MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0066097 9

ORNL-6305

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Eddy-Current Inspection of Ferromagnetic Tubing Using Pulsed Magnetic Saturation

C. V. Dodd
W. E. Deeds

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

[redacted]

OK per DR H 6/22/05

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
the Department of Energy
Technical Information Center
P.O. Box 62, Oak Ridge, Tennessee 37830

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL-6305
Distribution
Categories UC-79Th,
-Tk, -Tr

Metals and Ceramics Division

EDDY-CURRENT INSPECTION OF FERROMAGNETIC TUBING USING PULSED MAGNETIC SATURATION

C. V. Dodd and W. E. Deeds

Date Published – July 1986

[redacted]

Prepared for
Office of Technology Support Programs

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0066097 9

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

CONTENTS

ABSTRACT 1
INTRODUCTION 1
COMPUTER ANALYSIS OF PULSED EDDY CURRENTS 3
MAGNETIC SATURATION CALCULATIONS 6
DATA ACQUISITION SYSTEM 9
DATA PROCESSING SYSTEM 10
ACTUAL OPTIMIZED DESIGN 11
PERFORMANCE OF THE SYSTEM 13
FIELD DEMONSTRATION 19
CONCLUSIONS AND RECOMMENDATIONS 21
ACKNOWLEDGMENTS 21
REFERENCES 21
Appendix 1. PROGRAM PULSE 23
Appendix 2. PROGRAM MAGSCN 39
Appendix 3. PROGRAM PULRDG 41
Appendix 4. PROGRAM PULFIT 53
Appendix 5. PROGRAMS GRAPUL AND GRAPUF 69
Appendix 6. PROGRAM GPUL 79
Appendix 7. CIRCUIT DIAGRAMS 95

EDDY-CURRENT INSPECTION OF FERROMAGNETIC TUBING USING PULSED MAGNETIC SATURATION*

C. V. Dodd and W. E. Deeds

ABSTRACT

A pulsed eddy-current system has been designed and developed for nondestructive evaluation of 2.25Cr-1Mo steam generator tubing from the bore side. Since the tubing is ferromagnetic, a large current pulse is sent through a driver coil to produce magnetic saturation all the way through the tube wall. A pickup coil produces an output pulse that is dependent upon the tube properties as well as the driving pulse. The output pulse heights at selected times are used as data that are computer-correlated with calibration data taken from machined standards. Performance data, circuit diagrams, and computer programs are given for the system, which has been demonstrated to detect small flaws located near the outside of a thick ferromagnetic tube.

INTRODUCTION

Both ultrasonics and eddy currents are candidate methods for inspecting liquid metal reactor (LMR) steam generators. An ultrasonic probe and inspection system was successfully demonstrated for the Clinch River Breeder Reactor (CRBR)[1] and the Westinghouse-Nuclear Components Division (W-NCD) double-wall steam generator.[2] However, the inspection rate with ultrasonics is relatively slow and, in the the case of the W-NCD double-wall design, is ineffective in detecting defects in the outer tube behind the grooves used for leak detection. On the other hand, inspection with eddy currents is fast and not affected by the grooves. However, low-alloy steels, such as 2.25Cr-1Mo, are hard to inspect with eddy currents because they are ferromagnetic. Eddy-current penetration is inversely proportional to the square root of magnetic permeability, which can be quite high. Furthermore, the high permeability causes magnetic fields to be conducted around flaws, making them even harder to detect.

