A Microprocessor-based Compensator for Coreless Current Sensors

A Thesis
Presented to
the Faculty of Graduate Studies
the University of Manitoba

In Partial Fulfilment
of the requirements for the degree
of

Master of Science
in
Department of Electrical Engineering

by
Po For Leung

Winnipeg, Manitoba
April, 1990

ISBN 0-315-63394-8

A MICROPROCESSOR-BASED COMPENSATOR FOR CORELESS CURRENT SENSORS

BY

PO FOR LEUNG

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

Abstract

This study deals with the development of an alternative to the traditional current transformer. Current sensors including Hall Effect transducers and coreless coils are studied and their suitability as substitutes for the current transformer are evaluated. Both of these devices provide attractive alternatives to conventional current transformers in many applications. However, the output signals of them are very small and cross-talk between phases is significant. Therefore, a microprocessor-based compensator is designed for eliminating the phase-to-phase cross-talk. The procedures of designing the compensator are described in this report. The compensator operates with suitable current sensors to function as an accurate ammeter. The compensated ammeter is capable of displaying the actual magnitudes of the currents in the three phase bus-bars without the effect of phase-to-phase cross-talk.

Acknowledgements

The following paper could not have been completed without the support of Professor G. W. Swift who continuously showed an interest in terms of ideas and suggestions that clarified concepts I was dealing with. I would also like to thank Allan Mckay, Ken Biegun, G. Toole, W. Bourbonnais and A. Symmons for their technical support. Also, I would like to thank Siu Mei Liu who gave me constant encouragement. These people were an invaluable asset to my thesis paper and I would like to express my appreciation for their time and cooperation.

Financial assistance from Federal Pioneer Limited is gratefully acknowledged.

Po For Leung
April 1990

Table of Contents

Appendix page

List of Illustrations

Table page

Chapter I
Introduction

The purpose of this project is to develop an alternative to the traditional current transformer in certain special situations. For example, in three phase circuit breakers manufactured by Federal Pioneer Ltd, space is very limited within the breaker cases for installing a bulky current transformer. Properties of Hall Effect transducers and air-cored coils are studied. The suitability of these devices as substitutes for the current transformer is evaluated.

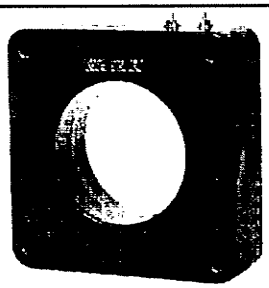


Figure 1 A current transformer

An important component of most protection or current measurement schemes is the current transformer. A typical current transformer (Crompton Instruments *Model 790*) is shown in Figure 1. Over a defined range of current and frequency, the transformer gives a signal proportional to the current flowing in the conductor being monitored. Also, the current transformer provides voltage isolation. At present, the current transformer is usually wound on a high permeability iron core with the secondary winding connected to a low impedance circuit. The current in the secondary winding is related to the current in primary winding by the turns ratio of the transformer. Although the performance of the conventional iron-cored current transformer is satisfactory and well-proven, it also has a few disadvantages [1]:

1. The output of the secondary winding may fail to indicate the actual current being monitored if the core iron becomes saturated.

2. The provision of a high quality iron core sufficiently large to avoid saturation adds to the size, weight and cost of the current transformer.
3. A current transformer capable of being temporarily fitted to a conductor must have an iron core which can be split for installation.

Hall Effect transducers have widely been used for current sensing in the last decade [2] [3] [4] [5] [6]. Previous works have proven that Hall Effect transducers have advantages when measuring high current: no iron saturation problem, measuring current of inaccessible conductors, easily attached to or removed from current carrying conductors without breaking the loop, light and compact.

Air-cored Rogowski coils [7] provide an attractive alternative to the conventional current transformers in many applications. A Rogowski coil is a toroidal winding that is commonly used to measure high currents [8]. Its chief advantages are cheapness, lightness, capability to be made flexible or in any shape, freedom from iron saturation problems, large dynamic range and satisfactory operation where there is a gap in the toroid.

However, the output signals of both these devices are very small and "cross-talk" between phases is significant. A microprocessor-based compensator is designed to eliminate the phase-to-phase cross-talk. This microprocessor-based system operates with suitable sensing devices to function as an accurate ammeter.

In this thesis, the magnetic field pattern around the three phase bus bars is first analyzed. Second, the properties of the Hall Effect transducers and the air-cored coils are described. The most suitable one for sensing current in the three phase bus bars will be selected. Third, a description of how the microprocessor-

based system was designed and constructed. The system provides the functions of amplifying the signals from the sensing devices, implementation of a mathematical model to compensate for the interference signals, and displaying the digital readings of the current for each phase. Finally, an accuracy analysis of the ammeter are given.

Chapter II
Analysis of the Magnetic Field Pattern

A magnetic field is generated around a conductor when electric current is passing through it. Also, the magnitude of magnetic field at a particular point near the conductor is linearly proportion to the magnitude of current in the conductor. Obviously, the magnitude of current can be obtained by measuring the magnitude of magnetic field. In a high current conductor, it is better to measure the magnetic field than the current itself.

For the best result, it is desirable to measure the field at the point with the highest flux density. To locate the point with the strongest magnetic field, the field pattern around a conductor is analyzed. Often, the conductor used has a rectangular cross-section and is called a *bus bar*. One standard dimension manufactured by Federal Pioneer Limited is 3 inches tall, a quarter inch thick and 4 inches spacing between bus bars.

A program **BIOT** was written for calculating the strength of magnetic field at points around the bus bar. The program is based on the Biot-Savart equation:

$$B = \frac{\mu I}{2\pi R} \qquad (1)$$

where **B** is the magnetic flux density, μ is the permeability, **I** is the current in an element of the bus bar, and **R** is the distance of the given point from the bus bar element. The units are SI (systéme internationale).

The program first divides the cross-sectional area of the copper bus bar into 48 square elements as shown in Figure 2. Secondly, assume each of these

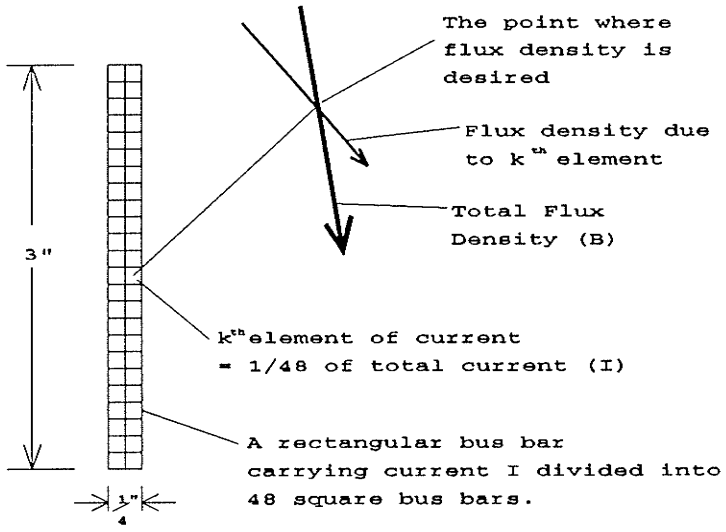


Figure 2 A rectangular bus-bar divided into 48 square bus-bars

square bars carries 1/48 the current of the entire bus bar. Thirdly, calculate the elementary flux density of each square bar at the given points by using the Biot-Savart equation. Lastly, the total flux density at the given points is computed by vector summation of the elementary fluxes. The source code of the program **BIOT.PAS** is listed in Appendix A.

For a current of 1000 amperes, the strength of magnetic flux at various points around the bus bar can be computed. The flux density versus vertical displacement of the sensing point along the vertical centre line of bus bars is plotted in Figure 3a and Figure 3b. The flux density versus displacement of the sensing point, from Bar A, along the horizontal centre line of bars is plotted in Figure 4. The results are verified by using a gauss meter (Model 610, F.W.Bell). It is obvious from the graphs that the maximum flux is detected at points close to the bus bar. After plotting lines with equal magnetic flux density, the magnetic field pattern around the bus bars is drawn in Figure 5.

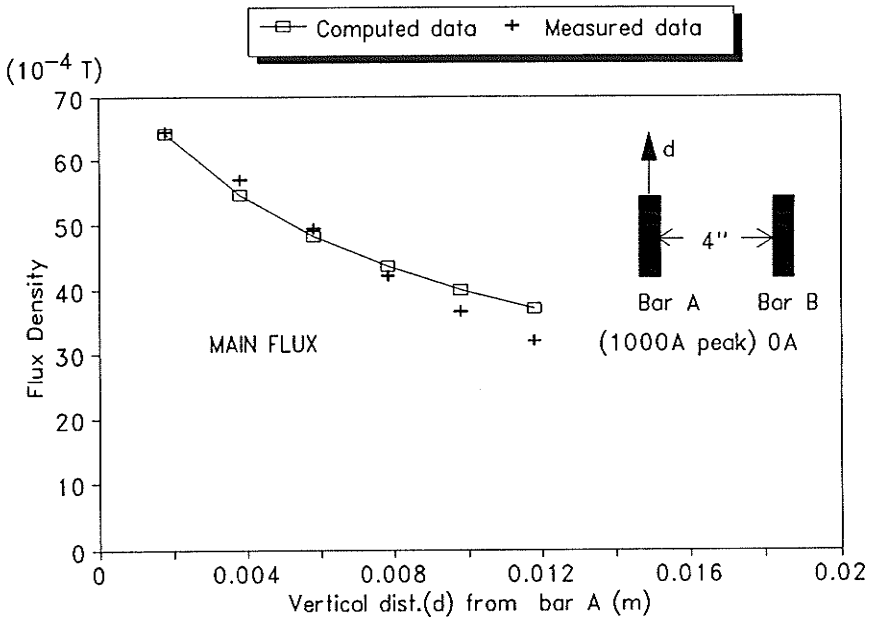


Figure 3a Main flux density versus vertical distance from bar A.

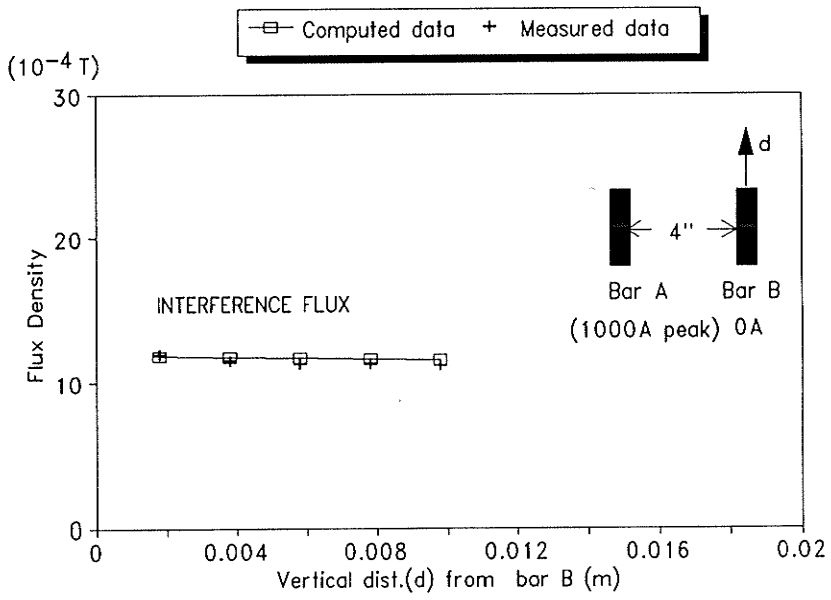


Figure 3b Interference flux density versus vertical distance from bar B

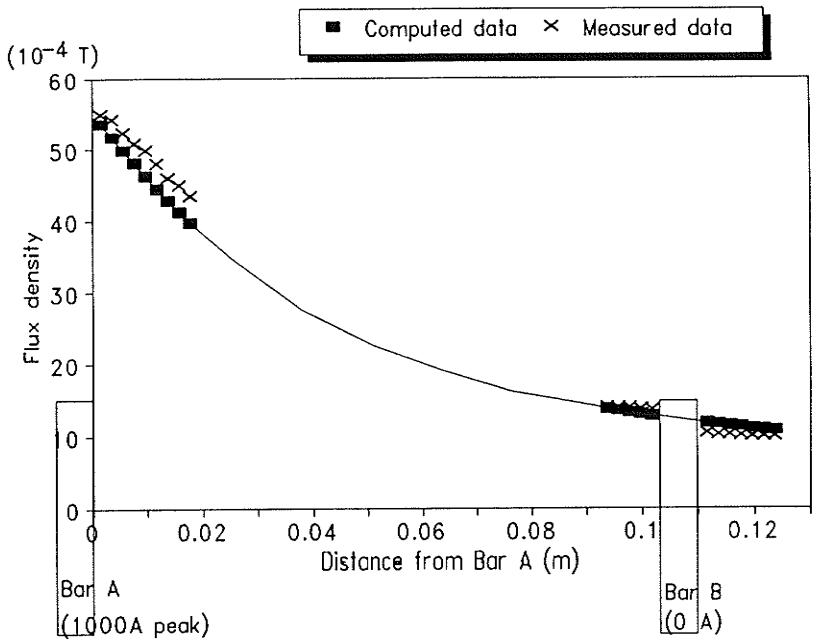


Figure 4 The plot of flux density versus horizontal displacement from Bar A.

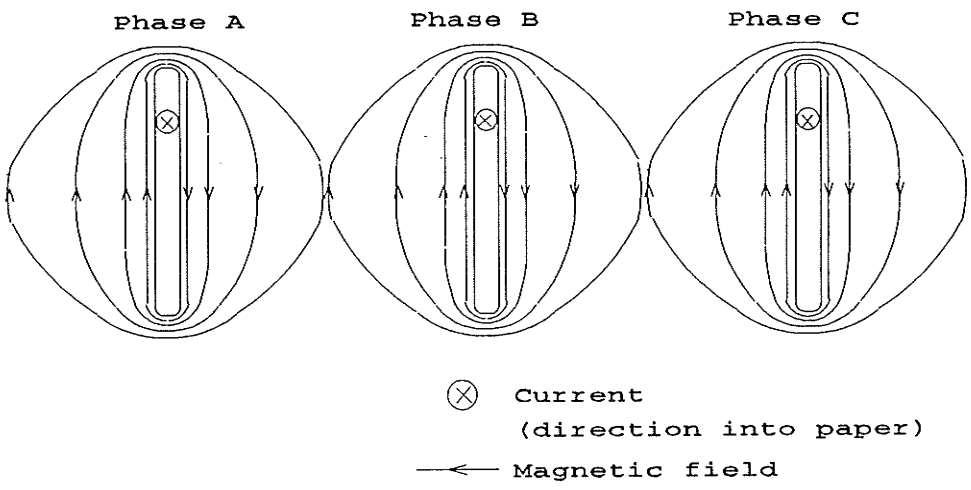


Figure 5 The magnetic field pattern around bus bars

Chapter III
Hall Effect Transducers

3.1 The Hall Effect

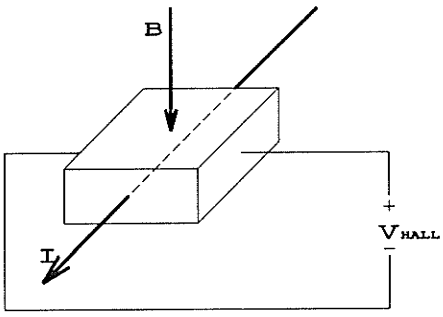


Figure 6 The Hall Element

The Hall Effect was discovered by Edward H. Hall in 1879. He found that a voltage is generated across a current-carrying conductor while current is passing through it in the presence of a transverse magnetic field. The voltage, called Hall voltage, is developed across the conductor in a direction perpendicular to both the initial current direction and the magnetic field, as shown in Figure 6. The magnitude of the voltage depends on the current, the strength of the magnetic field and the property of the material. Modern Hall effect devices use a semiconductor as the sensing material.

3.2 Device Assembly

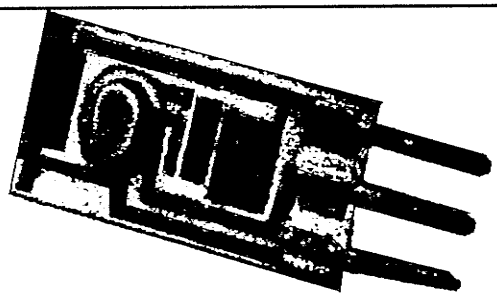


Figure 7 The 9SS LOHET

The transducer used in this project is called **9SS Linear output Hall effect transducers (LOHET™)**, shown in Figure 7, manufactured by Micro Switch. The output voltage of the transducer varies in proportion to the strength of the magnetic field. The specification for 9SS LOHET are in Appendix B.

3.2.1 The fixed-gain sensing device

A fixed-gain magnetic field sensing device based on Hall effect transducers were constructed as shown in Figure 8.

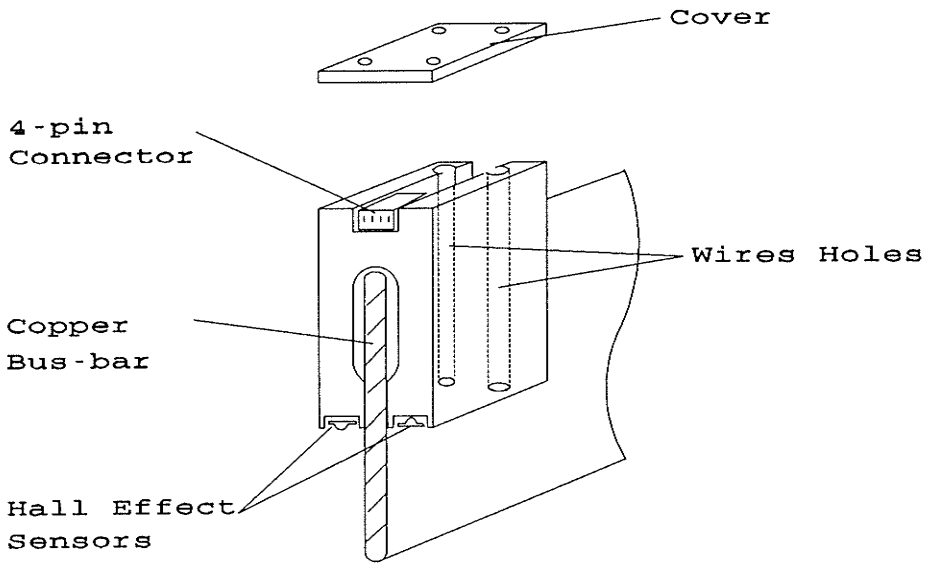


Figure 8 Diagram of the fixed-gain device

Hall effect transducers were mounted on each side of the device in opposite directions. This has the advantage of doubling the magnitude of the output signals and eliminating most of the cross-talk effect. Since the main flux circulates around a bus bar with the same polarity as the transducers, as shown in Figure 9, the output voltages are added. Conversely, since the polarity of the interference flux detected by a transducer on one side is opposite to that of the other side and their magnitudes are almost equal, the interference is almost cancelled.

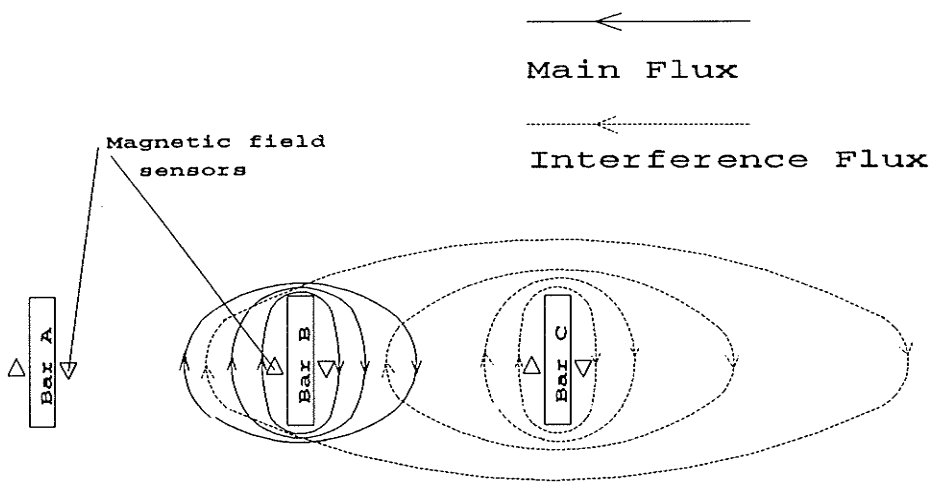


Figure 9 Magnetic fluxes generated by bus-bars

A summing circuit was built, as shown in Figure 10, for summing the output voltages of the LOHETs on the two sides of the unit. Outputs of both transducers are connected to the negative input of a 741 op-amp. A dc voltage was applied to the positive input of the op-amp for overcoming the dc bias.

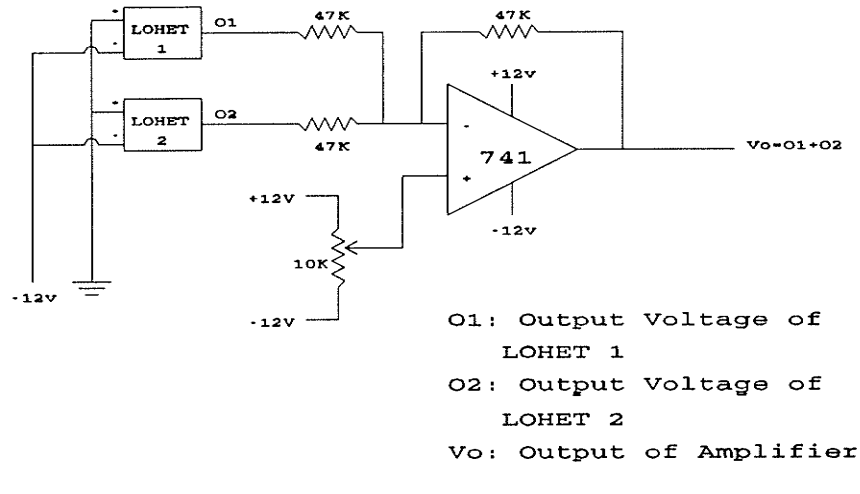


Figure 10 The circuit for summing LOHET outputs

3.2.2 The adjustable-gain sensing device

The fixed-gain magnetic field sensing device was the original design, however, it can measure up to a limited range of current. The adjustable gain device, as shown in Figure 11, is then constructed for measuring a higher range of current. Its basic structure is similar to the fixed-gain device, except the transducers are mounted on the rotatable knobs rather than fixed positions.

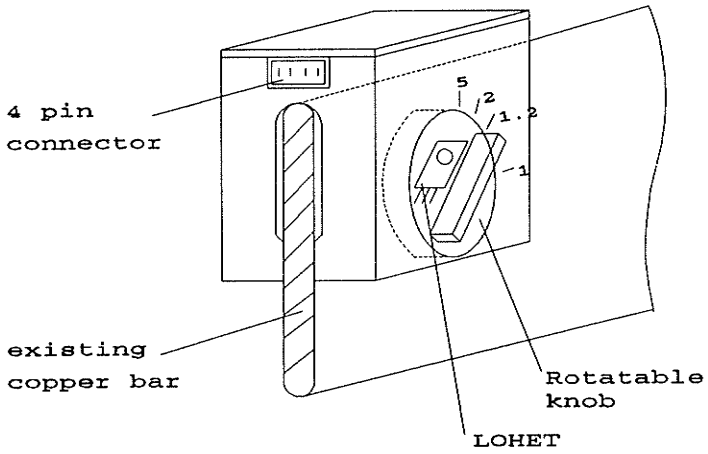


Figure 11 A diagram of the adjustable gain device

The transducers are angled to the horizontal by turning the knobs. The magnetic field incident to the transducers at an angle θ is shown in Figure 12. The strength of the magnetic field, induced by 1 unit of current flowing on the bus bar, is reduced by a multiplying factor of cos(θ). Therefore, this device can be adjusted to a higher measuring range than the fixed-position device.

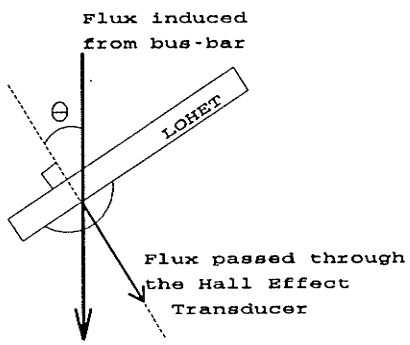


Figure 12 Magnetic flux to tilted LOHET

3.3 Device Testing

After assembly of the magnetic field sensing device, experiments are conducted to test the performance of the device. The aspects considered are linearity, accuracy and sensitivity of the device.

3.3.1 The Test Setup

In order to conduct experiments, a system was built to simulate the actual three phase bus bar situation in Federal Pioneer Ltd and the set up is illustrated in Figure 13. A copper bar with cross-section of ¼" by 3" is formed into a loop. The loop becomes the secondary winding of a current transformer with ratio of 200 to 1. One thousand amperes of current can be induced into the loop by applying 5 amperes to the primary winding of the current transformer. For convenience, the bar A is referred to the loop, the two adjacent bars are called bar B and bar C.

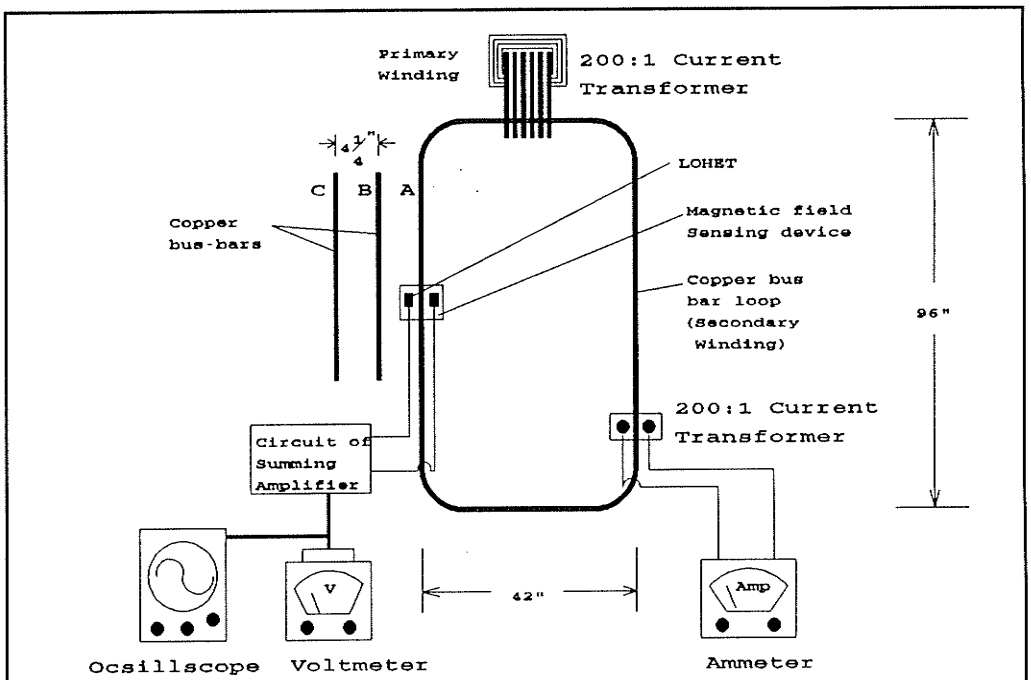


Figure 13 The experimental setup for testing the LOHET devices

3.3.2 Linearity

One important characteristic for metering purposes is linearity, so it was first examined. While current was gradually increased from 0A to 1000 A on the bar A and set the currents in the bar B and the bar C to 0A, the output voltages (induced by phase A current) of the Hall effect sensors on each phase were measured and plotted. The results, shown in Figure 14, indicate that the output voltages of sensors on each phase are linearly proportional to the current carrying in the phase A bus bar.

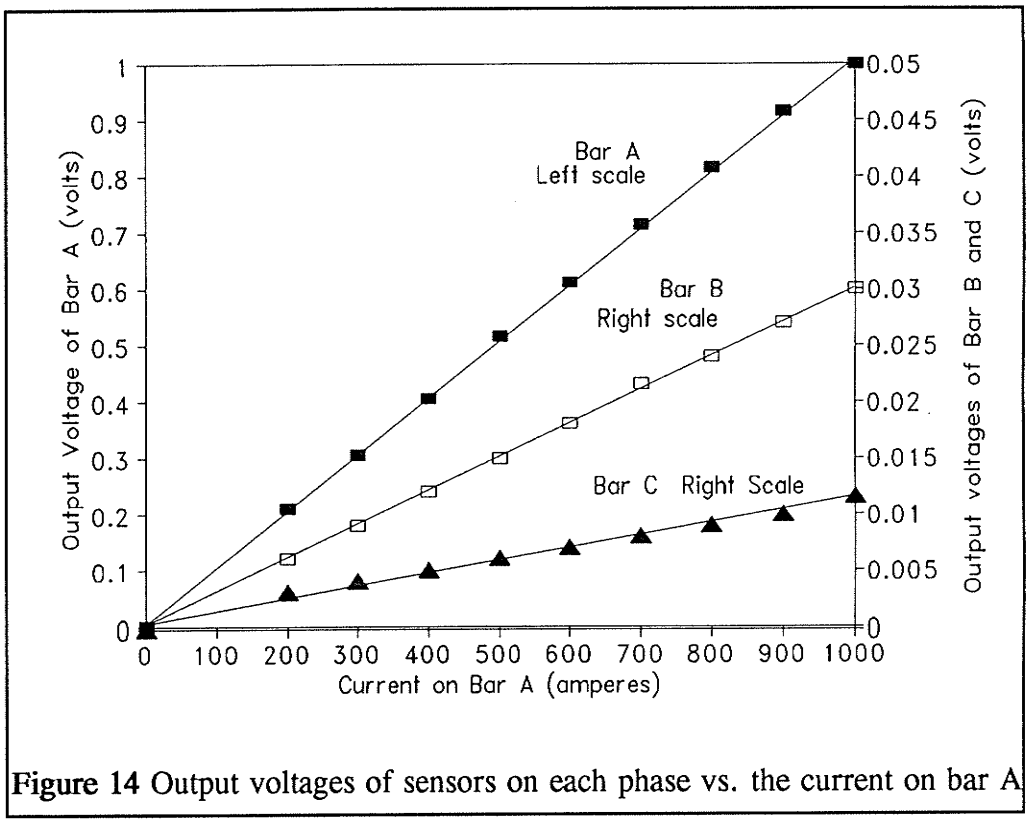


Figure 14 Output voltages of sensors on each phase vs. the current on bar A

3.3.3 Accuracy

Secondly, the accuracy of the device measurements as affected by "cross-talk" was examined. In the previous experiment, it was found that for 1 unit of voltage measured by the device on bar A, 0.03 unit of voltage would be measured on bar B and 0.012 unit would be measured on bar C. Referring to Figure 14, a correct reading of 1 volt is expected in phase A when 1000 amperes pass through bus bar A. However, interference from phase B and phase C causes the actual readings to be in error. Based on the following calculations, it is concluded that there is approximately 2 percent error in the actual phase A readings due to the interference from the other two phases, under balanced current conditions.

$$V_a(correct) = 1\angle 0^\circ \ volt$$

Assuming balanced currents in the three bus bars

$$\begin{aligned} V_a(actual) &= V_a(phaseA) + V_a(phaseB) + V_a(phaseC) \\ &= 1\angle 0^\circ + 0.03\angle -120^\circ + 0.012\angle 120^\circ \\ &= 1 - 0.015 - j0.026 - 0.006 + j0.01 \\ &= 0.98\angle -1^\circ \ volt \end{aligned}$$

$$\therefore \ Error(phaseA) = \frac{V_a(correct) - V_a(actual)}{V_a(actual)} \approx \frac{1 - 0.98}{1} = 2\%$$

Similarly,

$$Error(phaseB) \approx 3\%$$

$$Error(phaseC) \approx 2\%$$

3.3.4 Amount of cross-talk

The amount of cross-talk can be calculated, as shown in Figure 15 (a re-plot of Figure 4). The difference of flux densities between the left and right sides of bar B is the amount of cross-talk that cannot be cancelled out by LOHETs on both sides. The center of a LOHET is mounted one fifth of an inch from the bus-bar.

