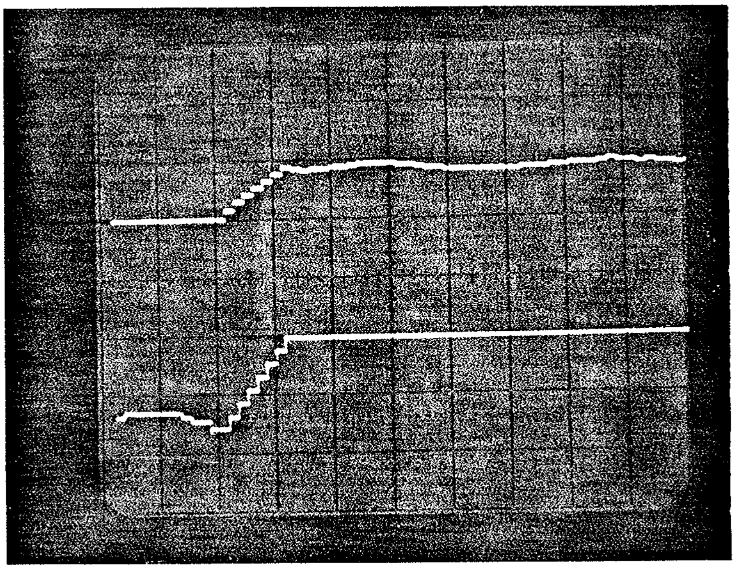


Fig. 5.1a

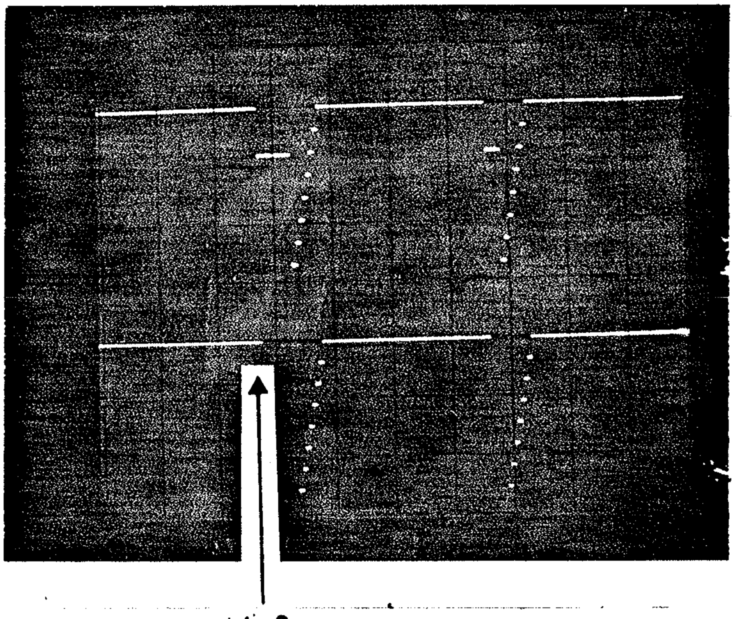


Fig. 5.1b

Figure 5.2a shows a plot of R versus X for 60-Hz input voltage signal and a 60.4-Hz input current signal. This difference in frequency is to simulate continuously variable phase shift. The amplitude of both signals is constant. The plot in figure 5.2b is an n-sided polygon. The reason for not being a circle is that the phase shift does not vary fast enough for the program to calculate new values for R and X and keep on outputing the same values till a new set of calculations are made. Another reason is the number saturation due to the multiplication and division subroutines. The effect of these sources of errors appears in fig.5.2d where the flat sides in the outermost distorted circle are due to number saturation and the limitation imposed on the minimum current signal to be $20. Figure 5.2c shows R and X versus time plot were flat peaks are due to the same reasons.

In the second test, the device was connected to the BICEPS machine as shown in figure 5.3. The current and voltage transformers are connected to phase 2. Line switching transients were created by switching the line length from 1.5pu to 2.75pu. This is accomplished by using a switch, not shown in the figure, to change line length to one of four different pre-set line lengths. As was explained in chapter 2, the power transfer capability of the system will decrease by switching off one of the lines in fig. 2.5a.

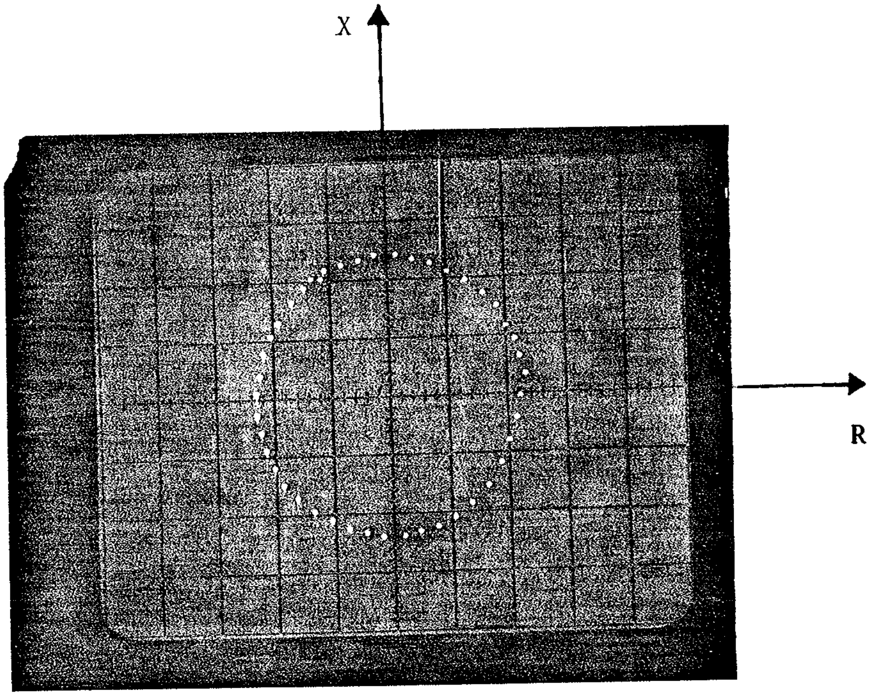


Fig. 5.2a

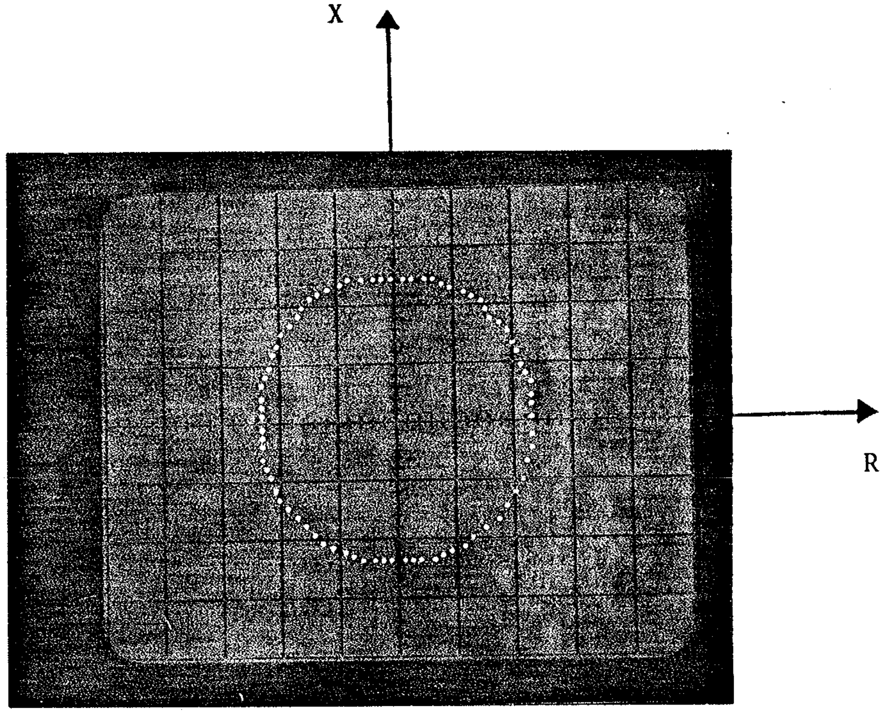


Fig. 5.2b

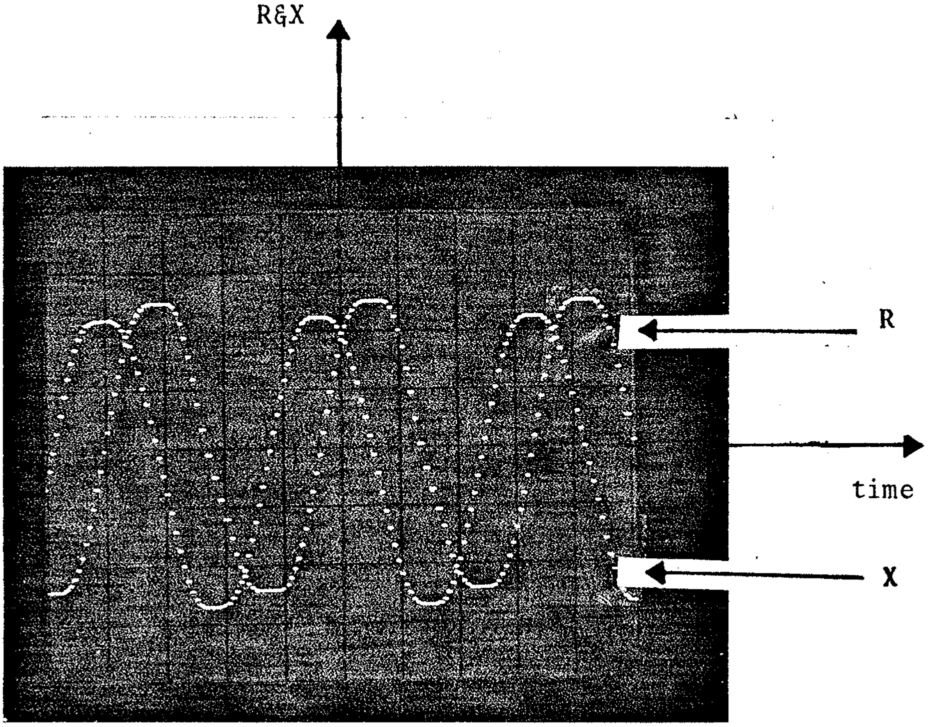


Fig. 5.2c

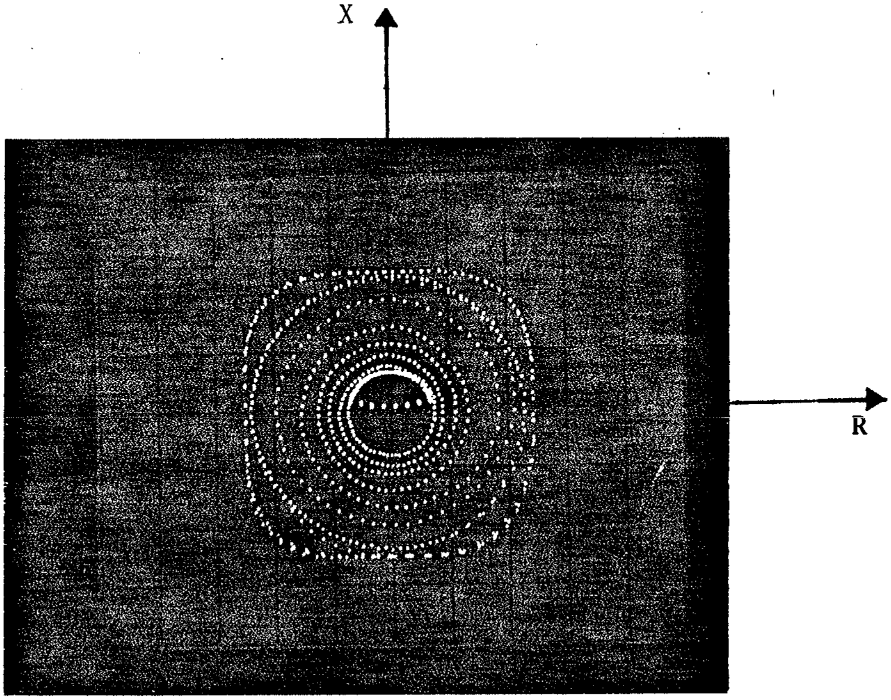


Fig 5.2d

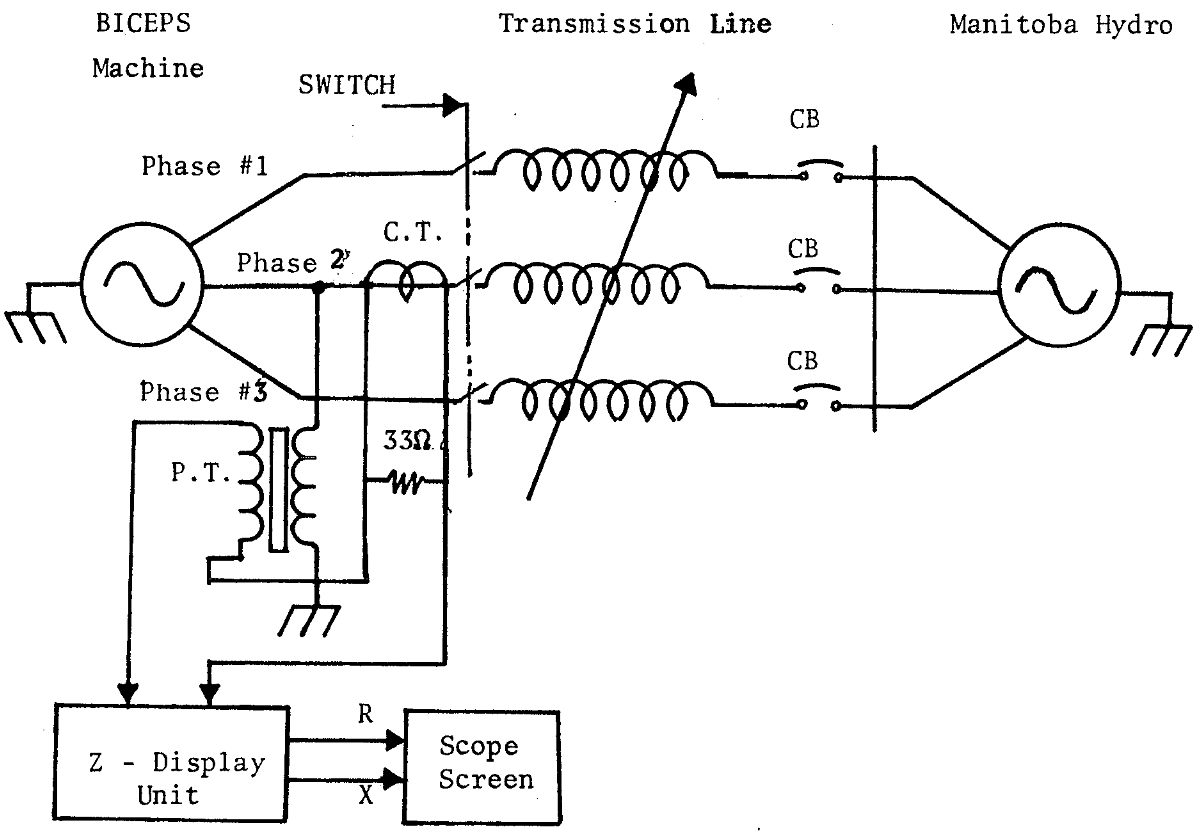


Fig. 5.3 Transient test connection on the BICEPS machine

Following the same argument, the generator was loaded to 0.65pu at a line length of 1.5pu. The corresponding system operating point on the impedance plane is shown in figure 5.4a at point t=0. After line switching to 2.75pu, the operating point was oscillating along the dotted curve as shown and finally stabilized at t=t . Each dot on this curve corresponds to one R and X calculation. Generator loading was increased to 0.68pu at line length of 1.5pu. After switching to 2.75pu, the system was stable with the operating point oscillating along the dotted locus as shown in Figure 5.4b.