*Research sponsored by the Office of Technology Support Programs, U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

A method often used to circumvent these difficulties is to magnetically saturate the ferromagnetic material, so that it will effectively act like air relative to further changes in the magnetic field. Magnetic saturation can be used with samples small enough to be placed within a magnet powerful enough to produce a field of about 1.5 T (15,000 gauss). Unfortunately, steam generator tubing must be inspected from the bore side, and, in the case of the W-NCD double-wall steam generator, the inner diameter of the tubing is small and the wall thickness great, so that a saturating coil inside the tube will saturate the core of the coil long before the wall of the tube is saturated. This means that the coil virtually has an air core, and the reluctance of the magnetic circuit is so high that the coil must carry very large currents (and get very hot) to saturate the tube all the way through the wall. Early attempts[3,4] to saturate the 2.25Cr-1Mo tubing from the bore side were unsuccessful because too much heat was generated in the coil. It became obvious that only a very large current in a bore-side coil could saturate the tubing, and the current could be applied for only a very short time if the coil was to survive. Therefore, it was evident that only a powerful current pulse on a low-duty cycle could possibly achieve the desired results. We also decided to use the current pulse, with its many frequency components, for interrogating as well as magnetically saturating the material. Thus, the pulse response could provide the additional information required to distinguish the various tubing properties of interest from extraneous variables, instead of requiring the usual multiple-fixed-frequency system to provide the data. This report describes the development of a system that accomplished these objectives.

Previous reports[5,6] have shown that, when several properties are changing, enough information must be collected to permit unambiguous identification of specific variables. This has usually been accomplished by using multiple fixed frequencies and measuring, for example, the magnitudes and phases of the output voltages at the various frequencies. However, the additional information can also be obtained by measuring the pulse height at selected times, or the times at which selected pulse heights are reached, or the magnitudes and phases of various Fourier

components of the signal pulse. Whatever types of information have been collected can then be processed in the same way as described in the multiple-property reports cited above.

In the past, pulsed systems have generally been simpler but less accurate than multifrequency systems. The advent of improved electronic modules has now made possible pulsed systems that have equal accuracy while retaining their simplicity. Thus, having to use a pulsed system no longer means a penalty in accuracy.

Figure 1 shows a cross-sectional view of a type of ferromagnetic tubing that must be inspected, and Table 1 lists the required and recommended levels of inspection for various types of flaws.

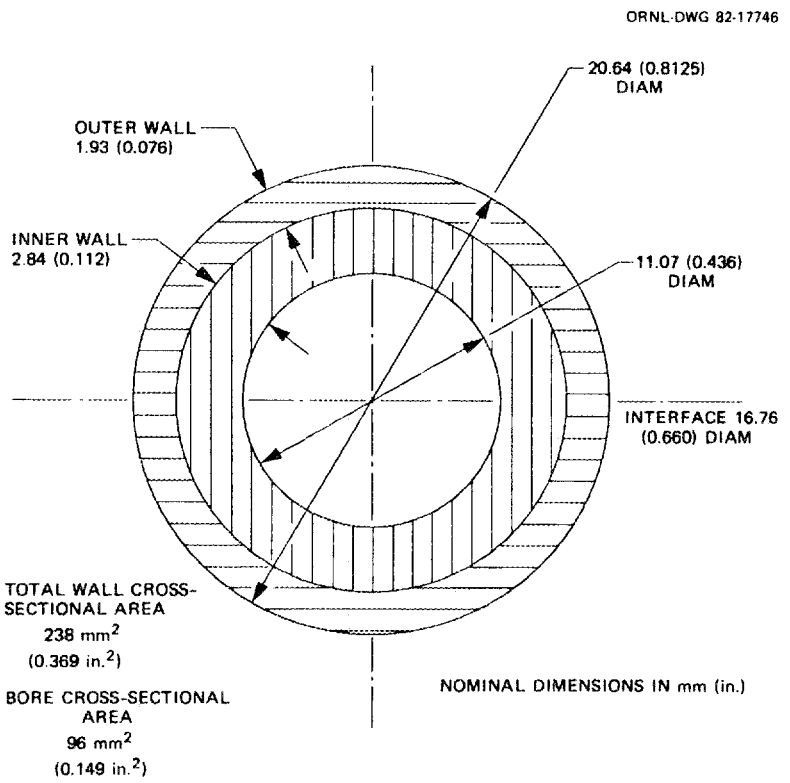


Fig. 1. Cross section of the nominal Westinghouse Intermediate "A" double-wall tube.