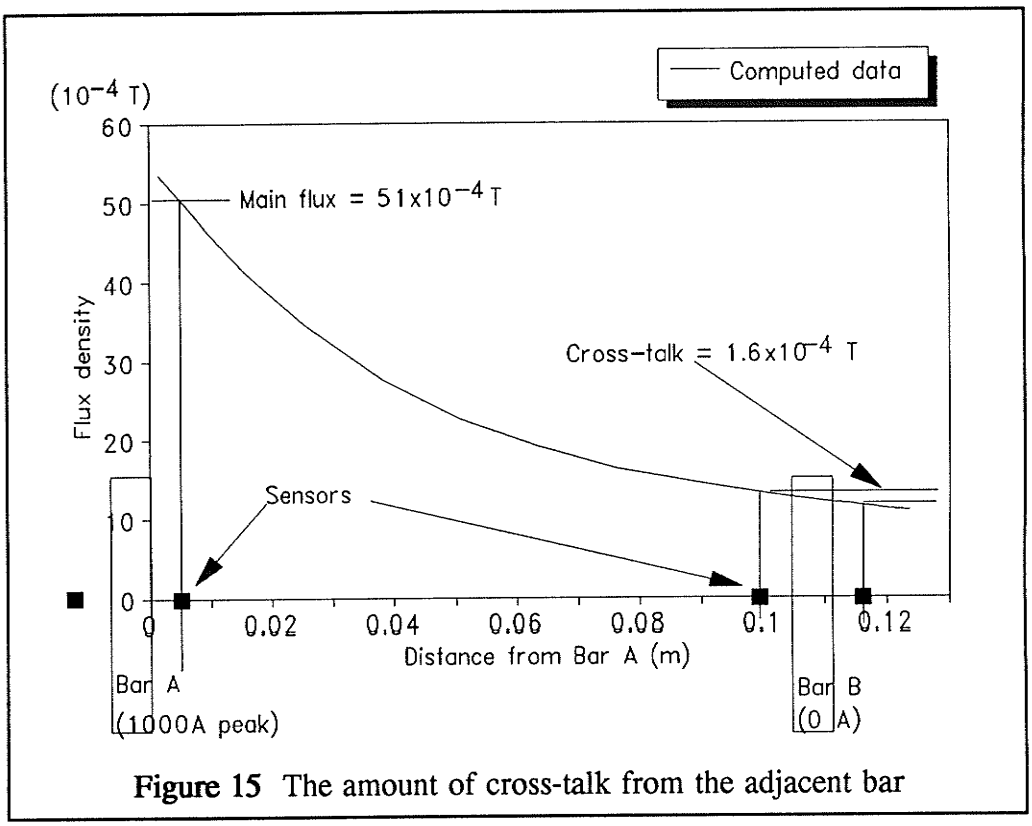


Figure 15 The amount of cross-talk from the adjacent bar

From the graph, the amount of cross-talk is about 1.6 x 10^{-4}T and the main flux on bar A is 51 x 10^{-4}T. By superposition, if same amount of current as bar A passes through bar B, the amount of cross-talk from bar B detected by bar A will also be 1.6 x 10^{-4}T. Therefore, the cross-talk from adjacent phase is:

$$Cross\text{-}talk = \frac{1.6\times10^{-4}\,T}{51\times10^{-4}\,T} \times 100\ \% \approx 3\ \%$$

3.3.5 Sensitivity

Thirdly, the sensitivity of flux density to the distance between sensors and bus-bars is also considered. The center of a LOHET is located approximately 5mm from a bus bar. Therefore, the sensitivity of this particular point is considered. From the curve in Figure 15, the flux density is 51 x 10^{-4}T at 5mm (0.2") and 50 x 10^{-4}T at 5.5mm (0.22") from a bus-bar (from 5mm to 5.5mm is 10% displacement). The sensitivity, S is:

$$S = \frac{\frac{change\ of\ flux}{flux}}{\frac{change\ of\ distance}{distance}}$$

$$= \frac{(51\times10^{-4}\,T - 50\times10^{-4}\,T)}{51\times10^{-4}\,T} \frac{5mm}{(5.5mm - 5mm)}$$

$$= 0.2$$

A sensitivity of 0.2 means that a 10% error in distance will cause only 0.2 x 10% = 2% of error in flux density. In other words, the manufacturing tolerance could be at most 0.25mm to maintain 1% accuracy.

$$The\ maximum\ tolerance = \frac{10\%}{2} \times 5\ mm = 0.25\ mm$$

3.3.6 Saturation

The fixed-gain device was designed for normal metering purposes, providing a measurement range from 0 to 4000 A with an accuracy of 2 percent. However, this rating is not sufficient for protection purposes. In the case of a fault, the extremely high current may saturate the fixed-gain device.

For protection purposes, the sensing device was modified to handled 10 times normal current. The adjustable-gain device can measure up to 40000 A before saturation. In the adjustable-gain device, the LOHETs are tilted instead of being mounted perpendicular to the magnetic field. Refer to Figure 12. As the angle θ increases, a larger current can be handled by the device.

3.3.7 Relationship between tilted angle and saturation limits

To verify that the relationship between the angle θ and the magnitude of flux detected by the LOHETs obeys the Cosine Rule, output voltages of the LOHETs were measured while the angle θ is gradually increased. The results are plotted in Figure 16.

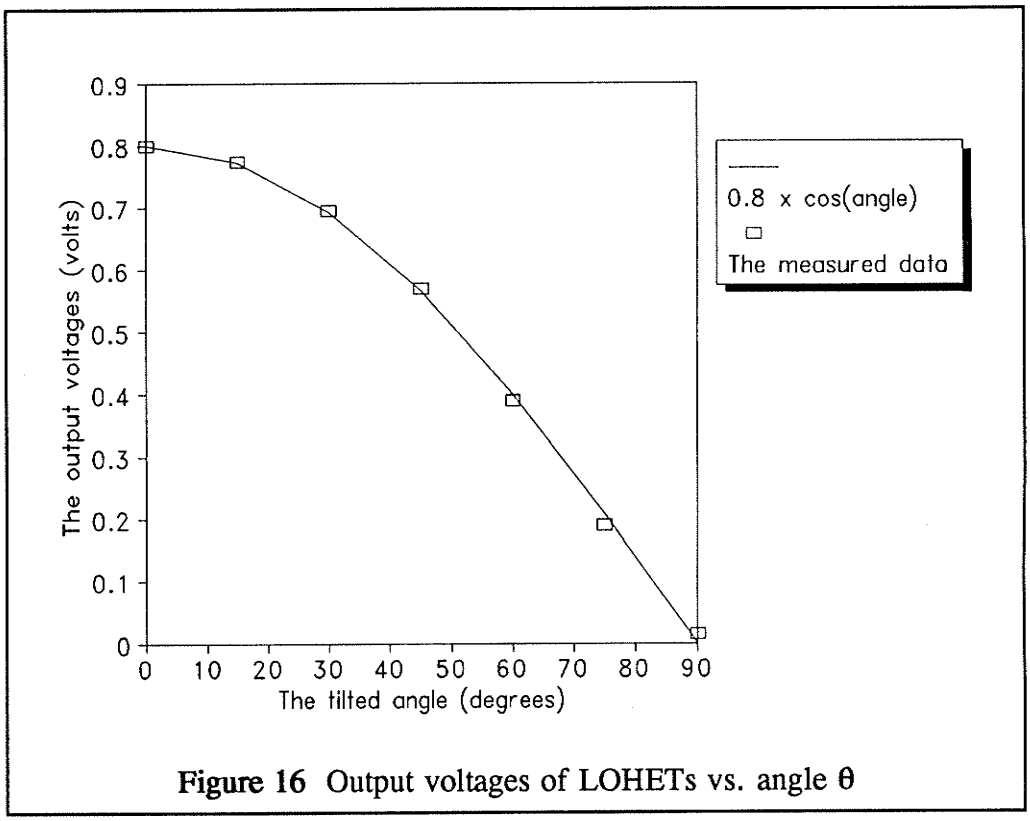


Figure 16 Output voltages of LOHETs vs. angle θ

Since the relationship follows the Cosine Rule, a desired angle for a device handling 40000 A is calculated, for example, as follows:

$$\theta = Cos^{-1}\left(\frac{4000}{40000}\right)$$
$$= 84.3°$$

Chapter IV
Air-cored Coils

4.1 Rogowski Coils

Almost 80 years ago, Rogowski described a current-voltage transducer consisting of a uniformly wound toroidal coil through which the current-carrying conductor is passed. As illustrated in Figure 17, the coil is wound back on itself to prevent it from picking up currents that run parallel to the plane of the Rogowski coil. The output voltage from the Rogowski coil is monitored by a high input impedance amplifier. The coil output is inherently differentiated so it is usually integrated by means of a passive RC circuit to provide a signal which is directly proportional to the current.

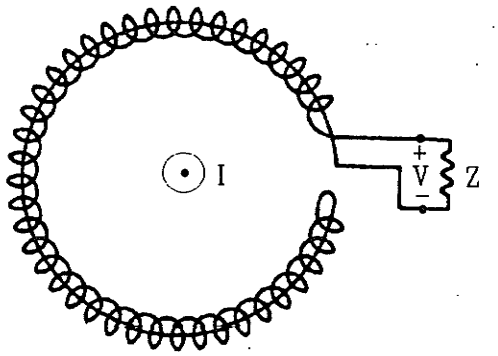


Figure 17 A Rogowski Coil

The transfer characteristic of the coil (V/I) depends on the dimensions and winding details of the coil, and to a smaller extent on the position of the conductor. It may be evaluated by calculating the flux linked by the toroid and applying Faraday's Law. The calculation is generally difficult and requires the use of a computer, however, equation (7) is true for situations where the toroid does not approach the conductor too closely:

$$\frac{V}{I} = 7.9\times10^{-5} \times f \times N \times A \quad mV/A \tag{7}$$

for sinusoidal waveforms where f is the frequency, N is the number turns per cm and A is the winding cross section area in cm^2. The derivation of equation (7) is shown in Appendix C.

The Rogowski coil has a plastic core which makes it substantially lighter than a current transformer. The device does not saturate, and hence continues to operate correctly even during the most severe faults. Since the Rogowski coil does not use an iron core, it is cheaper to manufacture. The output voltage of the coil is small (typically 1mV per ampere) and interfaces well with modern electronic metering and relay equipment. The output of a Rogowski coil is inherently linear and the dc offset is filtered out. There is negligible current flow in the coil; therefore there is no energy or heat dissipation problem. Since a small air gap will not adversely affect the performance of a Rogowski coil, it can be constructed to open slightly to encircle a current-carrying conductor.

The Rogowski coil produces a small voltage which makes the device more vulnerable to electronic noise. A Rogowski coil measures the first derivative of the current instead of the current itself; therefore, the output voltage is proportion to the frequency of the current and the phase of the voltage is shifted by 90 degrees from that of the current.

4.2 Rectangular Air-cored Coils

The performance of the toroidal solenoid Rogowski coil is satisfactory, but a high degree of quality control would have to be maintained during manufacturing to ensure accuracy. An uneven distribution of turns, a change in the cross-sectional area of the winding, or a slant in the coil's turn can cause a significant error.

The device considered here is a little out of the ordinary, in that **linear** rather than **toroidal** coils are considered, as illustrated in Figure 18. The idea is that these would have two advantages over toroidal coils:

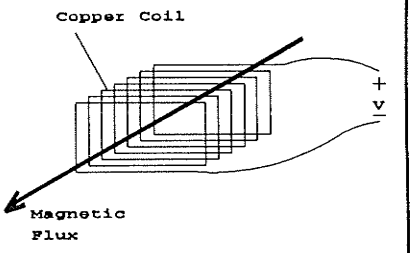


Figure 18 A linear coil

1. They would be easier and cheaper to wind.
2. They would take up less space, since they would fit flat against the bus bar rather than having to form a circle.

Unfortunately, the linear coil is more sensitive to interference from external magnetic fields due to its non-symmetrical shape. The major "external" magnetic fields are those caused by the currents of the other phases in the breaker case. Therefore, in the next chapter, a microprocessor-based compensator is designed for eliminating interference from other phases, sometimes called "cross-talk".

4.2.1 Device Assembly

A complete rectangular air-cored coil, as shown in Figure 19, composed of two linear windings. The upper winding is connected in series with the lower windings for doubling the reception of magnetic flux. Also, it has an advantage of cancelling a majority of the interference flux caused by current running parallel to it. Dimensions of the device are given in Figure 20. There are 600 turns of copper wire wound on each non-metallic bobbin. The width of the device is 1⅛ inches and it is mounted on a 1¼" x ⅛" bus bar.

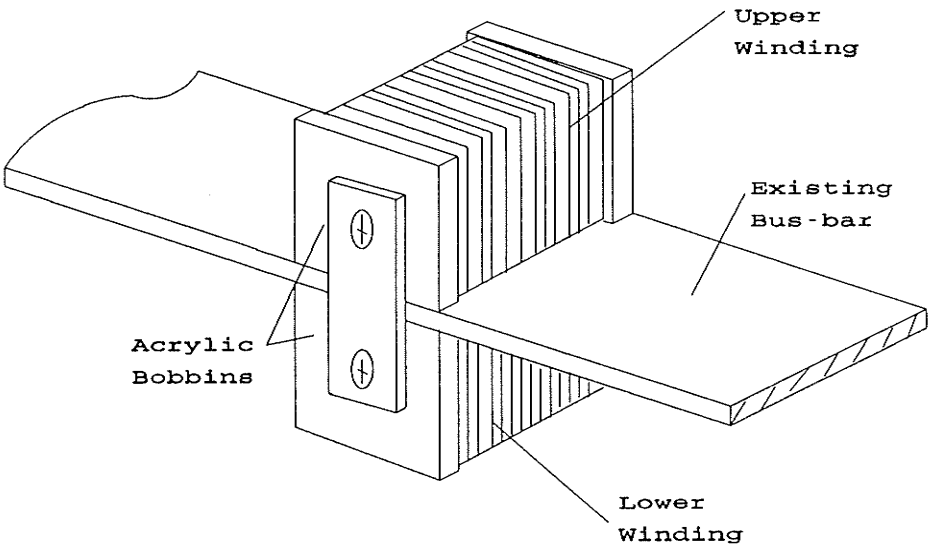


Figure 19 A rectangular air-cored coil

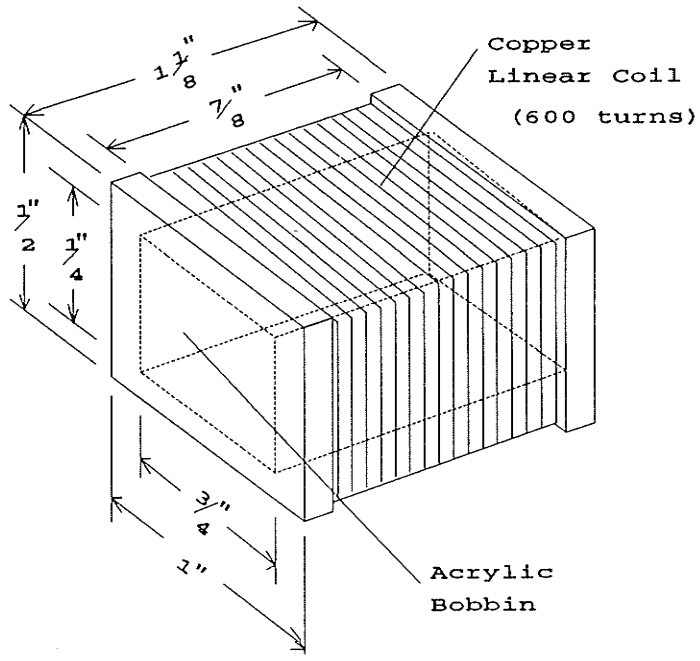


Figure 20 Dimensions of a rectangular coil

Chapter V

The Microprocessor-based Compensated Ammeter

The cross-sectional view of the three phase copper bus bars in a moulded-case breaker with magnetic field sensing coils is shown in Figure 21. Each lower winding is connected in series with the upper winding directly above it. While current I_a is passing through the Bar A, voltage V_a is generated by the copper coils attached on Bar A. In the same manner, voltage V_b is generated by passing current I_b through Bar B and V_c is generated by passing I_c through Bar C.

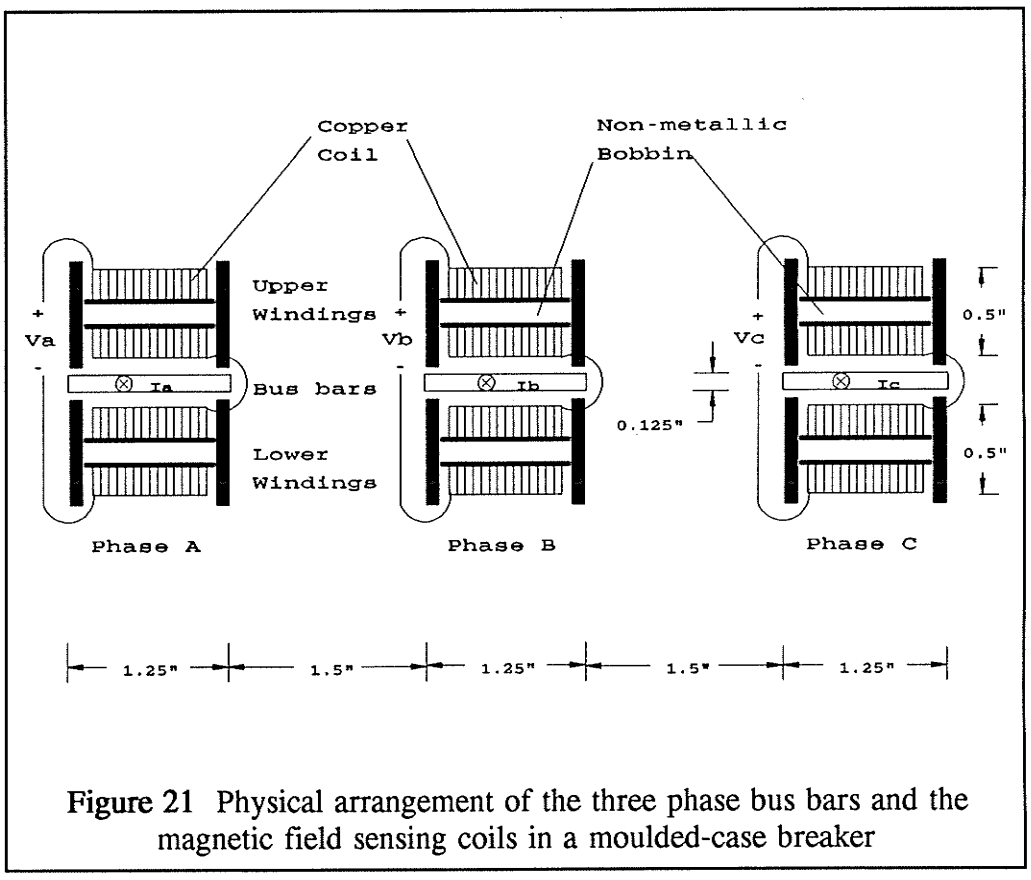


Figure 21 Physical arrangement of the three phase bus bars and the magnetic field sensing coils in a moulded-case breaker

Due to the necessarily close spacing of the bus bars, magnetic field *cross-talk* exists between the coils. The amount of the interference can be determined, so, a microprocessor-based compensator was developed to eliminate this interference. Three main stages are involved in developing the compensator. The stages are mathematical modelling, hardware design and software implementation. Each of these steps will be discussed in the following sections.

5.1 Mathematical Modelling

Referring to Figure 21, the following equation relates the sensing coil voltages V_a, V_b and V_c to the three phase bus bar currents I_a, I_b and I_c.

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = j\omega \begin{bmatrix} M_1 & M_2 & M_3 \\ M_2 & M_1 & M_2 \\ M_3 & M_2 & M_1 \end{bmatrix} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = j\omega \; \boldsymbol{M} \begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} \tag{8}$$

In the above equation, M_1 is the mutual inductance relating V_a and I_a, M_2 relates V_a and I_b, M_3 for V_a and I_c, and ω is the radian frequency, equal to 2πf where f is the frequency in Hertz.

By observation of equation (8), it is obvious that part of the voltage V_a is contributed by currents I_b and I_c. Similarly, part of V_b is contributed by the interference of I_a and I_c, and part of V_c is contributed by I_a and I_c. These interferences cause fallacious output readings of the air-cored coils.

The correct current of each phases can be obtained, as shown in equation (9), by multiplying the inverse of matrix $\boldsymbol{M}$ by the sensing coil output voltages V_a, V_b and V_c.

$$\begin{bmatrix} I_a \\ I_b \\ I_c \end{bmatrix} = \frac{1}{j\omega} M^{-1} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = \frac{1}{j\omega} \begin{bmatrix} K_1 & -K_3 & -K_4 \\ -K_3 & K_2 & -K_3 \\ -K_4 & -K_3 & K_1 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{9}$$

where K_1, K_2, K_3 and K_4 are:

$$K_1 = \frac{(M_1^2 - M_2^2)}{\Delta} \tag{10}$$

$$K_2 = \frac{(M_1^2 - M_3^2)}{\Delta} \tag{11}$$

$$K_3 = \frac{(M_1 M_2 - M_2 M_3)}{\Delta} \tag{12}$$

$$K_4 = \frac{(M_1 M_3 - M_2^2)}{\Delta} \tag{13}$$

$$\Delta = M_1^3 - 2M_1 M_2^2 + 2M_2^2 M_3 - M_1 M_3^2 \tag{14}$$

In order to apply these equations in real-time, an 8751 microprocessor-based compensator was designed to implement the formulae. The hardware design will be discussed in the next section.

5.1.1 Approximate Calculation of M_1, M_2 and M_3

The value of M_1 can only be found accurately by experiment, but it is useful to make a calculation based on the simplifying assumption shown in Figure 22. If a current I is carried by the bus bar, a magnetic field is generated around the bus bar and a mean flux path is indicated in Figure 22. For simplicity, the elliptical mean flux path is approximated by the equivalent circular path.

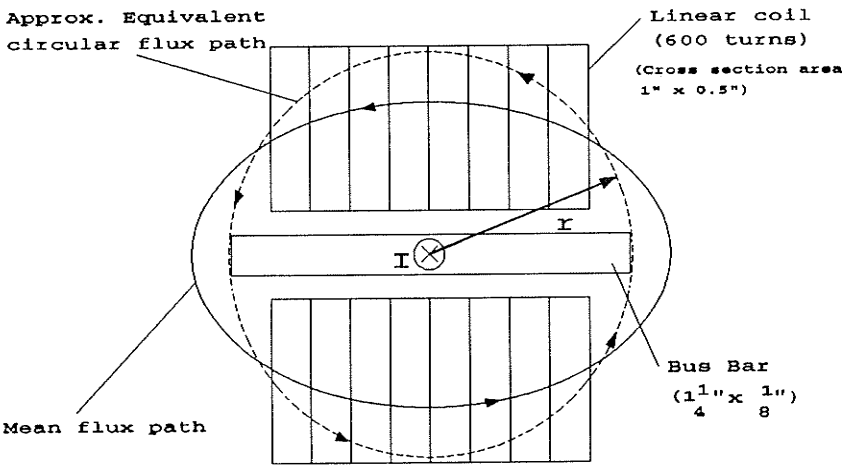


Figure 22 Approximate equivalent circular flux path around the bus bar

By Ampere's circuital law:

$$\oint H \cdot dL = i \tag{15}$$

where **H** is magnetic field intensity and **i** is the current flowing in the bus bar.

$$2\pi r H = i$$

$$H = \frac{i}{2\pi r} \tag{16}$$

where r is the radius of the circular close path.

$$B = \mu_o H = \frac{\mu_o i}{2\pi r} \tag{17}$$

where μ_o is the permeability of free space and equal to $4\pi \times 10^{-7}$ H/m

By Faraday's law:

$$v = N\frac{d\Phi}{dt} = NA\frac{dB}{dt} \tag{18}$$

where N is the number of turns, Φ is the magnetic flux, B is the flux density and A is the cross-sectional area of the coil.

Substituting (17) into (18):

$$v = \frac{NA\mu_o}{2\pi r}\frac{di}{dt} \tag{19}$$

Let $i = I \sin(2\pi ft)$:

$$v = \frac{NA\mu_o f}{r} I\cos(2\pi ft) = V\cos(2\pi ft) \tag{20}$$

Since N = 2 x 600 = 1200, f = 60Hz (current frequency) and A = the average cross-sectional area = 0.875" x 0.375" = 0.0222 x 0.0095 m^2 = 2.11 x 10^{-4} m^2,

$$V = \frac{NA\mu_o f}{r} I = 4\pi \times 10^{-7}\frac{NAf}{r} I = \frac{1.91 \times 10^{-5}}{r} I \tag{21}$$

For phase A, as illustrated in Figure 23, r = r1 = half of the bar width = 5/8" = 0.016m. The output voltage of the phase A coils is:

$$V_a = \frac{1.91\times10^{-5}}{0.016} I = 1.19\times10^{-3} I \qquad (22)$$

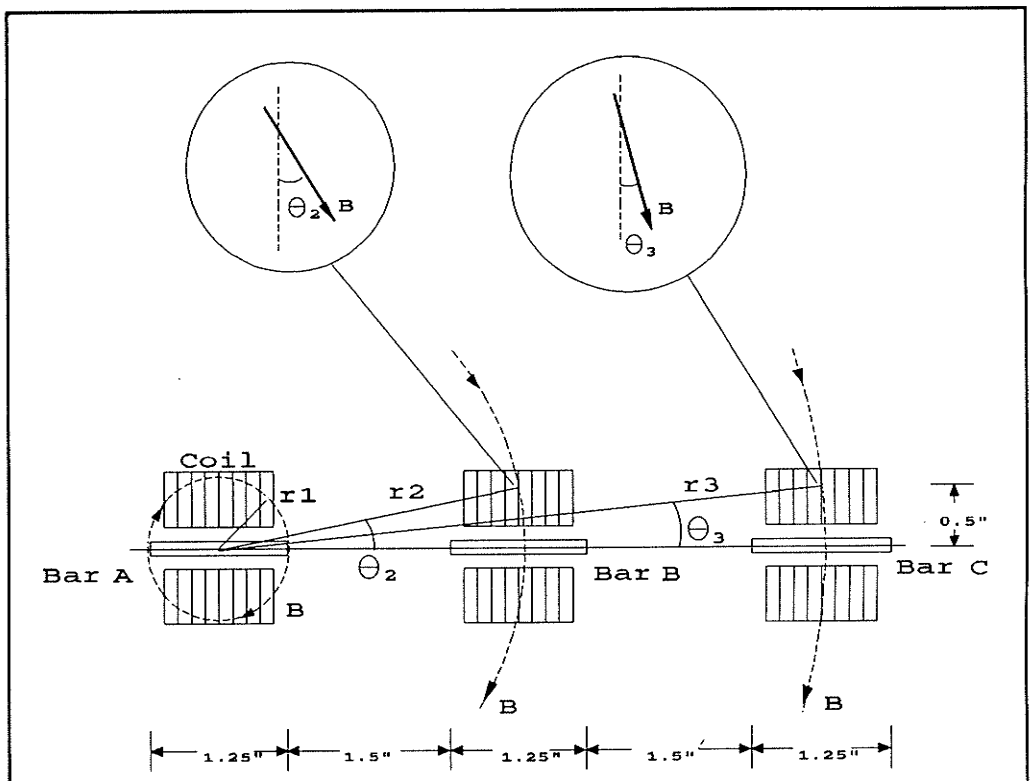


Figure 23 Magnetic flux generated when a current I is carried in phase A

As shown in Figure 23, the magnetic field incident to the phase B linear coils at an angle θ_2 so the flux density is reduced by a multiplying factor $\sin(\theta_2)$. Therefore, the output voltage of the phase B coils is approximately:

$$V_b = I \frac{1.91\times10^{-5}}{r2} \sin\theta_2 \qquad (23)$$

where r2 ≈ 2.75" = 0.07m and

$$\theta_2 = \tan^{-1}(\frac{0.5''}{2.75''}) = 10.3° \tag{24}$$

$$\therefore \quad V_b = I\frac{1.91\times10^{-5}}{0.07}\sin(10.3°) = 4.88\times10^{-5}I \tag{25}$$

Similarly for phase C, r3 ≈ 5.5" = 0.14m and

$$\theta_3 = \tan^{-1}(\frac{0.5''}{5.5''}) = 5.2° \tag{26}$$

then

$$V_c = I\frac{1.91\times10^{-5}}{0.14}\sin(5.2°) = 1.24\times10^{-5}I \tag{27}$$

Thus, based on the arrangement shown in Figure 21, 1A of current in bus bar A causes 1.19mV to be generated by the phase A coil, 0.0488mV by the phase B coil and 0.0124mV by the phase C. The mutual inductances M_1, M_2 and M_3 can be obtained by the following formulae:

$$M_1 = \frac{V_a}{\omega I} = \frac{1.19\times10^{-3}}{377} = 3.16\times10^{-6}H = 3160\ pH \tag{28}$$

$$M_2 = \frac{V_b}{\omega I} = \frac{4.88\times10^{-5}}{377} = 1.29\times10^{-7}H = 129\ pH \tag{29}$$

$$M_3 = \frac{V_c}{\omega I} = \frac{1.24\times10^{-5}}{377} = 3.29\times10^{-8}H = 33\ pH \tag{30}$$

5.1.2 Experimental determination of M_1, M_2 and M_3

In order to experimentally find the mutual inductances M_1, M_2 and M_3, we set I_a at a known value I and set I_b and I_c to zero. Measure the three main coil voltages V_a, V_b and V_c. The mutual inductances can be obtained by substituting the values of I, V_a, V_b and V_c into equations (28), (29) and (30).

A program called **K1234** was written based on the above equations. The program calculates the mutual inductances M_1, M_2, M_3 and the normalized values of K_1, K_2, K_3 and K_4 by inputting a set of data of I, V_a, V_b and V_c. The source code, **K1234.PAS,** is listed in Appendix D.

When I was set to 40A, V_a was measured as 47mV, V_b equal to 1.2mV and V_c equal to 0.4mV. The mutual inductances were found that M_1 equal to 3110pH, M_2 equal to 80pH and M_3 equal to 27pH. Comparing the calculated values with the measured values, indicates discrepancies within the tolerances of the approximate calculation. The values of K_1, K_2, K_3 and K_4 were found that equal to 128, 128, 3 and 1 respectively.

5.2 Hardware Design

The compensator, shown in Figure 24, is basically composed of three amplifier-integrators, three A/D convertors, an Intel 8751 microprocessor, and a display driver which drives a 6 digit display. Also, the schematic diagram of the system is illustrated in Appendix E. One of the reasons for selecting digital components rather than analog electronics is its flexibility. Tasks can be modified in a microprocessor-based system by simply changing the software. For example, the method of measuring the current can be changed from peak value sensing to root mean square (rms) computation by modifying the program.

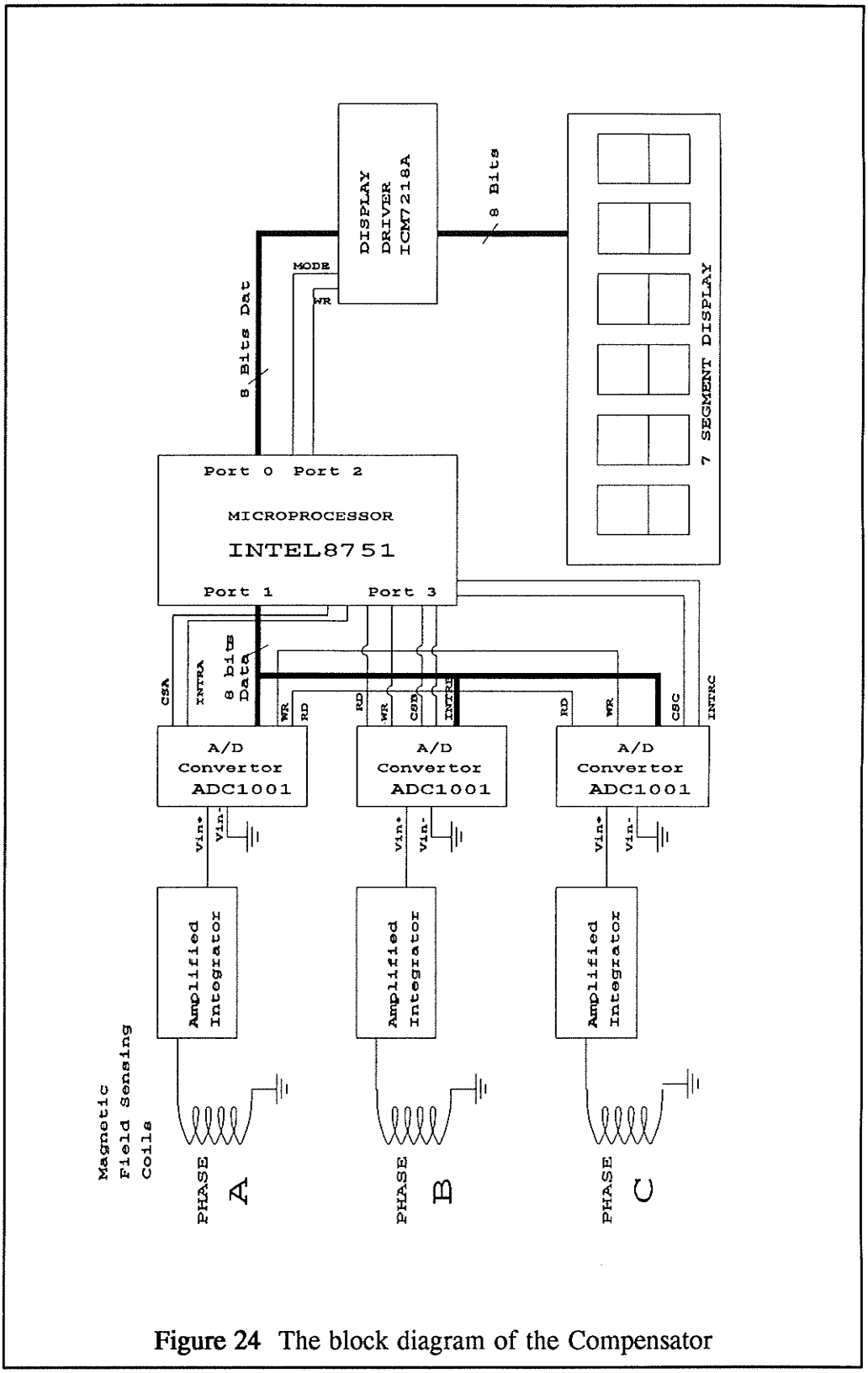


Figure 24 The block diagram of the Compensator

5.2.1 Designing the Amplifier-Integrator

The output voltage of the current sensing coil is very small and inherently differentiated respect to time, so the signal must be amplified and integrated before it is fed to the A/D convertor. Also, the A/D convertor will not accept negative voltage as an input. A circuit is designed, as shown in Figure 25, to be an amplifier-integrator. The functions of the amplifier-integrators include amplification, integration, and dc biasing.