To obtain an unstable transient, the generator was loaded to 0.75pu at line length =1.5pu. By switching to 2.75pu line length, the impedance locus was displayed on the scope screen as shown in Figure 5.5b in dotted line. The area shown by the 'unstable' sign indicates that the impedance display device could not generate the proper impedance locus. A smooth curve as shown on Figure 2.6a is the desired target. An 8-sample averaging routine was developed, to filter out the accumulated calculation error, and the device was tested one more time. One effect of this routine is to slow down the device response, a penalty for obtaining smooth curves under steady-state tests as shown in Figures 5.2c and 5.3a.

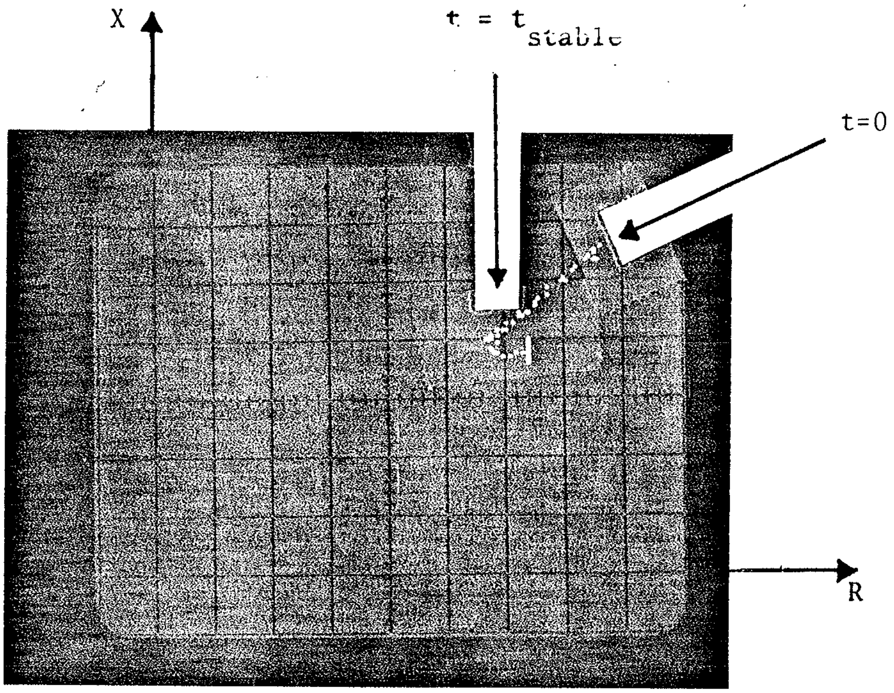


Fig. 5.4a Stable transient (p = .65 pu)

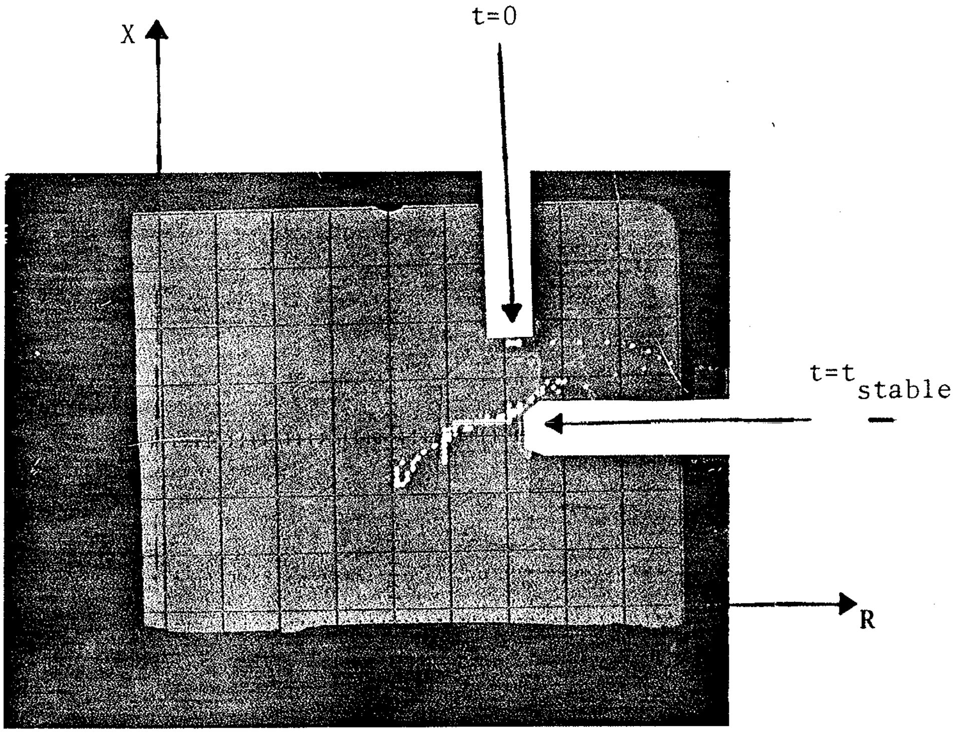


Fig. 5.4b Stable transient (p = .68 pu)

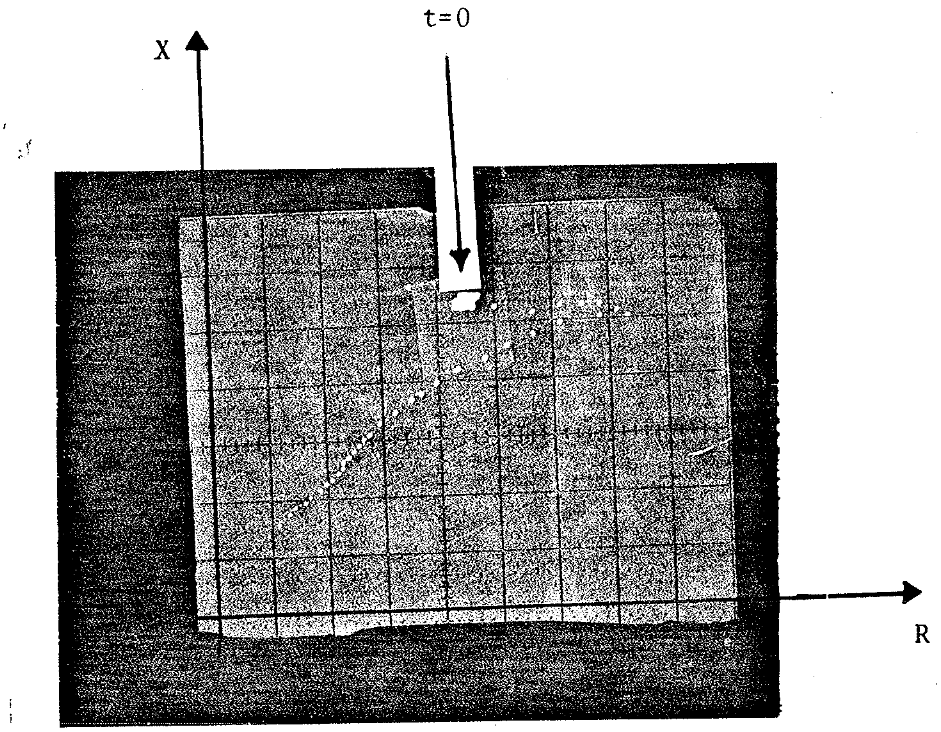


Fig 5.5a

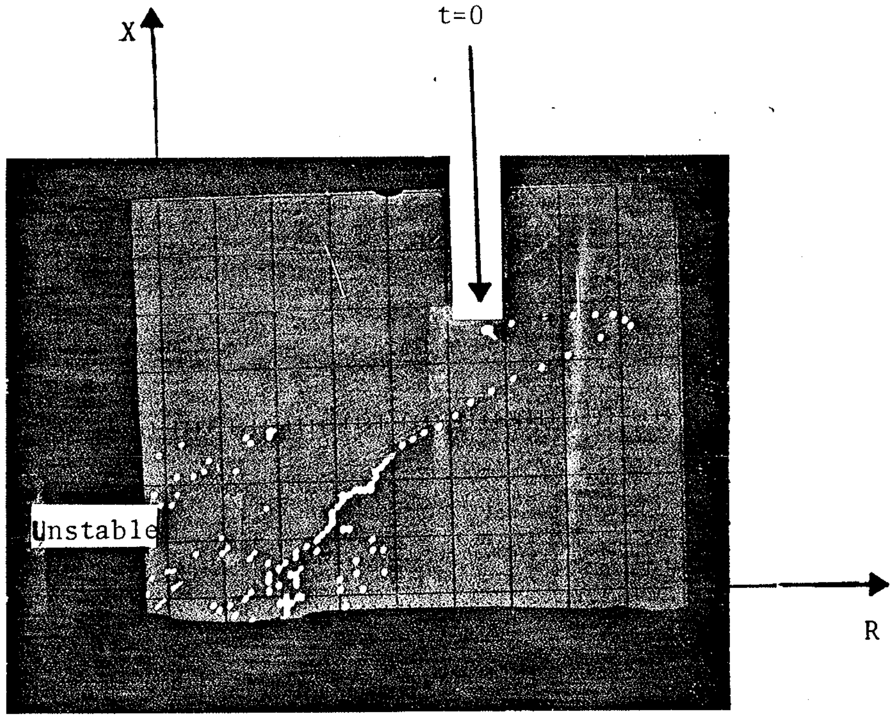


Fig. 5.5b Unstable power swing

5.2 DEVICE CALIBRATION PROCEDURE:

To calibrate the device, apply input sinusoidal voltages at the specified terminals analog input (V) and analog input (I) in Fig. 5.6. The analog input voltage range is $-8.9 < V < +8.9$ volts and the analog input current signal is $-4 < V <+4$ volts. The input signal conditioning circuits adjust these signals to the range accepted by the A/D converters $0 < V < +5$ Volt).

Trim pots TR3 and TR7 are used for signal biasing. Trim pots TR2 and TR6 are used to adjust operational amplifier gains.

The following steps are recommended for troubleshooting and device calibration:

1. This device runs on 5V DC and 12V DC. Test power supply circuit in Figure 5.1b for noise and ripple_free power supply
2. Apply a full scale 60 Hz analog voltage signal (-8.9v to +8.9v) to terminal Analog input (v) in figure 5.1a. A biased signal, (0v to +5v) should appear on terminal #6 on the A/D converter #1. If not, check the input signal conditioning circuit #1.
3. Apply a full scale 60 Hz analog current signal (-4v to +4v) to terminal Analog input (i). A biased signal (0v to +5v) should appear at terminal #6 on the A/D converter #2. If not, check the input signal conditioning circuit #2.

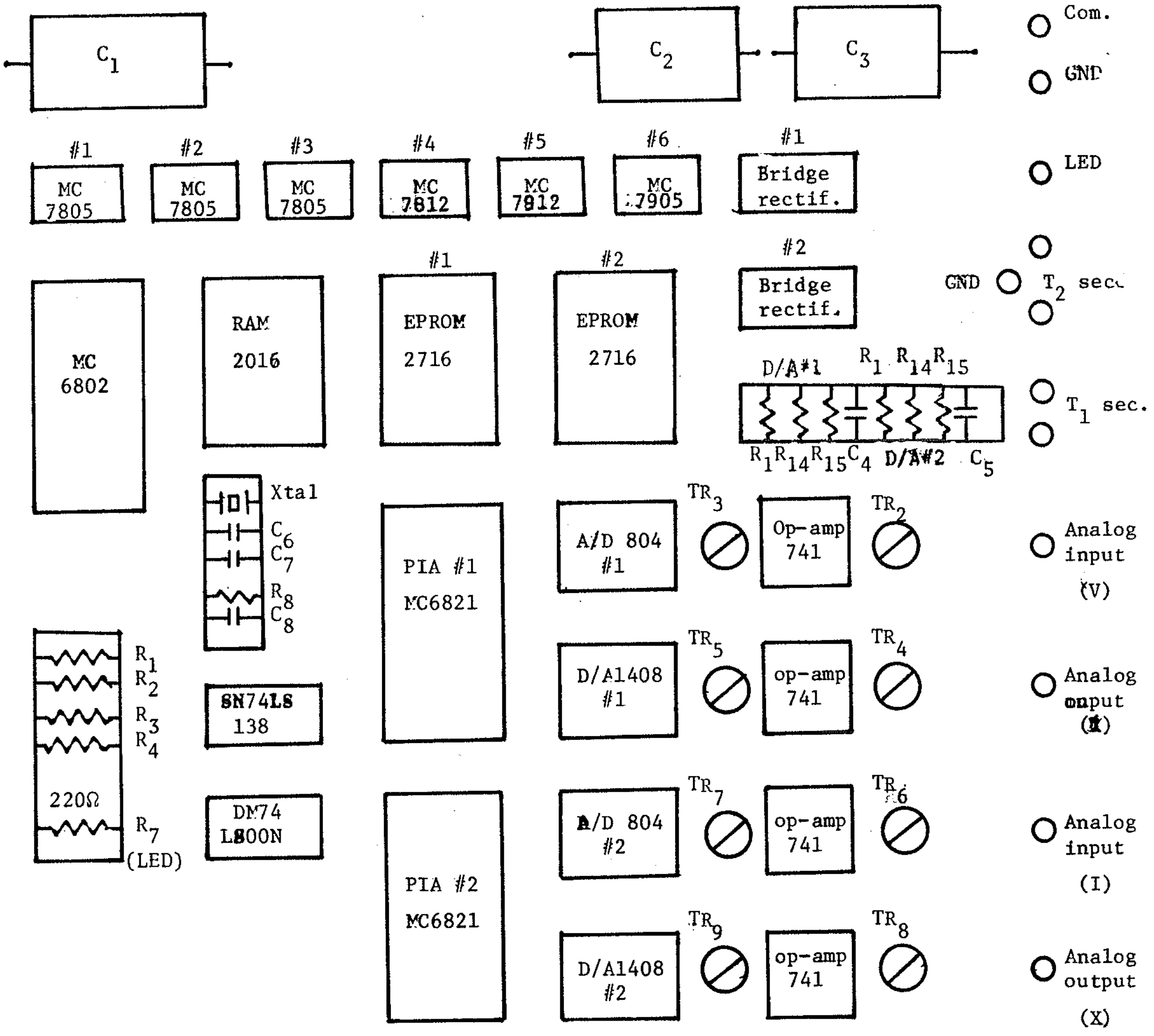


Fig. 5.6a Impedance display unit board layout

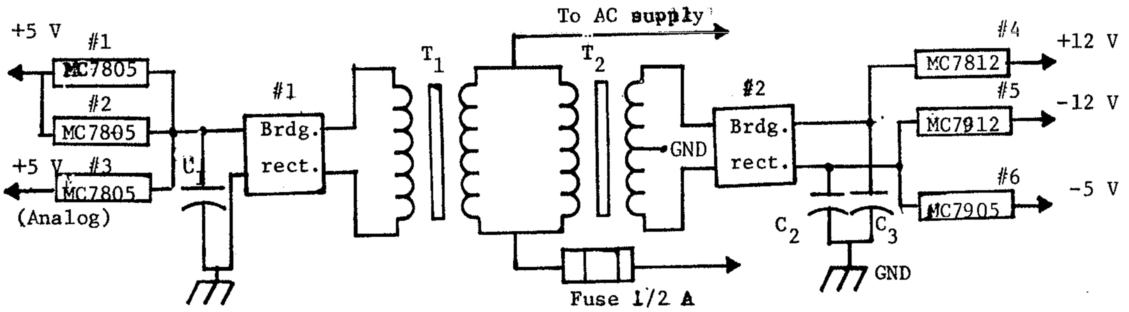


Fig. 5.6b power supply circuit

4. The microprocessor runs at a 1MHz frequency. This can be checked from terminal E by using an oscilloscope. The same clock signal shoul appear at pin 25 on PIAs #1 and #2, if not, check the crystal circuit.