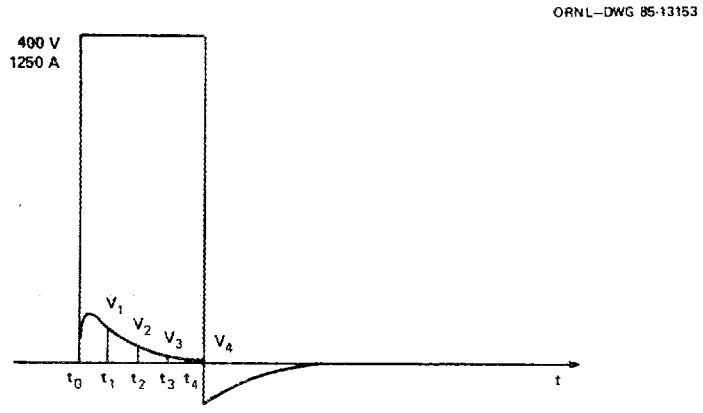
COMPUTER ANALYSIS OF PULSED EDDY CURRENTS

A piecewise continuous pulse can be considered as the sum of various sinusoidal (single-frequency) components. A δ-function pulse (infinitely

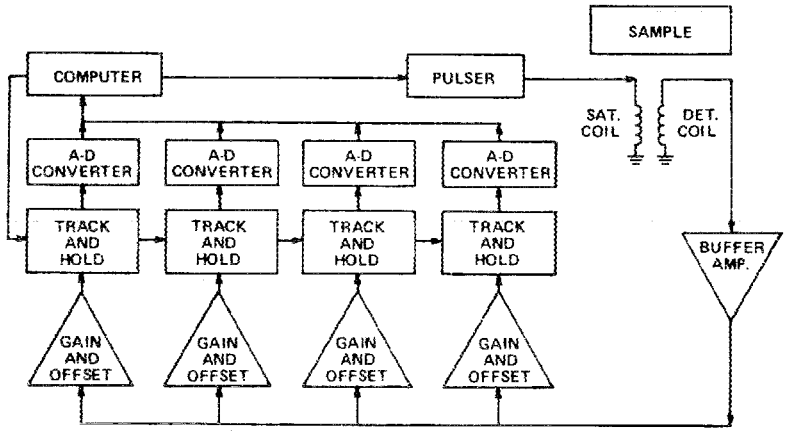
Table 1. Summary of required and recommended levels of sensitivity to artificial defects in double-wall tubing in Westinghouse Intermediate "A" prototype steam generator

Defect type	Depth, mm (in.)	
	Required	Recommended
Defects on tube outside diameter		
Circumferential band, 6 mm long (0.25 in.)	1.2 (0.047)	0.48 (0.019)
Axial or circumferential notch, 25 mm long by 1.6 mm wide (1 by 1/16 in.)	1.2 (0.047)	0.48 (0.019)
Hole, 1.6 mm in diameter (1/16 in.)	1.9 (0.076)	1.9 (0.076)
Defects on tube inside diameter		
Circumferential band, 6 mm long (0.25 in.)	1.2 (0.047)	1.2 (0.047)
Axial or circumferential notch, 25 mm long by 1.6 mm wide (1 by 1/16 in.)	1.2 (0.047)	1.2 (0.047)
Hole, 1.6 mm in diameter (1.16 in.)	2.8 (0.112)	2.8 (0.112)