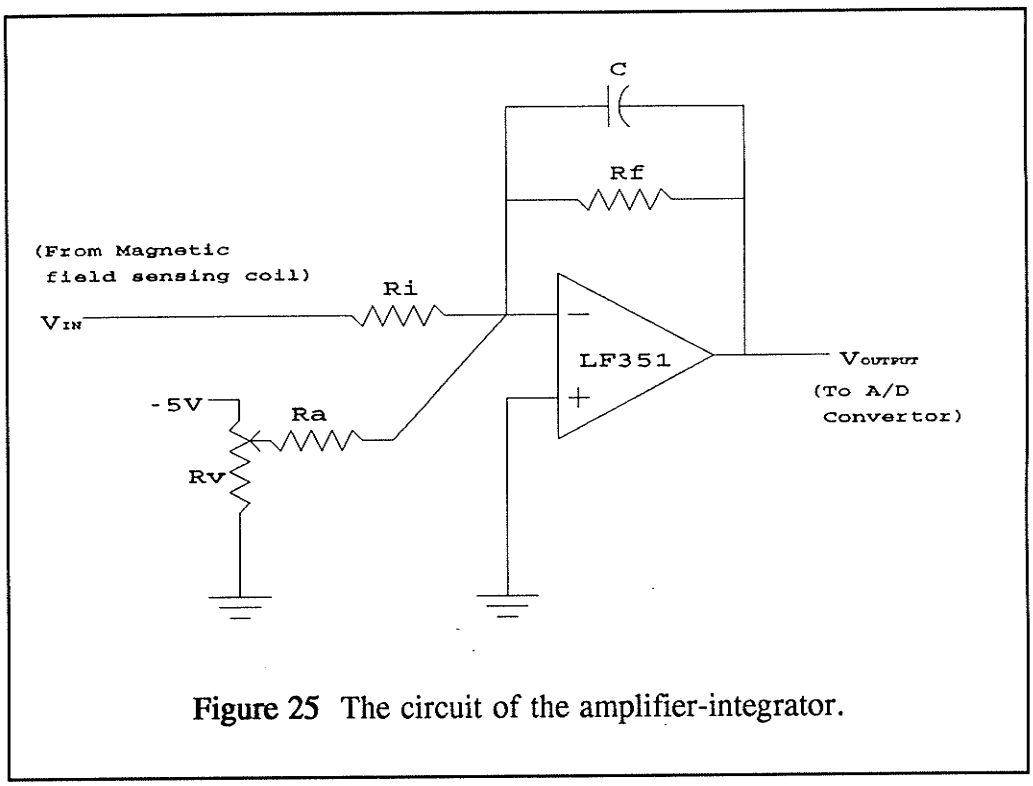


Figure 25 The circuit of the amplifier-integrator.

The ADC1001 AD converter has a full input range of 5 V and can only accept positive voltage. In order to measure ac voltage, the circuit dc biases the input voltage by 2.5 V so that the circuit has a zero reference of 2.5 V. Thus, the output voltage of this circuit represents 0 V if it is 2.5 V, represents +2.5 V if it is 5 V, and -2.5 V if it is 0 V. Therefore, the AD convertor has a full range of ± 2.5 V.

By measurement, the output voltage of the 600 turn coil was found to be typically 2mV (peak value) for 1 A (peak value) of bus bar current. A full measurement range of current is 200 A would therefore induce 0.4 V at the coil. To utilize the full range of AD convertors, a gain of 6.25 is therefore desired. For convenience, we set the desired gain to 7.

$$Desired\ Gain = \frac{Full\ range\ of\ AD\ convertor}{Full\ range\ of\ coil} = \frac{2.5\ volt}{0.4\ volt} = 6.25 \quad (31)$$

$$Gain\ (in\ dB) = 20\log(desired\ gain) = 20\log(7) = 16.9\,dB \quad (32)$$

Therefore, a gain of 16.9 dB at 60 Hertz of frequency is desired. Let's pick the cut-off frequency at 19Hz, so that 60Hz signals and higher will be effectively integrated.

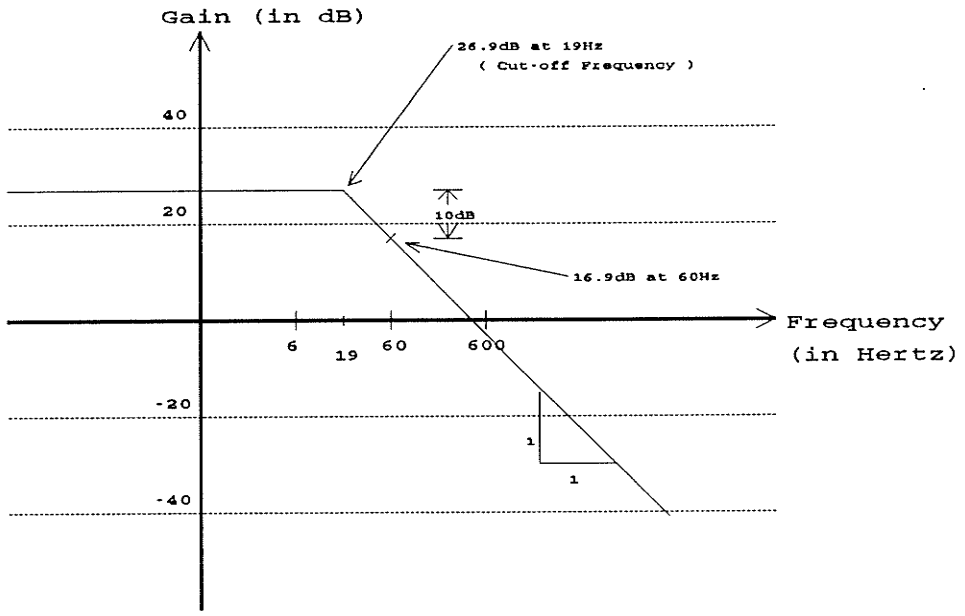


Figure 26 The log plot of the gain versus the frequency

From Figure 26, the gain at 19Hz is 26.9dB which is equal to 22.

$$desired\ gain\ at\ 19Hz = 10^{\frac{gain\ (in\ dB)}{20}} = 10^{\frac{26.9}{20}} \approx 22 \tag{33}$$

Refer to Figure 25, to achieve a gain of 22, Ri is selected as 1.5kΩ and Rf is 33kΩ for the convenience of standard resistor value. Ra is equal to Rf for a unit gain and the variable resistor Rv is 200kΩ.

The reason for selecting 1.5kΩ as Ri is that one of the sensing coils (phase C) was wound with 800 turns rather than 600 turns. This coil generates an extra 33% of voltage compared to the others. It is ideal to equalize the voltages by adjusting the gains of amplifiers. Since three-quarters of the output of phase C is desired, the ratio of the gain of phase C is 3:4 of the gain of other phase. To match the value of standard resistors, Ri of phases A and B is selected as 1.5kΩ and Ri of phase C is selected as 2kΩ.

The selection of the capacitance C is as follows:

$$\omega = 2\pi f_c = \frac{1}{R_f C} \tag{34}$$

where f_c is the cut-off frequency and R_f is the feed-back resistor.

$$C = \frac{1}{2\pi f_c R_f} = \frac{1}{2\pi(19)(33000)} = 0.25\mu F \tag{35}$$

In summary, refer to the circuit in Figure 25, the parameters of elements are:

Ri (for 600 turns coils)	=	1.5kΩ
Ri (for 800 turns coils)	=	2.0kΩ
Rf	=	33kΩ
Ra	=	33kΩ
Rv	=	200kΩ
C	=	0.25μF

Effectiveness of the Integration

For a current of 35 A on the bus bar, the output voltage of the coil (Figure 27) is 43mV peak amplitude. The signal is inherently differentiated, so harmonic content is emphasized. After integration and amplification by the circuit, as shown in Figure 28, the waveform is much more sinusoidal, as was the input current, and the peak value is 0.32 volt. The gain ratio is approximately 7 as desired.

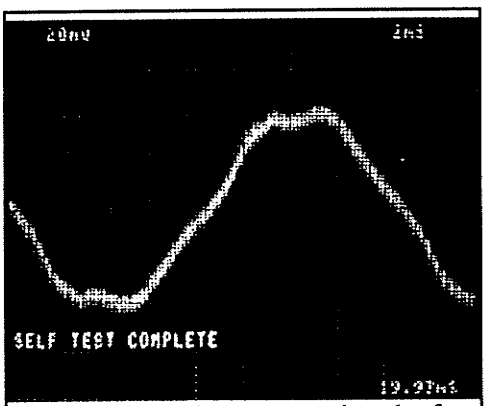


Figure 27 The output signal of a coil

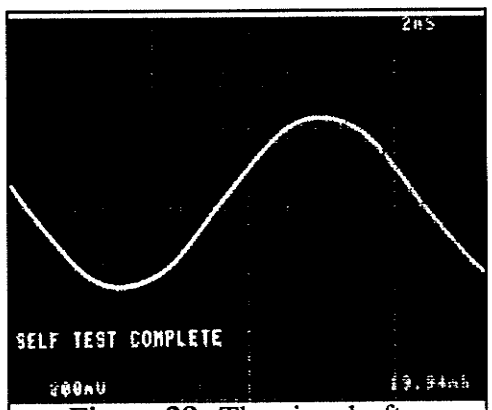


Figure 28 The signal after integration and amplification

5.2.2 Analog-to-Digital Convertors

After amplification and integration by the amplifier-integrator, the signal is applied to the ADC1001 A/D convertor. The ADC1001 is a 10-bit successive approximation A/D converter. The 10-bit data word is read in two 8-bit bytes, and formatted left justified with high byte first. The six least significant bits of the second byte are set to zero, as is proper for a 16-bit word.

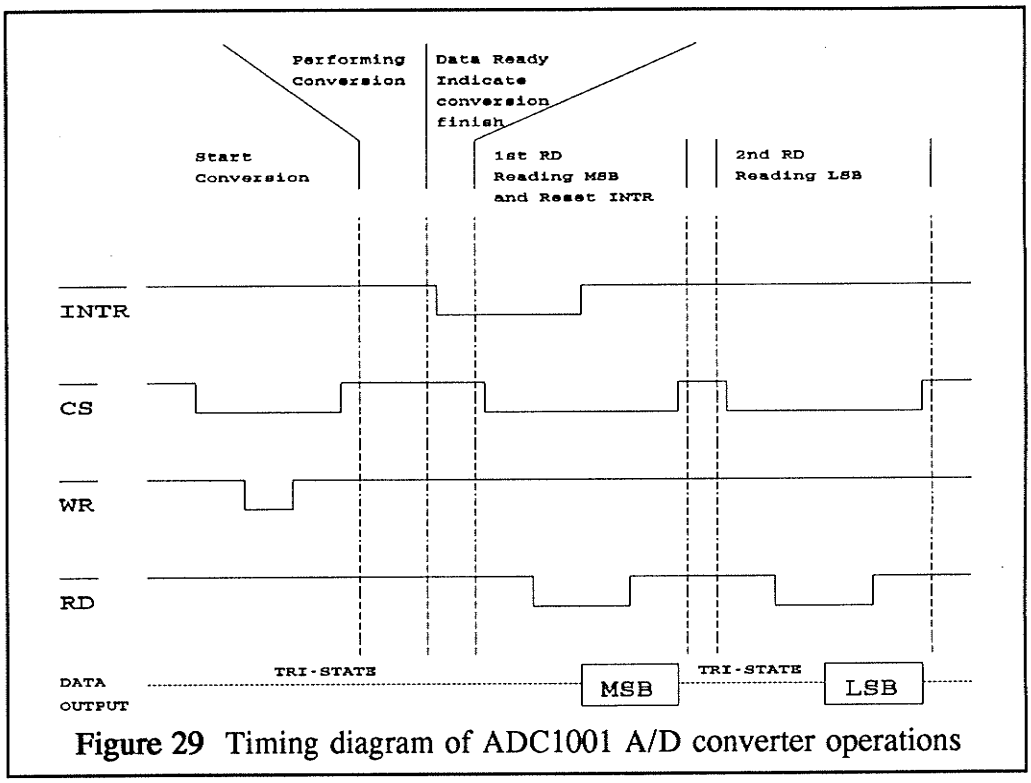


Figure 29 Timing diagram of ADC1001 A/D converter operations

The operation of the A/D converter is described in the timing diagram in Figure 29. The converter starts conversion while sensing negative pulses of chip select (CS) and Write (WR). A negative transition of interrupt (INTR), is sent back to the microprocessor, indicating conversion is complete and data is valid. The microprocessor starts to read the most significant byte (MSB) by sending negative pulses to CS and RD, and then reading the least significant byte (LSB) by sending negative pulses to CS and RD again.

5.2.3 The Intel 8751 Microprocessor

The main function of the 8751 microprocessor is to perform compensation by implementing the mathematical model in real time. In addition, the microprocessor controls the A/D convertors to sample signals at fixed intervals, and controls the display driver to display correct readings of current on each phase. A diagram that shows the interfacing of peripheral components is illustrated in Figure 30.

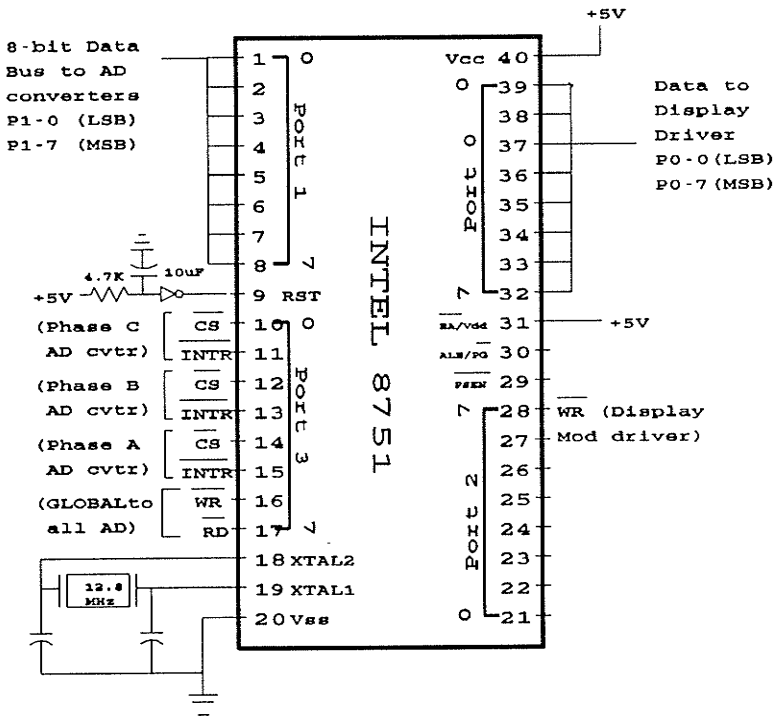


Figure 30 8751 microprocessor interfaces with peripheral components

There are 4 bidirectional I/O ports in the 8751 microprocessor. Port 0 is used for sending data to the display driver while Port 1 is used for receiving data from the AD converters. Bits of port 3 are used for controlling the AD converters. P3-6 is connected to WR lines of all AD converters and P3-7 is

connected to RD lines of all converters. CS lines are used for selecting which AD converter is functioning. INTR lines detect the data-ready signals from the corresponding converter. Port 2 is used for controlling the display driver. An oscillator is connected to the processor to make it run at 12MHz.

The Intel 8751 microprocessor has 4K bytes of on-chip EPROM (Erasable Programmable Read-Only Memory) for programming and 128 bytes of on-chip data memory. These features make it easy-to-use. Also, a system board using the 8751 can be more compact than one using a microprocessor which has no on-chip EPROM.

5.2.4 The ICM7218A Display Driver

The ICM7218A 8-digit multiplexed display driver is used for interfacing the 8751 microprocessor with a set of six NSN582 7-segment displays. A diagram of the ICM7218A pin configuration is shown in Figure 31. Each ICM7218A is capable of driving 8 digits of a 7-segment display. To accomplish this, each of the segment lines (a, b, c, ... ,D.P.) is connected to the corresponding segment and each 7-segment display is linked by a digit select line. Thus, to select a segment on a particular digit, the digit line corresponding to the digit and the segment line for the segment are selected.

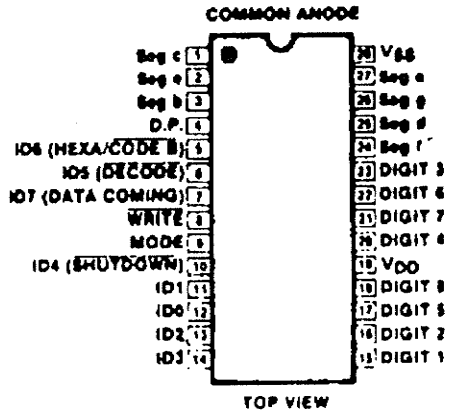


Figure 31 ICM7218A pin configuration

The relationship among bits of port0 of the Intel 8751, the input data lines of the display driver and the output segments of the 7-segment display is illustrated in Figure 32. Thus, to display a character or a number on a 7-segment display, a proper bit pattern must be written into port0. For example, the hexadecimal number *ED* (11101101 in binary) must be written into port0 to display a number "2" on the 7 segment display. Note that the decimal point (D.P.) light is on when pin 7 of port 0 goes low.

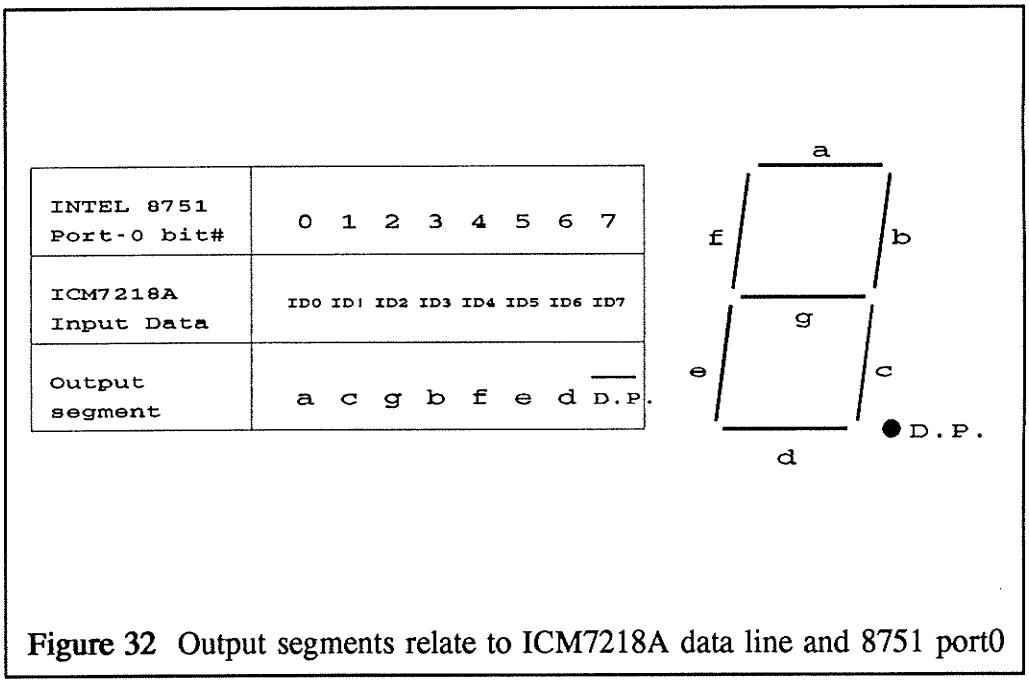


Figure 32 Output segments relate to ICM7218A data line and 8751 port0

5.3 Software Implementation

The software is basically composed of the main program and a timer interrupt routine. A simplified flow chart is shown in Figure 33. The main program initializes all variables, tests the display driver and 7-segment displays, and continuously detects and displays the correct reading of peak current on each phase. Upon receiving hardware interrupts at fixed intervals, the timer interrupt routine samples the voltage on each coil, performs compensation, calculates the actual current at the time and searches for peak values of current on each phase.

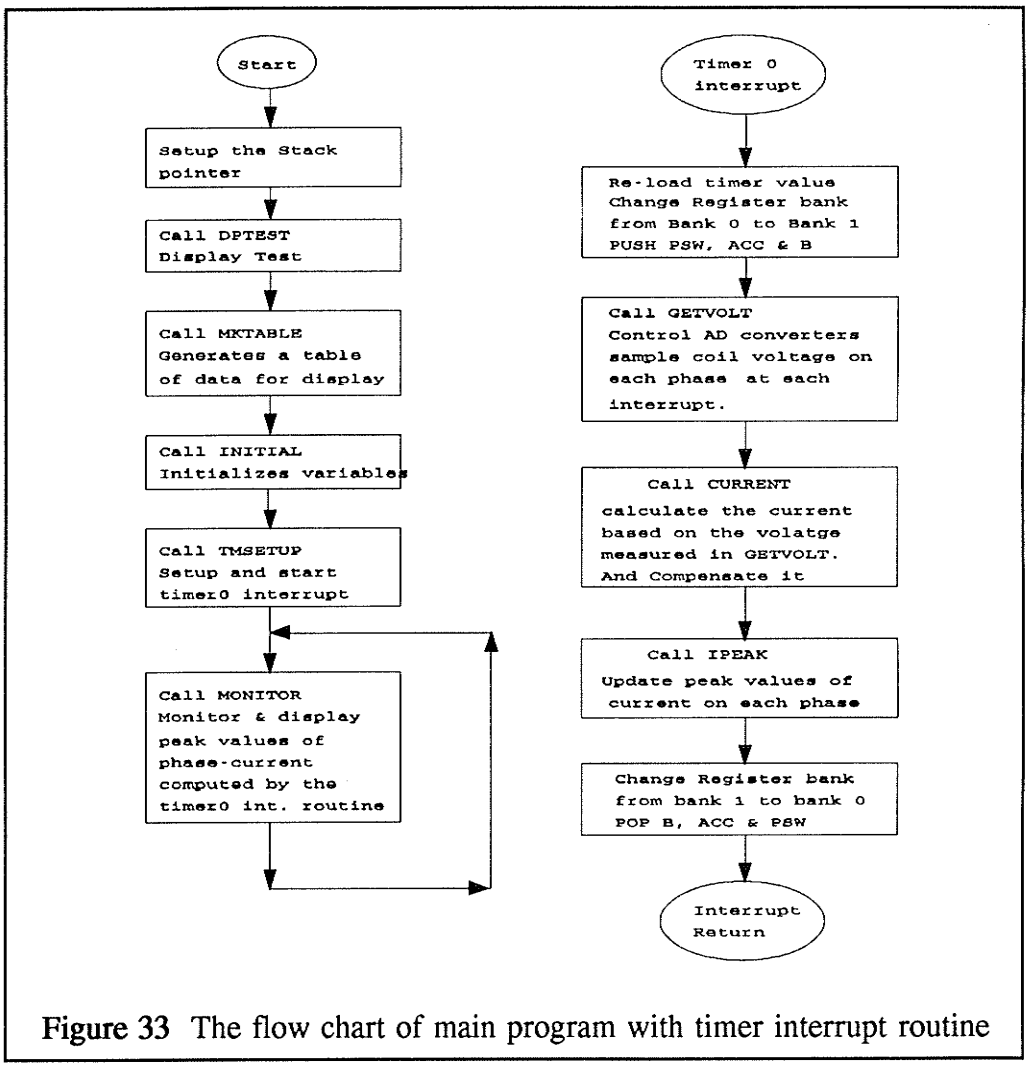


Figure 33 The flow chart of main program with timer interrupt routine

The source code is listed in Appendix F. The main program first assigns the location of the top of the stack pointer which is necessary for storing return addresses upon calling subroutines. Second, the display component is tested by subroutine DPTEST which sends a message "DPTEST" to the display driver. The 7-segment display should show the message **DPTEST**, indicating proper functions. Third, the subroutine MKTABLE generates a table which stores data for displaying numbers from 0 to 9. Fourth, the subroutine INITIAL initializes all declared variables in the program. Fifth, the Timer 0 of the 8751 microprocessor is set up as a 16-bit timer and started by the subroutine TMSETUP. Flow charts of subroutines DPTEST, MKTABLE, INITIAL and TMSETUP are illustrated in Figure 34. While the timer interrupt routine is running in the background, the subroutine MONITOR in the main program is still continuously monitoring and displaying values of current calculated by the timer interrupt routine. The subroutine MONITOR will be discussed in later sections.

The timer interrupt routine is separated into three stages: voltage sensing from each coil by subroutine GETVOLT; calculating the compensated current of each phase by subroutine CURRENT; and searching for the peak value of current of each phase by subroutine IPEAK. These subroutines will be described in the following sections.

The microprocessor is running at 12MHz so each machine cycle (takes 12 clock periods) is approximately 1μs. The length of time interval between interrupts of timer0 is dependent on the numbers stored in TM0H and TM0L. The frequency of the current being measured is 60Hz. The timer interrupt routine takes about 560 machine cycles (0.58ms) to complete. In order to obtain the maximum number of samples in each current cycle, hexadecimal numbers FD and 94 are stored in TM0H and TM0L respectively. Thus, the timer interrupt happens at every 620 (26C in hexadecimal) machine cycles which is approximately 0.64ms. In other words, the program samples 26 times in each 60Hz current cycle.

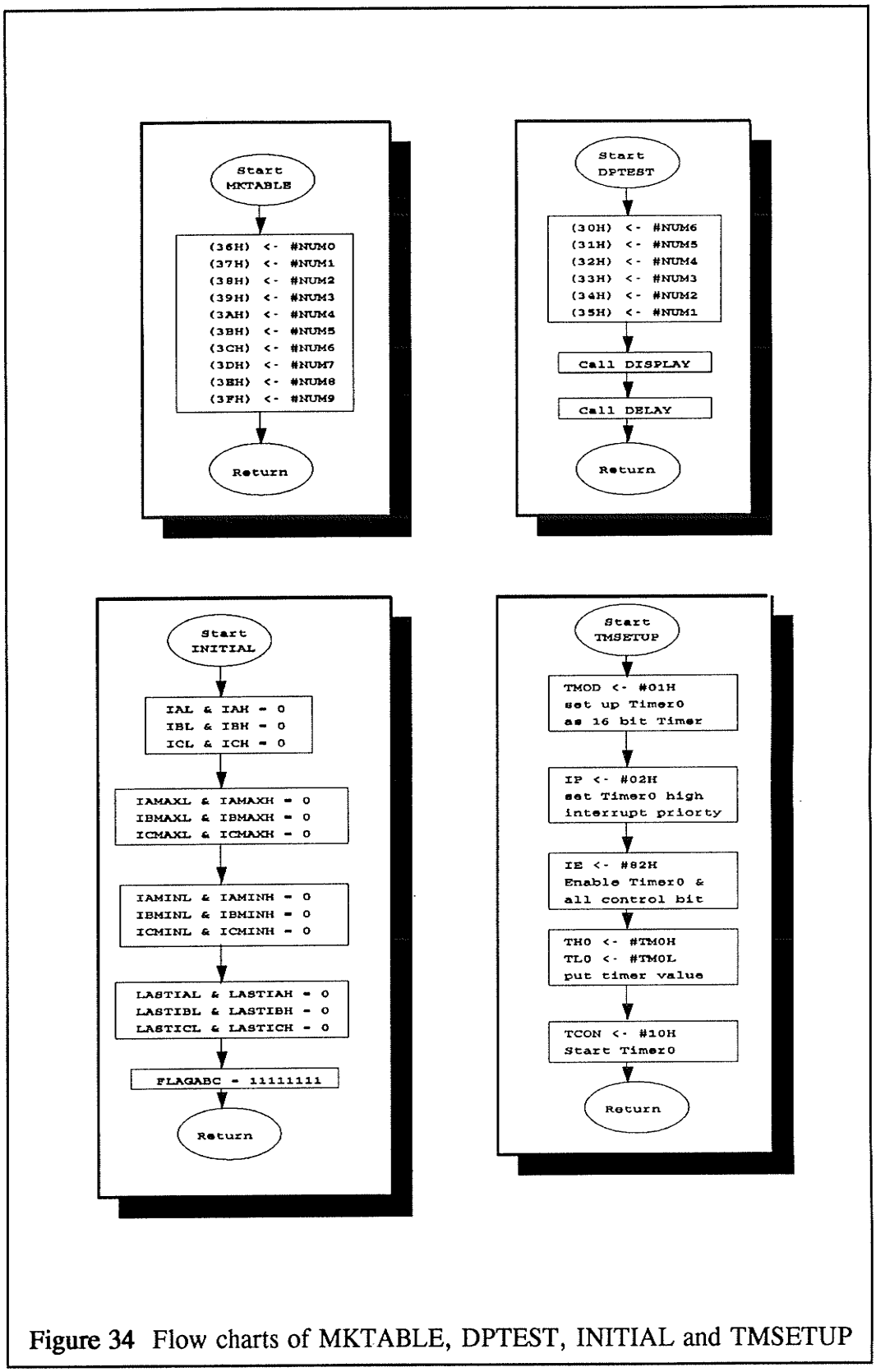


Figure 34 Flow charts of MKTABLE, DPTEST, INITIAL and TMSETUP

5.3.1 The subroutine GETVOLT

The functions of the subroutine GETVOLT include operating the AD converters to perform analog to digital conversions of voltages from the linear coils, reading the digitized data, and storing the most significant eight bits as variables VA, VB and VC for data from phase A, phase B and phase C respectively.

The flow chart of the subroutine GETVOLT is shown in Figure 35. At the beginning, the subroutine resets the data port Port1 and control port Port3. Next, as described in Section 2.2 of this chapter and Figure 31, the AD converters start conversions while sensing negative going pulses on the WR line and Chip Select lines CSA, CSB and CSC for phase A, phase B and phase C. After initializing the AD conversions, the subroutine starts polling the data ready signal INTA for phase A. Upon sensing a low transition on INTA, the most significant byte (MSB) of the digitized data is read into Accumulator A by sending negative going pulses on CSA and RD lines. The absolute value is taken by subroutine CHSIGN, and stored as variable VA. Then, the least significant byte (LSB) is read by sending negative going pulses on CSA and RD lines again. However, the LSB of data is not used and not stored. Then, data from phase B and phase C are stored as variables VB and VC in the same manner.

Since the ADC1001 converter only reads positive analog voltage, the input signals from the air-cored coils are dc biased 2.5V which is half of the full range of the converter input. Thus, the zero reference point of the converter is at 2.5V, any input value below 2.5V is actually negative and any value above 2.5V must subtract 2.5V to get the actually voltage. The subroutine CHSIGN is used for adjusting the data to obtain the actual value. An additional flow chart is shown in Figure 35 for subroutine CHSIGN.