5. For Chip selection and software changes, the Diagnostic Emulator should be used as follows:

a) Connect the emulator Probe into the microprocessor socket.

b) Switch Emulator ON. RESET and CLK indicators should be off; otherwise check the RESET or Crystal circuit as required.

c) With the RAMOVERLAY switches in the OFF position the Emulator display should read $F000 otherwise check decoder circuit.

d) When all above points are properly checked and corrected as required, decoding of the peripheral devices #1 and #2 should be tested as follows:

First: initialize both PIAs as shown in the software program. Call memory locations $E000 and $D000 on the address display. Pressing the Emulator EXAM key the DATA readout should give the digital word for each analog input. If not, check PIAs and A/D converters.

Second: call memory locations $E002 and #D002 and push $FF in each. A scope at Analog output (R) and analog output (X) on Fig. 5.6a should read +2.5 volt, if not, check PIAs and the output signal conditioning circuits for adjustments.

Third: Into the same memory locations, $E000 and $D000, push $7F, adjust trim pots TR4 and TR8 to obtain a zero volt output on the scope.

6. Apply the analog input signals (I) and (V) to the device then press the Emulator RUN key, the device should run giving the calculated resistance and reactance. If the device does not run repeat steps 1 through 6.

Chapter VI

CONCLUSIONS

1. This device is capable of computing and displaying, in real time, the impedance seen at a point in a power system.

2. Tests under steady-state conditions with simulated phase shift indicate that the device is able to generate a smooth impedance locus in spite of the errors accumulated during the R and X calculation process (mainly in the divide routine where the remainder of the division is discarded).

3. Stable transient tests on a model power system resulted in an acceptable swing locus, but the error was appreciable during an unstable transient test. Filtering out this error by an 8-sample averaging routine was effective in smoothing out the impedance locus but resulted in degraded dynamic response.

In light of these results, the author would like to recomend the following to enhance the device performance and duties:

1) Develope a software algorithm to plot the actual relay characteristics on the oscilloscope screen possibly by using the existing hardware.

2) Optimize the device performance by implementing multiprecision divide and multiply calculations.

3) Design a digital read-out using the existing software algorithm as an enhancement to the BICEPS teaching tool.

Appendix A

BIBLIOGRAPHY

1) Ronald L. Krutz, "Microprocessors And Logic Design," John Wiley & sons, Inc. 1980

2) General Electric, "Use of R-X diagram in relay works," switchgear puplications, GET - 2230b, 1966

3) W.A. Elmore, "Fundamentals of out-of-step Relaying," Westinghous electric Corporation, Relay - Instrument division , January 1979.

4) Edith Clarke, "Impedances seen by relays during power swings with and without faults," AIEE Transactions, Vol. 64, 1945, pp 372-384

5) Herbert Brunner, "Introduction to Microprocessors," Resto puplishing company, INc. 1981

6) National Semiconductors, " Data Conversion Acquisition Databook," 1980

7) Joseph D. Greenfeild, William Wray, " Using Microprocessors and Microcomputers: The 6800 Family," John Wiley & sons, 1981.

8) A.R. vanC. Warrington, " Protective Relays: Their Theory And Practice volume two," John Wiley & sons 1978

9) William D. Stevenson, Jr., " Elements of power systems," McGraw-Hill Book company, 1982

10) L.M. Wedepohl, " Polarized Mho Distance Relays, New

approach to the analysis of practical characteristics,"
IEE PROC. Vol 122, No. 3, March 1965

11) Lance A. Leventhal, " Introduction to Microprocessors Software, Hardware, Programing,"