high and narrow) would contain all frequencies with equal amplitude, but such a pulse is unobtainable and also undesirable in practice. We want to use the pulse to magnetically saturate the tubing, as well as to induce eddy currents in it and the pickup coil. Since a finite time is required for the magnetic field to penetrate and saturate the tubing, as well as for reading the response, a "square-wave" pulse is actually more useful than a δ-function one. A square-wave pulse is diagrammed in Figs. 2(*a*) and 3(*a*), which show a voltage of 400 V suddenly applied to the driver (saturating) coil at $t = 0$ and subsequently turned off suddenly. If the time between pulses is very long compared with the duration of the pulse, the frequency components will be mainly determined by the rise time (and fall time) of the pulse, as well as its total duration. The steeper the rise times, the greater are the high-frequency components. Figures 2(*a*) and 3(*a*) also show the voltages induced in a pickup coil near the driver coil. The pickup voltage will rise in such a way as to try to oppose the

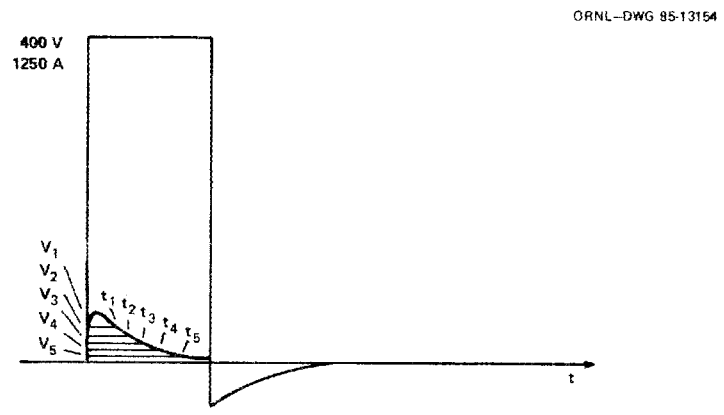


(a) **PULSE TO SATURATING COIL AND DETECTOR RESPONSE**

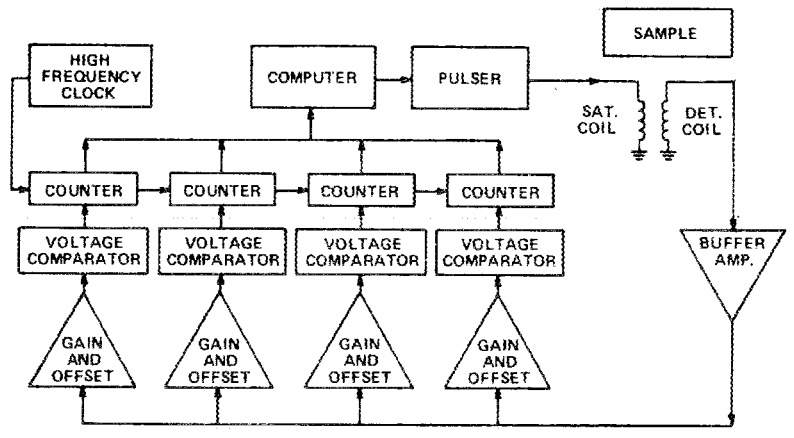


(b) **A PULSE AMPLITUDE INSTRUMENT**

Fig. 2. Pulse height at various times.



(a) **PULSE TO SATURATING COIL AND DETECTOR RESPONSE**



(b) **A PULSE TIME INTERVAL INSTRUMENT**

Fig. 3. Time-to-pulse height system.

changing flux produced by the driver current and the induced currents in the tube. Then when the driver voltage drops back to zero, there will again be a voltage induced in the pickup, trying to oppose that change. Figure 2(*b*) shows a block diagram of an instrument to measure the pickup voltages at certain times, t_1 to t_4. Figure 3(*b*) shows a similar diagram for an instrument to measure the times at which the pickup voltage reaches certain specified values.