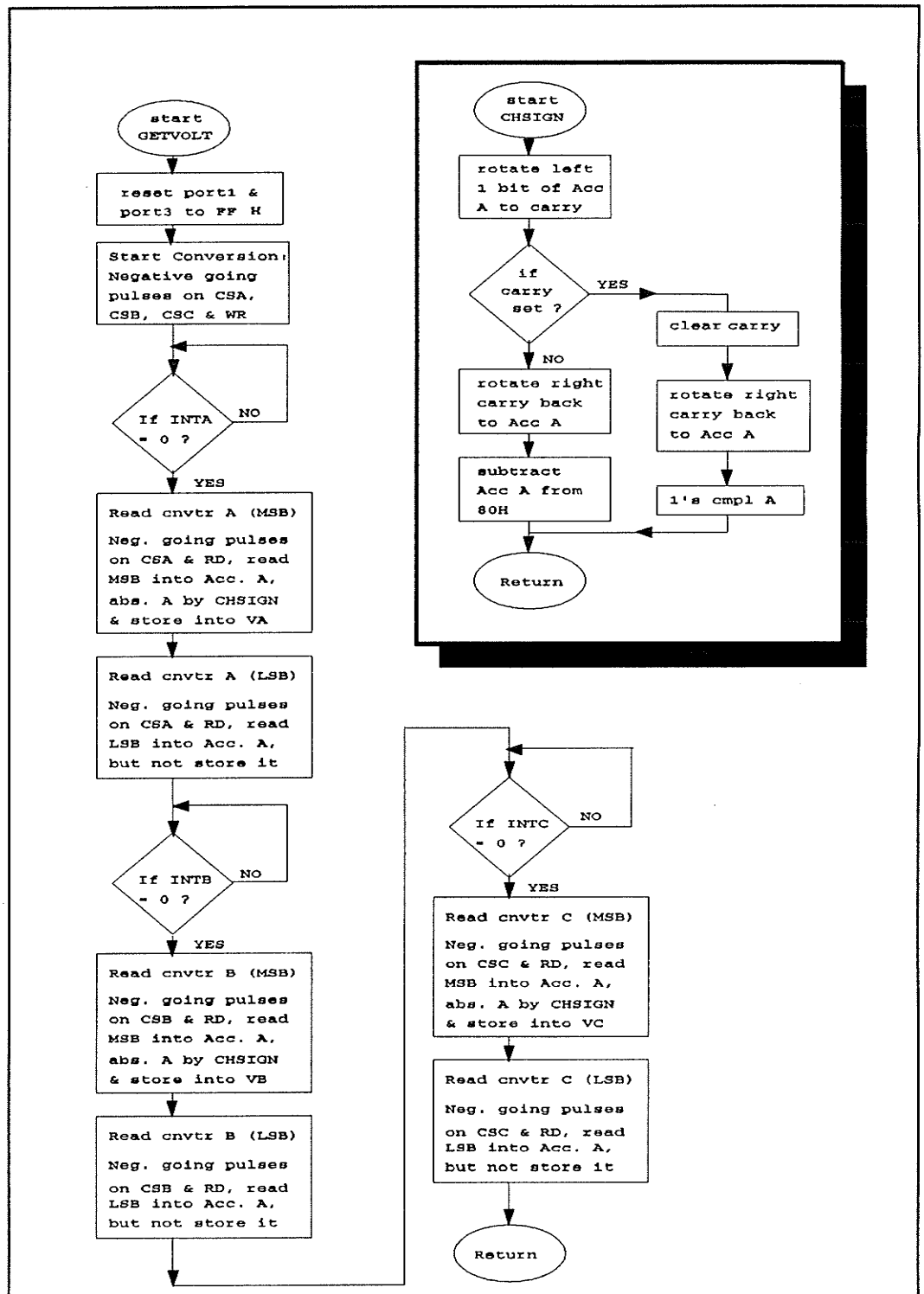


Figure 35 The flow chart of subroutine GETVOLT

5.3.2 The subroutine CURRENT

After reading the voltages of the three coils, the subroutine CURRENT compensates the readings by applying the mathematical model developed in Section 1 and calculates actual currents on each phase. The equations are:

$$IA = K1 \times VA - K3 \times VB - K4 \times VC$$
$$IB = K2 \times VB - K3 \times VA - K3 \times VC$$
$$IC = K1 \times VC - K3 \times VB - K4 \times VA$$

The program **K1234** yields the values of *K1, K2, K3* and *K4* from the experimentally measured values of *I, Va, Vb* and *Vc*. In the case used, *K1, K2, K3* and *K4* are equal to 128, 128, 3 and 1 respectively.

Figure 36 is a flow chart of the subroutine CURRENT. The subroutine first multiplies the single byte VA with constant K1 by calling the subroutine MULTI and the product is a two bytes result. The MSB is stored into IAH and the LSB is stored into IAL. Second, the MSB and LSB of the product of VB times K3 are subtracted from IAH and IAL. Third, the high and low bytes of the product of VC times K4 are subtracted from IAH and IAL. K1, K3 and K4 are calculated in advance and declared as constants at the beginning of the program. IAH and IAL are program variables which stand for the high byte and the low byte of current in phase A. Then currents in phase B and phase C are computed at the same manner. The results are stored as variables IBH, IBL, ICH and ICL which are the high byte of phase B current, the low byte of phase B current, the high byte of phase C current and the low byte of phase C current, respectively.

In addition, the multiplication of signed values must be arranged since negative values of VA, VB and VC are represented by 1's complement numbers. The subroutine MULTI is written for handling multiplications of signed values. A supplementary flow chart in Figure 36 explains its algorithm.

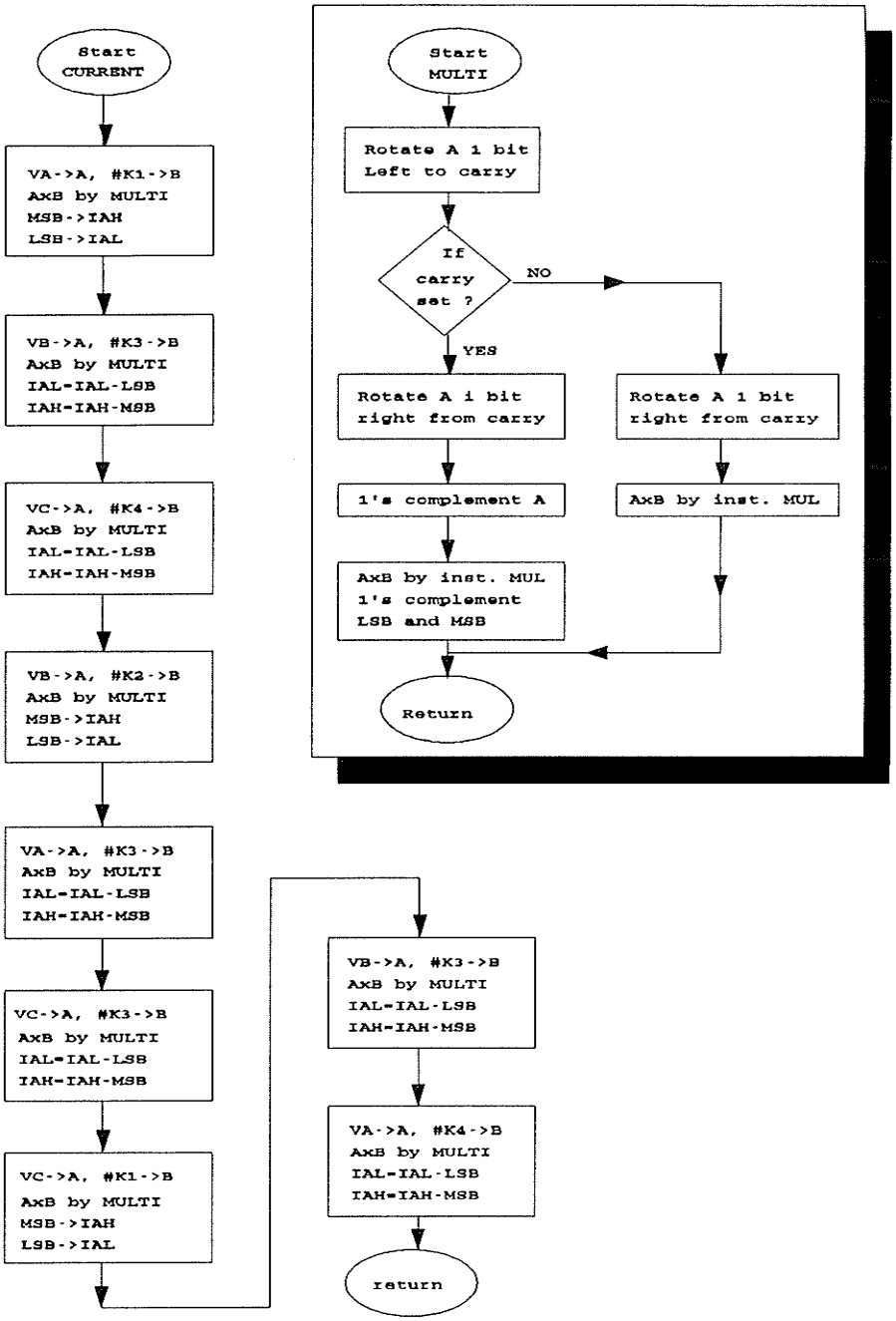


Figure 36 The flow chart of the subroutine CURRENT

5.3.3 The subroutine IPEAK

Timer 0 initializes the interrupt routine at fixed time intervals. Each time the routine captures a phase voltage and computes the actual current by the subroutine CURRENT. One of the best ways to represent these sinusoidal signals is by their peak amplitudes. Therefore, the subroutine IPEAK is written to search for the negative and positive peak values of each phase current in each cycle. The subroutine does not only look for the maximum, but also searches for the minimum point which is useful for checking dc offset in subroutines ISTBA, ISTBB and ISTBC which are discussed in later sections.

The flow chart of the subroutine is shown in Figure 38. The variable FLAGABC, as shown in Figure 37, is used for bookkeeping whether the maximum or the minimum point is currently being sought. Bit-0 of FLAGABC tracks phase A while bit-1 and bit 2 track phase B and phase C respectively. A set bit indicates a maximum search; a cleared bit indicates a minimum search.

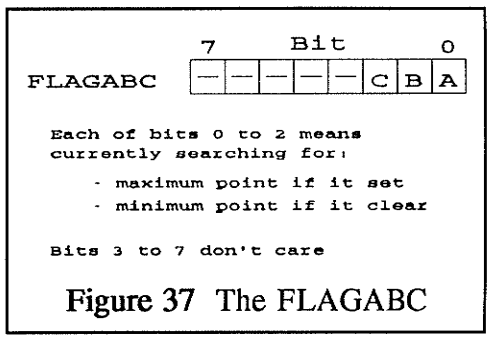


Figure 37 The FLAGABC

The subroutine first checks the flag bit (FLAGABC-0) for phase A. The subroutine then checks whether the previous current value (LASTIAH and LASTIAL) is larger or not than the present current value (IAH and IAL). If the previous value is larger than present value, then the maximum value is found and equal to the previous value. The high and low bytes of the maximum value are stored into variables IAMAXH and IAMAXL. Meanwhile, the subroutine clears FLAGABC-0 and stores the present value (IAH and IAL) as previous value (LASTIAH and LASTIAL) for the next interrupt. Conversely, if the previous value is smaller than the present value, a maximum value is not found yet. The

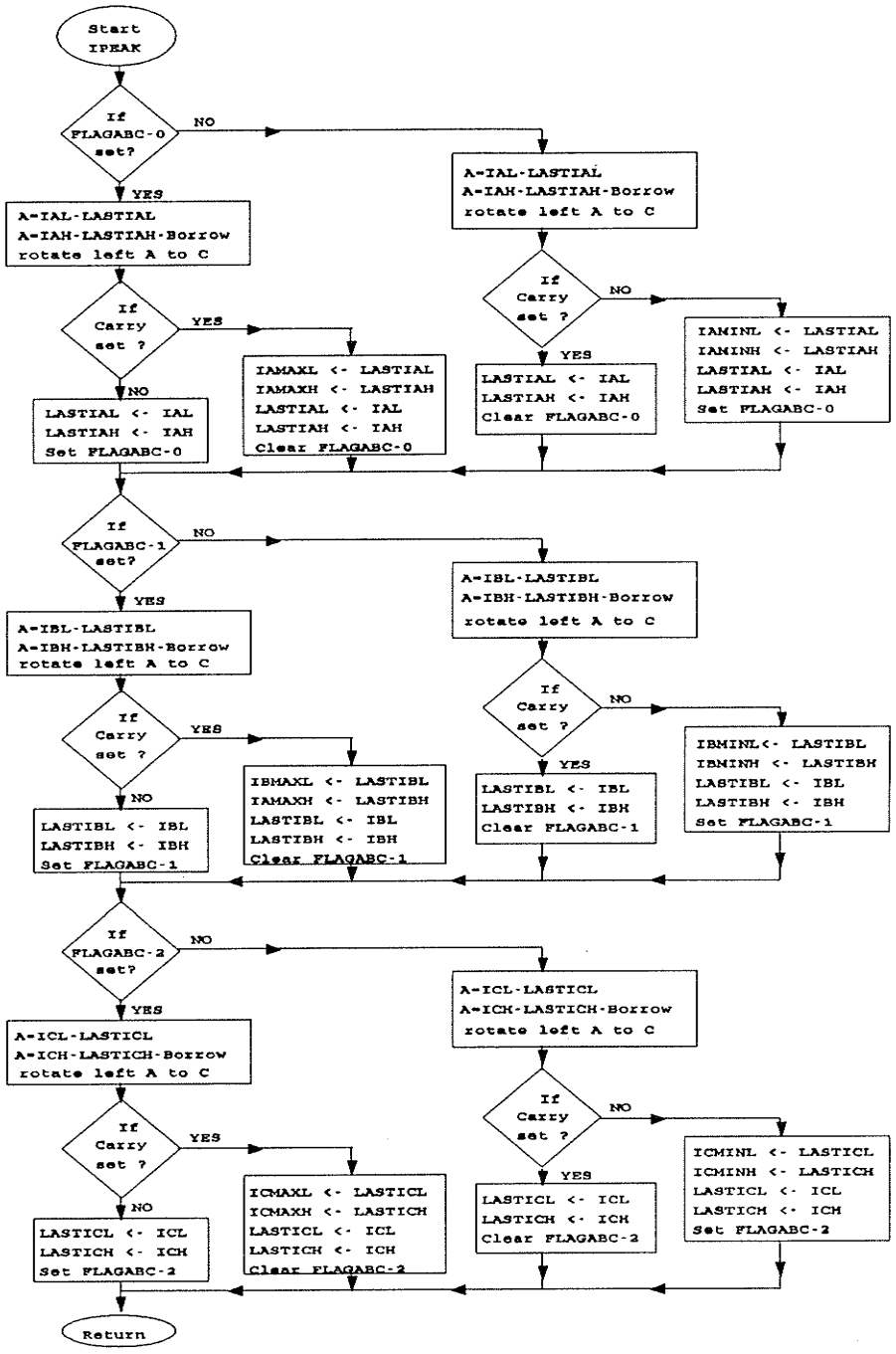


Figure 38 The flow chart of the subroutine IPEAK

subroutine then keeps the FLAGABC-0 set and updates the previous values.

On the other hand, if the subroutine found that the flag bit was clear at the beginning, it will search for the minimum point instead of the maximum, and a similar process is used.

Consequently, similar procedures are done in phase B and phase C to search for their maximum and minimum points. Variables used in this subroutine are:

IAH, IAL	- High and Low bytes of present phase A current
LASTIAH, LASTIAL	- High and Low bytes of previous phase A current
IAMAXH, IAMAXL	- High and Low bytes of maximum value of phase A current
IAMINH, IAMINL	- High and Low Bytes of minimum value of phase A current
IBH, IBL	- High and Low bytes of present phase B current
LASTIBH, LASTIBL	- High and Low bytes of previous phase B current
IBMAXH, IBMAXL	- High and Low bytes of maximum value of phase B current
IBMINH, IBMINL	- High and Low Bytes of minimum value of phase B current
ICH, ICL	- High and Low bytes of present phase C current
LASTICH, LASTICL	- High and Low bytes of previous phase C current
ICMAXH, ICMAXL	- High and Low bytes of maximum value of phase C current
ICMINH, ICMINL	- High and Low Bytes of minimum value of phase C current

5.3.4 The subroutine MONITOR

While the Timer 0 interrupt routine is sampling voltages and computing peak currents periodically, the subroutine MONITOR in the main program is continuously monitoring peak currents on each phase calculated by the timer interrupt routine and displaying them on the 7-segment display.

The flow chart of the subroutine MONITOR is shown in Figure 39. In fact, the subroutine MONITOR calls a series of other subroutines to achieve its functions. These subroutines are ISTBA, ISTBB, ISTBC, BITODEC, PUTDATA, DISPLAY and DELAY. Their functions and algorithms are discussed in following sections.

MONITOR first calls the subroutine ISTBA to eliminate any offset in phase A which is induced by truncation errors in numerical calculations. Secondly, the subroutine puts the most recent maximum values of phase A into registers R0 and R1 and calls the subroutine BITODEC. BITODEC converts the hexadecimal numbers in registers R0 and R1 to decimal numbers, and stores the decimal numbers into registers R2 and R3. Thirdly, MONITOR calls the subroutine PUTDATA to extract each digit at a time from the number stored in R2 and R3, matches each digit to its displaying data, arranges and stores these data into the displaying array. The display array is actually a set of six successive locations (from 30H to 35H) assigned in the memory. Fourth, the subroutine DISPLAY is called to display the data from the display array to the 7-segments displays. In addition, a delay loop is run, by calling the subroutine DELAY, to keep the data long enough in the buffer of the display driver to avoid display blanking. In the specifications of the ICM7218A display driver, a blanking time of approximately 10μsec occurs between digit strobes.

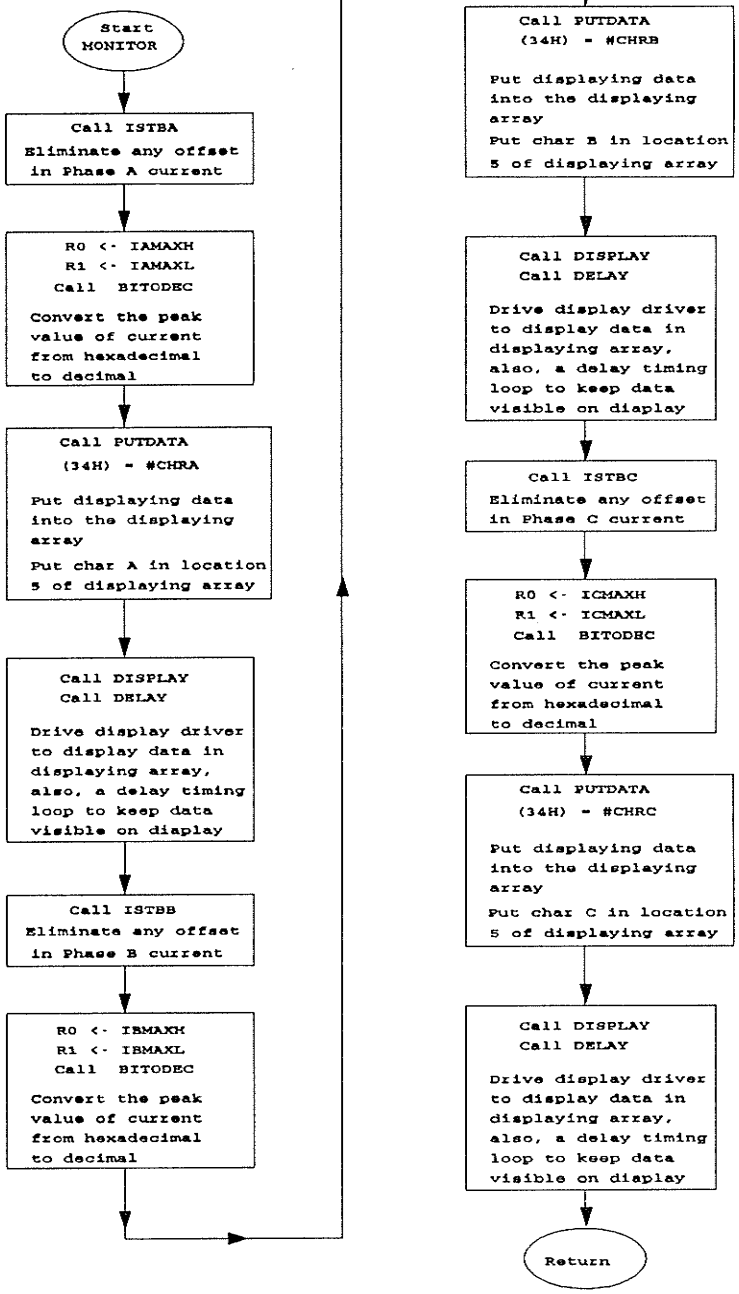


Figure 39 The flow chart of the subroutine MONITOR

5.3.5 Subroutines ISTBA, ISTBB, ISTBC and ISTBLZR

Truncation errors always exist in numerical computations. When truncation errors accumulate in calculations for a sinusoidal current signal, the signal gradually drifts and finally offset is imposed on the signal.

Basically, ISTBA, ISTBB and ISTBC are subroutines written for eliminating the offset from signals of their corresponding phase. Flow charts of these subroutines are shown in Figure 40. Each of them are tailored to meet variables of their phase. For example, ISTBA uses IAMAXH and IAMAXL as inputs but ISTBB uses IBMAXH and IBMAXL.

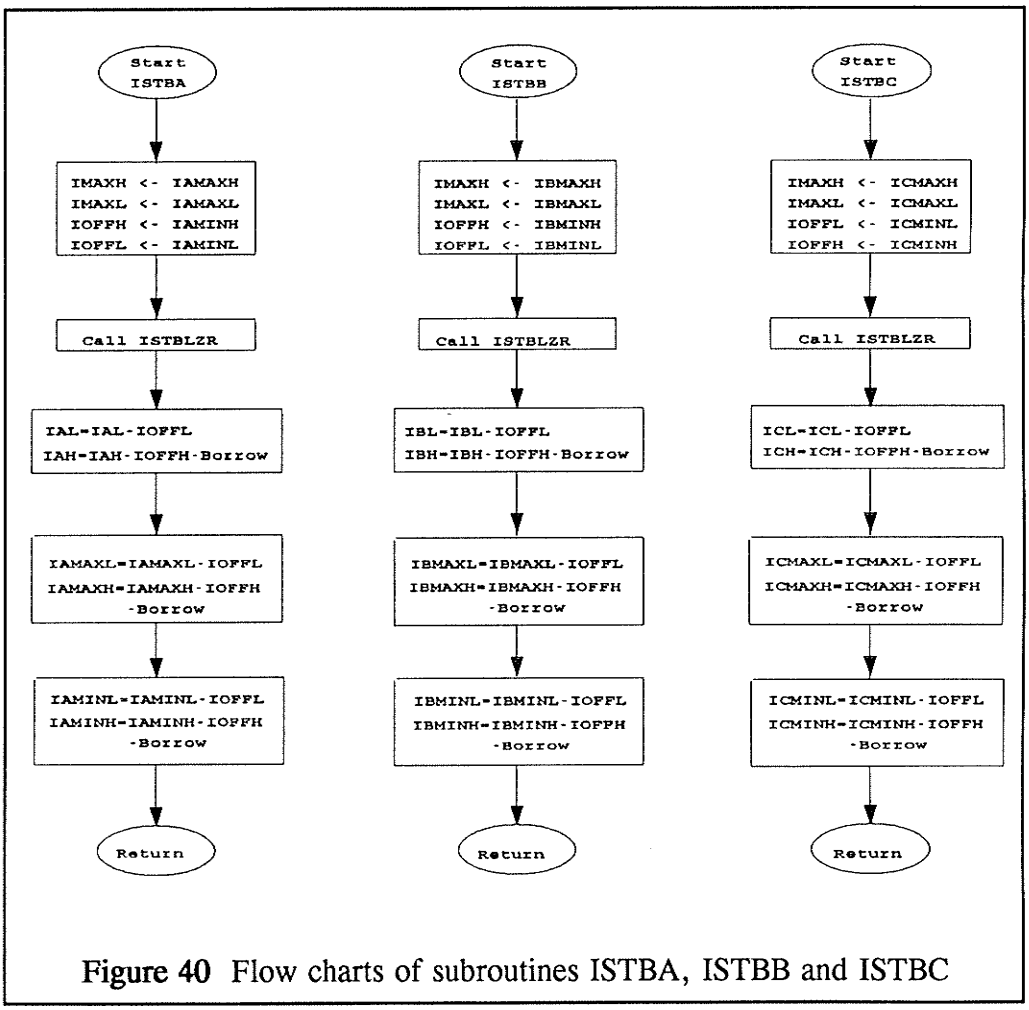


Figure 40 Flow charts of subroutines ISTBA, ISTBB and ISTBC

These subroutines first obtains maximum and minimum values of their phase and then calculates offset of their phase by calling the subroutine ISTBLZR. At last, they subtract the offset from their signals.

The flow chart of the subroutine ISTBLZR is shown in Figure 41. The function of this subroutine is to calculate the amount of offset in the current. The offset is actually the displacement of the mid-point, between maximum and minimum values of the sinusoid wave, from zero.

The subroutine ISTBLZR first checks the sign of the maximum value (IMAXH). If it is positive, then check whether the minimum value (IOFFH) is negative or not. If it is true, then adds the maximum value to the minimum value by calling SUMI. If the carry is set after the addition, then the subroutine simply calculates the mid-point by calling MIDPT. If the carry is not set, then the subroutine first calls CPLMT to take one's complement values, second calls MIDPT to calculate the midpoint and last calls CPLMT to take one's complement again. However, if the minimum value is positive, then it calls SUMI and MIDPT to find the mid-point. If the maximum value is checked as negative at begin, then the subroutine adds the values by calling SUMI and find the mid-point by MIDPT with one's complement values.

Flow charts of subroutines SUMI, MIDPT and CPLMT are shown in the supplementary diagram in Figure 41. SUMI adds the maximum current (IMAXH and IMAXL) with the minimum current (IOFFH and IOFFL). IOFFH and IOFFL have been assigned values of the minimum current in ISTBA, ISTBB or ISTBC. MIDPT calculates the midpoint by dividing IOFFH and IOFFL by 2. CPLMT takes the one's complement values of IOFFH and IOFFL.

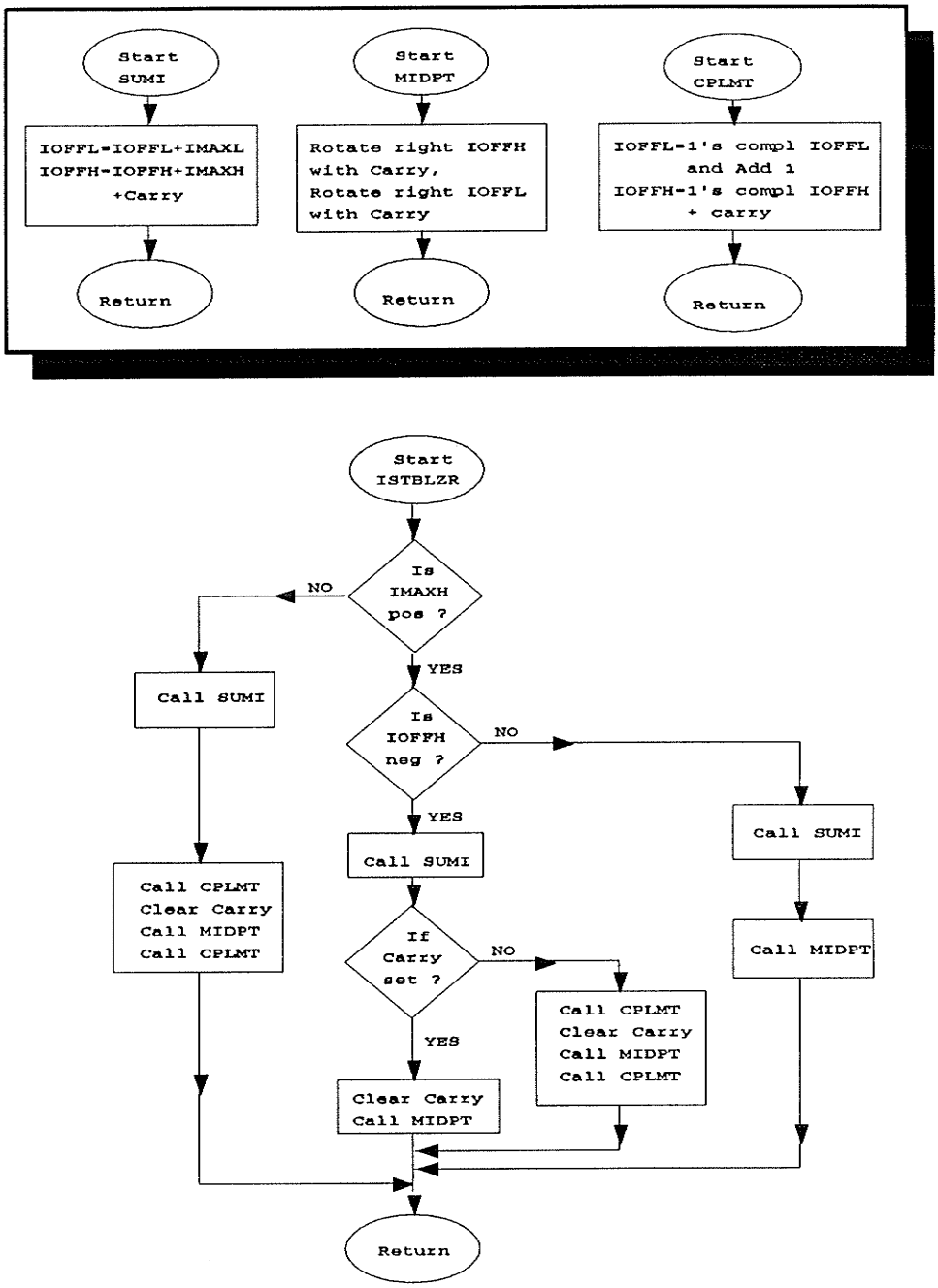


Figure 41 The flow chart of the subroutine ISTBLZR

5.3.6 The subroutine BITODEC

After the correct peak value of the current is obtained, the subroutine BITODEC converts this value from hexadecimal numbers to decimal numbers for the decimal readings on the 7-segment display.

The flow chart of the subroutine BITODEC is shown in Figure 44. The subroutine accepts the data in registers R0 and R1 as inputs where R0 is MSB and R1 is LSB. It converts the hexadecimal numbers in R0 and R1 to decimal numbers by following the conversion charts in Figure 42. It stores the decimal results into registers R2 and R3 where R2 is MSB and R3 is LSB. The flow chart may look complicated, in fact, the algorithm is simple. At the beginning, the subroutine calculates the absolute value of the data stored in R0 and R1 by calling the subroutine ABSR. Subsequently, it checks each of the bits in R0 and R1, and adds the weight of bits which are set. The weight of each bit is indicated below it in Figure 42. Bits 0 to 6 are neglected and bit 15 is the sign bit. If the carry is set while a weight being added to R3, BITODEC calls the subroutine CARRYIN to add the carry to R2. Flow charts of subroutines ABSR and CARRYIN are shown in Figure 43.