12) Motorola Incorporation, " Linear Interface Integrated Circuits," 1979

Source Programme

```
S0 06 0000 48 44 52 1B
S1 13 F000 CE 00 04 FF D0 00 CE FF 04 FF D0 02 CE 00 04 FF E8
S1 13 F010 E0 00 CE FF 04 FF E0 02 FE FA 10 7F C9 61 6F 00 3A
S1 13 F020 08 7C C9 61 B6 C9 61 81 17 2B F3 BE FA 02 FE FA E6
S1 13 F030 00 B6 D0 00 B1 FA 08 2B 02 20 F6 B6 D0 00 B1 FA 1F
S1 13 F040 08 2A 02 20 F6 F6 E0 00 F1 FA 08 2B 03 08 20 F5 5E
S1 13 F050 F6 E0 00 F1 FA 08 2A 03 08 20 F5 FF CA 02 FE FA D6
S1 13 F060 06 01 01 09 09 26 FC F6 E0 00 F7 CA 09 B6 D0 00 3A
S1 13 F070 B1 FA 08 2B 02 20 F6 B6 D0 00 B1 FA 08 2A 02 20 11
S1 13 F080 F6 FE FA 04 01 09 09 09 26 FA B6 D0 00 B7 CA 0A 3D
S1 13 F090 86 6E B7 CC 01 C6 90 F7 CC 02 B6 CA 02 B7 CC 03 D1
S1 13 F0A0 F6 CA 03 F7 CC 04 BD FD 00 B6 CC 06 B7 CA 03 F6 16
S1 08 F0B0 CC 06 F7 CA 02 7E F4 00 4E
S9 03 0000 FC
...

          1       MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE   1


         M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
            COPYRIGHT 1974 TO 1978 BY MOTOROLA INC

         MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                      MODIFIED IAMC VER: 1.0

00001                       NAM    PEAKPICK
C0002 F000                  ORG    $F000
00003                *
00004                ************************************************
00005                *
00006                *    PEAK PICKING ROUTINE & PHASE ANGLE CALCULATION
00007                *
00008                ************************************************
00009                *
00010 F000 CE 0004          LDX    #$04
00011 F003 FF D000          STX    $D000       INIT. PIA2 INPUT REGISTERS
00012 F006 CE FF04          LDX    #$FF04      LOAD IND. REG. WITH HEX  $FFO
00013 F009 FF D002          STX    $D002       INIT. PIA2 OUTPUT REGISTERS
00014 F00C CE 0004          LDX    #$004       LOAD IND. REGISTER WITH HEX.
00015 F00F FF E000          STX    $E000       INIT. PIA1 INPUT REGISTERS
00016 F012 CE FF04          LDX    #$FF04      LOAD INDEX REGISTER WITH $FFO
00017 F015 FF E002          STX    $E002       INIT. PIA1 OUTPUT REGISTERS
00018 F018 FE FA10          LDX    $FA10       LOAD IX WITH STARTING ADDRESS
 00019 F01B 7F C961          CLR    $C961       CLEAR BYTE COUNT
 00020 F01E 6F 00    CLEAR   CLR    0,X         CLEAR LOCATION INDICATED BY I
 00021 F020 08   INX
 00022 F021 7C C961          INC    $C961       INCREMENT COUNTER
 00023 F024 B6 C961          LDA A $C961       LOAD ACC.A WITH COUNT VALUE
 00024 F027 81 17            CMP A #$17
 00025 F029 2B F3            BMI    CLEAR      BRANCH TO CLEAR REST OF TABLE


------------------------------------------------------------------------------------------


 00026 F02B BE FA02 BIGIN  LDS    $FA02       INITIALISE STACK POINTER ( $C
 00027 F02E FE FA00        LDX    $FA00       CLEAR INDEX REGISTER
 00028 F031 B6 D000 PIA2VL LDA A $D000       LOAD ACC.A WITH INST. VALUE O
 00C29 F034 B1 FA08        CMP A $FA08       COMPARE ACC.A WITH HEX. $80
 00030 F037 2B 02          BMI    PIA2VH     BRANCH IF MINUS
 00031 F039 20 F6          BRA    PIA2VL     BRANCH ALWAYS
 00032 F03B B6 D000 PIA2VH LDA A $D000       LOAD ACC.A WITH INST. VALUE O
 00033 F03E B1 FA08        CMP A $FA08       COMPARE ACC.A WITH HEX. $80
 00034 F041 2A 02          BPL    PIA1IL     BRANCH TO CHECK I VALUE
 00035 F043 20 F6          BRA    PIA2VH     BRANCH BACK FOR V CHECK
 00036 F045 F6 E000 PIA1IL LDA B $E000       LOAD ACC.B BY INST.VALUE OF I
 00037 F048 F1 FA08        CMP B $FA08       COMPARE VALUE OF I WITH HEX.
 00038 F04B 2B 03          BMI    PIA1IH     BRANCH IF MINUS
 00039 F04D 08   INX               INCREMENT INDEX REGISTER
 00040 F04E 20 F5          BRA    PIA1IL     BRANCH ALWAYS
 00041 F050 F6 E000 PIA1IH LDA B $E000       LOAD ACC.B WITH INST. VALUE O
 00042 F053 F1 FA08        CMP B $FA08       COMPARE INST. VALUE OF I WITH
 00043 F056 2A 03          BPL    STX1       BRANCH TO STORE VALUE OF (N)
 00044 F058 08   INX               INCREMENT N.... N=N+1
 00045 F059 20 F5          BRA    PIA1IH     BRANCH ALWAYS
 00046 F05B FF CA02 STX1   STX    $CA02       STORE VALUE OF (N)
 00047 F05E FE FA06        LDX    $FA06       LOAD IND. REG. BY $015A
 00048 F061 01             NOP
 00049 F062 01             NOP
 00050 F063 09      DEX1   DEX                DECREMENT INDEX. REGISTER
 00051 F064 09             DEX                DECREMENT IND. REGISTER
 00052 F065 26 FC          BNE    DEX1       BRANCH TO DECREMENT IX LOOP
 00053 F067 F6 E000        LDA B $E000       LOAD ACC.B WITH 'I' MAX
 00054 F06A F7 CA09        STA B $CA09       STORE MAX. VALUE OF I
 ...

1 P A.P C.        MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE   2


 00055 F06D B6 D000 PIA2V1 LDA A $D000       LOAD ACC.A WITH INST. VALUE O
 00056 F070 B1 FA08        CMP A $FA08       COMPARE INST. VALUE OF V WITH
 00057 F073 2B 02          BMI    PIA2V2     BRANCH TO CHECK V>$80
 00058 F075 20 F6          BRA    PIA2V1     BRANCH ALWAYS
 00059 F077 B6 D000 PIA2V2 LDA A $D000       LOAD ACC.A WITH INST. VALUE O
 00060 F07A B1 FA08        CMP A $FA08       COMPARE INST. VALUE OF V WITH
 00061 F07D 2A 02          BPL    LDX1       BRANCH IF V > $80
 00062 F07F 20 F6          BRA    PIA2V2     BRANCH ALWAYS
 00063 F081 FE FA04 LDX1   LDX    $FA04       LOAD INDEX REGISTER WITH $010
 00064 F084 01      DEX2   NOP                NO OPERATION
 00065 F085 09             DEX                DECREMENT INDEX REGISTER
 00066 F086 09             DEX                DECREMENT INDEX REGISTER
 00067 F087 09             DEX                DECREMENT INDEX REGISTER
 00068 F088 26 FA          BNE    DEX2
 00069 F08A B6 D000        LDA A $D000       LOAD ACC.A WITH V MAX.
```

```
00070 F08D B7 CA0A          STA A $CA0A      STORE MAX. VALUE OF V
00071 F090 86 8E            LDA A #$8E       LOAD ACC.A WITH A CONSTANT $8
00072 F092 B7 CC01          STA A $CC01      INITIALISE MUL2RTN
00073 F095 C6 90            LDA B #$90
00074 F097 F7 CC02          STA B $CC02
00075 F09A B6 CA02          LDA A $CA02
00076 F09D B7 CC03          STA A $CC03
00077 F0A0 F6 CA03          LDA B $CA03
00078 F0A3 F7 CC04          STA B $CC04
00079 F0A6 BD FD00          JSR   MUL2RTN
00080 F0A9 B6 CC06          LDA A $CC06
00081 F0AC B7 CA03          STA A $CA03
00082 F0AF F6 CC05          LDA B $CC05
00083 F0B2 F7 CA02          STA B $CA02
00084 F0B5 7E F400          JMP   SINSTR     JUMP TO SINCOS SUBROUTINE
00085                *
00086 FD00                  ORG   $FD00
00087 FD00 0001      MUL2RT RMB   1
00088 F400                  ORG   $F400
00089 F400 0001      SINSTR RMB   1
00090                       END

SYMBOL TABLE

CLEAR  F01E  BIGIN  F028  PIA2VL F031  PIA2VH F03B  PIA1IL F045
PIA1IH F050  STX1   F058  DEX1   F063  PIA2V1 F08D  PIA2V2 F077
LDX1   F081  DEX2   F084  MUL2RT FD00  SINSTR F400
S0 06 0000 48 44 52 1B
S1 13 F400 BE FA 02 F6 FA 0D F7 CA 0F 7F CA 0E 7F CA 13 7F 3F
S1 13 F410 CA 1C 7F CA 1D 7F CA 1E 7F CA 1F 7F CA 24 7F CA 17
S1 13 F420 3A 7F CA 3B 7F CA 3C 7F CA 3D 7F CA 3E 7F CA 3F 00
S1 13 F430 7F CA 42 B6 CA 02 B7 CA 0B F6 CA 03 F7 CA 0C BD E2
S1 13 F440 F5 49 25 2D B6 CA 11 B1 FA 00 27 02 20 15 F6 CA CE
S1 13 F450 12 F1 FA 00 27 02 20 0B 4F B7 CA 18 43 B7 CA 17 94
S1 13 F460 7E F3 00 B7 CA 0B F6 CA 12 F7 CA 0C F7 CA AA 20 71
S1 13 F470 06 B6 CA 03 7E F5 8D BD F5 49 25 22 B6 CA 11 B1 7B
S1 13 F480 FA 00 27 02 20 24 F6 CA 12 F1 FA 00 27 02 20 1A F1
S1 13 F490 5F F7 CA 17 53 F7 CA 18 7C CA 3B 7E F3 00 F6 CA 53
S1 13 F4A0 AA 86 6A 10 7C CA 3B 7E F5 8D B7 CA 0B F6 CA 12 DF
S1 13 F4B0 F7 CA 0C F7 CA AB 20 03 01 01 01 BD F5 49 25 54 75
S1 13 F4C0 B6 CA 11 B1 FA 00 27 02 20 18 F6 CA 12 F1 FA 00 DE
S1 13 F4D0 27 02 20 0E 4F B7 CA 18 43 B7 CA 17 7C CA 3D 7E 0D
S1 13 F4E0 F3 00 B7 CA 0B F6 CA 12 F7 CA 0C F7 CA AC BD F5 DB
S1 13 F4F0 49 25 30 B6 CA 11 B1 FA 00 27 02 20 38 F6 CA 12 DB
S1 13 F500 F1 FA 00 27 02 20 2E 5F F7 CA 17 53 F7 CA 18 7E B4
S1 13 F510 F3 00 20 21 7C CA 3D 7C CA 3C F6 CA AB F7 CA 12 70
S1 13 F520 7E F5 8A 7C CA 3E 7C CA 3F F6 CA AC 86 5A 10 B7 BE
S1 13 F530 CA 12 7E F5 8A 4F 06 7F CA 24 B6 CA 12 B7 CA 03 16
S1 13 F540 F6 CA 11 F7 CA 02 7E F4 33 0C B6 CA 0C F6 CA 0B 1B
S1 13 F550 B0 CA 0F F2 CA 0E B7 CA 12 F7 CA 11 39 01 01 01 83

-------------------------------------------------------------------------------------------

S1 13 F560 C6 D0 FE FA 0E B6 CA 43 A1 00 26 06 EE FF FF CA B5
S1 13 F570 40 39 E1 00 27 04 08 7E F5 68 3E 7C CA 3B 7C CA 1A
S1 13 F580 24 F6 CA AA 86 5A 10 B7 CA 12 B6 CA 12 B7 CA 43 10
S1 13 F590 BD F5 60 7D CA 24 26 14 F6 CA 41 F7 CA 17 86 5A F7
S1 13 F5A0 B0 CA 43 B7 CA 43 7C CA 24 BD F5 60 FF CA 43 F6 58
S1 08 F5B0 CA 44 F7 CA 18 7E F3 00 F7
S9 03 0000 FC
 ---

1                  MOTOROLA M68SAM CROSS-ASSEMBLER            PAGE  1


           M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
              COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

           MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                     MODIFIED IAMC VER: 1.0

00001                       NAM   SINCOS
00002 F400                  ORG   $F400
00003                ********************************************
00004                *
00005                *  SINE-COSINE PHASE SHIFT ANGLE SUBROUTINE    *
00006                *
00007                ********************************************
00008                *
00009 F400 8E FA02 SINSTR LDS   $FA02      INITIALISE STACK POINTER
00010 F403 F6 FA0D        LDA B $FA0D
00011 F406 F7 CA0F        STA B $CA0F      STORE $D0 ( N=90 DEGREES) FIX
00012 F409 7F CA0E        CLR   $CA0E      CLEAR MEMORY LOCATION
00013 F40C 7F CA13        CLR   $CA13      CLEAR MEMORY LOCATION
00014 F40F 7F CA1C        CLR   $CA1C      CLEAR MEMORY LOCATION
00015 F412 7F CA1D        CLR   $CA1D      CLEAR MEMORY LOCATION
00016 F415 7F CA1E        CLR   $CA1E      CLEAR MEMORY LOCATION
00017 F418 7F CA1F        CLR   $CA1F      CLEAR MEMORY LOCATION
00018 F41B 7F CA24        CLR   $CA24      CLEAR SINE-COSINE LOOP FLAG
00019 F41E 7F CA3A        CLR   $CA3A      CLEAR RESISTANCE - REACTANCE
00020 F421 7F CA3B        CLR   $CA3B      CLEAR SEC QUADRANT COSINE FLA
00021 F424 7F CA3C        CLR   $CA3C
00022 F427 7F CA3D        CLR   $CA3D
00023 F42A 7F CA3E        CLR   $CA3E
00024 F42D 7F CA3F        CLR   $CA3F
00025 F430 7F CA42        CLR   $CA42
00026 F433 B6 CA02 SUB90  LDA A $CA02      LOAD ACC.A WITH ( N HIGH BYTE
00027 F436 B7 CA0B        STA A $CA0B      STORE PHASE SHIFT (HIGH BYTE)
00028 F439 F6 CA03        LDA B $CA03      LOAD ACC.B WITH ( N LOW BYTE
00029 F43C F7 CA0C        STA B $CA0C      STORE PHASE SHIFT (LOW BYTE)
00030 F43F BD F549        JSR   SUB22      JUMP TO SUBTRACT SUBROUTINE
00031 F442 25 2C          BCS   SUBEND     BRANCH IF ANGLE IS < 90 DEGRE
```

```
00032 F444 B6 CA11        LDA A $CA11
00033 F447 B1 FA00        CMP A $FA00
00034 F44A 27 02          BEQ   ZERO
00035 F44C 20 15          BRA   BR180
00036 F44E F6 CA12 ZERO   LDA B $CA12
00037 F451 F1 FA00        CMP B $FA00
00038 F454 27 02          BEQ   ANG90
00039 F456 20 0B          BRA   BR180
00040 F458 4F       ANG90 CLR A
00041 F459 B7 CA18        STA A $CA18     STORE COSINE 90 DEGREES
00042 F45C 43             COM A
00043 F45D B7 CA17        STA A $CA17     STORE SINE 90 DEGREES
00044 F460 7E F300        JMP   $F300     JUMP TO IMPEDANCE RTN
00045 F463 B7 CA0B BR180  STA A $CA0B     RE-INITIALISE SUBTRACT ROUTIN
00046 F466 F6 CA12        LDA B $CA12
00047 F469 F7 CA0C        STA B $CA0C     RE- INITIALISE SUBTRACT ROUTI
00048 F46C F7 CAAA        STA B $CAAA     SAVE ANGLE VALUE FOR FUTURE U
00049 F46F 20 06          BRA   SUB180    BRANCH TO SUBTACT 90 DEGREES
00050 F471 B6 CA03 SUBEND LDA A $CA03     LOAD ACC.A WITH PHASE ANGLE (
00051 F474 7E F58D        JMP   ANGLE
00052 F477 BD F549 SUB180 JSR   SUB22     JUMP TO SUBTRACT ROUTINE
00053 F47A 25 22          BCS   JUMP