To calculate the defect sensitivity of a pulsed coil system, a program for a reflection coil system near a multilayer conductor was modified to calculate the pulse response to a defect in a single-layer conductor. The pulse was a square wave synthesized from 80 frequencies at integral multiples of 100 Hz. The program, called PULSE, is listed in Appendix 1. The magnitude and phase of the pickup voltage at various lift-offs are printed for a metal sample with and without a defect. The metal in the example was assumed to be 5 mm thick (0.2 in.) and to have the resistivity of 2.25Cr-1Mo steel. The metal was assumed to have a relative permeability of 1, meaning that the steel was assumed to be saturated. The flaw was assumed to be located 0.127 mm (0.005 in.) in from the far side of the metal, and it produced small but definite changes in both the magnitude and the phase of the pickup signal. On the basis of this and other calculations, it seemed clear that defects could be detected with a pulsed driver system if the tube could be magnetically saturated. In earlier experiments[7] we obtained the values of relative permeability for 2.25Cr-1Mo steel at various levels of dc magnetic field, as shown in Table 2. These data show that the steel can be saturated with a magnetic field somewhat stronger than 1 T.

MAGNETIC SATURATION CALCULATIONS

About 2 T (20,000 gauss) is needed to saturate ordinary iron or steel, and Fig. 4 shows that the saturation magnetization for 2.25Cr-1Mo tubing is about 1.5 T. When that field is applied, the effective permeability for small changes in the field becomes essentially the same as that for air. Since we must perform the inspection from the bore side of the tube, a magnetizing coil must also be there. We have developed a

Table 2. Relative permeability for 2.25Cr-1Mo steel at various levels of dc magnetic field

dc Bias field		Relative permeability (μ/μ_0)
T	kilogauss	
0.73	7.3	51.1
0.91	9.1	19.6
1.00	10.0	13.1
1.05	10.5	0.9