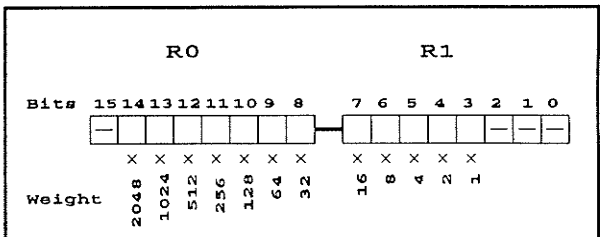


Figure 42 A conversion chart of BITODEC

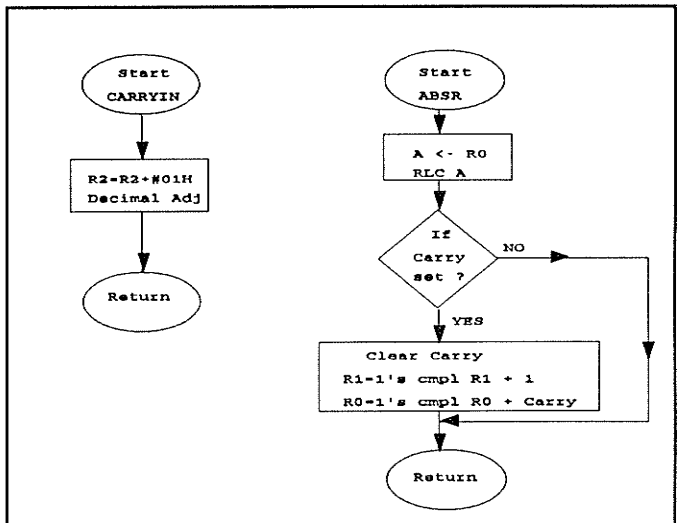


Figure 43 Flow charts of CARRYIN and ABSR

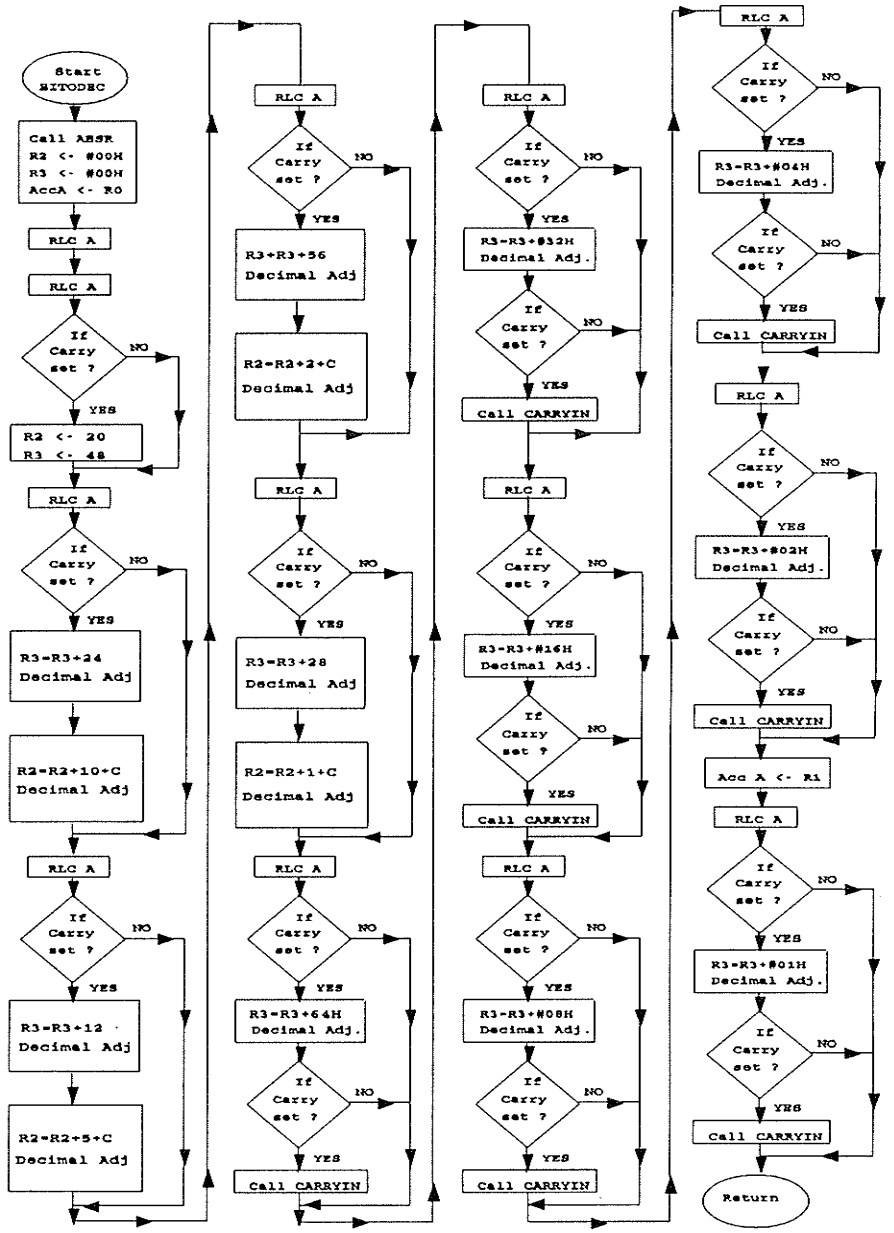


Figure 44 The flow chart of the subroutine BITODEC

5.3.7 The subroutine PUTDATA

The subroutine PUTDATA is written for extracting each of the binary coded decimal (BCD) numbers in the registers R2 and R3, matching these BCD numbers to their display data, and put these data into a display array. Each of the registers R2 and R3 contains two BCD numbers, as shown in Figure 45, one in the upper half-byte and one in the lower half-byte. These BCD numbers represent the four numerical digits being displayed on the 7-segment display.

The subroutine PUTDATA, as shown in Figure 46, extracts each BCD digit at a time from R2 and R3 by calling the subroutine EXTRACT. Also, EXTRACT matches the display data for a digit by calculating its location in the table generated by MKTABLE. In addition, EXTRACT puts the display data into the corresponding location in the display array. Finally, PUTDATA puts the display data of a character "I" into the last location (36H) of the display array as an abbreviation for "current".

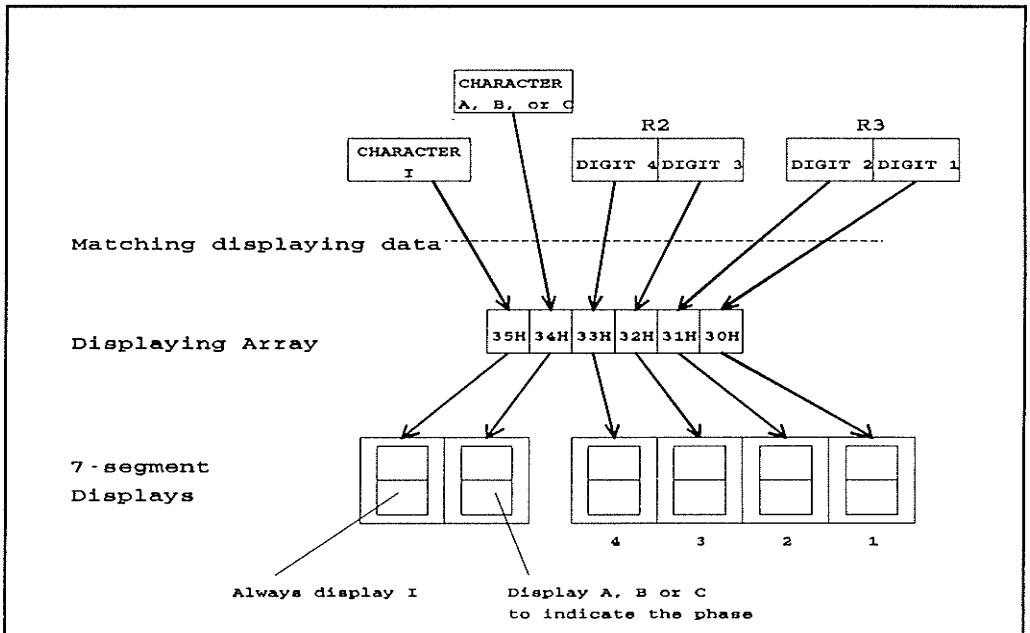


Figure 45 The displaying array related to R2, R3 and 7-segment displays

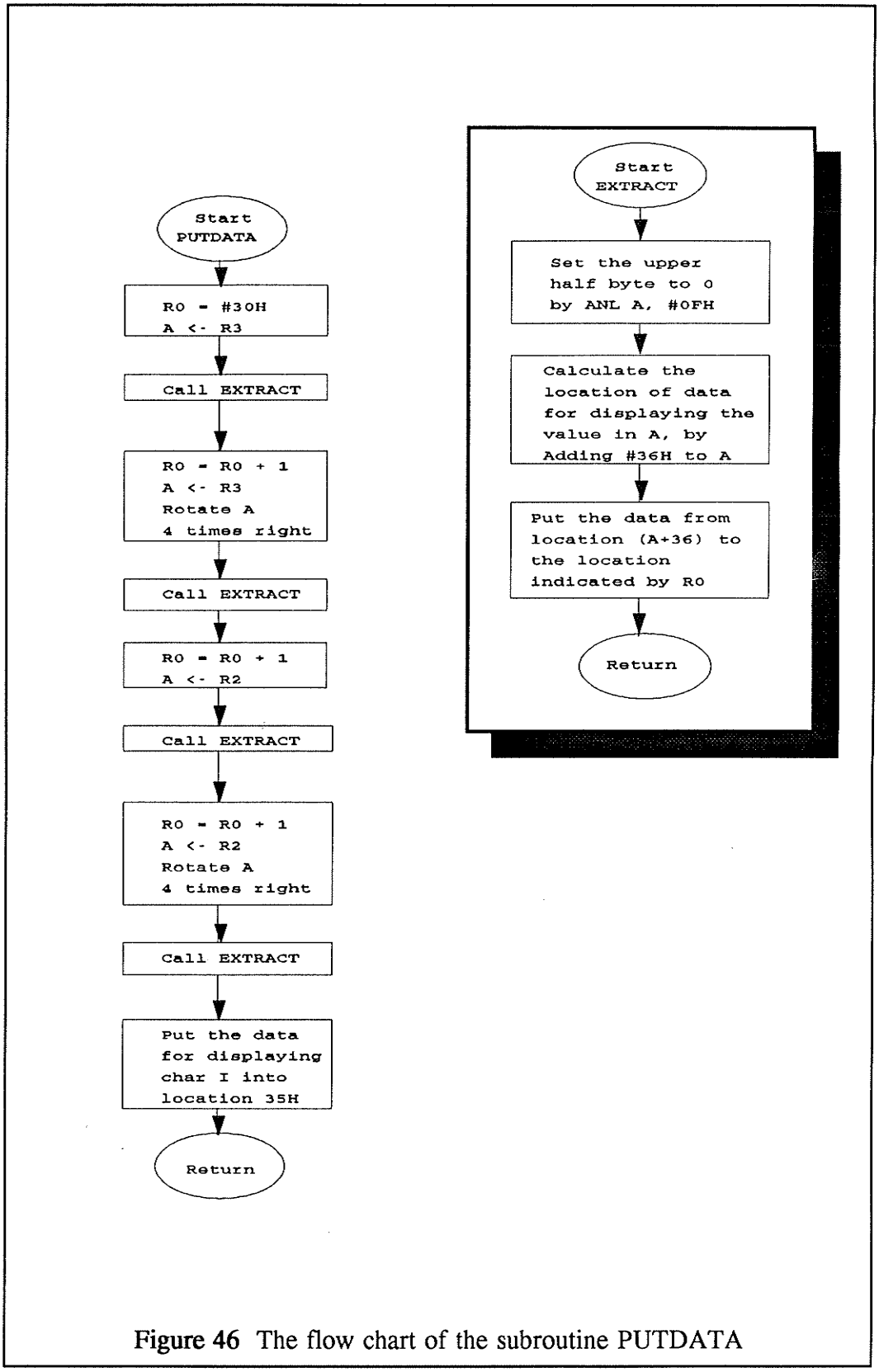


Figure 46 The flow chart of the subroutine PUTDATA

5.3.8 The subroutine DISPLAY

The final step of this software is to display the current on the 7-segment display. The key component for displaying data on 7-segment display is the multiplexed display driver. The subroutine DISPLAY follows the timing diagram in Figure 47 to operate the ICM7218A display driver.

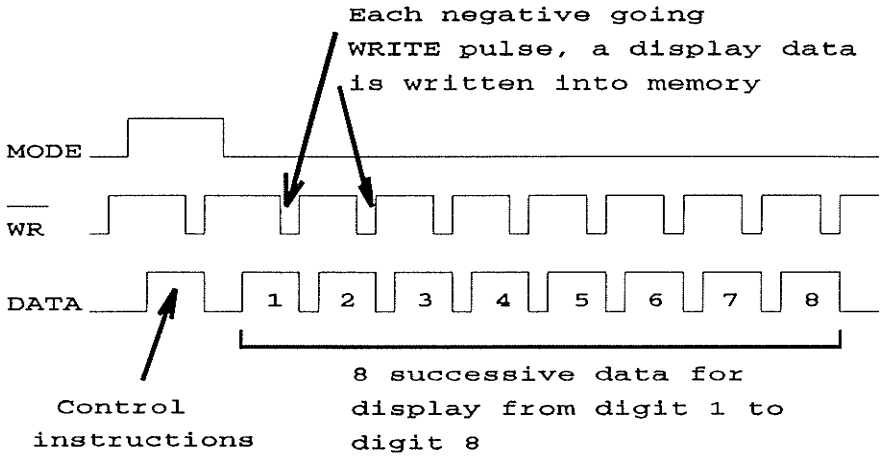
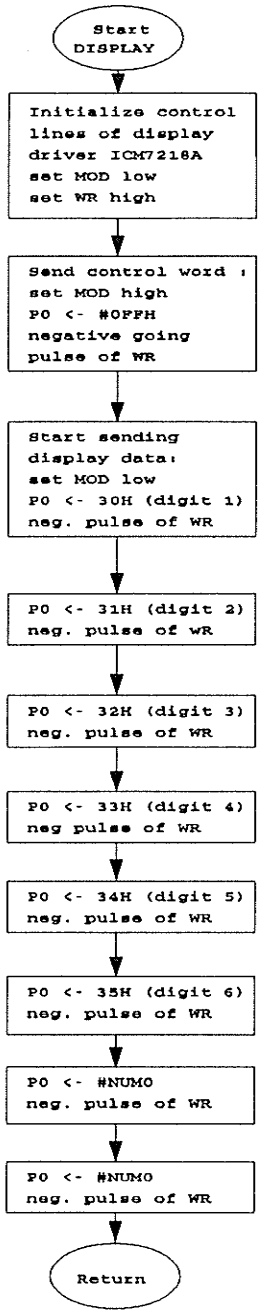


Figure 47 The timing of writing data to the ICM7218A display driver

The flow chart of the subroutine DISPLAY is shown in Figure 48. After initialization, the subroutine first sends the control word FF to the display driver by setting the MODE line high and generating negative pulse on the WRITE line. Consequently, with the MODE line set low, data of the six digits are sent successively to the display driver and each data is sent along with a negative going pulse on the WRITE line. The ICM7218A display driver is designed for handling eight digits and must be updated by refreshing all 8 digits. Thus, there are two extra dummy writes at the end.



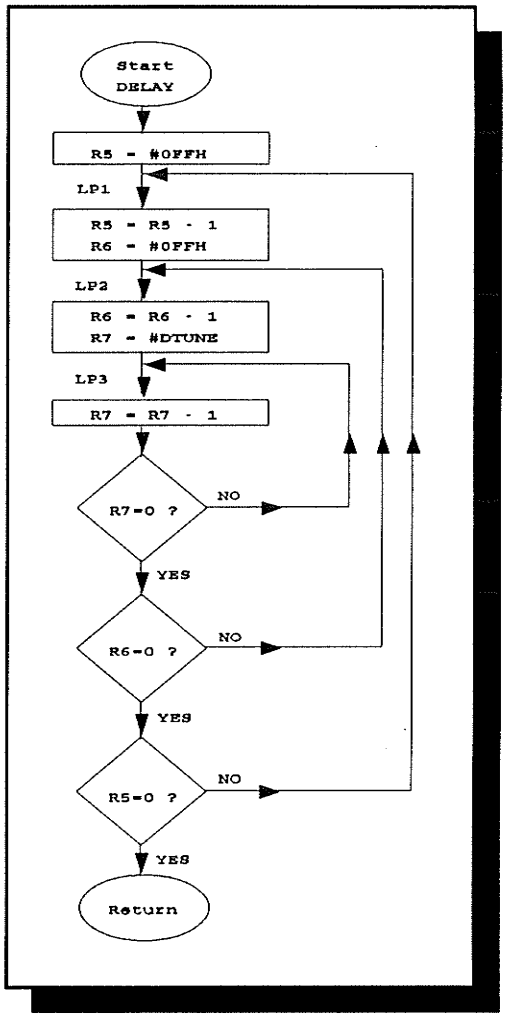


Figure 48 Flow charts of subroutines DISPLAY and DELAY

The subroutine DELAY shown in the supplementary diagram in Figure 48 is a timing delay loop to ensure the display does not blank. Based on the specifications of the ICM7218A display driver, it has a blanking time of approximately 10μs occurs between digit strobes. Referring to the flow chart, there is a delay tuning parameter DTUNE which is adjustable to change the duration of the time delay.

Chapter VI
Setup and Use of the Compensated Ammeter

6.1 Using the Compensated Ammeter

The actual layout of components on the system board is shown in Figure 49. The schematic diagram of the compensator is illustrated in Appendix E. In order to perform measurements on the three-phase bus-bar, users can simply connect the outputs of the current sensing devices to the input port of the compensator and provide it with a proper power supply. The magnitude of current then shows on the lower 4 digit display and the phase designation shows on the upper 2 digits. Currents are displayed in turn on the display with the sequence of phase A first, phase B, phase C and then restart from phase A. The reading of the current stays on for 0.5 second before the next phase is displayed. The system board can be reset by pressing the RESET button on the upper right hand corner.

6.2 Hardware Calibration

As shown in Figure 49, there are two sets of variable resistors that can be adjusted by a small screwdriver. The set on the right hand side of the board is used for adjusting dc offsets of the amplifiers. The set below the digital display is used for adjusting gains of the amplifiers.

6.2.1 Amplifier dc offsets

This set of three variable resistors is important for achieving precise measurements. As mention in Section 2.1 in Chapter V, the AD converter cannot

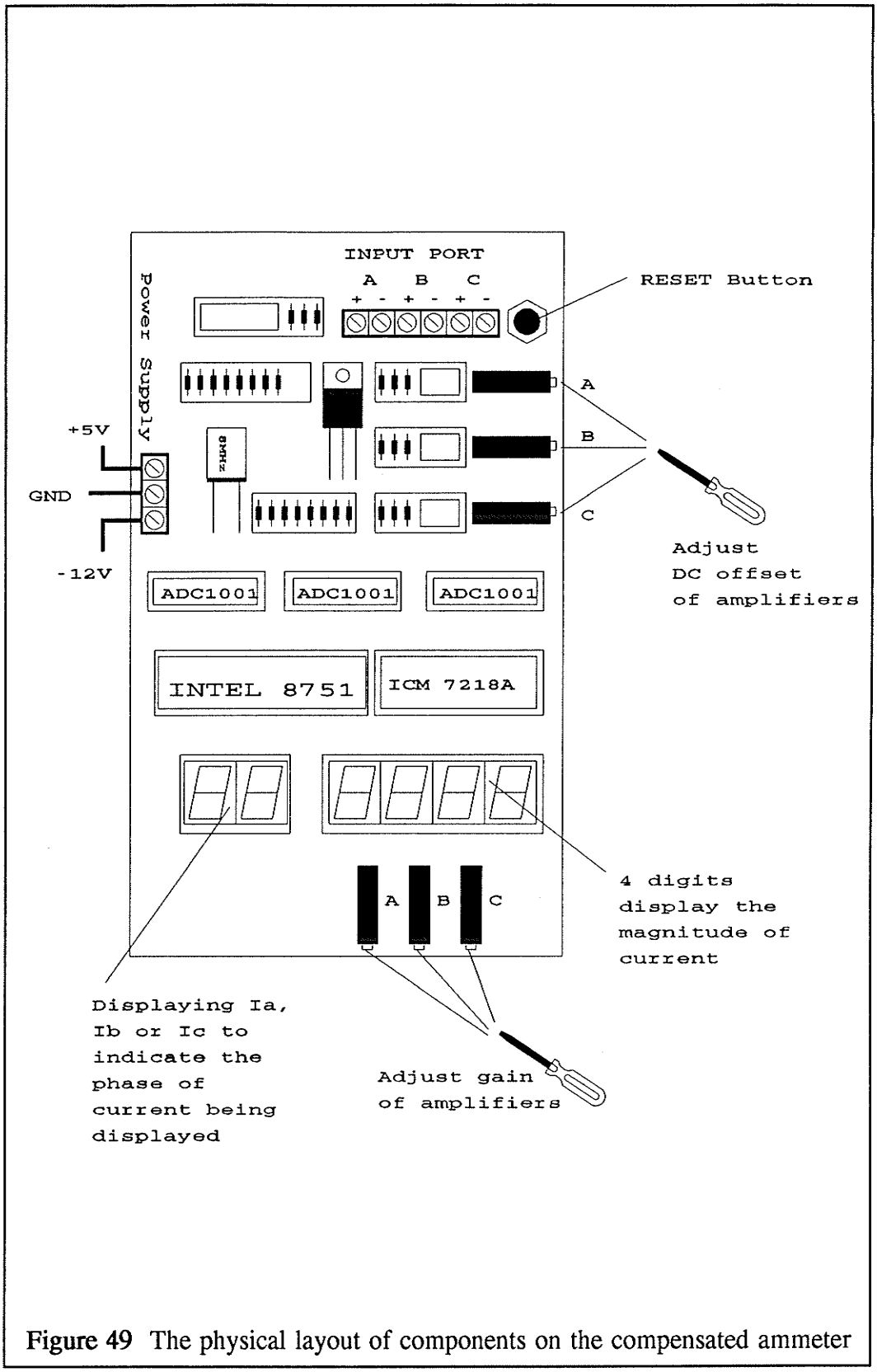


Figure 49 The physical layout of components on the compensated ammeter

accept negative input so that input signals are biased 2.5V at the amplifier-integrator. In order to obtain a correct reading of current, it must keep the dc offset of the amplifiers exactly 2.5V.

The procedure of adjusting the dc offset is to ground the input of the amplifier, then adjust the corresponding resistor while measuring the dc voltage by a voltmeter until the voltage reaches 2.5V.

6.2.2 Gains of Amplifiers

Scaling of the current readings by a factor of 2 is done in subroutine BITODEC (see Section 5.3.6). Finer adjustment is done by adjusting the set of resistors at the bottom of the board. Each of these 1kΩ variable resistors is in series with a 1kΩ resistor to form the input resistance for the corresponding amplifier-integrator (See Appendix E).

The fine adjustment was done as follows: First, the phase B and C bus bars were disconnected from the current source so that no voltage was induced into the phase A coil by the phase B and C currents. Second, the current in phase A was set to an arbitrary value using a reference ammeter as an assumed standard (See the Accuracy of the reference ammeters in p.68). Third, the phase A reading of the compensated ammeter was matched to the reading on the phase A reference ammeter by gain adjustment as described above. Fourth, the output voltage of the phase A amplifier was measured by a rms voltmeter (HP 3400A) and recorded.

Fifth, the phase A bus bar was disconnected from the current source and then the phase B bus bar was connected to the phase A current source. Note that the magnitude of current in phase A current source was not changed so that the same current was still flowing in phase B bus bar. By observing the rms voltmeter, the output voltage of the phase B amplifier was adjusted to match the

output voltage of the phase A amplifier.

Lastly, the phase B bus bar was replaced by the phase C bus bar. Similarly, the output voltage of the phase C amplifier was adjusted to match that of the phase A.

These adjustments not only calibrate the readings of the compensated ammeter, but also maintain the symmetry of the mutual inductance matrix $\boldsymbol{M}$ even though there are tolerances on the dimensions of the coils and inaccuracy of resistors used in the amplifiers.

6.3 Software Adjustable Parameters

In the program written for the compensator, there are two set parameter can be alternated. One set are the compensation coefficients K1, K3 and K4, the other one is the delay tuning DTUNE.

6.3.1 K1, K3 and K4

As described in Section 1 of Chapter V, K1, K3 and K4 are constants dependent on the properties of current sensing devices.

6.3.2 DTUNE

The delay tuning DTUNE is used for changing the time that a reading stays on the digital display.

Chapter VII
Accuracy of the Compensated Ammeter

7.1 The Test Setup

In order to evaluate the performance of the compensated ammeter, an experimental setup was built and is shown in Figure 50. The most important components are the three copper bus bars (1¼" x ⅛" cross-sectional area), which lie flat and are parallel with 1½" spacing, to simulate the actual bus-bars arrangement in a three phase circuit breaker manufactured by Federal Pioneer Limited.

The currents were induced in the three phase bus bars from the three phase ac power supply along with the three 4.2Ω resistance loads. The magnitudes of the currents were adjusted by the three phase autotransformer which had a maximum rating of 45 A on each phase. This rating restricted the current available in the experiments. The analog ammeters are used for current reference and the oscilloscope is used to monitor the output voltages of the current sensors and the amplifier-integrators.

The air-cored linear coils (current senors) were mounted above and below each bus bar although the lower coils do not show in Figure 50. The outputs of these coils are directly connected to the input port of the compensated ammeter.

In the following sections, a series of tests including linearity, effectiveness of compensation and saturation will be discussed as well as the studies of harmonic content, frequency and the demagnetizing effect.

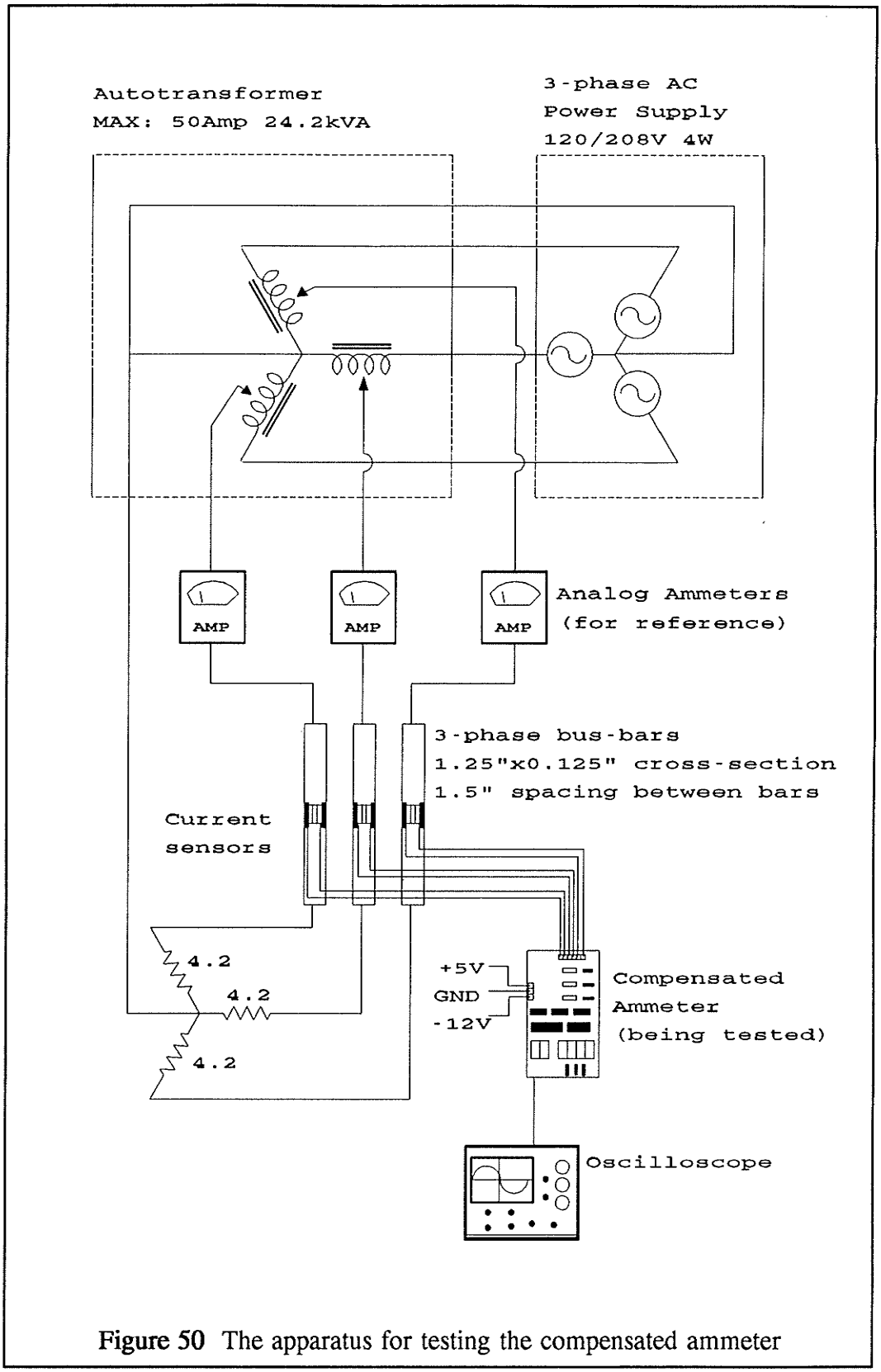


Figure 50 The apparatus for testing the compensated ammeter

Accuracy of the reference ammeters

As a partial check of the accuracy of the available reference ammeters, the consistency between three identical 50A full scale ammeters was checked by connecting them in series to the phase A current supply. The readings on the three ammeters were checked as the current in phase A is gradually increased from 0A to 45A. In the worst case, the reading on one of the ammeters was different from that of the others by 0.5A. This implies an inherent accuracy of about:

$$\frac{0.5\,A}{50\,A} \times 100\% = 1\%$$

7.2 Bit error

There are two sources of bit error: the A/D conversion and the output display.

A/D conversion bit error

Based on the specifications, the 10-bit A/D converter has a maximum full-scale error of 2 bits. Since the MSB is used for sign bit, the full scale of the A/D converter is 512 (9 bits). Therefore, the bit error is:

$$A/D\ bit\ error = \frac{2}{512} \times 100\% = 0.4\%$$

Output display bit error

The full range of the display is 256A. The bit error is:

$$Display\ bit\ error = \frac{1}{256} \times 100\% = 0.4\%$$

The total full-scale bit error

The total bit error is the sum of the A/D conversion bit error and the display bit error. Note that the total bit error is for full scale measurements. If a smaller measurement is made, the percentage of the bit error will increases.

$$The\ total\ bit\ error = 0.4\% + 0.4\% = 0.8\%$$

7.3 Linearity

One of the most important characteristics for metering purposes is the linearity. The linearity of the compensated ammeter was tested by comparing its readings and the readings of the analog ammeters when the magnitude of the currents in the bus-bars was gradually increased from 15 A to 45 A on each phase. The experimental data are shown in Table I and also plotted in Figure 51, Figure 52 and Figure 53. In the first column of Table I, the readings of the analog ammeters and the compensated ammeter were deliberately matched. Then, assuming the analog ammeters are linear, the values of higher currents verify the linearity of the compensated ammeter.

Table I The results of the linearity tests

The Current Readings on Phase A, B and C (Amperes)							
A	Analog Ammeter	18	24	30	35.5	41.5	44.5
	Compen. Ammeter	18	24	30	36	42	44
B	Analog Ammeter	15	20	25	30	35	37.5
	Compen. Ammeter	15	19	25	29	35	37
C	Analog Ammeter	14	19	24	29	34	36
	Compen. Ammeter	14	18	24	28	34	36

By observing the measurement data and the graphs, it is seen that the current readings on the compensated ammeter are linearly proportional to the magnitudes of the currents carrying by the bus-bars. Although the range of measurement is limited in this experiment, the system is fundamentally linear, so the compensated ammeter is capable of handling a higher range of current.