00054 F47C B6 CA11        LDA A $CA11

...

.: N 03          MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  2


00055 F47F B1 FA00        CMP A $FA00
00056 F482 27 02          BEQ   ZERO1
00057 F484 20 24          BRA   BR270
00058 F486 F6 CA12 ZERO1  LDA B $CA12
00059 F489 F1 FA00        CMP B $FA00
00060 F48C 27 02          BEQ   ANG180
00061 F48E 20 1A          BRA   BR270
00062 F490 5F      ANG180 CLR B
00063 F491 F7 CA17        STA B $CA17
00064 F494 53             COM B
00065 F495 F7 CA18        STA B $CA18
00066 F498 7C CA3B        INC   $CA3B
00067 F49B 7E F300        JMP   $F300     JMP TO IMPEDANCE RTN
00068 F49E F6 CAAA JUMP   LDA B $CAAA     PHASE ANG.( 90 < ANGLE < 180
00069 F4A1 86 5A          LDA A #$5A
00070 F4A3 10             SBA
00071 F4A4 7C CA3B        INC   $CA3B     FLAG FOR COSINE IS NEGATIVE
00072 F4A7 7E F58D        JMP   ANGLE
00073 F4AA B7 CA0B BR270  STA A $CA0B     RE- INITIALISE SUBTRACT ROUTI
00074 F4AD F6 CA12        LDA B $CA12
00075 F4B0 F7 CA0C        STA B $CA0C     RE- INITIALISE SUBTRACT ROUTI
00076 F4B3 F7 CAAB        STA B $CAAB     SAVE ANGLE VALUE FOR FUTURE U

------------------------------------------------------------------------

00077 F4B6 20 03          BRA   SUB270    BRANCH TO SUBTRACT 90 DEGREES
00078 F4B8 01             NOP
00079 F4B9 01             NOP
00080 F4BA 01             NOP
00081 F4BB BD F549 SUB270 JSR   SUB22
00082 F4BE 25 54          BCS   FLAG2
00083 F4C0 B6 CA11        LDA A $CA11
00084 F4C3 B1 FA00        CMP A $FA00
00085 F4C6 27 02          BEQ   ZERO2
00086 F4C8 20 18          BRA   BR360
00087 F4CA F6 CA12 ZERO2  LDA B $CA12
00088 F4CD F1 FA00        CMP B $FA00
00089 F4D0 27 02          BEQ   ANG270
00090 F4D2 20 0E          BRA   BR360     270 < ANGLE
00091 F4D4 4F      ANG270 CLR A
00092 F4D5 B7 CA18        STA A $CA18
00093 F4D8 43             COM A
00094 F4D9 B7 CA17        STA A $CA17
00095 F4DC 7C CA3D        INC   $CA3D     SIN 270 DGREES IS NEGATIVE
00096 F4DF 7E F300        JMP   $F300
00097 F4E2 B7 CA0B BR360  STA A $CA0B     RE- INITIALISE SUBTRACT ROUTI
00098 F4E5 F6 CA12        LDA B $CA12
00099 F4E8 F7 CA0C        STA B $CA0C     RE - INITIALISE SUBTRACT ROUT
00100 F4EB F7 CAAC        STA B $CAAC     SAVE ANGLE VALUE FOR FUTURE U
00101 F4EE BD F549        JSR   SUB22     JUMP TO SUBTRACT ROUTINE
00102 F4F1 25 30          BCS   FLAG3
00103 F4F3 B6 CA11        LDA A $CA11     ANGLE = 360
00104 F4F6 B1 FA00        CMP A $FA00
00105 F4F9 27 02          BEQ   ZERO3
00106 F4FB 20 38          BRA   NEWCYL
00107 F4FD F6 CA12 ZERO3  LDA B $CA12
00108 F500 F1 FA00        CMP B $FA00
00109 F503 27 02          BEQ   ANG360
00110 F505 20 2E          BRA   NEWCYL
00111 F507 5F      ANG360 CLR B
00112 F506 F7 CA17        STA B $CA17
00113 F50B 53             COM B
00114 F50C F7 CA18        STA B $CA18

...

1 S N 03          MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  3


00115 F50F 7E F300        JMP   $F300
00116 F512 20 21          BRA   NEWCYL    BRANCH TO NEW CYCLE
00117 F514 7C CA3D FLAG2  INC   $CA3D     SET FLAG, SIN IS NEGATIVE, 3R
00118 F517 7C CA3C        INC   $CA3C     SET FLAG, COS IS NEGATIVE, 3R
00119 F51A F6 CAAB        LDA B $CAAB
```

```
00120 F51D F7 CA12        STA B $CA12     180 < PHASE SHIFT < 270
00121 F520 7E F58A        JMP   ANGLE2
00122 F523 7C CA3E FLAG3  INC   $CA3E     SET FLAG, SIN IS NEGATIVE, 4T
00123 F526 7C CA3F        INC   $CA3F     SET FLAG     MAY BE NOT NEEDE
00124 F529 F6 CAAC        LDA B $CAAC
00125 F52C 86 5A          LDA A #$5A
00126 F52E 10             SBA
00127 F52F B7 CA12        STA A $CA12
00128 F532 7E F58A        JMP   ANGLE2
00129 F535 4F      NEWCYL CLR A
00130 F536 06             TAP             PUSH VALUE IN ACC.A INTO C.C.
00131 F537 7F CA24        CLR   $CA24     INCREMENT SIN-COS FLAG
00132 F53A B6 CA12        LDA A $CA12
00133 F53D B7 CA03        STA A $CA03
00134 F540 F6 CA11        LDA B $CA11

00135 F543 F7 CA02        STA B $CA02
00136 F546 7E F433        JMP   SUB90     JUMP TO START A NEW CYCLE
00137 F549 0C      SUB22  CLC             CLEAR CARRY BIT
00138 F54A B6 CA0C        LDA A $CA0C     LOAD ACC.A WITH DATA IN $CA0C
00139 F54D F6 CA0B        LDA B $CA0B     LOAD ACC.B WITH DATA IN $FA0B
00140 F550 B0 CA0F        SUB A $CA0F     ACC.A - $CA0F
00141 F553 F2 CA0E        SBC B $CA0E     SUBTRACT WITH CARRY, ACC.B -
00142 F556 B7 CA12        STA A $CA12
00143 F559 F7 CA11        STA B $CA11
00144 F55C 39             RTS             RETURN FROM SUBROUTINE
00145 F55D 01             NOP
00146 F55E 01             NOP
00147 F55F 01             NOP
00148 F560 C6 D0   SINTBL LDA B #$D0      LOAD ACC.B WITH END OF TABLE
00149 F562 FE FA0E        LDX   $FA0E     LOAD IX WITH SIN-COS TABLE ST
00150 F565 B6 CA43        LDA A $CA43     LOAD AAC.A WITH DESIRED VALUE
00151 F568 A1 00   SINTB1 CMP A 00,X
00152 F56A 26 06          BNE   SINTB2
00153 F56C EE FF          LDX   #$FF,X
00154 F56E FF CA40        STX   $CA40     $CA41 CONTAINS SINE,THE ANGLE
00155 F571 39             RTS
00156 F572 E1 00   SINTB2 CMP B ,X
00157 F574 27 04          BEQ   SINTB3
00158 F576 08             INX             INCREMENT INDEX REGISTER
00159 F577 7E F568        JMP   SINTB1
00160 F57A 3E      SINTB3 WAI             WAIT
00161 F57B 7C CA3B FLAG1  INC   $CA3B
00162 F57E 7C CA24        INC   $CA24     INCREMENT SIN-COS FLAG
00163 F581 F6 CAAA        LDA B $CAAA
00164 F584 86 5A          LDA A #$5A
00165 F586 10             SBA
00166 F587 B7 CA12        STA A $CA12
00167 F58A B6 CA12 ANGLE2 LDA A $CA12     LOAD ACC.A WITH PHASE
00168 F58D B7 CA43 ANGLE  STA A $CA43     $CA43 CONTAINS DESIRED VALUE
00169 F590 BD F560        JSR   SINTBL    JUMP FOR SIN-TABLE SEARCH
00170 F593 7D CA24        TST   $CA24     TEST IF $CA24 CONTAINS ZERO
00171 F596 26 14          BNE   COSAGL    BRANCH TO STORE COSINE ANGLE
00172 F598 F6 CA41        LDA B $CA41     LOAD ACC.A BY SINE PHASE ANGL

--------------------------------------------------------------------------------

00173 F59B F7 CA17        STA B $CA17     STORE SINE PHASE ANGLE IN MUL
00174 F59E 86 5A          LDA A #$5A

...

1 S N 03         MOTOROLA M68SAM CROSS-ASSEMBLER         PAGE   4


00175 F5A0 B0 CA43        SUB A $CA43
00176 F5A3 B7 CA43        STA A $CA43
00177 F5A6 7C CA24        INC   $CA24     INCREMENT $CA24
00178 F5A9 BD F560        JSR   SINTBL    JUMP TO SIN TABLE ROUTINE
00179 F5AC FF CA43 COSAGL STX   $CA43     STORE COSINE PHASE ANGLE
00180 F5AF F6 CA44        LDA B $CA44     LOAD ACC.B WITH COSINE PHASE
00181 F5B2 F7 CA18        STA B $CA18     STORE PHASE ANGLE
00182 F5B5 7E F300        JMP   $F300     JUMP TO IMPEDANCE RTN
00183              *
00184 F400                ORG   $F400
00185 F400 0001    SINSTR RMB   1
00186              *
00187                     END
SYMBOL TABLE

SINSTR F400  SUB90  F433  ZERO   F44E  ANG90  F458  BR180  F463
SUBEND F471  SUB180 F477  ZERO1  F486  ANG180 F490  JUMP   F49E
BR270  F4AA  SUB270 F4BB  ZERO2  F4CA  ANG270 F4D4  BR360  F4E2
ZERO3  F4FD  ANG360 F507  FLAG2  F514  FLAG3  F523  NEWCYL F535
SUB22  F549  SINTBL F560  SINTB1 F568  SINTB2 F572  SINTB3 F57A
FLAG1  F57B  ANGLE2 F58A  ANGLE  F58D  COSAGL F5AC
```

```
S0 06 0000 48 44 52 1B
S1 13 F300 B6 CA 09 B0 FA 08 25 68 B1 FA 0B 22 03 B6 FA 08 9B
S1 13 F310 B7 CB 01 86 04 B7 CB 02 BD FD A0 B6 CB 03 B7 CA F9
S1 13 F320 52 F6 CB 04 F7 CA 53 7D CA 3A 27 1E B6 CA 17 B7 8A
S1 13 F330 CB 01 F6 CA 54 F7 CB 02 BD FD A0 B6 CB 03 B7 CA C8
S1 13 F340 50 F6 CB 04 F7 CA 51 7E F3 80 F6 CA 18 F7 CB 02 05
S1 13 F350 B6 CA 0A B0 FA 08 25 18 B7 CA 54 B7 CB 01 BD FD 1E
S1 13 F360 A0 F6 CB 03 F7 CA 50 B6 CB 04 B7 CA 51 7E F3 80 DC
S1 06 F370 7E F0 28 FD
S9 03 0000 FC
 ---
```

```
1                  MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE   1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1978 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                       MODIFIED IAMC VER: 1.0
 00001                          NAM    IMPED2
 00002 F300                     ORG    $F300
 00003                  *
 00004                  *********************************
 00005                  *                               *
 00006                  *  IMPEDANCE CALCULATION SUB-RTN  *
 00007                  *                               *
 00008                  *********************************
 00009                  *
 00010 F300 B6 CA09 IMPED  LDA A $CA09      LOAD ACC.A WITH I - MAX
 00011 F303 B0 FA08        SUB A $FA08      ACC.A CONTAINS I PEAK
 00012 F306 25 68          BCS   CHECK
 00013 F308 B1 FA0B        CMP A $FA0B      COMPARE I PEAK WITH $20
 00014 F30B 22 03          BHI   MINIM
 00015 F30D B6 FA08        LDA A $FA08      PUSH $20 AS I - MINIMUM
 00016 F310 B7 CB01 MINIM  STA A $CB01      INITIALISE MUL1RTN
 00017 F313 86 04   JUMP1  LDA A #$04
 00018 F315 B7 CB02        STA A $CB02
 00019 F318 BD FDA0        JSR   MUL1RTN    JUMP TO MUL1RTN
 00020 F31B B6 CB03        LDA A $CB03      LOAD HIH BYTE
 00021 F31E B7 CA52        STA A $CA52      INITIALISE DIVIDE RTN
 00022 F321 F6 CB04        LDA B $CB04      LOAD WITH LOW BYTE
 00023 F324 F7 CA53        STA B $CA53
 00024 F327 7D CA3A IMPED1 TST   $CA3A      TEST FLAG FOR FOR RESISTANCE
 00025 F32A 27 1E          BEQ   RESIST
 00026 F32C B6 CA17        LDA A $CA17      LOAD WITH SIN PHASE ANGLE
 00027 F32F B7 CB01        STA A $CB01
 00028 F332 F6 CA54        LDA B $CA54      LOAD WITH V - PEAK
 00029 F335 F7 CB02        STA B $CB02

--------------------------------------------------------------------------------

 00030 F338 BD FDA0        JSR   MUL1RTN
 00031 F33B B6 CB03        LDA A $CB03
 00032 F33E B7 CA50        STA A $CA50
 00033 F341 F6 CB04        LDA B $CB04
 00034 F344 F7 CA51        STA B $CA51
 00035 F347 7E F380        JMP   DIVIDE
 00036 F34A F6 CA18 RESIST LDA B $CA18      LOAD ACC.B WITH COS PHASE  ..
 00037 F34D F7 CB02        STA B $CB02
 00038 F350 B6 CA0A        LDA A $CA0A
 00039 F353 B0 FA08        SUB A $FA08
 00040 F356 25 18          BCS   CHECK      BRANCH TO STOP, SOME THING IS
 00041 F358 B7 CA54        STA A $CA54      SOTRE V PEAK
 00042 F35B B7 CB01        STA A $CB01
 00043 F35E BD FDA0        JSR   MUL1RTN
 00044 F361 F6 CB03        LDA B $CB03
 00045 F364 F7 CA50        STA B $CA50
 00046 F367 B6 CB04        LDA A $CB04
 00047 F36A B7 CA51        STA A $CA51
 00048 F36D 7E F380        JMP   DIVIDE
 00049 F370 7E F028 CHECK  JMP   $F028
 .0050                  *
 00051 F380                ORG   $F380
 00052 F380 0001   DIVIDE  RMB   1
 00053 FDA0                ORG   $FDA0
 00054 FDA0 0001   MUL1RT  RMB   1
 ---

1 I(P D            MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE   2


 00055                  *
 00056                          END

SYMBOL TABLE

IMPED  F300  MINIM  F310  JUMP1  F313  IMPED1 F327  RESIST F34A
CHECK  F370  DIVIDE F380  MUL1RT FDA0
```

```
S0 06 0000 48 44 52 1B
S1 13 F380 BE FA 02 F6 CA 51 37 B6 CA 50 36 FE FA 09 A6 00 CA
S1 13 F390 E6 01 37 36 34 30 86 01 6D 01 2B 0B 4C 68 02 69 67
S1 13 F3A0 01 2B 04 81 11 26 F5 A7 00 A6 03 E6 04 6F 03 6F 61
S1 13 F3B0 04 E0 02 A2 01 24 07 EB 02 A9 01 0C 20 01 0D 69 5B
S1 13 F3C0 04 69 03 64 01 66 02 6A 00 26 E6 31 31 31 32 33 8E
S1 13 F3D0 7D CA 3A 27 08 B7 CA 32 F7 CA 33 20 0C 7C CA 3A 26
S1 0F F3E0 B7 CA 30 F7 CA 31 7E F3 27 7E F2 60 12
S9 03 0000 FC
 ---

1                  MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  1


           M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
              COPYRIGHT 1974 TO 1978 BY MOTOROLA INC

           MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                     MODIFIED IAMC VER: 1.0

 00001                        NAM   DIVIDE
 00002 F380                   ORG   $F380
 00003                 *
 00004                 ***********************************
 00005                 *                                 *
 00006                 *     SYSTEM DIVIDE SUBROUTINE    *