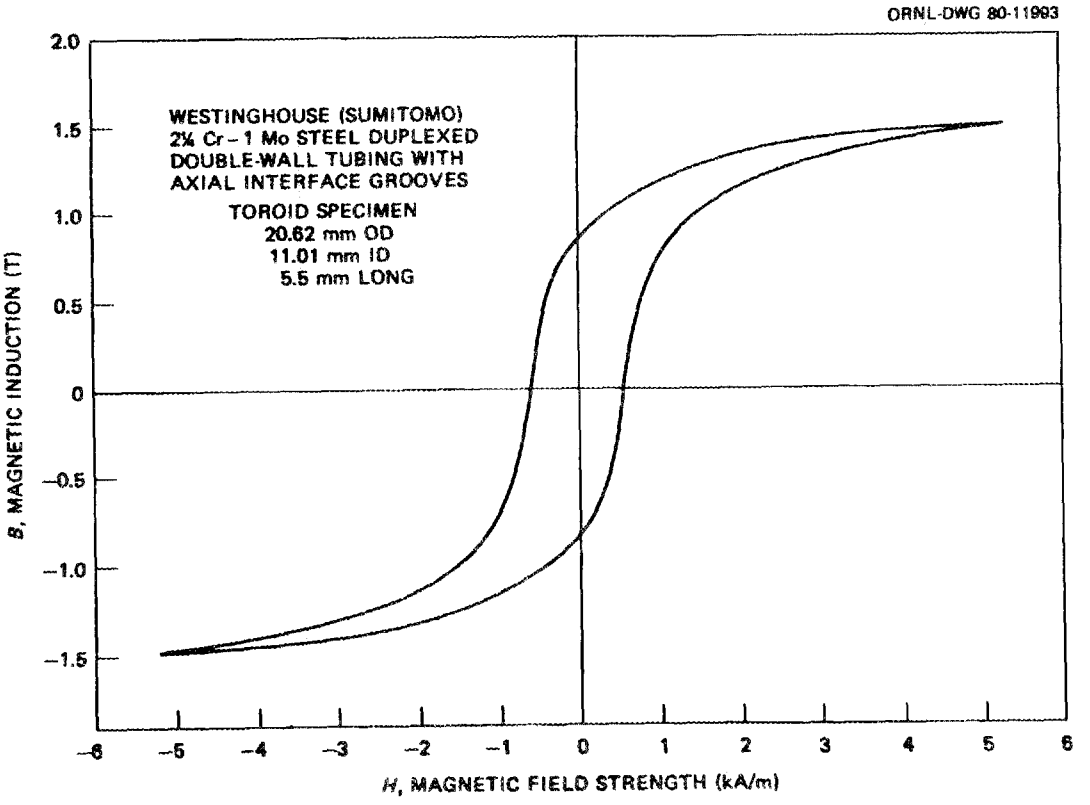


Fig. 4. B-H curve for 2.25Cr-1Mo steel tubing.

program, MAGSCN, listed in Appendix 2, that calculates the axial and radial components of the magnetic field at various points in the neighborhood of an air-core solenoid. This program was used to calculate the axial field strength, B_Z, at a point opposite the middle of the solenoid at the radius where the outside of the tube would be. The outside of the solenoid was just small enough to fit inside the tube, and the inside diameter and length of the solenoid were varied to optimize the design. The quality factor used in the optimization was the magnetic energy (proportional to B_Z^2) divided by the power supplied to the coil. Since both B_Z^2 and the power are proportional to $(NI)^2$, the square of the number of ampere-turns, the quality factor depends mainly upon the length and inner diameter of the coil. Too short a coil or too large an inner diameter will produce insufficient field strength at the outside of the tube; too long a coil or too small an inner diameter will produce greater field strength but consume too much power. Also, a short coil has a better resolution as an eddy-current detector. Figure 5 shows the quality factor for coils of various lengths and inner diameters; the outer diameter is fixed at 10.67 mm (0.420 in.) in order to pass through the 2.25Cr-1Mo tubing. In all cases, the number of ampere-turns was 200,000 and peak power was 1.03 MW. The optimum coil appears to have a length of about 7.87 mm (0.310 in.) and an inner diameter of about 4.57 mm (0.180 in.). The magnetic field at a point corresponding to the outside of the tube opposite the middle of the coil turns out to be 0.839 T for the optimum coil shape. Therefore, a larger peak current would presumably be required for saturation. However, the ferromagnetic tubing itself will contribute a considerable amount of spontaneous magnetization current, so that the final power requirements are best obtained experimentally with the actual material.

The coil could not long withstand such a high power, which is mainly dissipated by ohmic heating. Therefore, the pulse duration must be kept very short, about 0.435 ms, or just long enough to take the necessary number of readings. In addition, unless the pulse repetition rate or duty cycle is kept very small, some additional means of cooling the coil assembly is needed. Therefore, cooling passages are provided for coolant to flow through the center of the coil. Various coolants have been tried,