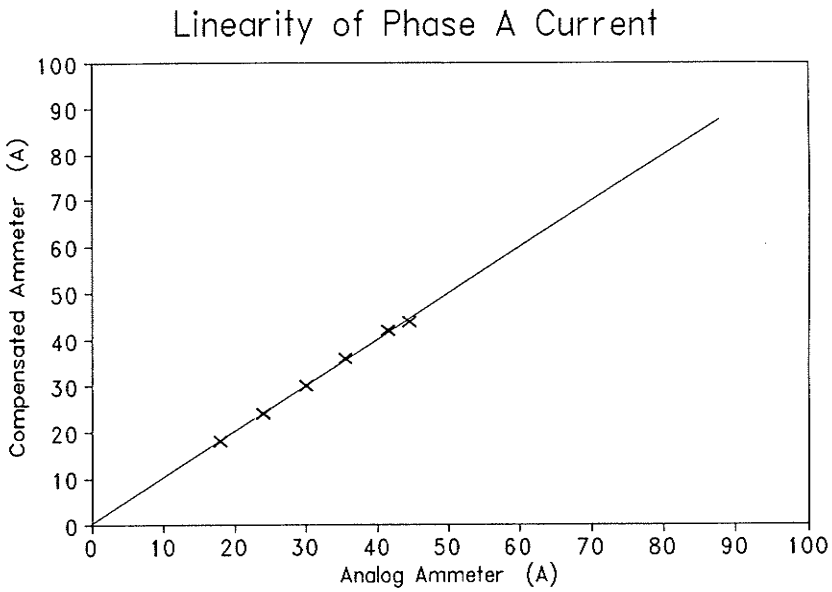


Figure 51 The phase A current readings on the compensated ammeter vs. that of the analog ammeter

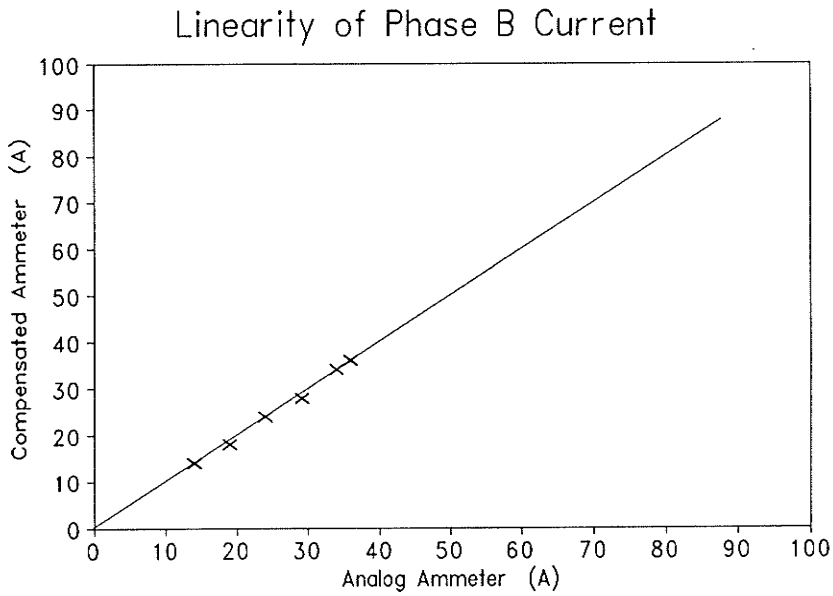


Figure 52 The phase B current readings on the compensated ammeter vs. that of the analog ammeter

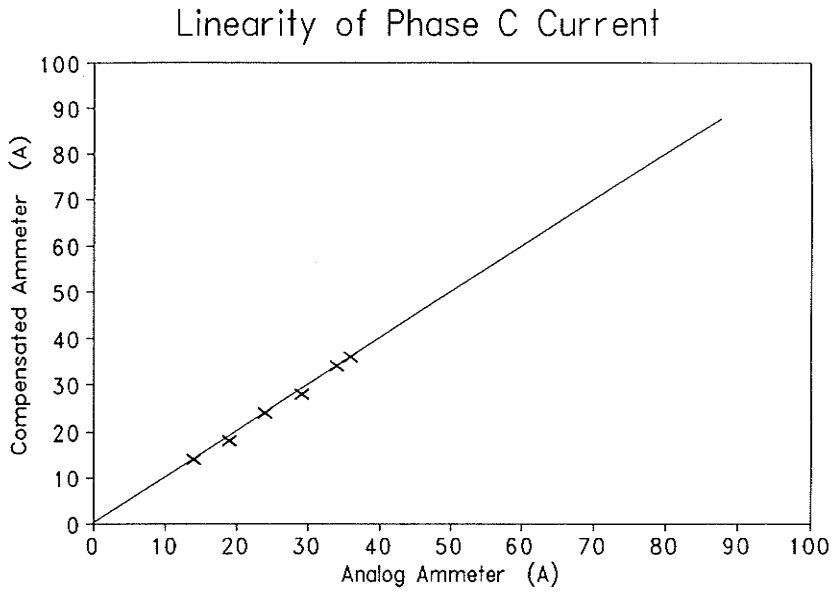


Figure 53 The phase C current readings on the compensated ammeter vs. that of the analog ammeter

7.4 Effectiveness of Compensation

Experiments were done to verify the effectiveness of the compensation of the ammeter. While a known current is circulated through only one of the three phase bus bars, the current readings for the other two phases are recorded. However, the amount of "cross-talk" between phases is comparatively small within the testing range (0 to 40A), sometimes it is less than the bit error of the AD conversions. Therefore, the gain of the amplifier-integrators is adjusted to 5 times larger so that a larger current is simulated and fed into the compensator. The simulated results are shown in Tables II, III and IV. Table II shows the experimental data when only the phase A current is present. The data for the phase B current only is in Table III, and that of the phase C only is in Table IV. Based on these experimental data, the performance of the compensation of the ammeter is proven satisfactory.

Table II The current readings on each phase while only the phase A current is conducting

Phase A (A)	Phase B (A)	Phase C (A)
100	0	0
151	0	0
200	0	0

Table III The current readings on each phase while only the phase B current is conducting.

Phase A (A)	Phase B (A)	Phase C (A)
0	98	0
0	150	0
0	185	0

Table IV The current readings on each phase while only the phase C current is conducting

Phase A (A)	Phase B (A)	Phase C (A)
0	0	102
0	0	148
0	0	180

7.5 Saturation

The saturation limit of the compensator is dependent on the input voltage range of the AD converter and the gain of the amplifier-integrator. The ADC1001 AD converter has an input range of ± 2.5V. The input of the AD converter is directly connected from the output of the amplifier-integrator. Thus, the maximum current that the compensated ammeter can handle is the current which generates a peak voltage of 2.5V to the input of the AD converter.

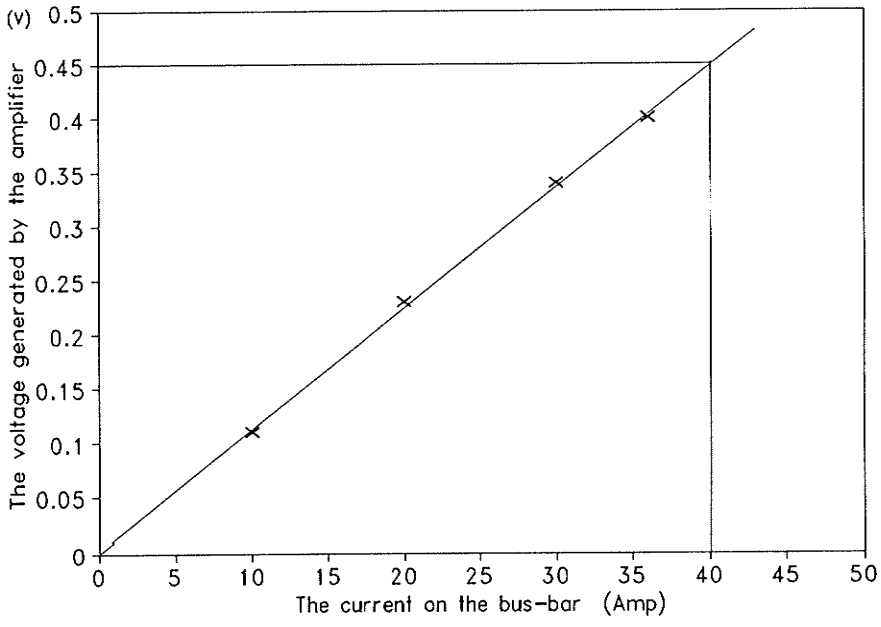


Figure 54 The output voltages of the amplifier-integrator vs. the current on the bus-bar

The maximum current that can be generated in the test setup is only 45 A and it is not sufficient to cause the ammeter to saturate. However, the output voltage of the amplifier-integrator, as shown in Figure 54, is linearly proportional to the current being measured. Thus, the saturation current of the compensated

ammeter can be calculated as follows:

$$The\ saturation\ current = \frac{2.5\ volt}{0.45\ volt} \times 40\ A \approx 222\ A$$

From Figure 54, a current of 40 A on the bus-bar causes the amplifier-integrator to generate 0.45V as an input to the AD converter. The saturation current is calculated and approximately equal to 222 A.

7.6 Harmonic Content

The compensated ammeter determines the magnitude of a current by detecting the peak value of the current instead of calculating its root mean square (rms) value. In fact, the rms value is an accurate and standard way to represent a current. However, for simplicity of assembly language programming, and since the primary objective of this thesis was to show that compensation would work, the algorithm first detects the peak value of the current at the most recent cycle and then scales the peak value to be the rms value.

Consider for example, a pure sinusoidal current with a peak value of 10 A, as shown in Figure 55, then its rms value is 7.07 A. The compensated ammeter would correctly display the current magnitude as 7 A. As shown in Figure 56, the rms value of a square wave with peak value of 10 A is also 10 A, but the ammeter would still display the current magnitude as 7 A and which is incorrect. Therefore, the ammeter can only handle a sinusoidal current signal with small harmonic content.

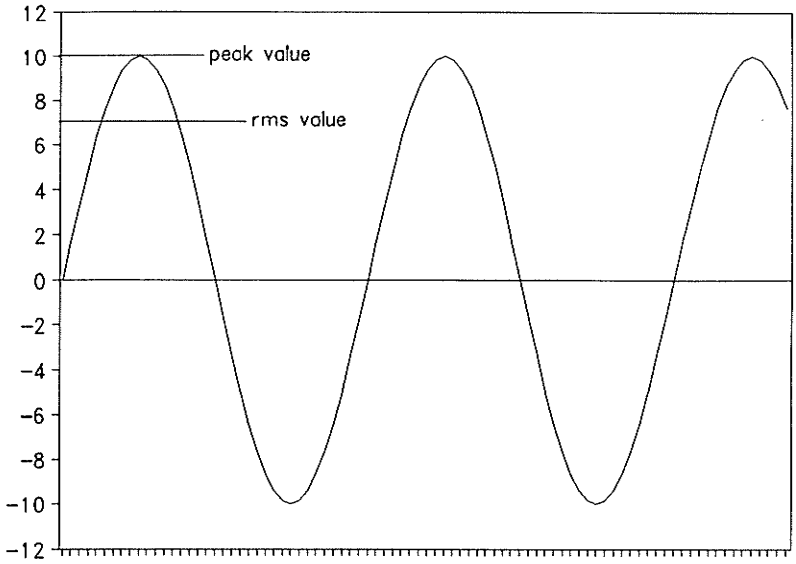


Figure 55 A sinusoidal current signal

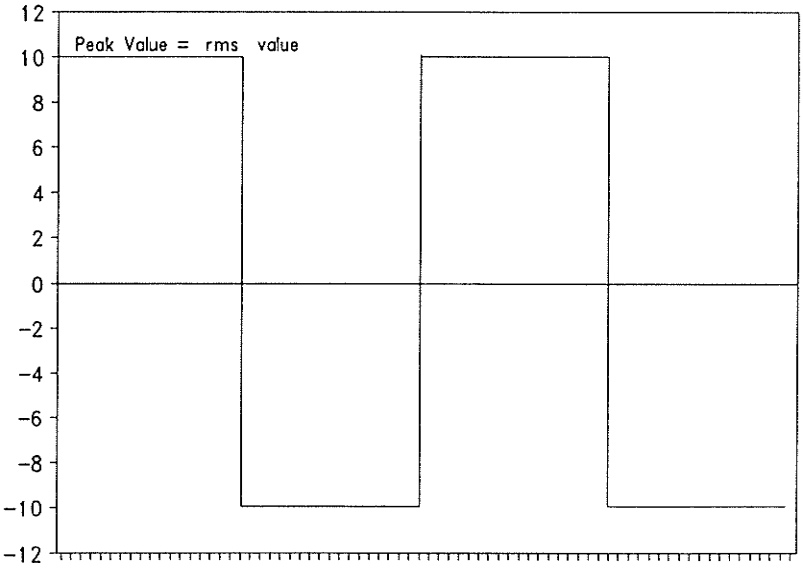


Figure 56 A square wave current

In order to handle a current with harmonic content, the algorithm should calculate the true rms value of the current by using the following equation.

$$\textit{The rms current} = I_{rms} = \sqrt{\frac{1}{N}\sum_{k=1}^{N} i_k^2}$$

where N is the number of samples in one cycle and i_k is the sampling current.

7.7 Frequency

The sampling error is dependent on both of the current frequency (F_o) and the sampling frequency (F_s). The maximum sampling error happens when the peak current value lies between two successive samples as shown in Figure 57.

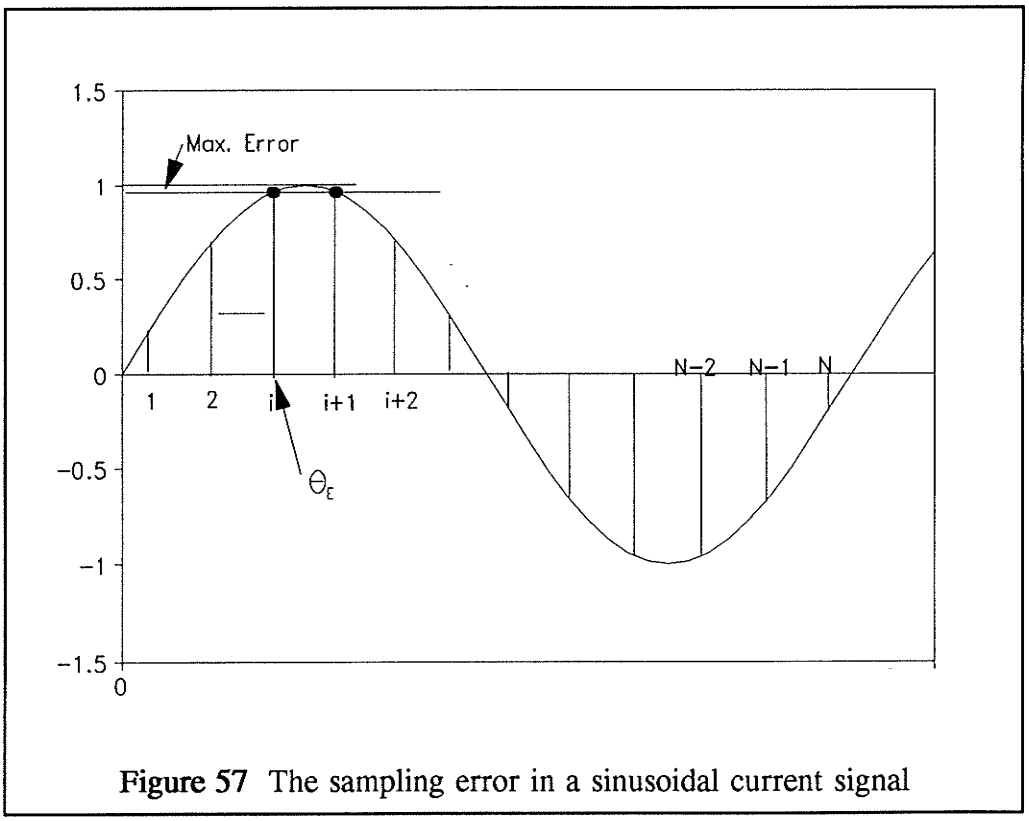


Figure 57 The sampling error in a sinusoidal current signal

Refer to Figure 57, N is the total number of samples in each current cycle and θ_E is the angle that the maximum error happens.

$$N = \frac{sampling\ frequency}{current\ frequency} = \frac{F_s}{F_o} \tag{36}$$

$$\theta_E = \frac{\pi}{2} - \frac{\pi}{N} = (\frac{N-2}{N})(\frac{\pi}{2}) = (1-2\frac{F_o}{F_s})(\frac{\pi}{2}) \tag{37}$$

Assuming the current signal is sinusoidal, the maximum sampling error is:

$$\begin{aligned} The\ maximum\ error &= [\,1 - \sin(\theta_E)\,] \times 100\% \\ &= \left[\,1 - \sin[\,(1 - 2\frac{F_o}{F_s})\,\frac{\pi}{2}\,]\,\right] \times 100\% \end{aligned} \tag{38}$$

The sampling frequency F_s relates to the clock (oscillator) frequency directly. The timer counts 620 machine cycles for each interrupt and 1 machine cycle consists of 12 oscillator period. Therefore, the sampling frequency is:

$$F_s = \frac{clock\ frequency}{620 \times 12} \tag{39}$$

Substituting (39) into (38):

$$Max.\ sampling\ error = \left[\,1 - \sin[\,(\,1 - \frac{14880\,F_o}{clock\ frequency}\,)\,\frac{\pi}{2}\,]\,\right] \times 100\% \tag{40}$$

7.7.1 Clock Frequency

If the current frequency is kept constant at 60Hz, the relationship between the maximum sampling error and the clock frequency is shown in Figure 58. The maximum error for the 12MHz clock frequency is 0.68%. Also, the error reduces if a higher frequency clock is used.

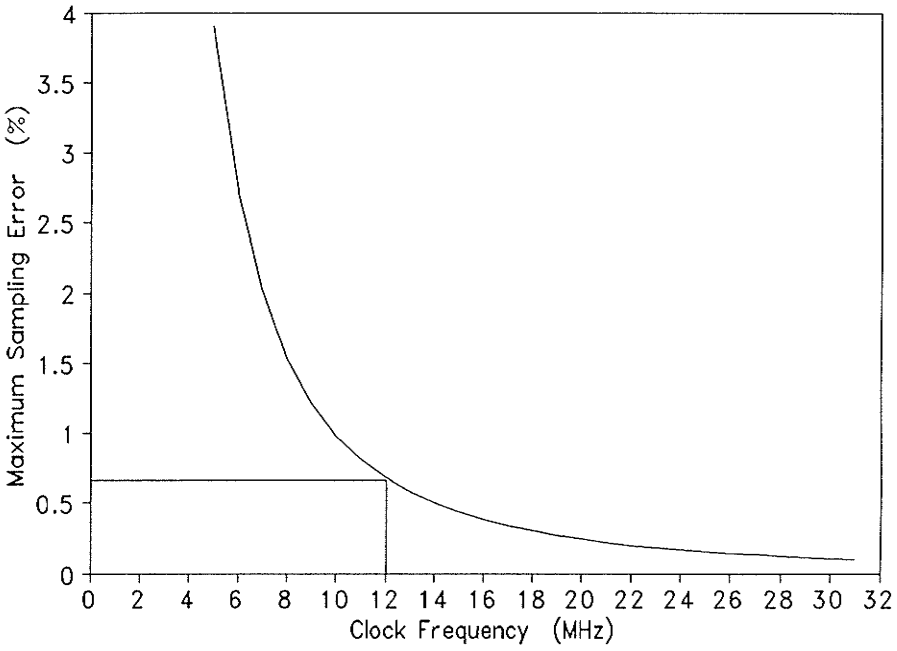


Figure 58 The maximum sampling error versus the clock frequency

7.7.2 Current Frequency

On the other hand, if the clock frequency is fixed at 12MHz and the frequency of the current being measured varies, the sampling error is plotted in Figure 59. The sampling error for a 60Hz current is 0.68%, and the error increases if the current frequency increases. For a 12Mhz clock, the sampling error stays under 1% for a current frequency up to 70Hz.

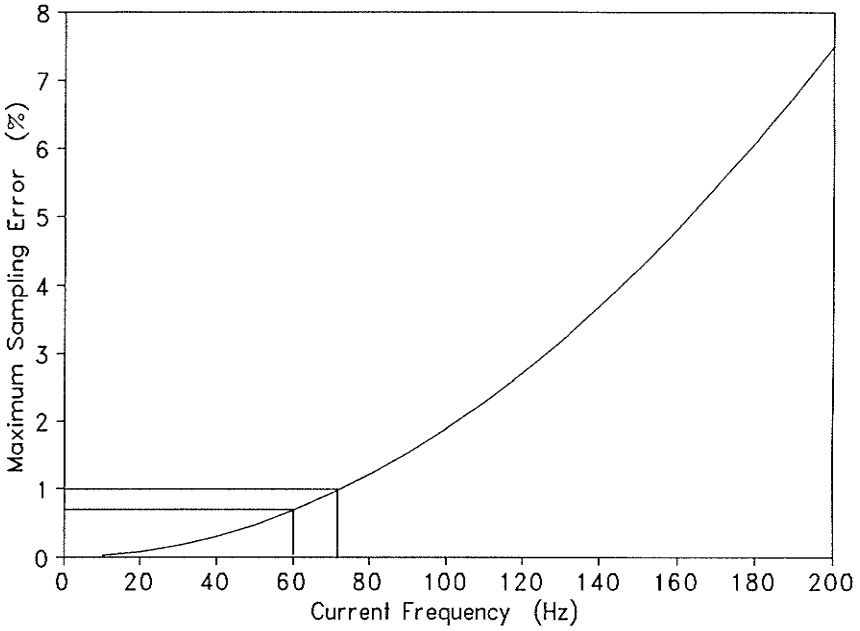


Figure 59 The maximum sampling error versus the current frequency

However, the amplifier-integrator was designed to effectively integrate the 60Hz current signals and higher. Therefore, the frequency range of the current being measured is limited to between 60Hz and 70Hz, so the error is under 1%.

7.8 The demagnetizing effect

When an ac current i is being measured by the linear sensing coil, the coil can be considered as a voltage source v in series with a resistor R_W (the coil resistance). Therefore, a small current i_b flows in the coil. Then, an un-desired backward magnetic flux B_b is induced by the current i_b. The magnetic flux B_b is opposite in direction to the rate of change of the main flux B and affects the output voltages of the coil. This effect is called the *demagnetizing effect*. In order to minimize the effect, a high input impedance was used to reduce the current i_b.

The linear coil resistance R_W is 25Ω and the input resistance R_i is 1.5kΩ. The coil reactance is negligible. The demagnetizing flux B_b can be calculated as follows.

Let $i = I\sin(\omega t) = I\sin(2\pi f t)$, the main flux B and the induced voltage v were derived in Section 5.1.1. From equations (17) and (20),

$$B = \frac{\mu_o i}{2\pi r} = \frac{4\pi\times10^{-7}}{2\pi(0.016m)} I\sin(2\pi f t) = 1.25\times10^{-5} I\sin(2\pi f t)\ T$$

$$v = \frac{1.91\times10^{-5}}{0.016m} I\cos(2\pi f t) = 1.19\times10^{-3} I\cos(2\pi f t)\ V$$

$$i_b = \frac{v}{R_i + R_W} = \frac{1.19\times10^{-3} I\cos(2\pi f t)}{1500\Omega + 25\Omega} = 7.8\times10^{-7} I\cos(2\pi f t)\ A$$

If the linear coil is approximated by a long solenoid, at the center of the coil, the magnetic flux induced by the current i_b is approximately equal to:

$$B_b = \frac{\mu_o N i_b}{l}$$

where N is the number of turns in a coil and l is the length of the coil ($N = 600$ and $l = 7/8" = 0.022m$). Therefore,

$$B_b = \frac{(4\pi\times10^{-7})(600)}{0.022m}\, 7.8\times10^{-7} I\cos(2\pi f t) = 2.67\times10^{-8} I\cos(2\pi f t)\ T$$

The demagnetizing effect error produces a magnitude error and a phase error. Using phasors:

$$\begin{aligned} Magnitude\ Error &= 1 - \left|\frac{\overline{B} + \overline{B_b}}{\overline{B}}\right| \\ &= 1 - \left|\frac{1.25\times10^{-5} I\angle 0^\circ + 2.67\times10^{-8} I\angle 90^\circ}{1.25\times10^{-5} I\angle 0^\circ}\right| \\ &= 1 - |1.00000228\angle 0.12^\circ| \\ &= -0.00000228 = -2.28\times10^{-4}\,\% \end{aligned}$$

In a similar way,

$$Phase\ Error = 0.12^\circ$$

Therefore, the error due to the demagnetizing effect is negligible.

Chapter VIII
Conclusions

The objective of this project was to develop an alternative to the traditional current transformer. Current transformers usually function as the current sensors of most protection or current measurement schemes. Although its performance is well-proven and satisfactory, the current transformer also has a few disadvantages such as iron core saturation problems, bulk, cost and inconvenience for temporary installations.

The characteristics of other possible current sensors: the Linear Output Hall Effect Transducer (LOHET) and the coreless Rogowski coil have been studied. They both provide attractive alternatives to conventional current transformers.

The current sensor considered in this project was modified from the Rogowski coil and is a little out of the ordinary: linear coreless coils were used rather than the toroidal coils. The idea is that the linear coil has two advantages over the toroidal Rogowski coil:

1. It is easier to wind the linear coil so that its cost is lower.
2. The linear coil takes up less space, since it fits flat against the bus bar rather than forming a circle.

Unfortunately, the linear coil is more sensitive to interference from external magnetic fields due to its non-symmetrical shape. The major "external" magnetic fields are those caused by the currents of the other phases. Therefore, a compensator was built to eliminate the phase-to-phase "cross-talk".

The compensator was designed based on an Intel 8751H microprocessor which controls the AD converters to sample the output voltages of the linear coils, implements the derived formulae to compensate the current, and displays the peak value of the compensated current on the 7-segment display. Therefore, the compensator is also considered as a compensated ammeter. The compensated ammeter has the following features:

1. It effectively integrates the differentiated output signals of the linear coils for 60Hz and higher frequency currents.
2. By using a 12Mhz crystal, the compensator samples the signals at every 640μs so that there are 26 samples in each 60Hz cycle.
3. It compensates the current in real time by applying the developed mathematical model.
4. It detects the peak current of each phase in every cycle.
5. It displays the digital readings of each phase current in turn and each reading lasts for 0.5s. Therefore, the reading on the display for one of the phase currents is updated at every 1.5s.

The accuracy of the compensated ammeter was analyzed in relation to such areas as bit error, linearity, effectiveness of compensation, saturation, harmonic content, frequency and the demagnetizing effect. The results are summarized as follows:

1. The full scale bit error is 0.8%.
2. The current readings on the compensated ammeter are linearly proportional to the magnitudes of the currents being measured.
3. The performance of the compensation of the ammeter is proven satisfactory.

4. The saturation limit of the compensator is dependent on the analog input voltage range of the AD converters and the gains of the amplifiers. In this case, the compensated ammeter will saturate at a current of 222 A. If the gain of each amplifier is reduced or a higher input range AD converter is used, the saturation limit will increase.
5. Since the ammeter detects the peak current rather than the rms value of current, it cannot handle a current signal with harmonic content.
6. The accuracy of the ammeter increases if a higher frequency clock is used, since the sampling error reduces while the clock frequency increases.
7. On the other hand, the sampling error increases if a higher frequency current is being measured.
8. At present, a 12MHz clock is being used, the sampling rate is 26 times per cycle and the maximum sampling error is 0.68% while the ammeter is measuring a 60Hz current. Also, the maximum sampling error is kept below 1% for the current frequency up to 70Hz.
9. The compensator effectively integrates signals equal to or higher than 60Hz. Therefore, the frequency range for proper operation of the compensator is between 60 and 70 Hertz.
10. The error caused by the demagnetizing effect is negligible.

In general, the objective was met. The compensated ammeter operates with the rectangular coreless coils (linear coils) providing a good alternative to conventional current transformers for normal metering purposes. Its chief advantages are that it is light and compact, easily attached to or removed from current carrying conductors without breaking the loop, and freedom from iron saturation problems. Also, the compensated ammeter is capable of working with the linear Hall effect transducers by changing the constants K1, K2, K3 and K4.

Recommendations

For the future improvement, I have the following suggestions:

1. A faster microprocessor with a faster clock may be used to enhance the accuracy.
2. The algorithm of obtaining the current magnitude should be changed to calculate the rms value instead of searching the peak value. Thus, the current signals with harmonic content could be handled.
3. Higher resolution AD converters should be used so that the accuracy of the compensator is increased or maintained even if the gain of the amplifiers was reduced to raise the saturation limit.
4. During calibration of the compensated ammeter, standard laboratory ammeters should be used for reference.

References

[1] S. Kearley, "Electronics revives the Rogowski coil c.t.," *Electrical Review*, Vol. 216, No. 4, pp. 22-23, February 1,1985.

[2] K. Sato, O. Miki, M. Iida and H. Nishikawa, "A current sensor using a Hall Generator for a low-voltage circuit breaker," *IEEE Transactions on Power Apparatus and Systems,* Vol. 104, No.7, pp. 1890-96, July 1985.

[3] P. Galluzzi, "Current monitor uses Hall sensor," *EDN*, pp. 205, March 31, 1987.

[4] A. Smith, "Hall-effect current detector," *Electronics and Wireless World,* pp. 62, October 1985.

[5] H. Stratman, "Clamp-on DC Ammeter," *Radio Electronics,* Vol. 55, No. 7, pp. 61-64, July 1984.

[6] A. M. Ferendeci, "Hall probe for measuring high currents in superconducting coils," *Rev. Sci. Instrum.,* Vol. 57, No. 6, pp. 1163-64, June 1986.

[7] W. Rogowski and W. Steinhaus, *Arch. Elektrotech,* Vol. 1, pp. 141, 1912.

[8] W. Stygar and G. Gerdin, "High Frequency Rogowski Coil Response Characteristics," *IEEE Transactions on Plasma Science,* Vol. 10, No. 1, March 1982.

Appendix A

The Source Code of the Program BIOT.PAS

```
PROGRAM BIOT;

Const
    maxsize = 48;
    U = 0.00000125664;
    curr = 1000;

Type
   HallElem = Record
            X : Real;
            Y : Real;
            R : Real;
            B : Real;
            Phi : Real;
            BV : Real;
            BH : Real;
            end;

   HalArray = Array[1..maxsize] of HallElem;

Var
  targetfile : text;
  sourcefile : text;
  Hlist : HalArray;
  Flux, Beta : Real;
  X2, Y2 : Real;
  index : integer;

Procedure Position( Var  Hlist:HalArray );

Var
  Delta, initial : Real;
  index : integer;

Begin
    Delta := 0.003175;
    initial := 0.0015875;
    For index := 1 to 24 Do begin
        Hlist[index].X := initial;
        Hlist[Index+24].X := initial + Delta;
        Hlist[index].Y := index*delta - initial;
        Hlist[index+24].Y := Hlist[index].Y;
    end;
end;
```

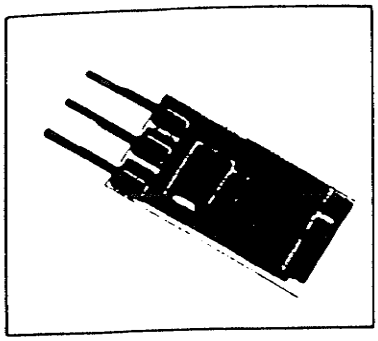
```
Procedure  EleBiot( Var Hlist : HalArray; X2, Y2 : Real) ;

Var
  index : integer;

Begin
    For index := 1 to MAxsize do begin
       with Hlist[ index] do begin
           R := Sqrt( Sqr(X2-X) + Sqr(Y2-Y) );
           If Y2 > Y then
              Phi := ArcTan( (X-X2) / (Y2-Y) ) * 360 / 6.283185
           else
               Phi := ArcTan( (X-X2) / (Y2-Y) ) * 360 / 6.283185 - 180;
           B := (( U * curr ) / ( 301.59 * R ))*10000;     (* 48 * 2 * Pi *)
           BV := B * sin( Phi*6.283185/360 );
           BH := B * Cos( Phi*6.283185/360 );
       end;
    end;
end;


Procedure  Biot( Hlist : HalArray; Var Magn, Angle : Real );

Var
  index : integer;
  Vsum, Hsum : Real;

Begin
    Vsum := 0;
    Hsum := 0;
    For index := 1 to maxsize do begin
       Vsum := Hlist[index].BV + Vsum;
       Hsum := Hlist[index].BH + Hsum;
    end;
    Magn := Sqrt( Sqr(Vsum) + Sqr(Hsum) );
    Angle := ArcTan( Vsum/Hsum ) * 360 / 6.283185;
end;


Begin
    assign( targetfile, 'outda.pas' );
    assign( sourcefile, 'inda.pas' );
    reset( sourcefile );
```

```
    rewrite( targetfile );
 (*    writeln(' input x/y coordinates of test point : '); *)
  while not eof(sourcefile) do begin
    readln(sourcefile, X2, Y2 );
    write( targetfile, 'X=', X2:10:7, '  Y=', Y2:10:7 );
    position( Hlist );
    EleBiot( Hlist, X2, Y2 );
    Biot( Hlist, Flux, Beta );
 (*    writeln( targetfile, '     X=     ', '    Y=     ',
     '    R=     ', '    B=     ', '     Phi=  ','    Bv=    ',
     '    Bh=    ');
    For index := 1 to 48 do begin
       with Hlist[index] do begin
          writeln( targetfile, X:10:7, Y:10:7, R:10:7,
                   B:10:7, Phi:10:5, Bv:11:7,
                   BH:11:7 );
       end;
    end;       *)
    writeln( targetfile, '  Flux= ', Flux:12:8, '  Angle= ', Beta:12:8 );
    writeln( 'flux=', flux:12:8, 'angle=', beta:12:8 );
  end;
    close(targetfile);
    close(sourcefile);
end.
```

Appendix B

The Specifications of the 9SS LOHET

9SS Linear output Hall effect transducers (LOHET™)



Actuation and interface information: see pages 22 thru 24.