 00007                 *                                 *
 00008                 ***********************************
 00009                 *
 00010 F380 BE FA02 DIVIDE LDS   $FA02      INITIALISE STACK POINTER
 00011 F383 F6 CA51        LDA B $CA51      LOAD ACC.B WITH DIVIDEND LOW
 00012 F386 37             PSH B            PUSH DIVIDEND LOW BYTE INTO S
 00013 F387 B6 CA50        LDA A $CA50
 00014 F38A 36             PSH A            PUSH DIVIDEND HIGH BYTE INTO
 00015 F38B FE FA09        LDX   $FA09      LOAD $FA09 BY $CA , AND $FA0A
 00016 F38E A6 00          LDA A X          LOAD ACC.A WITH DIVISOR HI-BY
 00017 F390 E6 01          LDA B 1,X        LOAD ACC.B WITH DIVISOR LOW B
 00018 F392 37             PSH B            PUSH DIVISOR INTO STACK (LOWB
 00019 F393 36             PSH A            PUSH DIVISOR INTO STACK (HIGH
 00020 F394 34             DES              DECREMENT STACK POINTER
 00021 F395 30             TSX              STORE STACK POINTER IN INDEX
 00022 F396 86 01          LDA A #01
 00023 F398 6D 01          TST   1,X
 00024 F39A 2B 0B          BMI   DIVB       BRANCH IF MINUS
 00025 F39C 4C      DIVA   INC A
 00026 F39D 68 02          ASL   2,X        ARITHMETIC SHIFT LEFT
 00027 F39F 69 01          ROL   1,X        ROTATE RIGHT
 00028 F3A1 2B 04          BMI   DIVB
 00029 F3A3 81 11          CMP A #017
 00030 F3A5 26 F5          BNE   DIVA
------------------------------------------------------------------------------------------

 00031 F3A7 A7 00   DIVB   STA A 0,X        SAVE COUNT
 00032 F3A9 A6 03          LDA A 3,X
 00033 F3AB E6 04          LDA B 4,X
 00034 F3AD 6F 03          CLR   3,X
 00035 F3AF 6F 04          CLR   4,X
 00036 F3B1 E0 02   DIVC   SUB B 2,X
 00037 F3B3 A2 01          SBC A 1,X
 00038 F3B5 24 07          BCC   DIVD
 00039 F3B7 EB 02          ADD B 2,X        DIVISOR TOO LARGE
 00040 F3B9 A9 01          ADC A 1,X
 00041 F3BB 0C             CLC              CLEAR CARRY
 00042 F3BC 20 01          BRA   DIVE
 00043 F3BE 0D      DIVD   SEC              SET CARRY
 00044 F3BF 69 04   DIVE   ROL   4,X
 00045 F3C1 69 03          ROL   3,X
 00046 F3C3 64 01          LSR   1,X        ADJUST DIVISOR
 00047 F3C5 66 02          ROR   2,X
 00048 F3C7 6A 00          DEC   0,X
 00049 F3C9 26 E6          BNE   DIVC
 00050 F3CB 31             INS              INCREMENT STACK POINTER TO CL
 00051 F3CC 31             INS
 00052 F3CD 31             INS
 00053 F3CE 32             PUL A
 00054 F3CF 33             PUL B

 ---

1 D V D            MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  2


 00055 F3D0 7D CA3A        TST   $CA3A      CHECK FLAGE = ZERO
 00056 F3D3 27 08          BEQ   RESIS
 00057 F3D5 B7 CA32        STA A $CA32      STORE SYSTEM REACTANCE
 00058 F3D8 F7 CA33        STA B $CA33      STORE REACTANCE HIGH BYTE
 00059 F3DB 20 0C          BRA   SIGNXR
 00060 F3DD 7C CA3A RESIS  INC   $CA3A
 00061 F3E0 B7 CA30        STA A $CA30      STORE SYSTEM RESISTANCE
 00062 F3E3 F7 CA31        STA B $CA31
 00063 F3E6 7E F327        JMP   IMPED1
 00064 F3E9 7E F260 SIGNXR JMP   $F260
 00065                *
 00066 F327                ORG   $F327
 00067 F327 0001    IMPED1 RMB   1
 00068                *
 00069                     END

SYMBOL TABLE

DIVIDE F380  DIVA   F39C  DIVB   F3A7  DIVC   F3B1  DIVD   F3BE
DIVE   F3BF  RESIS  F3DD  SIGNXR F3E9  IMPED1 F327
```

```
S0 06 0000 48 44 52 1B
S1 13 F260 81 FA 00 27 07 86 7F B7 CA 33 20 05 F1 FA 0C 22 CA
S1 13 F270 F4 B6 CA 30 81 FA 00 27 07 86 7F B7 CA 31 20 08 2E
S1 13 F280 F6 CA 31 F1 FA 0C 22 F1 7D CA 3B 27 14 B6 FA 08 0A
S1 13 F290 B0 CA 31 B7 CA 31 F6 FA 08 FB CA 33 F7 CA 33 20 09
S1 13 F2A0 44 7D CA 3D 27 14 B6 FA 08 F6 FA 08 B0 CA 31 F0 0C
S1 13 F2B0 CA 33 B7 CA 31 F7 CA 33 20 2B 7D CA 3E 27 14 B6 E6
S1 13 F2C0 FA 08 B0 CA 33 B7 CA 33 F6 FA 08 FB CA 31 F7 CA 28
S1 13 F2D0 31 20 12 B6 FA 08 F6 FA 08 BB CA 31 FB CA 33 B7 B2
S1 08 F2E0 CA 31 F7 CA 33 7E F1 5C 6B
S9 03 0000 FC
 ...

1                 MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                     MODIFIED IAMC VER: 1.0

00001                         NAM   SIGNRX
00002 F260                    ORG   $F260
00003                  *
00004                  ****************************************************
00005                  *
00006                  *   RESISTANCE - REACTANCE SIGN DETERMINATION    *
00007                  *
00008                  ****************************************************
00009                  *
00010 F260 81 FA00 SIGN   CMP A $FA00      CHECK OUOTIENT HIGH BYTE = ZE
00011 F263 27 07          BEQ   LOBYTE     BRANCH TO CHECK LOW - BYTE
00012 F265 86 7F   LIMIT  LDA A #$7F
00013 F267 B7 CA33        STA A $CA33      STORE $7F UPPER LIMIT ON RESI
00014 F26A 20 05          BRA   CHEKX
00015 F26C F1 FA0C LOBYTE CMP B $FA0C      CHECK OUOTIENT LOW - BYTE = S
00016 F26F 22 F4          BHI   LIMIT      BRANCH IF C + Z = 0
00017 F271 B6 CA30 CHEKX  LDA A $CA30
00018 F274 81 FA00        CMP A $FA00
00019 F277 27 07          BEQ   HIBYTE
00020 F279 86 7F   LIMITX LDA A #$7F
00021 F27B B7 CA31        STA A $CA31
00022 F27E 20 08          BRA   SIGN2
00023 F280 F6 CA31 HIBYTE LDA B $CA31
00024 F283 F1 FA0C        CMP B $FA0C
00025 F286 22 F1          BHI   LIMITX
00026 F288 7D CA3B SIGN2  TST   $CA3B      TEST IF COSINE IS NEGATIVE
00027 F28B 27 14          BEQ   SIGN3      BRANCH TO TEST THIRD QUADRANT
00028 F28D B6 FA08        LDA A $FA08

---------------------------------------------------------------------------------------


00029 F290 B0 CA31        SUB A $CA31      R- = $80 - R
00030 F293 B7 CA31        STA A $CA31      OUTPUT RESISTANCE TO D/A CONV
00031 F296 F6 FA08        LDA B $FA08
00032 F299 FB CA33        ADD B $CA33      X+ = $80 + X
00033 F29C F7 CA33        STA B $CA33      OUTPUT REACTANCE TO D/A CONVE
00034 F29F 20 44          BRA   FINISH
00035 F2A1 7D CA3D SIGN3  TST   $CA3D
00036 F2A4 27 14          BEQ   SIGN4      TEST FOR SIGN IN THIRC OUDRAN
00037 F2A6 B6 FA08        LDA A $FA08
00038 F2A9 F6 FA08        LDA B $FA08
00039 F2AC B0 CA31        SUB A $CA31      R- = $80 - R
00040 F2AF F0 CA33        SUB B $CA33      X- = $80 - X
00041 F2B2 B7 CA31        STA A $CA31      OUTPUT RESISTANCE TO D/A
00042 F2B5 F7 CA33        STA B $CA33      OUTPUT REACTANCE TO D/A
00043 F2B8 20 2B          BRA   FINISH     JUMP FOR NEW CYCLE
00044 F2BA 7D CA3E SIGN4  TST   $CA3E      TEST FOR SIGN IN FOURTH OUADR
00045 F2BD 27 14          BEQ   SIGN1      BRANCH TO TEST FOR FIRST OUAD
00046 F2BF B6 FA08        LDA A $FA08
00047 F2C2 B0 CA33        SUB A $CA33      X- = $80 - X
00048 F2C5 B7 CA33        STA A $CA33
00049 F2C8 F6 FA08        LDA B $FA08
00050 F2CB FB CA31        ADD B $CA31      R+ = $80 = R
00051 F2CE F7 CA31        STA B $CA31      OUTPUT RESISTANCE TO D/A CONV
00052 F2D1 20 12          BRA   FINISH
00053 F2D3 B6 FA08 SIGN1  LDA A $FA08
00054 F2D6 F6 FA08        LDA B $FA08

 ...

1 S G+R8          MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  2


00055 F2D9 BB CA31        ADD A $CA31      R+ = $80 + R
00056 F2DC FB CA33        ADD B $CA33      X+ = $80 + X
00057 F2DF B7 CA31        STA A $CA31
00058 F2E2 F7 CA33        STA B $CA33
00059 F2E5 7E F15C FINISH JMP   $F15C
00060                  *
00061                         END

SYMBOL TABLE

SIGN   F260  LIMIT  F265  LOBYTE F26C  CHEKX  F271  LIMITX F279
HIBYTE F280  SIGN2  F288  SIGN3  F2A1  SIGN4  F2BA  SIGN1  F2D3
FINISH F2E5
```

```
S0 06 0000 48 44 52 1B
S1 13 FDA0 C6 08 F7 CB 00 F6 CB 01 4F 74 CB 02 24 01 1B 46 E7
S1 12 FDB0 76 CB 04 7A CB 00 26 F1 B7 CB 03 F6 CB 04 39 1C
S9 03 0000 FC
 ---

1                 MOTOROLA M68SAM CROSS-ASSEMBLER         PAGE  1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                    MODIFIED IAMC VER: 1.0

 00001                          NAM   MUL1RTN
 00002 FDA0                     ORG   $FDA0
 00003                   *
 00004                   ******************************************
 00005                   *
 00006                   *    MULTIPLY RTN   ONE BYTE * ONE BYTE      *
 00007                   *
 00008                   ******************************************
 00009                   *
 00010 FDA0 C6 08   MUL1RT LDA B #$08       PUT 08 IN B REGISTER
 00011 FDA2 F7 CB00        STA B $CB00      SAVE SHIFT COUNT (08)
 00012 FDA5 F6 CB01        LDA B $CB01      LOAD MULTIPLICAND INTO ACC.B
 00013 FDA8 4F             CLR A            CLEAR A
 00014 FDA9 74 CB02 SKIP2  LSR   $CB02      SHIFT MULTIPLIER RIGHT
 00015 FDAC 24 01          BCC   SKIP1      GO TO SKIP 1 IF CARRY CLEAR
 00016 FDAE 1B             ABA              ADD ACC.'S
 00017 FDAF 46      SKIP1  ROR A            SHIFT PRODUCT MS BYTE RIGHT
 00018 FDB0 76 CB04        ROR   $CB04      SHIFT PRODUCT REG.LS BYTE
 00019 FDB3 7A CB00        DEC   $CB00      DECREMENT COUNT
 00020 FDB6 26 F1          BNE   SKIP2      BRANCH IF NOT 0
 00021 FDB8 B7 CB03        STA A $CB03      IF 0 DONE. STORE MS BYTE
 00022 FDBB F6 CB04        LDA B $CB04      LOAD ACC.B WITH PRODUCT LS BY
 00023 FDBE 39             RTS
 00024                   *
 00025                          END

 SYMBOL TABLE

 MUL1RT FDA0  SKIP2  FDA9  SKIP1  FDAF
S0 06 0000 48 44 52 1B
S1 13 FD00 7F CC 05 7F CC 06 7F CC 07 7F CC 08 B6 CC 02 F6 2F
S1 13 FD10 CC 04 F7 CB 01 B7 CB 02 BD FD A0 F7 CC 08 B7 CC 20
S1 13 FD20 07 B6 CC 02 F6 CC 03 F7 CB 01 B7 CB 02 BD FD A0 DE
```

```
S1 13 FD30 FB CC 07 F7 CC 07 89 00 B7 CC 06 B6 CC 04 F6 CC CD
S1 13 FD40 01 F7 CB 01 B7 CB 02 BD FD A0 FB CC 07 F7 CC 07 75
S1 13 FD50 B9 CC 06 B7 CC 06 86 00 89 00 B7 CC 05 B6 CC 01 71
S1 13 FD60 F6 CC 03 F7 CB 01 B7 CB 02 BD FD A0 FB CC 06 F7 65
S1 0C FD70 CC 06 B9 CC 05 B7 CC 05 39 69
S9 03 0000 FC
 ---

1                 MOTOROLA M68SAM CROSS-ASSEMBLER         PAGE  1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                    MODIFIED IAMC VER. 1.0

 00001                          NAM   MUL2RTN
 00002 FD00                     ORG   $FD00
 00003                   *
 00004                   *****************************************
 00005                   *
 00006                   *   TWO BYTE BY TWO BYTE MULTIPLY ROUTINE   *
 00007                   *
 00008                   *****************************************
 00009                   *
 00010 FD00 7F  CC05            CLR   PRODHB    CLEAR PRODUCT REGISTER HIGH B
 00011 FD03 7F  CC06            CLR   PROMHB    CLEAR PRODUCT REGISTER MIDDLE
 00012 FD06 7F  CC07            CLR   PROMLB    CLEAR PRODUCT REGISTER MIDDLE
 00013 FD09 7F  CC08            CLR   PRODLB    CLEAR PRODUCT REGISTER LOW BY
 00014 FD0C B6  CC02            LDA A MULDLB    LOAD ACC.A WITH MULTIPLICAND
 00015 FD0F F6  CC04            LDA B MULRLB    LOAD ACC.B WITH MULTIPLIER LO
 00016 FD12 F7  CB01            STA B MULTD1    STORE MULTIPLICAND IN MUL1RTN
 00017 FD15 B7  CB02            STA A MULTR1    STORE MULTIPLIER
 00018 FD18 BD  FDA0            JSR   MUL1RTN   JUMP TO MUL1RTN SUBROUTINE
 00019 FD1B F7  CC08            STA B PRODLB    STORE PRODUCT LOW BYTE
 00020 FD1E B7  CC07            STA A PROMLB    STORE PRODUCT MIDDLE LOW BYTE
 00021 FD21 B6  CC02            LDA A MULDLB    LOAD ACC.A WITH MULTIPLICAND
 00022 FD24 F6  CC03            LDA B MULRHB    LOAD ACC.B WITH MULTIPLIER HI
 00023 FD27 F7  CB01            STA B MULTD1
 00024 FD2A B7  CB02            STA A MULTR1
 00025 FD2D BD  FDA0            JSR   MUL1RTN
 00026 FD30 FB  CC07            ADD B PROMLB
 00027 FD33 F7  CC07            STA B PROMLB
 00028 FD36 89  00              ADC A #00000
 00029 FD38 B7  CC06            STA A PROMHB    PRODUCT REGISTER HIGH BYTE
 00030 FD3B B6  CC04            LDA A MULRLB
 00031 FD3E F6  CC01            LDA B MULDHB    LOAD ACC.B WITH MULTIPLICAND
 00032 FD41 F7  CB01            STA B MULTD1
 00033 FD44 B7  CB02            STA A MULTR1
```

```
00034 FD47 BD FDA0          JSR    MUL1RTN
00035 FD4A FB CC07          ADD B PROMLB
00036 FD4D F7 CC07          STA B PROMLB
00037 FD50 B9 CC06          ADC A PROMHB
00038 FD53 B7 CC06          STA A PROMHB
00039 FD56 86 00            LDA A #00000
00040 FD58 89 00            ADC A #00000
00041 FD5A B7 CC05          STA A PRODHB     PRODUCT HIGH BYTE
00042 FD5D B6 CC01          LDA A MULDHB
00043 FD60 F6 CC03          LDA B MULRHB
00044 FD63 F7 CB01          STA B MULTD1
00045 FD66 B7 CB02          STA A MULTR1
00046 FD69 BD FDA0          JSR    MUL1RTN
00047 FD6C FB CC06          ADD B PROMHB
00048 FD6F F7 CC06          STA B PROMHB
00049 FD72 B9 CC05          ADC A PRODHB
00050 FD75 B7 CC05          STA A PRODHB
00051 FD78 39               RTS              RETURNE FROM JSR
00052                *
00053 CC01                  ORG    $CC01
00054 CC01 0001      MULDHB RMB    1

...

1 MSL R4N        MOTOROLA M68SAM CROSS-ASSEMBLER        PAGE  2


00055 CC02 0001      MULDLB RMB    1
00056 CC03 0001      MULRHB RMB    1
00057 CC04 0001      MULRLB RMB    1
00058 CC05 0001      PRODHB RMB    1
00059 CC06 0001      PROMHB RMB    1
00060 CC07 0001      PROMLB RMB    ·1
00061 CC08 0001      PRODLB RMB    1
00062 CB01                  ORG    $CB01
00063 CB01 0001      MULTD1 RMB    1
00064 CB02 0001      MULTR1 RMB    1
00065 FDA0                  ORG    $FDA0
00066 FDA0 0001      MUL1RT RMB    1
00067                *
00068                       END

SYMBOL TABLE

MULDHB CC01  MULDLB CC02  MULRHB CC03  MULRLB CC04  PRODHB CC05
PROMHB CC06  PROMLB CC07  PRODLB CC08  MULTD1 CB01  MULTR1 CB02
MUL1RT FDA0
```

```
S0 06 0000 48 44 52 1B
S1 07 F15C FE C9 70 86 EE
S1 13 F160 C9 B7 C9 70 86 5E B7 C9 71 F6 CA 31 54 54 54 F7 29
S1 13 F170 C9 51 B6 C9 4D F6 C9 58 10 F6 C9 51 1B B7 C9 4D 86
S1 13 F180 F6 CA 33 54 54 54 F7 C9 59 B6 C9 4C F6 C9 60 10 79
S1 13 F190 F6 C9 59 1B B7 C9 4C 7F C9 62 A6 01 7C C9 62 A7 CD
S1 13 F1A0 02 09 B6 C9 62 81 0F 27 02 20 EF B6 C9 4D F6 C9 1C
S1 0D F1B0 4C F7 D0 02 B7 E0 02 7E F0 2B 0A
S9 03 0000 FC
 ...

1                MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                     MODIFIED IAMC VER: 1.0

 00001                        NAM   AVERAGE1
 00002 F15C                   ORG   $F15C
 00003                  *
 00004                  ********************************************
 00005                  *
 00006                  *    R - X ERROR FILTERING RTN 8 SAMPLES      *
 00007                  *
 00008                  ********************************************
 00009                  *
 00010 F15C FE C970             LDX   $C970      Initialize IX
 00011 F15F 86 C9               LDA A #$C9
 00012 F161 B7 C970             STA A $C970      Push $C9 into $C970
 00013 F164 86 5E               LDA A #$5E
 00014 F166 B7 C971             STA A $C971      Push $5E into $C971
 00015 F169 F6 CA31             LDA B $CA31      Get R - last calc.
 00016 F16C 54                  LSR B            divide R by 2
 00017 F16D 54                  LSR B
 00018 F16E 54                  LSR B
 00019 F16F F7 C951             STA B $C951      Save R/8
 00020 F172 B6 C94D             LDA A $C94D       Get R oldest sample
 00021 F175 F6 C958             IDA B $C958      Get R last output
 00022 F178 10                  SBA              subtr. old. sample
 00023 F179 F6 C951             LDA B $C951       Get newest sample
 00024 F17C 1B                  ABA               Add to R - out
 00025 F17D B7 C94D             STA A $C94D       Store R - out
 00026 F180 F6 CA33             LDA B $CA33       Get Last X calc.
 00027 F183 54                  LSR B             Divide by 2
 00028 F184 54                  LSR B
 00029 F185 54                  LSR B
 00030 F186 F7 C959             STA B $C959        save X - Sample
 00031 F189 B6 C94C             LDA A $C94C        Get X - last outp.
 00032 F18C F6 C960             LDA B $C960        Get X oldet samples

-------- ---------------------------------------------------------------------------------


 00033 F18F 10                  SBA                subt. oldest samples
 00034 F190 F6 C959             LDA B $C959        get newest sample
 00035 F193 1B                  ABA                Add to X - out
 00036 F194 B7 C94C             STA A $C94C        Save X - out
 00037 F197 7F C962             CLR   $C962        clear sample cnt.
 00038 F19A A6 01   SHIFT       LDA A $1,X           incr  IX
 00039 F19C 7C C962             INC   $C962
 00040 F19F A7 02               STA A $2,X
 00041 F1A1 09                  DEX
 00042 F1A2 B6 C962             LDA A $C962
 00043 F1A5 81 0F               CMP A #$0F            finish 16 samples
 00044 F1A7 27 02               BEQ   OUTPUT
 00C45 F1A9 20 EF               BRA   SHIFT
 00046 F1AB B6 C94D OUTPUT      LDA A $C94D
 00047 F1AE F6 C94C             LDA B $C94C
 00048 F1B1 F7 D002             STA B $D002          R - output
 00049 F1B4 B7 E002             STA A $E002          X - output
 00050 F1B7 7E F02B             JMP   CYCLE          new cycle
 00051                  *
 00052 F02B                     ORG   $F02B
 00053 F02B 0001        CYCLE   RMB   1
 00054                  *

 ...

1 A6E2A E          MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  2


 00055                          END

 SYMBOL TABLE

 SHIFT   F19A   OUTPUT F1AB   CYCLE   F02B
```

```
S0 06 0000 48 44 52 1B
S1 13 F200 00 04 09 0D 12 16 1B 1F 24 28 2C 31 35 39 3E 42 E7
S1 13 F210 46 4B 4F 53 57 5C 60 64 68 6C 70 74 78 7C 80 84 90
S1 13 F220 88 8B 8F 93 96 9A 9D A1 A4 A8 AB AE B2 B5 B8 BB B8
S1 13 F230 BE C1 C4 C6 CA CC CF D2 D4 D7 D9 DB DE DF E2 E4 A8
S1 13 F240 E6 E8 EA EC ED EF F0 F2 F3 F5 F6 F7 F8 F9 FA FB 9D
S1 0F F250 FC FD FD FE FE FF FF FF FF FF FF D0 F2
S1 13 FA00 00 00 CF FF 00 E7 01 5A 80 CA 52 20 7F 5A F1 00 5C
S1 05 FA10 C9 4A DD
S1 13 F100 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 83
S1 13 F110 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 73
S1 13 F120 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 63
S1 13 F130 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 53
S1 13 F140 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 43
S1 0E F150 50 51 52 53 54 55 56 57 58 59 5A 09
S9 03 0000 FC
 ...

1                MOTOROLA M68SAM CROSS-ASSEMBLER           PAGE  1


          M68SAM IS THE PROPERTY OF MOTOROLA SPD. INC.
             COPYRIGHT 1974 TO 1976 BY MOTOROLA INC

          MOTOROLA M6800 CROSS ASSEMBLER, RELEASE 1.3
                    MODIFIED IAMC VER: 1.0

 00001                        NAM   CONST
 00002 F200                   ORG   $F200
 00003                  *
 00004                  ***********************************
 00005                  *
 00006                  *   SINE - COSINE TABLE CONSTANTS     *
 00007                  *
 00008                  ***********************************
 00009                  *
 00010 F200 00                FCB   $00,$04,$09,$0D,$12,$16,$1B,$1F,$24,$28
       F201 04
       F202 09
       F203 0D
       F204 12
       F205 16
       F206 1B
       F207 1F
       F208 24
       F209 28
 00011 F20A 2C                FCB   $2C,$31,$35,$39,$3E,$42,$46,$4B,$4F,$53
       F20B 31
       F20C 35
       F20D 39
       F20E 3E

---------------------------------------------------------------------------------------

       F20F 42
       F210 46
       F211 4B
       F212 4F
       F213 53
 00012 F214 57                FCB   $57,$5C,$60,$64,$68,$6C,$70,$74   120.
       F215 5C
       F216 60
       F217 64
       F218 68
       F219 6C
       F21A 70
       F21B 74
 00013 F21C 78                FCB   $78,$7C,$80,$84,$88,$8B,$8F,$93,$96,$9A
       F21D 7C
       F21E 80
       F21F 84
       F220 88
       F221 8B
       F222 8F
       F223 93
       F224 96
       F225 9A
 00014 F226 9D                FCB   $9D,$A1,$A4,$A8,$AB,$AE,$B2,$B5,$B8,$BB
       F227 A1
       F228 A4
       F229 A8
       F22A AB
       F22B AE
       F22C B2

 ...

1 C¦N3T          MOTOROLA M68SAM CROSS-ASSEMBLER           PAGE  2


       F22D B5
       F22E B8
       F22F BB
 00015 F230 BE                FCB   $BE,$C1,$C4,$C6,$CA,$CC,$CF,$D2,$D4,$D7
       F231 C1
       F232 C4
       F233 C6
       F234 CA
       F235 CC
       F236 CF
       F237 D2
       F238 D4
       F239 D7
```

```
00016 F23A D9                    FCB   $D9,$DB,$DE,$DF,$E2,$E4,$E6,$E8,$EA,$EC
      F23B DB
      F23C DE
      F23D DF
      F23E E2
      F23F E4
      F240 E6
      F241 E8
      F242 EA
      F243 EC
00017 F244 ED                    FCB   $ED,$EF,$F0,$F2,$F3,$F5,$F6,$F7,$F8,$F9
      F245 EF
      F246 F0
      F247 F2
      F248 F3
      F249 F5
      F24A F6
      F24B F7
      F24C F8
      F24D F9
00018 F24E FA                    FCB   $FA,$FB,$FC,$FD,$FD,$FE,$FE,$FF,$FF,$FF
      F24F FB
      F250 FC
      F251 FD
      F252 FD
      F253 FE
      F254 FE
      F255 FF
      F256 FF
      F257 FF
00019 F258 FF                    FCB   $FF,$FF,$FF,$D0
      F259 FF
      F25A FF
      F25B D0
00020 FA00                       ORG   $FA00
00021 FA00 00                    FCB   $00,$00,$CF,$FF,$00,$E7,$01,$5A,$80   21
      FA01 00
      FA02 CF
      FA03 FF
      FA04 00
      FA05 E7
      FA06 01
      FA07 5A
      FA08 80
00022 FA09 CA                    FCB   $CA,$52,$20,$7F,$5A,$F1,$00,$C9,$4A   22
      FA0A 52
      FA0B 20

...

1 C¦N3T            MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  3

-----------------------------------------------------------------------------------


      FA0C 7F
      FA0D 5A
      FA0E F1
      FA0F 00
      FA10 C9
      FA11 4A
00023 F100                       ORG   $F100
00024 F100 00                    FCB   00,01,02,03,04,05,06,07,08,09   240.
      F101 01
      F102 02
      F103 03
      F104 04
      F105 05
      F106 06
      F107 07
      F108 08
      F109 09
00025 F10A 0A                    FCB   10,11,12,13,14,15,16,17,18,19   250.
      F10B 0B
      F10C 0C
      F10D 0D
      F10E 0E
      F10F 0F
      F110 10
      F111 11
      F112 12
      F113 13
00026 F114 14                    FCB   20,21,22,23,24,25,26,27,28,29   260.
      F115 15
      F116 16
      F117 17
      F118 18
      F119 19
      F11A 1A
      F11B 1B
      F11C 1C
      F11D 1D
00027 F11E 1E                    FCB   30,31,32,33,34,35,36,37,38,39   270.
      F11F 1F
      F120 20
      F121 21
      F122 22
      F123 23
      F124 24
      F125 25
      F126 26
      F127 27
00028 F128 28                    FCB   40,41,42,43,44,45,46,47,48,49   280.
      F129 29
```

```
          F12A  2A
          F12B  2B
          F12C  2C
          F12D  2D
          F12E  2E
          F12F  2F
          F130  30
          F131  31
   00029  F132  32                FCB    50,51,52,53,54,55,56,57,58,59  290.
          F133  33
          F134  34

 ---

1 C¦N3T            MOTOROLA M68SAM CROSS-ASSEMBLER          PAGE  4


          F135  35
          F136  36
          F137  37
          F138  38
          F139  39
          F13A  3A
          F13B  3B
   00030  F13C  3C                FCB    60,61,62,63,64,65,66,67,68,69  300.
          F13D  3D
          F13E  3E
          F13F  3F
          F140  40
          F141  41
          F142  42
          F143  43
          F144  44
          F145  45
   00031  F146  46                FCB    70,71,72,73,74,75,76,77,78,79  310.
          F147  47
          F148  48
          F149  49
          F14A  4A
          F14B  4B
          F14C  4C
          F14D  4D
          F14E  4E
          F14F  4F
   00032  F150  50                FCB    80,81,82,83,84,85,86,87,88,89  320.
          F151  51
          F152  52
          F153  53
          F154  54
          F155  55
          F156  56

-----  -----------------------------------------------------------------------

          F157  57
          F158  58
          F159  59
   00033  F15A  5A                FCB    90
   00034                  *
   00035                          END
```