Application Notes are available through your local MICRO SWITCH sales office.
Solid State Application Note 1 - Applying Linear Output Hall Effect Transducers (84-05704)
Solid State Application Note 2 - Interfacing Digital Hall Effect Transducers (84-05706)
Solid State Application Note 3 - Using 9SS LOHET™ Specifications (84-05729)
Solid State Application Note 4 - Current Sensing with the 9SS LOHET™ (84-05732)
Solid State Application Note 5 - Position Sensing with the 9SS LOHET™ (84-05736)
Solid State Application Note 6 - Interfacing the 9SS LOHET™ with Comparators and Op Amps (84-05737)

FEATURES

- *Hall effect sensing*
- *Single [redacted] output*
- *3 pin in-line printed circuit board terminals . . . standard .100" mounting centers*
- *Thin ceramic package*
- *4.5 to 8 VDC or 8 to 16 VDC supply voltages*
- *Laser trimmed thick film resistors . . . minimize sensitivity variations, compensate for temperature variations*
- *Tight specifications on output as function of magnetic induction*

9SS series Linear Output Hall Effect Transducers (LOHET) are operated by the magnetic field from a permanent magnet or an electromagnet. The output voltage varies in proportion to the strength of the magnetic field. A Hall effect integrated circuit chip is mounted on a ceramic base. Laser trimmed thick film resistors on the ceramic substrate result in consistent sensitivity from one device to the next, and provide compensation for temperature variations.

Three supply voltage types are available: a 4.5 to 8 VDC listing, and two 8 to 16 VDC listings. The regulated 8 to 16 V listing's output varies from 1.5 to 4.5 volts as magnetic flux density varies. The ratiometric output of the other 8 to 16 V listing varies from 25% to 75% of the supply voltage as magnetic flux varies.

9SS's can be used to provide linear feedback for analog control systems. In DC motors they can sense rotor position. As current sensors, 9SS combined with a coil presents very low impedance to sensed current, and high isolation.

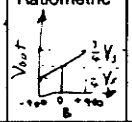
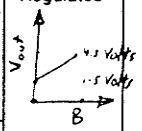
In order to use 9SS LOHET in a linear position sensor, the magnetic field should vary in proportion to position. Contact your nearest MICRO SWITCH sales office for assistance.

9SS ORDER GUIDE

Supply Voltage (VDC)	Supply Current (mA max.)	Catalog Listing	Output Type*
8 to 16	19	[redacted]91SS12-2	Ratiometric
	16	[redacted]92SS12-2	Regulated
4.5 to 8	17.5	[redacted]93SS12-2	Unregulated

*Current sourcing

9SS MAGNETIC CHARACTERISTICS

Supply Voltage = V_s (VDC)	Output Type	Span ① (-400 to +400 gauss)	Null ① (Offset at 0 gauss)	① Sensitivity (mV/gauss)	① Linearity (% span)	Temperature Error: Null Shift (Max. %)	Temperature Error: Typical Sensitivity (%/° C)	Catalog Listing
8 to 16 (Performance at 12 VDC)	Ratiometric 	6.0V ②	6.0 ± 0.6V	7.5 ± 0.2	± 1.5	-40 to 150°C ±5	-40 to 0°C -0.034	91SS12-2 ②
						-25 to 85°C ±3	0 to 150°C -.077	
						0 to 50°C ±2		
8 to 16	Regulated 	3.0V ②	3.0 ± 0.6 V	3.75 ± 0.1	± 5.0	-40 to 150°C ±7	0 to -40°C, -.03	92SS12-2
							25 to 0°C, +.004	
						0 to 150°C ±5	25 to 50°C, -.02	
							50 to 85°C, -.043	
							85 to 120°C, -.057	
							120 to 150°C, -.073	
4.5 to 8 (Performance at 5 VDC)	Unregulated	1.5 V	2.25 ± 0.5 V	1.875 ± 0.1	± 2.5	0 to 85°C ±5	0 to 50°C -0.048	93SS12-2 ③
							50 to 85°C -0.07	

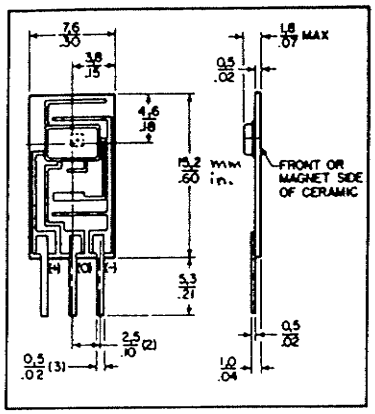
① At 25°C, 2200 Ohm load
② Span, null, sensitivity vary in proportion to supply voltage over range of 12 ± 4VDC.
③ Span, null, sensitivity approximately proportional to supply voltage over range of 4.5 to 8 VDC.

ABSOLUTE MAXIMUM RATINGS*

Parameter	4.5 to 8 VDC Circuit	8 to 16 VDC Circuit
Supply Voltage (Vs)	-0.5 to +10 VDC	-0.5 to +18 VDC
Output Current	10mA	10mA
Temperature		
Storage	-40° C to +150° C (-40° F to +302° F)	
Operating	0° C to +85° C (32° F to +185° F)	-40° C to +150° C (-40° F to +302° F)
Magnetic Flux	No limit. Circuit cannot be damaged by magnetic overdrive.	

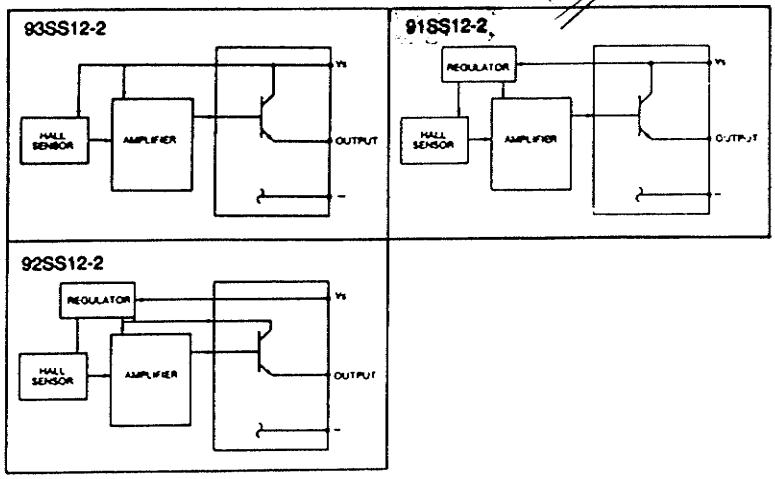
*Hall effect sensors may not work at these ratings, but will not be damaged unless the ratings are exceeded. Preferred operating characteristics are given in the 9SS Magnetic Characteristics chart.

MOUNTING DIMENSIONS



BLOCK DIAGRAMS

Internal



Application Information

CURRENT SINKING AND CURRENT SOURCING

Two types of output circuits are common to solid state circuitry . . . current sinking and current sourcing. The names are derived from the location of the load in the output circuit.

Current sourcing (open emitter): The load is connected between the output and ground. The load is isolated from the supply voltage when the sensor is Off. When the sensor turns On, current flows from the power supply through the sensor, into the load. The sensor supplies a "source" of power to the load. A current sourcing output is Normally Low and goes High when the sensor is On.

Current sinking (open collector): The load is connected between the power supply and the sensor. In this circuit, the load is isolated from ground when the sensor is Off. When the sensor turns On, the circuit is complete and current flows from the supply through the load, through the sensor and then to ground. The sensor switches, or "sinks" the output current to ground. A current sinking output is Normally High and goes Low when the sensor is On.

INTERFACING FOR MICRO SWITCH HALL EFFECT SENSORS

The schematics shown are typical of the outputs with which MICRO SWITCH Hall effect sensors can be interfaced. Values shown are representative only.

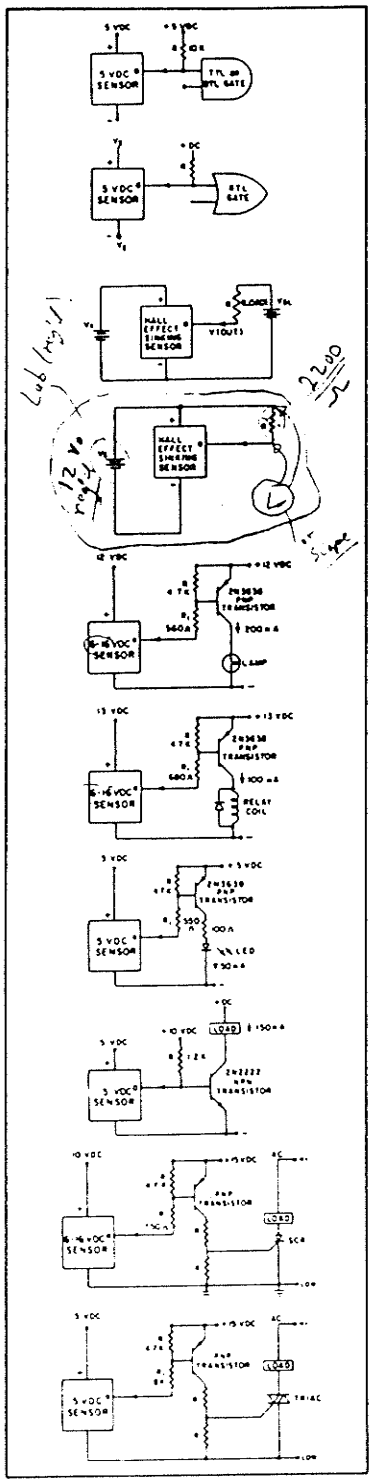
PULL-UP AND PULL-DOWN RESISTORS

It is common practice to use a pull-up resistor for current sinking and a pull-down resistor for current sourcing. These resistors minimize the effect of small leakage currents from the output of the device or from the electronics with which it is interfaced. In addition, they provide better noise immunity and faster rise and fall times.

The current sinking output is open collector and the current sourcing is open emitter. These outputs are floating, so the pull-up or pull-down resistor helps establish a solid quiescent voltage level. When checking sensors, a load is essential. See interface drawings.

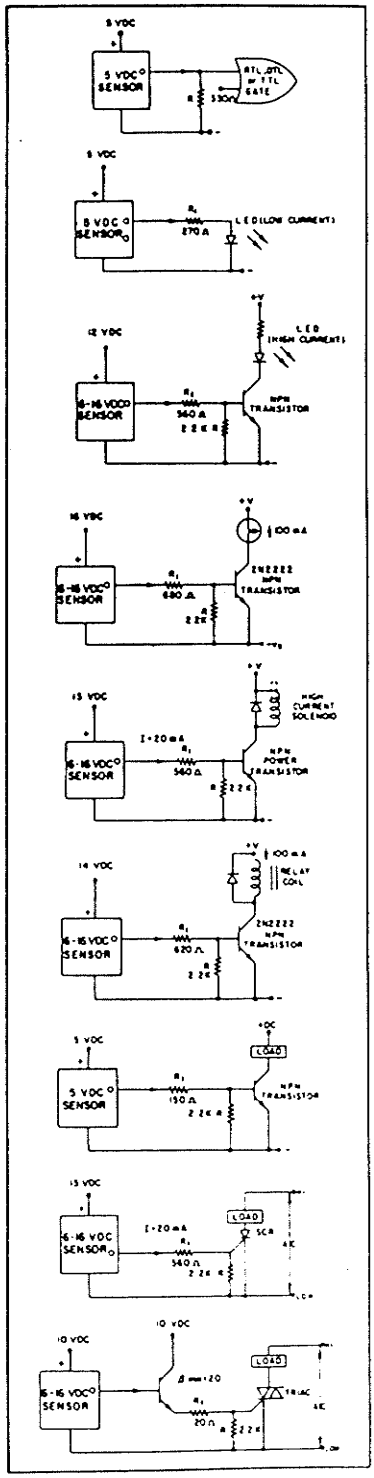
CURRENT SINKING OUTPUTS

(Current flow through load into sensor). Output terminal is open collector. In the unoperated condition (I_L = 0), the output voltage is normally High.



CURRENT SOURCING OUTPUTS

(Current flow from sensor through the load). Output terminal is open emitter. In the unoperated condition (I_L = 0), the output voltage is normally Low.



Appendix C

The Transfer Characteristics of the Rogowski Coil

By Ampere's Law

$$\oint H dL = i$$

where H is the magnetic field intensity and i is the current encircled by the Rogowski coil.

$$2\pi r H = i$$

where r is the radius of the closed path and also is the radius of Rogowski coil.

$$H = \frac{i}{2\pi r}$$

$$B = \mu_o H = \frac{\mu_o i}{2\pi r}$$

By Faraday's Law

$$v = N' \frac{d\Phi}{dt} = N' A \frac{dB}{dt} = \frac{N' A \mu_o}{2\pi r} \frac{di}{dt}$$

where Φ is the magnetic flux, B is the flux density, A is the cross-sectional area of the Rogowski coil, N´ is the number of turns in the Rogowski coil and $\mu_o = 4\pi \times 10^{-7}$.

Let $i = I \sin(\omega t) = I \sin(2\pi f t)$:

$$v = \frac{N' A \mu_o I f}{r} \cos(2\pi f t) = V \cos(2\pi f t)$$

where

$$V = \frac{N' A \mu_o I f}{r}$$

$$\therefore \qquad \frac{V}{I} = \frac{N' A \mu_o f}{r} = 4\pi \times 10^{-7} \frac{N' A f}{r} \ volts/amp$$

Let N is the number of turns per cm = turn density :

$$N = \frac{N'}{2\pi r(100)}$$

and the area A is in cm^2, then:

$$\frac{V}{I} = 4\pi \times 10^{-7} \frac{N 2\pi r(100) A f}{r(10000)}$$

$$= 7.9 \times 10^{-8} NAf \quad V/A$$

$$= 7.9 \times 10^{-5} NAf \quad mV/A$$

Appendix D

Using the Program K1234.PAS and its Source Code

Functions

The program **K1234.PAS** calculates normalized values of *K1, K2, K3* and *K4* by inputting a set of data of *I, Va, Vb* and *Vc*.

Using the Program

An executable version of **K1234.PAS** called **K1234.EXE** is included in the software diskette. To run the program, just simply type *K1234* at the DOS prompt as follow:

A> ***K1234*** (Type K1234 at DOS prompt)

then input the set of data in the format given, the first is current *I*, the second is *Va*, then *Vb* and the last is *Vc*. For example:

Please input the data <I,Va,Vb,Vc>:

20 0.082 0.009 0.003 (Input the first set data, *I* is 20, *Va* is 0.82, *Vb* is 0.009 and *Vc* is 0.003)

40 0.192 0.012 0.005 (Input the second set of data)

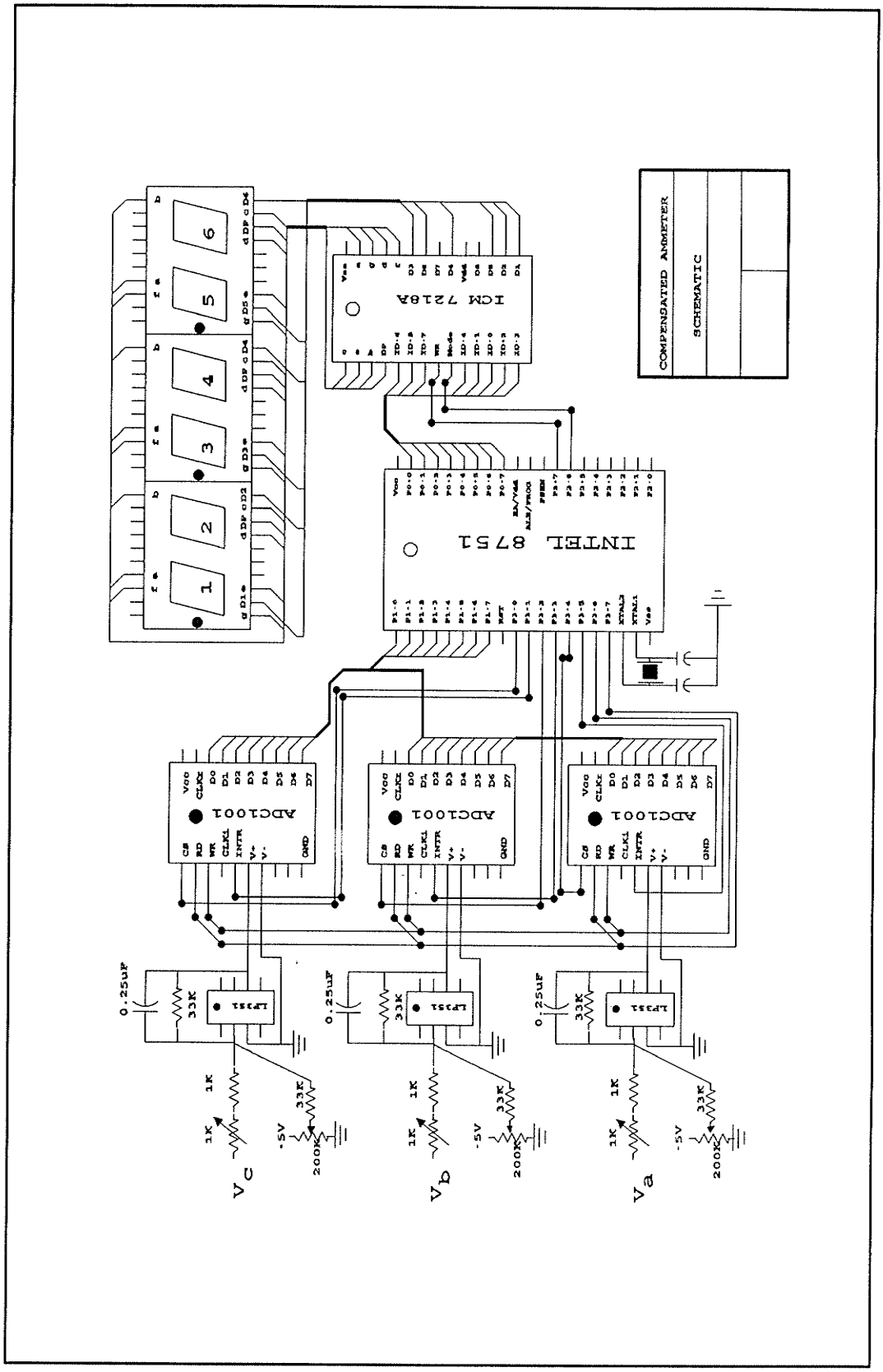
There is no limit of number of data sets. To indicate the end of the input list, just type:

0 0 0 0 (Indicates no more data input)

```
Program  K1234;

VAR
  Ia, Va, Vb, Vc : real;
  count : integer;
  M1, M2, M3, sm1, sm2, sm3 : real;
  K1, K2, K3, K4, delta : real;
  K11, K22, K33, K44 : integer;

BEGIN
    Writeln('Please input < Ia, Va, Vb, Vc >:');
    Readln( Ia, Va, Vb, Vc );
    count := 0;
    M1 := 0;
    M2 := 0;
    M3 := 0;
    While (Ia < > 0) do begin
        count := count + 1;
        sm1 := Va / (377 * Ia);
        M1 := M1 + SM1;
        sm2 := Vb / (377 * Ia);
        M2 := M2 + sm2;
        sm3 := Vc / (377 * Ia);
        M3 := M3 + sm3;
        writeln('M1,M2,M3 are: ',sm1:10:7,sm2:10:7,sm3:10:7);
        Readln( Ia, Va, Vb, Vc );
    end;
    M1 := M1 / count;
    M2 := M2 / count;
    M3 := M3 / count;
    Delta := M1*M1*M1 - 2*M1*M2*M2 + 2*M2*M2*m3 - M1*M3*M3;
    K1 := (M1*M1 - M2*M2) / delta;
    K2 := (M1*M1 - M3*M3) / delta;
    K3 := (M1*M2 - M2*M3) / delta;
    K4 := (M1*M3 - M2*M2) / delta;
    K11 := ROUND( K1/K4 );
    K22 := ROUND( K2/K4 );
    K33 := ROUND( K3/K4 );
    K44 := 1;
    Writeln( 'K1 = ', K1:10:4, '   or   ', K11:6 );
    Writeln( 'K2 = ', K2:10:4, '   or   ', K22:6 );
    Writeln( 'K3 = ', K3:10:4, '   or   ', K33:6 );
    Writeln( 'K4 = ', K4:10:4, '   or   ', K44:6 );
END.
```