Object Programme

```
        NAM    PEAKPICK
        ORG    $F000
*
*******************************************************
*                                                     *
*   PEAK PICKING ROUTINE & PHASE ANGLE CALCULATION    *
*                                                     *
*******************************************************
*
        LDX    #$04
        STX    $D000        INIT. PIA2 INPUT REGISTERS
        LDX    #$FF04       LOAD IND. REG  WITH HEX. $FF04
        STX    $D002        INIT. PIA2 OUTPUT REGISTERS
        LDX    #$004        LOAD IND. REGISTER WITH HEX. $004
        STX    $E000        INIT. PIA1 INPUT REGISTERS
        LDX    #$FF04       LOAD INDEX REGISTER WITH $FF04
        STX    $E002        INIT. PIA1 OUTPUT REGISTERS
        LDX    $FA10        LOAD IX WITH STARTING ADDRESS ( $C94A )
        CLR    $C961        CLEAR BYTE COUNT
CLEAR   CLR    0,X          CLEAR LOCATION INDICATED BY IX
        INX
        INC    $C961        INCREMENT COUNTEP
        LDAA   $C961        LOAD ACC.A WITH COUNT VALUE
        CMPA   #$17
        BMI    CLEAR        BRANCH TO CLEAR REST OF TABLE
BIGIN   LDS    $FA02        INITIALISE STACK POINTER ( $CFFF )
        LDX    $FA00        CLEAR INDEX REGISTER
PIA2VL  LDAA   $D000        LOAD ACC.A WITH INST. VALUE OF V
        CMPA   $FA08        COMPARE ACC.A WITH HEX. $80
        BMI    PIA2VH       BRANCH IF MINUS
        BRA    PIA2VL       BRANCH ALWAYS
PIA2VH  LDAA   $D000        LOAD ACC.A WITH INST  VALUE OF V
        CMPA   $FA08        COMPARE ACC.A WITH HEX. $80
        BPL    PIA1IL       BRANCH TO CHECK I VALUE
        BRA    PIA2VH       BRANCH BACK FOR V CHECK
PIA1IL  LDAB   $E000        LOAD ACC.B BY INST.VALUE OF I
        CMPB   $FA08        COMPARE VALUE OF I WITH HEX. $80
        BMI    PIA1IH       BRANCH IF MINUS
        INX                 INCREMENT INDEX REGISTER
        BRA    PIA1IL       BRANCH ALWAYS
PIA1IH  LDAB   $E000        LOAD ACC.B WITH INST. VALUE OF I
        CMPB   $FA08        COMPARE INST  VALUE OF I WITH $80
        BPL    STX1         BRANCH TO STORE VALUE OF (N)
        INX                 INCREMENT N ... N=N+1
        BRA    PIA1IH       BRANCH ALWAYS
STX1    STX    $CA02        STORE VALUE OF (N)
        LDX    $FA06        LOAD IND  REG  BY $015A
        NOP
        NOP
DEX1    DEX                 DECREMENT INDEX. REGISTER
        DEX                 DECREMENT IND  REGISTER
        BNE    DEX1         BRANCH TO DECREMENT IX LOOP


-------------------------------------------------------------------------------


        LDAB   $E000        LOAD ACC.B WITH 'I' MAX
        STAB   $CA09        STORE MAX. VALUE OF I
PIA2V1  LDAA   $D000        LOAD ACC.A WITH INST  VALUE OF V
        CMPA   $FA08        COMPARE INST  VALUE OF V WITH $80
        BMI    PIA2V2       BRANCH TC CHECK V>$80
        BRA    PIA2V1       BRANCH ALWAYS
PIA2V2  LDAA   $D000        LOAD ACC.A WITH INST. VALUE OF V
        CMPA   $FA08        COMPARE INST. VALUE OF V WITH $80
        BPL    LDX1         BRANCH IF V > $80
        BRA    PIA2V2       BRANCH ALWAYS
LDX1    LDX    $FA04        LOAD INDEX REGISTER WITH $0E7
DEX2    NOP                 NO OPERATION
        DEX                 DECREMENT INDEX REGISTER
        DEX                 DECREMENT INDEX REGISTER
        DEX                 DECREMENT INDEX REGISTER
        BNE    DEX2
        LDAA   $D000        LOAD ACC.A WITH V MAX
        STAA   $CA0A        STORE MAX  VALUE OF V
        LDAA   #$6E         LOAD ACC.A WITH A CONSTANT $6E
        STAA   $CC01       INITIALISE MUL2RTN
        LDAB   #$90
        STAB   $CC02
        LDAA   $CA02
        STAA   $CC03
        LDAB   $CA03
        STAB   $CC04
        JSR    MUL2RTN
        LDAA   $CC06
        STAA   $CA03
        LDAB   $CC05
        STAB   $CA02
        JMP    SINSTR         JUMP TO SINCOS SUBROUTINE
*
        ORG    $FD00
MUL2RTN RMB    1
        ORG    $F400
SINSTR  RMB    1
        END
```

```
         NAM    SINCOS
         ORG    $F400
*********************************************
*                                           *
*  SINE-COSINE PHASE SHIFT ANGLE SUBROUTINE *
*                                           *
*********************************************
*
SINSTR LDS    $FA02       INITIALISE STACK POINTER
       LDAB   $FA0D
       STAB   $CA0F       STORE $D0 ( N=90 DEGREES) FIXED DATA
       CLR    $CA0E       CLEAR MEMORY LOCATION
       CLR    $CA13       CLEAR MEMORY LOCATION
       CLR    $CA1C       CLEAR MEMORY LOCATION
       CLR    $CA1D       CLEAR MEMORY LOCATION
       CLR    $CA1E       CLEAR MEMORY LOCATION
       CLR    $CA1F       CLEAR MEMORY LOCATION
       CLR    $CA24       CLEAR SINE-COSINE LOOP FLAG
       CLR    $CA3A       CLEAR RESISTANCE - REACTANCE FLAG
       CLR    $CA3B       CLEAR SEC.QUADRANT COSINE FLAG
       CLR    $CA3C
       CLR    $CA3D
       CLR    $CA3E
       CLR    $CA3F
       CLR    $CA42
SUB90  LDAA   $CA02       LOAD ACC.A WITH ( N HIGH BYTE )
       STAA   $CA0B       STORE PHASE SHIFT (HIGH BYTE) IN SUBRTN
       LDAB   $CA03       LOAD ACC.B WITH ( N LOW BYTE )
       STAB   $CA0C       STORE PHASE SHIFT (LOW BYTE) IN SUBRTN
       JSR    SUB22       JUMP TO SUBTRACT SUBROUTINE
       BCS    SUBEND      BRANCH IF ANGLE IS < 90 DEGREES
       LDAA   $CA11
       CMPA   $FA00
       BEQ    ZERO
       BRA    BR180
ZERO   LDAB   $CA12
       CMPB   $FA00
       BEQ    ANG90
       BRA    BR180
ANG90  CLRA
       STAA   $CA18       STORE COSINE 90 DEGREES
       COMA
       STAA   $CA17       STORE SINE 90 DEGREES
       JMP    $F300       JUMP TO IMPEDANCE RTN
BR180  STAA   $CA0B       RE-INITIALISE SUBTRACT ROUTINE
       LDAB   $CA12
       STAB   $CA0C       RE- INITIALISE SUBTRACT ROUTINE
       STAB   $CAAA       SAVE ANGLE VALUE FOR FUTURE USE
       BRA    SUB180      BRANCH TO SUBTACT 90 DEGREES
SUBEND LDAA   $CA03       LOAD ACC.A WITH PHASE ANGLE (N)
       JMP    ANGLE
SUB180 JSR    SUB22       JUMP TO SUBTRACT ROUTINE
```

```
------------------------------------------------------------------------
```

```
       BCS    JUMP
       LDAA   $CA11
       CMPA   $FA00
       BEQ    ZERO1
       BRA    BR270
ZERO1  LDAB   $CA12
       CMPB   $FA00
       BEQ    ANG180
       BRA    BR270
ANG180 CLRB
       STAB   $CA17
       COMB
       STAB   $CA18
       INC    $CA3B
       JMP    $F300         JMP TO IMPEDANCE RTN
JUMP   LDAB   $CAAA       PHASE ANG.( 90 < ANGLE < 180 )
       LDAA   #$5A
       SBA
       INC     $CA3B       FLAG FOR COSINE IS NEGATIVE
       JMP    ANGLE
BR270  STAA   $CA0B       RE- INITIALISE SUBTRACT ROUTINE
       LDAB   $CA12
       STAB   $CA0C       RE- INITIALISE SUBTRACT ROUTINE
       STAB   $CAAB       SAVE ANGLE VALUE FOR FUTURE USE
       BRA    SUB270      BRANCH TO SUBTRACT 90 DEGREES
       NOP
       NOP
       NOP
SUB270 JSR    SUB22
       BCS    FLAG2
       LDAA   $CA11
       CMPA   $FA00
       BEQ    ZERO2
       BRA    BR360
ZERO2  LDAB   $CA12
       CMPB   $FA00
       BEQ    ANG270
       BRA    BR360       270 < ANGLE
ANG270 CLRA
       STAA   $CA18
       COMA
       STAA   $CA17
       INC    $CA3D        SIN 270 DGREES IS NEGATIVE
       JMP    $F300
BR360  STAA   $CA0B       RE- INITIALISE SUBTRACT ROUTINE
       LDAB   $CA12
       STAB   $CA0C       RE - INITIALISE SUBTRACT ROUTINE
       STAB   $CAAC       SAVE ANGLE VALUE FOR FUTURE USE
       JSR    SUB22       JUMP TO SUBTRACT ROUTINE
       BCS    FLAG3
       LDAA   $CA11        ANGLE = 360
       CMPA   $FA00
```

```
        BEQ    ZERO3
        BRA    NEWCYL
ZERO3   LDAB   $CA12
        CMPB   $FA00
        BEQ    ANG360
        BRA    NEWCYL
ANG360  CLRB
        STAB   $CA17
        COMB
        STAB   $CA18
        JMP    $F300
        BRA    NEWCYL          BRANCH TO NEW CYCLE
FLAG2   INC    $CA3D           SET FLAG, SIN IS NEGATIVE, 3RD QUADRANT
        INC    $CA3C           SET FLAG, COS IS NEGATIVE, 3RD QUADRANT
        LDAB   $CAAB
        STAB   $CA12           180 < PHASE SHIFT < 270
        JMP    ANGLE2
FLAG3   INC    $CA3E           SET FLAG, SIN IS NEGATIVE, 4TH QUADRANT
        INC    $CA3F           SET FLAG     MAY BE NOT NEEDED ?
        LDAB   $CAAC
        LDAA   #$5A
        SBA
        STAA   $CA12
        JMP    ANGLE2
NEWCYL  CLRA
        TAP                    PUSH VALUE IN ACC.A INTO C.C.R.
        CLR    $CA24           INCREMENT SIN-COS FLAG
        LDAA   $CA12
        STAA   $CA03
        LDAB   $CA11
        STAB   $CA02
        JMP    SUB90           JUMP TO START A NEW CYCLE
SUB22   CLC                    CLEAR CARRY BIT
        LDAA   $CA0C           LOAD ACC.A WITH DATA IN $CA0C
        LDAB   $CA0B           LOAD ACC.B WITH DATA IN $FA0B
        SUBA   $CA0F           ACC.A - $CA0F
        SBCB   $CA0E           SUBTRACT WITH CARRY, ACC.B - $CA0E
        STAA   $CA12
        STAB   $CA11
        RTS                    RETURN FROM SUBROUTINE
        NOP
        NOP
        NOP
SINTBL  LDAB   #$D0            LOAD ACC.B WITH END OF TABLE
        LDX    $FA0E           LOAD IX WITH SIN-COS TABLE STARTING ADDRESS
        LDAA   $CA43           LOAD AAC.A WITH DESIRED VALUE OF ANGLE
SINTB1  CMPA   00,X
        BNE    SINTB2
        LDX    #$FF,X
        STX    $CA40            $CA41 CONTAINS SINE THE ANGLE
        RTS
SINTB2  CMPB   ,X

------------------------------------------------------------------------

        BEQ    SINTB3
        INX                     INCREMENT INDEX REGISTER
        JMP    SINTB1
SINTB3  WAI                     WAIT
FLAG1   INC    $CA3B
        INC    $CA24            INCREMENT SIN-COS FLAG
        LDAB   $CAAA
        LDAA   #$5A
        SBA
        STAA   $CA12
ANGLE2  LDAA   $CA12            LOAD ACC.A WITH PHASE
ANGLE   STAA   $CA43            $CA43 CONTAINS DESIRED VALUE
        JSR    SINTBL           JUMP FOR SIN-TABLE SEARCH
        TST    $CA24            TEST IF $CA24 CONTAINS ZERO
        BNE    COSAGL           BRANCH TO STORE COSINE ANGLE
        LDAB   $CA41            LOAD ACC.A BY SINE PHASE ANGLE
        STAB   $CA17            STORE SINE PHASE ANGLE IN MUL1RTN
        LDAA   #$5A
        SUBA   $CA43
        STAA   $CA43
        INC    $CA24            INCREMENT $CA24
        JSR    SINTBL           JUMP TO SIN TABLE ROUTINE
COSAGL  STX    $CA43            STORE COSINE PHASE ANGLE
        LDAB   $CA44            LOAD ACC.B WITH COSINE PHASE ANGLE
        STAB   $CA18            STORE PHASE ANGLE
        JMP    $F300            JUMP TO IMPEDANCE RTN
*
        ORG    $F400
SINSTR  RMB    1
*
        END
        NAM    IMPED2
        ORG    $F300
*
*********************************
*                               *
*  IMPEDANCE CALCULATION SUB-RTN  *
*                               *
*********************************
*
IMPED   LDAA   $CA08      LOAD ACC.A WITH I - MAX
        SUBA   $FA08      ACC.A CONTAINS I PEAK
        BCS    CHECK
        CMPA   $FA0B      COMPARE I PEAK WITH $20
        BHI    MINIM
        LDAA   $FA0B       PUSH $20 AS I - MINIMUM
MINIM   STAA   $CB01      INITIALISE MUL1RTN
JUMP1   LDAA   #$04
        STAA   $CB02
        JSR    MUL1RTN     JUMP TO MUL1RTN
        LDAA   $CB03      LOAD HIH BYTE
        STAA   $CA52      INITIALISE DIVIDE RTN
        LDAB   $CB04      LOAD WITH LOW BYTE
        STAB   $CA53
IMPED1  TST    $CA3A       TEST FLAG FOR FOR RESISTANCE OR REACTANCE
        BEQ    RESIST
        LDAA   $CA17       LOAD WITH SIN PHASE ANGLE
        STAA   $CB01
        LDAB   $CA54       LOAD WITH V - PEAK
        STAB   $CB02
        JSR    MUL1RTN
        LDAA   $CB03
        STAA   $CA50
        LDAB   $CB04
        STAB   $CA51
        JMP    DIVIDE
RESIST  LDAB   $CA18      LOAD ACC.B WITH COS PHASE ANGLE
        STAB   $CB02
        LDAA   $CA04
        SUBA   $FA08
        BCS    CHECK      BRANCH TO STOP, SOME THING IS WRONG?
        STAA   $CA54      SOTRE V PEAK
        STAA   $CB01
        JSR    MUL1RTN
        LDAB   $CB03
        STAB   $CA50
        LDAA   $CB04
        STAA   $CA51
        JMP    DIVIDE
CHECK   JMP    $F02B
*
        ORG    $F380
DIVIDE  RMB    1

----------------------------------------------------------------------------

        ORG    $FDA0
MUL1RTN RMB    1
*
        END
        NAM    DIVIDE
        ORG    $F380
*
***********************************
*                                 *
*     SYSTEM DIVIDE SUBROUTINE    *
*                                 *
***********************************
*
DIVIDE  LDS    $FA02      INITIALISE STACK POINTER
        LDAB   $CA51       LOAD ACC.B WITH DIVIDEND LOW BYTE
        PSHB              PUSH DIVIDEND LOW BYTE INTO STACK
        LDAA   $CA50
        PSHA               PUSH DIVIDEND HIGH BYTE INTO STACK
        LDX    $FA09      LOAD $FA09 BY $CA , AND $FA0A BY $52
        LDAA   X          LOAD ACC.A WITH DIVISOR HI-BYTE
        LDAB   1,X        LOAD ACC.B WITH DIVISOR LOW BYTE
        PSHB              PUSH DIVISOR INTO STACK (LOWBYTE)
        PSHA              PUSH DIVISOR INTO STACK (HIGH BYTE)
        DES               DECREMENT STACK POINTER
        TSX               STORE STACK POINTER IN INDEX REG.
        LDAA   #01
        TST    1,X
        BMI    DIVB        BRANCH IF MINUS
DIVA    INCA
        ASL    2,X         ARITHMETIC SHIFT LEFT
        ROL    1,X         ROTATE RIGHT
        BMI    DIVB
        CMPA   #017
        BNE    DIVA
DIVB    STAA   0,X          SAVE COUNT
        LDAA   3,X
        LDAB   4,X
        CLR    3,X
        CLR    4,X
DIVC    SUBB   2,X
        SBCA   1,X
        BCC    DIVD
        ADDB   2,X            DIVISOR TOO LARGE
        ADCA   1,X
        CLC                   CLEAR CARRY
        BRA    DIVE
DIVD    SEC                   SET CARRY
DIVE    ROL    4,X
        ROL    3,X
        LSR    1,X             ADJUST DIVISOR
        ROR    2,X
        DEC    0,X
        BNE    DIVC
        INS                    INCREMENT STACK POINTER TO CLEAN UP
        INS
        INS
        PULA
        PULB
        TST    $CA3A      CHECK FLAGE = ZERO
        BEQ    RESIS
        STAA   $CA32       STORE SYSTEM REACTANCE
        STAB    $CA33      STORE REACTANCE HIGH BYTE
        BRA     SIGNXR
RESIS   INC     $CA3A
        STAA    $CA30       STORE SYSTEM RESISTANCE
        STAB    $CA31
        JMP     IMPED1
SIGNXR  JMP     $F260
*
        ORG     $F327
IMPED1  RMB     1
*
        END
        NAM    SIGNRX
        ORG    $F260
*
**********************************************
*                                            *
*   RESISTANCE - REACTANCE SIGN DETERMINATION   *
*                                            *
**********************************************
*
SIGN    CMPA   $FA00      CHECK QUOTIENT HIGH BYTE = ZERO
        BEQ    LOBYTE     BRANCH TO CHECK LOW - BYTE
LIMIT   LDAA   #$7F
        STAA   $CA33       STORE $7F UPPER LIMIT ON RESISTANCE VALUE
        BRA    CHEKX
LOBYTE  CMPB   $FA0C       CHECK QUOTIENT LOW - BYTE = $7F
        BHI    LIMIT       BRANCH IF C + Z = 0
CHEKX   LDAA   $CA30
        CMPA   $FA00
        BEQ    HIBYTE
LIMITX  LDAA   #$7F
        STAA   $CA31
        BRA    SIGN2
HIBYTE  LDAB   $CA31
        CMPB   $FA0C
        BHI    LIMITX
SIGN2   TST    $CA3B      TEST IF COSINE IS NEGATIVE
        BEQ    SIGN3      BRANCH TO TEST THIRD QUADRANT SIGN
        LDAA   $FA08
        SUBA   $CA31      R- = $80 - R
        STAA   $CA31      OUTPUT RESISTANCE TO D/A CONVERTER
        LDAB   $FA08
        ADDB   $CA33      X+ = $80 + X
        STAB   $CA33      OUTPUT REACTANCE TO D/A CONVERTER

-----------------------------------------------------------------------------------------

        BRA    FINISH
SIGN3   TST    $CA3D
        BEQ    SIGN4      TEST FOR SIGN IN THIRD QUDRANT
        LDAA   $FA08
        LDAB   $FA08
        SUBA   $CA31       R- = $80 - R
        SUBB   $CA33       X- = $80 - X
        STAA   $CA31       OUTPUT RESISTANCE TO D/A
        STAB   $CA33       OUTPUT REACTANCE TO D/A
        BRA    FINISH      JUMP FOR NEW CYCLE
SIGN4   TST    $CA3E       TEST FOR SIGN IN FOURTH QUADRANT
        BEQ    SIGN1       BRANCH TO TEST FOR FIRST QUADRANT
        LDAA   $FA08
        SUBA   $CA33       X- = $80 - X
        STAA   $CA33
        LDAB   $FA08
        ADDB   $CA31       R+ = $80 = R
        STAB   $CA31       OUTPUT RESISTANCE TO D/A CONVERTER
        BRA    FINISH
SIGN1   LDAA   $FA08
        LDAB   $FA08
        ADDA   $CA31        R+ = $80 + R
        ADDB   $CA33        X+ = $80 + X
        STAA   $CA31
        STAB   $CA33
FINISH  JMP    $F15C
*
          END
        NAM    MUL1RTN
        ORG    $FDA0
*
******************************************
*                                        *
*    MULTIPLY RTN   ONE BYTE * ONE BYTE      *
*                                        *
******************************************
*
MUL1RTN LDAB #$08       PUT 08 IN B REGISTER
        STA B $CB00     SAVE SHIFT COUNT (08)
        LDA B $CB01     LOAD MULTIPLICAND INTO ACC.B
        CLR A           CLEAR A
SKIP2   LSR    $CB02    SHIFT MULTIPLIER RIGHT
        BCC    SKIP1    GO TO SKIP 1 IF CARRY CLEAR
        ABA             ADD ACC,'S
SKIP1   ROR A           SHIFT PRODUCT MS BYTE RIGHT
        ROR    $CB04    SHIFT PRODUCT REG.LS BYTE
        DEC    $CB00    DECREMENT COUNT
        BNE    SKIP2    BRANCH IF NOT 0
        STA A $CB03     IF 0 DONE. STORE MS BYTE
        LDAB   $CB04    LOAD ACC.B WITH PRODUCT LS BYTE
        RTS
*
        END
        NAM    MUL2RTN
        ORG    $FDOO
*
*****************************************
*                                       *
*    TWO BYTE BY TWO BYTE MULTIPLY ROUTINE   *
*                                       *
*****************************************
*
        CLR    PRODHB     CLEAR PRODUCT REGISTER HIGH BYTE
        CLR    PROMHB     CLEAR PRODUCT REGISTER MIDDLE HIGH BYTE
        CLR    PROMLB     CLEAR PRODUCT REGISTER MIDDLE LOW BYTE
        CLR    PRODLB     CLEAR PRODUCT REGISTER LOW BYTE
        LDAA   MULDLB     LOAD ACC.A WITH MULTIPLICAND LOW BYTE
        LDAB   MULRLB     LOAD ACC.B WITH MULTIPLIER LOW BYTE
        STAB   MULTD1     STORE MULTIPLICAND IN MUL1RTN
        STAA   MULTR1     STORE MULTIPLIER
        JSR    MUL1RTN    JUMP TO MUL1RTN SUBROUTINE
        STAB   PRODLB     STORE PRODUCT LOW BYTE
        STAA   PROMLB     STORE PRODUCT MIDDLE LOW BYTE
        LDAA   MULDLB     LOAD ACC.A WITH MULTIPLICAND LOW BYTE
        LDAB   MULRHB     LOAD ACC.B WITH MULTIPLIER HIGH BTYE
        STAB   MULTD1
        STAA   MULTR1
        JSR    MUL1RTN
        ADDB   PROMLB
        STAB   PROMLB
        ADCA   #00000
        STAA   PROMHB      PRODUCT REGISTER HIGH BYTE
        LDAA   MULRLB
        LDAB   MULDHB      LOAD ACC.B WITH MULTIPLICAND HIGH BYTE
        STAB   MULTD1
        STAA   MULTR1
        JSR    MUL1RTN
        ADDB   PROMLB
        STAB   PROMLB
        ADCA   PROMHB
        STAA   PROMHB
        LDAA   #00000
        ADCA   #00000
        STAA   PRODHB       PRODUCT HIGH BYTE
        LDAA   MULDHB
        LDAB   MULRHB
        STAB   MULTD1
        STAA   MULTR1
        JSR    MUL1RTN
        ADDB   PROMHB
        STAB   PROMHB
        ADCA   PRODHB
        STAA   PRODHB
        RTS                  RETURNE FROM JSR
*
        ORG    $CC01

------------------------------------------------------------------------

MULDHB RMB    1
MULDLB RMB    1
MULRHB RMB    1
MULRLB RMB    1
PRODHB RMB    1
PROMHE RMB    1
PROMLB RMB    1
PRODLB RMB    1
       ORC    $CB01
MULTD1 RMB    1
MULTR1 RMB    1
        ORG   $FDAO
MUL1RTN RMB    1
*
       END
```

```
        NAM    AVERAGE1
        ORG    $F15C
*
*****************************************************
*                                                   *
*     R - X ERROR FILTERING RTN & SAMPLES           *
*                                                   *
*****************************************************
*
        LDX    $C970       INITIALIZE IX
        LDAA   #$C9
        STAA   $C970        PUSH $C9 INTO $C970
        LDAA   #$5E
        STAA   $C971        PUSH $5E INTO $C971
        LDAB   $CA31        GET R - LAST CALC.
        LSRB                DIVIDE R BY 2
        LSRB
        LSRB
        STAB   $C951        SAVE R/8
        LDAA   $C94D        GET R OLDEST SAMPLE
        LDAB   $C958        GET R LAST OUTPUT
        SBA                 SUBTR. OLD SAMPLE
        LDAB   $C951        GET NEWEST SAMPLE
        ABA                 ADD TO R - OUT
        STAA   $C94D        STORE R - OUT
        LDAB   $CA33        GET LAST X CALC.
        LSRB                DIVIDE BY 2
        LSRB
        LSRB
        STAB   $C959        SAVE X - SAMPLE
        LDAA   $C94C        GET X - LAST OUTP.
        LDAB   $C960        GET X OLDEST SAMPLES
        SBA                 SUBT. OLDEST SAMPLES
        LDAB   $C959        GET NEWEST SAMPLE
        ABA                 ADD TO X - OUT
        STAA   $C94C        SAVE X - OUT
        CLR    $C962        CLEAR SAMPLE CNT.
SHIFT   LDAA    $1,X        INCR. IX
        INC    $C962
        STAA   $2,X
        DEX
        LDAA   $C962
        CMPA   #$0F         FINISH 16 SAMPLES
        BEQ    OUTPUT
        BRA    SHIFT
OUTPUT  LDAA   $C94D
        LDAB   $C94C
        STAB   $D002         R - OUTPUT
        STAA   $E002         X - OUTPUT
        JMP    CYCLE         NEW CYCLE
*
        ORG    $F028

-----------------------------------------------------------------------

CYCLE   RMB    1
*
        END
        NAM    CONST
        ORG    $F200
*
**************************************
*                                    *
*   SINE - COSINE TABLE CONSTANTS    *
*                                    *
**************************************
*
        FCB    $00,$04,$09,$0D,$12,$16,$1B,$1F,$24,$28
        FCB    $2C,$31,$35,$39,$3E,$42,$46,$4B,$4F,$53
        FCB    $57,$5C,$60,$64,$68,$6C,$70,$74
        FCB    $78,$7C,$80,$84,$88,$8B,$8F,$93,$96,$9A
        FCB    $9D,$A1,$A4,$A8,$AB,$AE,$B2,$B5,$B8,$BB
        FCB    $BE,$C1,$C4,$C6,$CA,$CC,$CF,$D2,$D4,$D7
        FCB    $D9,$DB,$DE,$DF,$E2,$E4,$E6,$E8,$EA,$EC
        FCB    $ED,$EF,$F0,$F2,$F3,$F5,$F6,$F7,$F8,$F9
        FCB    $FA,$FB,$FC,$FD,$FD,$FE,$FE,$FF,$FF,$FF
        FCB    $FF,$FF,$FF,$D0
        ORG    $FA00
        FCB    $00,$00,$CF,$FF,$00,$E7,$01,$5A,$80
        FCB    $CA,$62,$20,$7F,$5A,$F1,$00,$C8,$4A
        ORG    $F100
        FCB    00,01,02,03,04,05,06,07,08,09
        FCB    10,11,12,13,14,15,16,17,18,19
        FCB    20,21,22,23,24,25,26,27,28,29
        FCB    30,31,32,33,34,35,36,37,38,39
        FCB    40,41,42,43,44,45,46.47,48,49
        FCB    50,51,52,53,54,55,56,57,58,59
        FCB    60,61,62,63,64,65,66,67,68,69
        FCB    70,71,72,73,74,75,76,77,78,79
        FCB    80,81,82,83,84,85,86,87,88,89
        FCB    90

        END
```