Appendix E

The Schematic Diagram of the Compensator


COMPENSATED AMMETER
SCHEMATIC
ICM 7218A
INTEL 8751
ADC1001
LF351
0.25uF
33K
1K
200K
-5V
Vc
Vb
Va

Appendix F

The Source Code of Software for the Compensater

```
                ;
                ;           NEWSOFT2.ASM          March 26, 1990
                ;
                ;           The program for the 8751 based
                ;           compensated-ammeter.
                ;
                ;
0070            K1      EQU     80H             ; K1 = K2 = 128
0003            K3      EQU     03H             ; K3 = 3
0001            K4      EQU     01H             ; K4 = 1
                ;
001F            DTUNE   EQU     1FH             ; Display delay tunning
                ;
00FD            TM0H    EQU    0FDH             ; Set Timer=26CH=620=0.64ms
00D5            TM0L    EQU     94H             ; to sample 26 times/period
0061            STACK   EQU     61H             ; TOP OF STACK = 61H
                ;
00FB            NUM0    EQU    0FBH
008A            NUM1    EQU     8AH
00ED            NUM2    EQU    0EDH
00CF            NUM3    EQU    0CFH
009E            NUM4    EQU     9EH
00D7            NUM5    EQU    0D7H
00F7            NUM6    EQU    0F7H
008B            NUM7    EQU     8BH
00FF            NUM8    EQU    0FFH
009F            NUM9    EQU     9FH
00EF            CHRA    EQU    0EFH
00F6            CHRB    EQU    0F6H
00E4            CHRC    EQU    0E4H
00B0            CHRI    EQU    0B0H
                ;
0040            VA      EQU     40H             ; variables in GETVOLT
0041            VB      EQU     41H
0042            VC      EQU     42H
                ;
0043            ITEMP   EQU     43H             ; variables in CURRENT
0044            IAH     EQU     44H
0045            IAL     EQU     45H
0046            IBH     EQU     46H
0047            IBL     EQU     47H
0048            ICH     EQU     48H
0049            ICL     EQU     49H
                ;
004A            LASTIAH EQU     4AH             ; variables used in IPEAK
004B            LASTIAL EQU     4BH
004C            IAMAXH  EQU     4CH
004D            IAMAXL  EQU     4DH
004E            IAMINH  EQU     4EH
004F            IAMINL  EQU     4FH
0050            LASTIBH EQU     50H
0051            LASTIBL EQU     51H
0052            IBMAXH  EQU     52H
0053            IBMAXL  EQU     53H
0054            IBMINH  EQU     54H
0055            IBMINL  EQU     55H
0056            LASTICH EQU     56H
0057            LASTICL EQU     57H
0058            ICMAXH  EQU     58H
0059            ICMAXL  EQU     59H
005A            ICMINH  EQU     5AH
005B            ICMINL  EQU     5BH
```

```
005C            FLAGABC  EQU     5CH
                ;
005D            IMAXH    EQU     5DH             ; VARIABLES IN ISTBLZR
005E            IMAXL    EQU     5EH
005F            IOFFH    EQU     5FH
0060            IOFFL    EQU     60H
                ;
0000                     ORG     RESET
0000 020100              JMP     MAIN
                ;
0003                     ORG     03H
0003 FD                  DB      0FDH          ; because demaged byte at 0003
                ;
000B                     ORG     TIMER0
000B 02010F              JMP     TIMINT
                ;
0200                     ORG     200H
                ;
                ;-------------------------------------------------------------
                ;   GETVOLT IS A SUBROUTINE TO CONTROL AD CONVERTERS TO MONITOR
                ;   THE INPUT VOLTAGE AND STORE THE VOLTAGE INTO R0 AND R1
                ;
0200 33         CHSIGN:  RLC     A               ; CHECK THE MSB BIT
0201 5007               JNC     POSITIVE        ; IF MSB=0 THEN POSITIVE NUMBER
0203 C3                  CLR     C               ; IF NEGATIVE, DEDUCT #80H
0204 13                  RRC     A
0205 F4                  CPL     A               ; 1'S COMPLEMENT
0206 04                  INC     A               ; Add 1
0207 020210              JMP     ENDSIGN
020A 13         POSITIVE:RRC     A
020B F4                  CPL     A               ; CALCULATE #80H - A
020C 2480                ADD     A, #80H
020E 04                  INC     A
020F C3                  CLR     C
0210 22         ENDSIGN: RET
                ;
0211 75B0FF     GETVOLT: MOV     P3, #0FFH         ; RESET ALL CONTROL BITS
0214 7590FF              MOV     P1, #0FFH
0217 C2B4                CLR     P3.4              ; CS A -TRANS
0219 C2B2                CLR     P3.2              ; CS B -TRANS
021B C2B0                CLR     P3.0              ; CS C -TRANS
021D C2B6                CLR     P3.6              ; WR IMPULSE
021F D2B6                SETB    P3.6
0221 D2B4                SETB    P3.4              ; CS A +TRANS
0223 D2B2                SETB    P3.2              ; CS B +TRANS
0225 D2B0                SETB    P3.0              ; CS C +TRANS
0227 30B502     CHKA:    JNB     P3.5, GETA        ; CHECK CONVERTER A INTR
022A 80FB                JMP     CHKA              ; POLLING
022C C2B4       GETA:    CLR     P3.4              ; CSA -TRANSITION
022E C2B7                CLR     P3.7              ; RD DOWN-TRANS
0230 E590                MOV     A, P1             ; MODIFIED THE SIGN & NUMBER
0232 5100                CALL    CHSIGN
0234 F540                MOV     VA, A             ; STORE MSB INTO VA
0236 D2B7                SETB    P3.7              ; RD UP-TRANS
0238 D2B4                SETB    P3.4              ; CS A UP-TRANS
023A C2B4                CLR     P3.4              ; CS A DOWN-TRANS
023C C2B7                CLR     P3.7              ; RD DOWN-TRANS
023E E590                MOV     A, P1             ; Dummy read the LSB
0240 D2B7                SETB    P3.7              ; RD UP-TRANS
0242 D2B4                SETB    P3.4              ; CS A UP-TRANS
0244 30B302     CHKB:    JNB     P3.3, GETB        ; CHECK CONVERTER B INTR
0247 80FB                JMP     CHKB
```

```
0249 C2B2      GETB:    CLR    P3.2              ; CS B -TRANS
024B C2B7               CLR    P3.7              ; RD -TRANS
024D E590               MOV    A, P1             ; MODIFIED SIGN & NUMBER
024F 5100               CALL   CHSIGN
0251 F541               MOV    VB, A             ; MSB -> VB
0253 D2B7               SETB   P3.7              ; RD +TRANS
0255 D2B2               SETB   P3.2              ; CS B +TRANS
0257 C2B2               CLR    P3.2              ; CS B -TRANS
0259 C2B7               CLR    P3.7              ; RD -TRANS
025B E590               MOV    A, P1             ; dummy read
025D D2B7               SETB   P3.7              ; RD +TRANS
025F D2B2               SETB   P3.2              ; CS B +TRANS
0261 30B102    CHKC:    JNB    P3.1, GETC        ; CHECK CONVERTER C INTR
0264 80FB               JMP    CHKC              ; POLLING
0266 C2B0      GETC:    CLR    P3.0              ; CS C -TRANS
0268 C2B7               CLR    P3.7              ; RD -TRANS
026A E590               MOV    A, P1
026C 5100               CALL   CHSIGN            ; MODIFIED SIGN & NUMBER
026E F542               MOV    VC, A             ; MSB -> VC
0270 D2B7               SETB   P3.7              ; RD +TRANS
0272 D2B0               SETB   P3.0              ; CS C +TRANS
0274 C2B0               CLR    P3.0              ; CS C -TRANS
0276 C2B7               CLR    P3.7              ; RD -TRANS
0278 E590               MOV    A, P1             ; Dummy Read
027A D2B7               SETB   P3.7              ; RD +TRANS
027C D2B0               SETB   P3.0              ; CS C +TRANS
027E 22                 RET
                ;
                ;  ------------------------------------------------------
                ;    CURRENT is a subroutine to calculate the current
                ;    Ia, Ib & Ic and each with double byte precison
                ;
                ;    MULTI is used to perform multiplication of signed value
                ;    in A with unsigned value B, and given a signed result
                ;
                ;
027F 33        MULTI:   RLC    A                 ; CHECK SIGN ON 1ST BIT
0280 4005               JC     NEGMUL            ; IF -VE JMP TO NEGMUL
0282 13                 RRC    A                 ;
0283 A4                 MUL    AB                ; OTHERWISE, SIMPLY A x B
0284 020293             JMP    ENDMUL            ; FINISH
0287 13        NEGMUL:  RRC    A
0288 F4                 CPL    A                 ; IF -VE, 1'S COMPLEMENT
0289 04                 INC    A                 ; Add 1
028A A4                 MUL    AB                ; A x B
028B F4                 CPL    A                 ; 1'S COMPLEMENT A
028C 04                 INC    A                 ; Add 1
028D C5F0               XCH    A, B
028F F4                 CPL    A                 ; 1'S COMPLEMENT B
0290 04                 INC    A                 ; Add 1
0291 C5F0               XCH    A, B
0293 22        ENDMUL:  RET
                ;
                ;    (    Ia = K1*Va - K3*Vb + K4*Vc    )
                ;
0294 E540      CURRENT: MOV    A, VA             ; PUT Va INTO A
0296 75F070             MOV    B, #K1            ; K1 -> B
0299 517F               CALL   MULTI             ; Va x K1
029B 85F044             MOV    IAH, B            ; MSB -> IAH
029E F545               MOV    IAL, A            ; LSB -> IAL
02A0 E541               MOV    A, VB             ; Vb -> A
02A2 75F003             MOV    B, #K3            ; K3 -> B
```

```
02A5 517F               CALL    MULTI           ; Vb x K3
02A7 C545               XCH     A, IAL
02A9 C3                 CLR     C
02AA 9545               SUBB    A, IAL          ; IAL - LSB
02AC F545               MOV     IAL, A
02AE E544               MOV     A, IAH          ;
02B0 95F0               SUBB    A, B            ; IAH - MSB
02B2 F544               MOV     IAH, A
02B4 E542               MOV     A, VC           ; Vc -> A
02B6 75F001             MOV     B, #K4          ; K4 -> B
02B9 517F               CALL    MULTI           ; Vc x K4
02BB C545               XCH     A, IAL
02BD C3                 CLR     C
02BE 9545               SUBB    A, IAL          ; IAL - LSB
02C0 F545               MOV     IAL, A
02C2 E544               MOV     A, IAH
02C4 95F0               SUBB    A, B            ; IAH - MSB
02C6 F544               MOV     IAH, A
                ;
                ;       Ib = K2*Vb - K3*Va - K3*Vc
                ;
02C8 E541               MOV     A, VB           ; Vb -> A
02CA 75F070             MOV     B, #K1          ; K2 = K1 -> B
02CD 517F               CALL    MULTI           ; K2 * Vb
02CF 85F046             MOV     IBH, B          ; MSB -> IBH
02D2 F547               MOV     IBL, A          ; LSB -> IBL
02D4 E540               MOV     A, VA           ; Va -> A
02D6 75F003             MOV     B, #K3          ; K3 -> B
02D9 517F               CALL    MULTI           ; K3 * Va
02DB C547               XCH     A, IBL
02DD C3                 CLR     C
02DE 9547               SUBB    A, IBL          ; IBL - LSB
02E0 F547               MOV     IBL, A
02E2 E546               MOV     A, IBH
02E4 95F0               SUBB    A, B            ; IBH - MSB - BORROW
02E6 F546               MOV     IBH, A
02E8 E542               MOV     A, VC           ; Vc -> A
02EA 75F003             MOV     B, #K3          ; K3 -> B
02ED 517F               CALL    MULTI           ; K3 * Vc
02EF C547               XCH     A, IBL
02F1 C3                 CLR     C
02F2 9547               SUBB    A, IBL          ; IBL - LSB
02F4 F547               MOV     IBL, A
02F6 E546               MOV     A, IBH
02F8 95F0               SUBB    A, B            ; IBH - MSB - BORROW
02FA F546               MOV     IBH, A
                ;
                ;       Ic = K1*Vc - K3*Vb + K4*Va
                ;
02FC E542               MOV     A, VC
02FE 75F070             MOV     B, #K1
0301 517F               CALL    MULTI
0303 85F048             MOV     ICH, B
0306 F549               MOV     ICL, A
0308 E541               MOV     A, VB
030A 75F003             MOV     B, #K3
030D 517F               CALL    MULTI
030F C549               XCH     A, ICL
0311 C3                 CLR     C
0312 9549               SUBB    A, ICL
0314 F549               MOV     ICL, A
0316 E548               MOV     A, ICH
```

```
 0318 95F0              SUBB   A, B
 031A F548              MOV    ICH, A
 031C E540              MOV    A, VA
 031E 75F001            MOV    B, #K4
 0321 517F              CALL   MULTI
 0323 C549              XCH    A, ICL
 0325 C3                CLR    C
 0326 9549              SUBB   A, ICL
 0328 F549              MOV    ICL, A
 032A E548              MOV    A, ICH
 032C 95F0              SUBB   A, B
 032E F548              MOV    ICH, A
 0330 22                RET
                ;
                ;
                                                                        ;
-----------------------------------------------------------------------
                ;     IPEAK   is a routine keep tracking the maximum and the
minimum
                ;    points of the waves of Ia, Ib and Ic.
                ;
                ;
 0331 E55C      IPEAK:   MOV    A, FLAGABC
 0333 5401               ANL    A, #01H            ; CHECK BIT0 IN FLAGABC
 0335 B4012A             CJNE   A, #01H, MINA      ; IF = 0, GOTO MINA
 0338 E545               MOV    A, IAL
 033A C3                 CLR    C
 033B 954B               SUBB   A, LASTIAL         ; IAL - LASTIAL
 033D E544               MOV    A, IAH
 033F 954A               SUBB   A, LASTIAH         ; IAH - LASTIAH - C
 0341 33                 RLC    A                  ; CHECK SIGN OF RESULT
 0342 400C              JC     FINDMAXA           ; IF IA < IALAST, MAX IA FOUND
 0344 85454B             MOV    LASTIAL, IAL       ; OTHERWISE, IALAST = IANEW
 0347 85444A             MOV    LASTIAH, IAH
 034A 435C01            ORL    FLAGABC, #01H      ; NEXT TIME, GOTO MAXA AGAIN
 034D 020389             JMP    IPEAKB
 0350 854B4D    FINDMAXA:MOV    IAMAXL, LASTIAL    ; IF MAX IA FOUND
 0353 854A4C             MOV    IAMAXH, LASTIAH
 0356 85454B             MOV    LASTIAL, IAL
 0359 85444A             MOV    LASTIAH, IAH
 035C 535CFE             ANL    FLAGABC, #0FEH     ; NEXT TIME GOTO MINA
 035F 020389             JMP    IPEAKB
 0362 E545      MINA:    MOV    A, IAL
 0364 C3                 CLR    C
 0365 954B               SUBB   A, LASTIAL         ; IAL - LASTIAL
 0367 E544               MOV    A, IAH
 0369 954A               SUBB   A, LASTIAH         ; IAH - LASTIAH - C
 036B 33                 RLC    A
 036C 500C              JNC    FINDMINA           ; IF IALAST < IA, MIN IA FOUND
 036E 85454B             MOV    LASTIAL, IAL
 0371 85444A             MOV    LASTIAH, IAH
 0374 535CFE              ANL    FLAGABC, #0FEH     ; OTHERWISE, GO BACK MINA
AGAIN
 0377 020389             JMP    IPEAKB
 037A 854B4F    FINDMINA:MOV    IAMINL, LASTIAL
 037D 854A4E             MOV    IAMINH, LASTIAH
 0380 85454B             MOV    LASTIAL, IAL
 0383 85444A             MOV    LASTIAH, IAH
 0386 435C01             ORL    FLAGABC, #01H          ; NEXT TIME GOTO MAXA
                ;
 0389 E55C      IPEAKB:  MOV    A, FLAGABC
 038B 5402               ANL    A, #02H
```

```
038D B4022A             CJNE    A, #02H, MINB
0390 E547               MOV     A, IBL
0392 C3                 CLR     C
0393 9551               SUBB    A, LASTIBL
0395 E546               MOV     A, IBH
0397 9550               SUBB    A, LASTIBH
0399 33                 RLC     A
039A 400C               JC      FINDMAXB
039C 854751             MOV     LASTIBL, IBL
039F 854650             MOV     LASTIBH, IBH
03A2 435C02             ORL     FLAGABC, #02H
03A5 0203E1             JMP     IPEAKC
03A8 855153     FINDMAXB:MOV    IBMAXL, LASTIBL
03AB 855052             MOV     IBMAXH, LASTIBH
03AE 854751             MOV     LASTIBL, IBL
03B1 854650             MOV     LASTIBH, IBH
03B4 535CFD             ANL     FLAGABC, #0FDH
03B7 0203E1             JMP     IPEAKC
03BA E547       MINB:   MOV     A, IBL
03BC C3                 CLR     C
03BD 9551               SUBB    A, LASTIBL
03BF E546               MOV     A, IBH
03C1 9550               SUBB    A, LASTIBH
03C3 33                 RLC     A
03C4 500C               JNC     FINDMINB
03C6 854751             MOV     LASTIBL, IBL
03C9 854650             MOV     LASTIBH, IBH
03CC 535CFD             ANL     FLAGABC, #0FDH
03CF 0203E1             JMP     IPEAKC
03D2 855155     FINDMINB:MOV    IBMINL, LASTIBL
03D5 855054             MOV     IBMINH, LASTIBH
03D8 854751             MOV     LASTIBL, IBL
03DB 854650             MOV     LASTIBH, IBH
03DE 435C02             ORL     FLAGABC, #02H
                ;
03E1 E55C       IPEAKC: MOV     A, FLAGABC
03E3 5404               ANL     A, #04H
03E5 B4042A             CJNE    A, #04H, MINC
03E8 E549               MOV     A, ICL
03EA C3                 CLR     C
03EB 9557               SUBB    A, LASTICL
03ED E548               MOV     A, ICH
03EF 9556               SUBB    A, LASTICH
03F1 33                 RLC     A
03F2 400C               JC      FINDMAXC
03F4 854957             MOV     LASTICL, ICL
03F7 854856             MOV     LASTICH, ICH
03FA 435C04             ORL     FLAGABC, #04H
03FD 020439             JMP     IPEAKEND
0400 855759     FINDMAXC:MOV    ICMAXL, LASTICL
0403 855658             MOV     ICMAXH, LASTICH
0406 854957             MOV     LASTICL, ICL
0409 854856             MOV     LASTICH, ICH
040C 535CFB             ANL     FLAGABC, #0FBH
040F 020439             JMP     IPEAKEND
0412 E549       MINC:   MOV     A, ICL
0414 C3                 CLR     C
0415 9557               SUBB    A, LASTICL
0417 E548               MOV     A, ICH
0419 9556               SUBB    A, LASTICH
041B 33                 RLC     A
041C 500C               JNC     FINDMINC
```

```
041E 854957            MOV    LASTICL, ICL
0421 854856            MOV    LASTICH, ICH
0424 535CFB            ANL    FLAGABC, #0FBH
0427 020439            JMP    IPEAKEND
042A 85575B  FINDMINC:MOV     ICMINL, LASTICL
042D 85565A            MOV    ICMINH, LASTICH
0430 854957            MOV    LASTICL, ICL
0433 854856            MOV    LASTICH, ICH
0436 435C04            ORL    FLAGABC, #04H
0439 22      IPEAKEND:RET
             ;
             ;
             ;  ------------------------------------------------------------
             ;    ISTBLZR is a routine to ensure the DC offset is eliminated
             ;
043A E55E    SUMI:     MOV    A, IMAXL
043C 2560              ADD    A, IOFFL
043E F560              MOV    IOFFL, A
0440 E55D              MOV    A, IMAXH
0442 355F              ADDC   A, IOFFH
0444 F55F              MOV    IOFFH, A
0446 22                RET
             ;
0447 E55F    MIDPT:    MOV    A, IOFFH
0449 13                RRC    A
044A F55F              MOV    IOFFH, A
044C E560              MOV    A, IOFFL
044E 13                RRC    A
044F F560              MOV    IOFFL, A
0451 22                RET
             ;
0452 E560    CPLMT:    MOV    A, IOFFL
0454 F4                CPL    A
0455 2401              ADD    A, #01H
0457 F560              MOV    IOFFL, A
0459 E55F              MOV    A, IOFFH
045B F4                CPL    A
045C 3400              ADDC   A, #00H
045E F55F              MOV    IOFFH, A
0460 22                RET
             ;
0461 E55D    ISTBLZR:  MOV    A, IMAXH
0463 33                RLC    A
0464 4020              JC     NEGCAL
0466 E55F              MOV    A, IOFFH
0468 33                RLC    A
0469 4007              JC     LEVEL1
046B 913A              CALL   SUMI
046D 9147              CALL   MIDPT
046F 02048F            JMP    ISTBEND
0472 913A    LEVEL1:   CALL   SUMI
0474 400A              JC     LEVEL2
0476 9152              CALL   CPLMT
0478 C3                CLR    C
0479 9147              CALL   MIDPT
047B 9152              CALL   CPLMT
047D 02048F            JMP    ISTBEND
0480 C3      LEVEL2:   CLR    C
0481 9147              CALL   MIDPT
0483 02048F            JMP    ISTBEND
0486 913A    NEGCAL:   CALL   SUMI
0488 9152              CALL   CPLMT
```

```
048A C3                 CLR     C
048B 9147               CALL    MIDPT
048D 9152               CALL    CPLMT
048F 22         ISTBEND: RET
                ;
0490 854C5D     ISTBA:  MOV     IMAXH, IAMAXH
0493 854D5E             MOV     IMAXL, IAMAXL
0496 854E5F             MOV     IOFFH, IAMINH
0499 854F60             MOV     IOFFL, IAMINL
049C 9161               CALL    ISTBLZR
049E C3                 CLR     C
049F E545               MOV     A, IAL
04A1 9560               SUBB    A, IOFFL
04A3 F545               MOV     IAL, A
04A5 E544               MOV     A, IAH
04A7 955F               SUBB    A, IOFFH
04A9 F544               MOV     IAH, A
04AB C3                 CLR     C
04AC E54D               MOV     A, IAMAXL
04AE 9560               SUBB    A, IOFFL
04B0 F54D               MOV     IAMAXL, A
04B2 E54C               MOV     A, IAMAXH
04B4 955F               SUBB    A, IOFFH
04B6 F54C               MOV     IAMAXH, A
04B8 C3                 CLR     C
04B9 E54F               MOV     A, IAMINL
04BB 9560               SUBB    A, IOFFL
04BD F54F               MOV     IAMINL, A
04BF E54E               MOV     A, IAMINH
04C1 955F               SUBB    A, IOFFH
04C3 F54E               MOV     IAMINH, A
04C5 22                 RET
                ;
04C6 85525D     ISTBB:  MOV     IMAXH, IBMAXH
04C9 85535E             MOV     IMAXL, IBMAXL
04CC 85545F             MOV     IOFFH, IBMINH
04CF 855560             MOV     IOFFL, IBMINL
04D2 9161               CALL    ISTBLZR
04D4 C3                 CLR     C
04D5 E547               MOV     A, IBL
04D7 9560               SUBB    A, IOFFL
04D9 F547               MOV     IBL, A
04DB E546               MOV     A, IBH
04DD 955F               SUBB    A, IOFFH
04DF F546               MOV     IBH, A
04E1 C3                 CLR     C
04E2 E553               MOV     A, IBMAXL
04E4 9560               SUBB    A, IOFFL
04E6 F553               MOV     IBMAXL, A
04E8 E552               MOV     A, IBMAXH
04EA 955F               SUBB    A, IOFFH
04EC F552               MOV     IBMAXH, A
04EE C3                 CLR     C
04EF E555               MOV     A, IBMINL
04F1 9560               SUBB    A, IOFFL
04F3 F555               MOV     IBMINL, A
04F5 E554               MOV     A, IBMINH
04F7 955F               SUBB    A, IOFFH
04F9 F554               MOV     IBMINH, A
04FB 22                 RET
                ;
04FC 85585D     ISTBC:  MOV     IMAXH, ICMAXH
```

```
04FF 85595E             MOV     IMAXL, ICMAXL
0502 855A5F             MOV     IOFFH, ICMINH
0505 855B60             MOV     IOFFL, ICMINL
0508 9161               CALL    ISTBLZR
050A C3                 CLR     C
050B E549               MOV     A, ICL
050D 9560               SUBB    A, IOFFL
050F F549               MOV     ICL, A
0511 E548               MOV     A, ICH
0513 955F               SUBB    A, IOFFH
0515 F548               MOV     ICH, A
0517 C3                 CLR     C
0518 E559               MOV     A, ICMAXL
051A 9560               SUBB    A, IOFFL
051C F559               MOV     ICMAXL, A
051E E558               MOV     A, ICMAXH
0520 955F               SUBB    A, IOFFH
0522 F558               MOV     ICMAXH, A
0524 C3                 CLR     C
0525 E55B               MOV     A, ICMINL
0527 9560               SUBB    A, IOFFL
0529 F55B               MOV     ICMINL, A
052B E55A               MOV     A, ICMINH
052D 955F               SUBB    A, IOFFH
052F F55A               MOV     ICMINH, A
0531 22                 RET
                ;
                ; ------------------------------------------------------------
                ;   BITODEC IS A SUBROUTINE TO CONVERT THE BINARY BITS IN R0 TO
                ;   BCD DECIMAL NUMBER STORE INTO R2 (MSB) AND R3 (LSB)
                ;
                ;   CARRYIN  is used to increment R2 when carry is set
                ;
0532 CA         CARRYIN: XCH    A,  R2              ; EXCHANGE A WITH R2
0533 2401                ADD    A,  #01H            ; INCREMENT A
0535 D4                  DA     A                   ; WITH DECIMAL ADJUSTED
0536 CA                  XCH    A,  R2              ; EXCHANGE A BACK TO R2
0537 22                  RET
                ;
0538 E8         ABSR:    MOV    A,  R0
0539 33                  RLC    A
053A 500B                JNC    POSI
053C C3                  CLR    C
053D E9                  MOV    A,  R1
053E F4                  CPL    A
053F 2401                ADD    A,  #01H
0541 F9                  MOV    R1, A
0542 E8                  MOV    A,  R0
0543 F4                  CPL    A
0544 3400                ADDC   A,  #00H
0546 F8                  MOV    R0, A
0547 22         POSI:    RET
                ;
0548 B138       BITODEC: CALL   ABSR            ; Take absolute value of R0 R1
054A 7A00                MOV    R2, #00H        ; initialize R2 & R3
054C 7B00                MOV    R3, #00H
054E E8                  MOV    A,  R0               ; Put IaMax(MSB) into A
054F 33                  RLC    A                    ; Bit15 is sign bit, always
0 (abs)
0550 33         BIT14:   RLC    A
0551 5004                JNC    BIT13
0553 7A24                MOV    R2, #24H
```

```
0555 7B10              MOV    R3, #10H
0557 33        BIT13:  RLC    A
0558 500A              JNC    BIT12
055A CB                XCH    A, R3
055B 2412              ADD    A, #12H
055D D4                DA     A
055E CB                XCH    A, R3
055F CA                XCH    A, R2
0560 3405              ADDC   A, #05H
0562 D4                DA     A
0563 CA                XCH    A, R2
0564 33        BIT12:  RLC    A
0565 500A              JNC    BIT11
0567 CB                XCH    A, R3
0568 2456              ADD    A, #56H
056A D4                DA     A
056B CB                XCH    A, R3
056C CA                XCH    A, R2
056D 3402              ADDC   A, #02H
056F D4                DA     A
0570 CA                XCH    A, R2
0571 33        BIT11:  RLC    A                ; check bit13
0572 500A              JNC    BIT10            ; if set
0574 CB                XCH    A, R3            ; R3 + 64
0575 2428              ADD    A, #28H
0577 D4                DA     A                ; Decimal adjust
0578 CB                XCH    A, R3
0579 CA                XCH    A, R2
057A 3401              ADDC   A, #01H
057C D4                DA     A
057D CA                XCH    A, R2
057E 33        BIT10:  RLC    A                ; check bit12
057F 5009              JNC    BIT9
0581 CB                XCH    A, R3
0582 2464              ADD    A, #64H          ; if set add 32
0584 D4                DA     A
0585 CB                XCH    A, R3
0586 5002              JNC    BIT9            ; if carry set
0588 B132              CALL   CARRYIN          ; inc R2
058A 33        BIT9:   RLC    A
058B 5009              JNC    BIT8
058D CB                XCH    A, R3
058E 2432              ADD    A, #32H
0590 D4                DA     A
0591 CB                XCH    A, R3
0592 5002              JNC    BIT8
0594 B132              CALL   CARRYIN
0596 33        BIT8:   RLC    A
0597 5009              JNC    BIT7
0599 CB                XCH    A, R3
059A 2416              ADD    A, #16H
059C D4                DA     A
059D CB                XCH    A, R3
059E 5002              JNC    BIT7
05A0 B132              CALL   CARRYIN
05A2 E9        BIT7:   MOV    A, R1
05A3 33                RLC    A
05A4 5009              JNC    BIT6
05A6 CB                XCH    A, R3
05A7 2408              ADD    A, #08H
05A9 D4                DA     A
05AA CB                XCH    A, R3
```

```
05AB 5002               JNC     BIT6
05AD B132               CALL    CARRYIN
05AF 33         BIT6:   RLC     A
05B0 5009               JNC     BIT5
05B2 CB                 XCH     A, R3
05B3 2404               ADD     A, #04H
05B5 D4                 DA      A
05B6 CB                 XCH     A, R3
05B7 5002               JNC     BIT5
05B9 B132               CALL    CARRYIN
05BB 33         BIT5:   RLC     A
05BC 5009               JNC     BIT4
05BE CB                 XCH     A, R3
05BF 2402               ADD     A, #02H
05C1 D4                 DA      A
05C2 CB                 XCH     A, R3
05C3 5002               JNC     BIT4
05C5 B132               CALL    CARRYIN
05C7 33         BIT4:   RLC     A
05C8 5009               JNC     BITEND
05CA CB                 XCH     A, R3
05CB 2401               ADD     A, #01H
05CD D4                 DA      A
05CE CB                 XCH     A, R3
05CF 5002               JNC     BITEND
05D1 B132               CALL    CARRYIN
05D3 22         BITEND: RET
                ;
                ;                                                       ;
-----------------------------------------------------------------------
                ;    EXTRACT AND PUTDATA ARE SUBROUTINES TO EXTRACT EACH BCD
DECIMAL
                ;    DIGIT FROM R2 AND R3 AND PUT THEM INTO THE DISPLAY MEMORY
LOCATION
                ;
05D4 540F       EXTRACT: ANL    A,  #0FH        ; ZERO THE MSB 4 BITS IN A
05D6 2436                ADD    A,  #36H        ; A+36 IS LOCATION IN MEMORY
05D8 F9                  MOV    R1, A
05D9 E7                  MOV    A,  @R1         ; PUT THE CONTENT(A+40) INTO
A
05DA F6                    MOV    @R0,A             ; PUT THIS CONTENT INTO
DISPLAY MEMORY
05DB 22                  RET
                ;
05DC 7830       PUTDATA: MOV    R0, #30H         ; LOCATION OF FIRST DISLAY
DIGIT
05DE EB                  MOV    A,  R3           ; PUT LSB INTO A
05DF B1D4                CALL   EXTRACT
05E1 08                   INC    R0              ; LOCATION  OF 2ND DISPLAY
DIGIT
05E2 EB                  MOV    A,  R3           ; GET THE LSB
05E3 03                  RR     A                ; ROTATE IT RIGHT 4 TIMES
05E4 03                  RR     A
05E5 03                  RR     A
05E6 03                  RR     A
05E7 B1D4                CALL   EXTRACT
05E9 08                   INC    R0              ; LOCATION OF 3RD DISPLAY
DIGIT
05EA EA                  MOV    A,  R2
05EB B1D4                CALL   EXTRACT
05ED 08                   INC    R0              ; LOCATION OF 4TH DISPLAY
```

```
DIGIT
 05EE EA                MOV     A,  R2
 05EF 03                RR      A
 05F0 03                RR      A
 05F1 03                RR      A
 05F2 03                RR      A
 05F3 B1D4              CALL    EXTRACT
 05F5 7535B0            MOV     35H ,#CHRI          ; 6TH DIGIT IS CHAR I
 05F8 22                RET
                  ;
                  ;
                  ;  ------------------------------------------------------------
                  ;   DISPLAY IS A SUBROUTINE TO RUN THE DISPLAY DRIVER TO DISPLAY
                  ;   THE CONTENTS FROM MEMORY 30 TO 37 INTO THE 7 SEGMENT DISPLAY
                  ;
 05F9 C2A6        DISPLAY: CLR    P2.6            ; SET MOD LOW
 05FB D2A7                 SETB   P2.7            ; SET WR HIGH
 05FD D2A6                 SETB   P2.6            ; SET MOD HIGH
 05FF 7580FF               MOV    P0, #0FFH       ; LOAD CONTROL WORD
 0602 C2A7                 CLR    P2.7            ; WR IMPULSE
 0604 D2A7                 SETB   P2.7
 0606 C2A6                 CLR    P2.6            ; SET MOD LOW
 0608 853080               MOV    P0, 30H         ; DIGIT 1
 060B C2A7                 CLR    P2.7            ; WR IMPULSE
 060D D2A7                 SETB   P2.7
 060F 853180               MOV    P0, 31H         ; DIGIT 2
 0612 C2A7                 CLR    P2.7            ; WR IMPULSE
 0614 D2A7                 SETB   P2.7
 0616 853280               MOV    P0, 32H         ; DIGIT 3
 0619 C2A7                 CLR    P2.7            ; WR IMPULSE
 061B D2A7                 SETB   P2.7
 061D 853380               MOV    P0, 33H         ; DIGIT 4
 0620 C2A7                 CLR    P2.7            ; WR IMPULSE
 0622 D2A7                 SETB   P2.7
 0624 853480               MOV    P0, 34H         ; DIGIT 5
 0627 C2A7                 CLR    P2.7            ; WR IMPULSE
 0629 D2A7                 SETB   P2.7
 062B 853580               MOV    P0, 35H         ; DIGIT 6
 062E C2A7                 CLR    P2.7            ; WR IMPULSE
 0630 D2A7                 SETB   P2.7
 0632 7580FB               MOV    P0, #NUM0       ; DIGIT 7
 0635 C2A7                 CLR    P2.7            ; WR IMPULSE
 0637 D2A7                 SETB   P2.7
 0639 7580FB               MOV    P0, #NUM0       ; DIGIT 8
 063C C2A7                 CLR    P2.7            ; WR IMPULSE
 063E D2A7                 SETB   P2.7
 0640 22                   RET
                  ;
                  ;
                  ;  ------------------------------------------------------------
                  ;   DELAY is a routine to delay an interval of time to
                  ;   allow the display has enough time to display
                  ;
 0641 7D1F        DELAY:   MOV    R5, #DTUNE          ;
 0643 1D          LP1:     DEC    R5                  ; < Loop 1 Begin
 0644 7EFF                 MOV    R6, #0FFH
 0646 1E          LP2:     DEC    R6                  ; < Loop 2 Begin
 0647 BE00FC               CJNE   R6, #00H, LP2       ;           Loop 2 End >
 064A BD00F6               CJNE   R5, #00H, LP1       ;           Loop 1 End >
 064D 22                   RET
                  ;
 064E 9190        MONITOR: CALL   ISTBA
```

```
0650 A84C               MOV     R0, IAMAXH          ; Get IaMax(MSB)
0652 A94D               MOV     R1, IAMAXL          ; Get IaMax(LSB)
0654 B148               CALL    BITODEC             ; convert Ia into decimal
0656 B1DC               CALL    PUTDATA             ; Put Dec value into display
0658 7534EF             MOV     34H, #CHRA
065B B1F9               CALL    DISPLAY             ; display Ia
065D D141               CALL    DELAY
                ;
065F 91C6               CALL    ISTBB
0661 A852               MOV     R0, IBMAXH          ; Get IbMax(MSB)
0663 A953               MOV     R1, IBMAXL          ; IbMax(LSB)
0665 B148               CALL    BITODEC
0667 B1DC               CALL    PUTDATA
0669 7534F6             MOV     34H, #CHRB
066C B1F9               CALL    DISPLAY             ; display Ib
066E D141               CALL    DELAY
                ;
0670 91FC               CALL    ISTBC
0672 A858               MOV     R0, ICMAXH          ; Get IcMax(MSB)
0674 A959               MOV     R1, ICMAXL          ; IcMax(LSB)
0676 B148               CALL    BITODEC
0678 B1DC               CALL    PUTDATA
067A 7534E4             MOV     34H, #CHRC
067D B1F9               CALL    DISPLAY             ; display Ic
067F D141               CALL    DELAY
0681 22                 RET
                ;
                ;
                ;  ------------------------------------------------------
                ;   MAKE A TABLE WHICH STORES NUM0 TO NUM9 IN MEMORY
                ;   FROM 36H TO 3FH
                ;
0682 7536FB     MKTABLE: MOV    36H, #NUM0          ; put data of '0' to (36H)
0685 75378A              MOV    37H, #NUM1          ; put data of '1' to (37H)
0688 7538ED              MOV    38H, #NUM2          ; put data of '2' to (38H)
068B 7539CF              MOV    39H, #NUM3          ; put data of '3' to (39H)
068E 753A9E              MOV    3AH, #NUM4          ; put data of '4' to (3AH)
0691 753BD7              MOV    3BH, #NUM5          ; put data of '5' to (3BH)
0694 753CF7              MOV    3CH, #NUM6          ; put data of '6' to (3CH)
0697 753D8B              MOV    3DH, #NUM7          ; put data of '7' to (3DH)
069A 753EFF              MOV    3EH, #NUM8          ; put data of '8' to (3EH)
069D 753F9F              MOV    3FH, #NUM9          ; put data of '9' to (3FH)
06A0 22                  RET
                ;
                ;   ------------------------------------------------
                ;         DPTEST is used for testing the displaying
                ;         components
                ;
06A1 7530FB     DPTEST:  MOV    30H, #0FBH     ; character D
06A4 7531BD              MOV    31H, #0BDH     ; character P
06A7 75328B              MOV    32H, #8BH      ; character T
06AA 7533F5              MOV    33H, #0F5H     ; character E
06AD 7534D7              MOV    34H, #0D7H     ; character S
06B0 75358B              MOV    35H, #8BH      ; character T
06B3 B1F9                CALL   DISPLAY
06B5 D141                CALL   DELAY
06B7 22                  RET
                ;
                ;
06B8 754500     INITIAL: MOV    IAL,  #00H       ; Ia(MSB) = 0
06BB 754400              MOV    IAH,  #00H       ; Ia(LSB) = 0
06BE 754700              MOV    IBL,  #00H       ; Ib(MSB) = 0
```

```
06C1 754600              MOV    IBH,  #00H      ; Ib(LSB) = 0
06C4 754900              MOV    ICL,  #00H      ; Ic(MSB) = 0
06C7 754800              MOV    ICH,  #00H      ; Ic(LSB) = 0
06CA 754D00              MOV    IAMAXL,  #00H
06CD 754C00              MOV    IAMAXH,  #00H
06D0 754F00              MOV    IAMINL,  #00H
06D3 754E00              MOV    IAMINH,  #00H
06D6 754B00              MOV    LASTIAL, #00H
06D9 754A00              MOV    LASTIAH, #00H
06DC 755300              MOV    IBMAXL,  #00H
06DF 755200              MOV    IBMAXH,  #00H
06E2 755500              MOV    IBMINL,  #00H
06E5 755400              MOV    IBMINH,  #00H
06E8 755100              MOV    LASTIBL, #00H
06EB 755000              MOV    LASTIBH, #00H
06EE 755900              MOV    ICMAXL,  #00H
06F1 755800              MOV    ICMAXH,  #00H
06F4 755B00              MOV    ICMINL,  #00H
06F7 755A00              MOV    ICMINH,  #00H
06FA 755700              MOV    LASTICL, #00H
06FD 755600              MOV    LASTICH, #00H
0700 755CFF              MOV    FLAGABC, #0FFH
0703 22                  RET
                  ;
0704 758901      TMSETUP: MOV    TMOD, #01H      ; Timer0 Mode 1 operation
0707 75B802              MOV    IP,   #02H      ; setup interrupt priorty
070A 75A882              MOV    IE,   #82H      ; Enable timer0 interrupt
070D 758CFD              MOV    TH0,  #TM0H     ; load the timer number
0710 758AD5              MOV    TL0,  #TM0L
0713 758810              MOV    TCON, #10H      ; START TIMER0
0716 22                  RET
                  ;
0100                     ORG    100H
0100 758161      MAIN:   MOV    SP, #STACK
0103 D1A1                CALL   DPTEST       ; DISPLAY TESTING
0105 D182                CALL   MKTABLE      ; MAKE the display data table
0107 D1B8                CALL   INITIAL      ; Initialize all variables
0109 F104                CALL   TMSETUP      ; Setup the timer0 interrupt
010B D14E        HERE:   CALL   MONITOR
010D 80FC                JMP    HERE
                  ;
                  ;
010F 758CFD      TIMINT: MOV    TH0, #TM0H   ; reload the timer count
0112 758AD5              MOV    TL0, #TM0L
0115 C0D0                PUSH   PSW          ; Push the status registers
0117 C0E0                PUSH   ACC          ; Push Accumulator A
0119 C0F0                PUSH   B            ; Push Accumulator B
011B D2D3                SETB   RS0          ; change register bank from 0 to 1
011D 5111                CALL   GETVOLT      ; sampling the input voltage
011F 5194                CALL   CURRENT      ; calculate the compensated current
0121 7131                CALL   IPEAK        ; search for peak current
0123 C2D3                CLR    RS0          ; change reg. bank back to 1
0125 D0F0                POP    B            ; Pop Accumulator B
0127 D0E0                POP    ACC          ; Pop Accumulator A
0129 D0D0                POP    PSW          ; Pop the status registers
012B 32                  RETI
0000                     END
ABSR     0538          DISPLAY  05F9          IBMINH   0054          LASTIAH
004A          NUM8     00FF
ACC      00E0          DPTEST   06A1          IBMINL   0055          LASTIAL
004B          NUM9     009F
B        00F0          DTUNE    001F          ICH      0048          LASTIBH
```

0050 P0 0080
BIT10 057E ENDMUL 0293 ICL 0049 LASTIBL
0051 P1 0090
BIT11 0571 ENDSIGN 0210 ICMAXH 0058 LASTICH
0056 P2 00A0
BIT12 0564 EXTRACT 05D4 ICMAXL 0059 LASTICL
0057 P3 00B0
BIT13 0557 FINDMAXA 0350 ICMINH 005A LEVEL1
0472 POSI 0547
BIT14 0550 FINDMAXB 03A8 ICMINL 005B LEVEL2
0480 POSITIVE 020A
BIT4 05C7 FINDMAXC 0400 IE 00A8 LP1
0643 PSW 00D0
BIT5 05BB FINDMINA 037A IMAXH 005D LP2
0646 PUTDATA 05DC
BIT6 05AF FINDMINB 03D2 IMAXL 005E MAIN
0100 RESET 0000
BIT7 05A2 FINDMINC 042A INITIAL 06B8 MIDPT
0447 RS0 00D3
BIT8 0596 FLAGABC 005C IOFFH 005F MINA
0362 SP 0081
BIT9 058A GETA 022C IOFFL 0060 MINB
03BA STACK 0061
BITEND 05D3 GETB 0249 IP 00B8 MINC
0412 SUMI 043A
BITODEC 0548 GETC 0266 IPEAK 0331 MKTABLE
0682 TCON 0088
CARRYIN 0532 GETVOLT 0211 IPEAKB 0389 MONITOR
064E TH0 008C
CHKA 0227 HERE 010B IPEAKC 03E1 MULTI
027F TIMER0 000B
CHKB 0244 IAH 0044 IPEAKEND 0439 NEGCAL
0486 TIMINT 010F
CHKC 0261 IAL 0045 ISTBA 0490 NEGMUL
0287 TL0 008A
CHRA 00EF IAMAXH 004C ISTBB 04C6 NUM0
00FB TM0H 00FD
CHRB 00F6 IAMAXL 004D ISTBC 04FC NUM1
008A TM0L 00D5
CHRC 00E4 IAMINH 004E ISTBEND 048F NUM2
00ED TMOD 0089
CHRI 00B0 IAMINL 004F ISTBLZR 0461 NUM3
00CF TMSETUP 0704
CHSIGN 0200 IBH 0046 ITEMP 0043 NUM4
009E VA 0040
CPLMT 0452 IBL 0047 K1 0070 NUM5
00D7 VB 0041
CURRENT 0294 IBMAXH 0052 K3 0003 NUM6
00F7 VC 0042
DELAY 0641 IBMAXL 0053 K4 0001 NUM7
008B