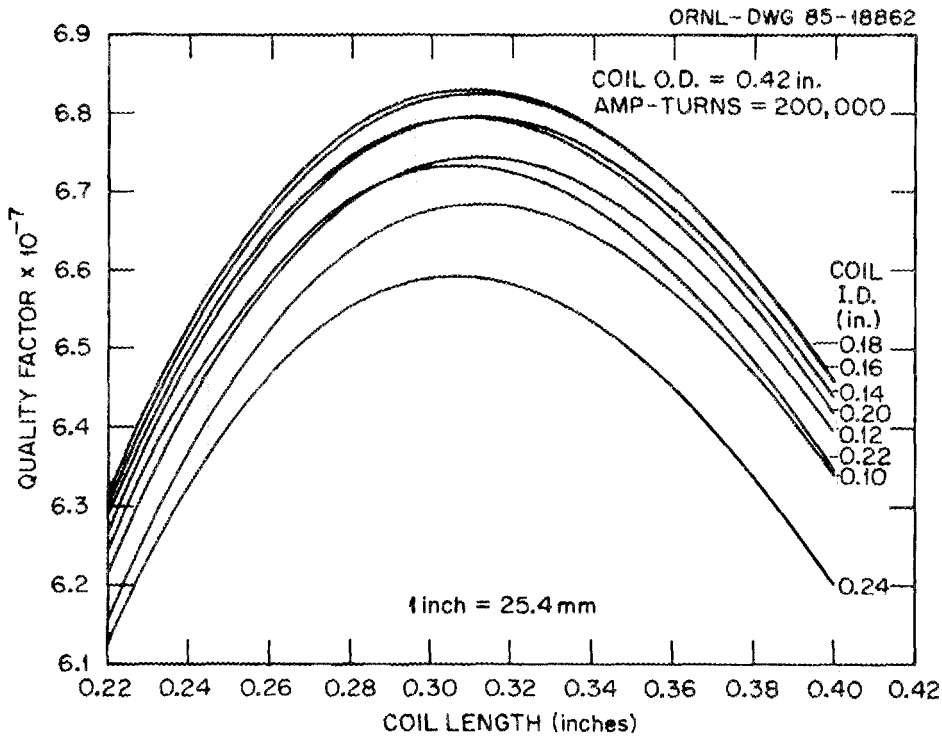


Fig. 5. Curves for optimizing the shape of the saturating coil. The quality factor is proportional to the ratio of magnetic energy to input power.

including air, water, and a mixture of water and alcohol; other coolants were also considered, but environmental or financial considerations ruled out many of them.

In addition to less heating, another advantage of a pulsed saturation system is the decreased "grabbing" of the saturating magnet, which would otherwise tend to be attracted strongly to one side or another of the ferromagnetic tube. With pulses only 435 μs long at intervals of about 1 s, the grabbing effect is hardly noticeable, since the inertia of the coil assembly prevents it from moving appreciably during the pulse.

DATA ACQUISITION SYSTEM

The signal from the driving coil saturates the tube wall and also induces currents in the tubing and in a pickup coil, whose output voltage depends on the tube properties as well as the driving pulse. The driving

and detection circuits had to be designed to allow a fast rise and a high-power pulse in the driver coil without too much ringing in the pickup coil. There had to be a careful balance between high-frequency components, which are easily transmitted, and low-frequency components, which are not.

The pickup output voltage can be considered as a "signature" characteristic of the original pulse shape modified by the tube effects. This signature contains much information, which can be extracted in various ways. One way is to measure the magnitude of the output voltages at selected times; this is the method used in Fig. 2. Another way is to measure the times at which the pulse reaches certain preset voltages; this is the method shown in Fig. 3. Still another way would be to pass the pulse through various filters to analyze the pulse into its Fourier components. The last method is complicated if many frequencies are involved and is also unnecessary and quite expensive. If the frequency components are desired, a fast Fourier transform (FFT) of the pulse height data can be obtained. But even this procedure is unnecessary, as we shall see.

Integrated circuit modules are available to determine and record either the pulse heights at selected times or the times to reach selected pulse heights. At present, the former method seems to offer advantages in terms of cost and accuracy with existing modules. Appendix 7 shows the circuit diagrams for a system to measure pulse heights at predetermined time intervals and send the data to a computer. In essence, this system is a multichannel analyzer adapted for this special application.

DATA PROCESSING SYSTEM

The use of multiple bits of information to determine various properties of a sample has already been described in several reports.[5,6] The basic method used at ORNL is to form a polynomial for each property desired. The polynomial for a given property consists of the bits of information raised to various powers and multiplied by coefficients determined by a least-squares method to produce the best fit to that particular property for data measured on known standard samples. It does not really matter what type of information is used for the polynomial variables, so long as the bits of information depend upon the desired

property in a different way from the other properties. The bits of information might be the output magnitudes and phases measured at selected frequencies, the pulse heights at selected times, or any other set of variables sufficient to distinguish variations in the given property.

Since this procedure has been thoroughly described in previous reports,[5,6] we shall say only that the program PULRDG, which is listed in Appendix 3, records the pulse information and PULFIT, which is listed in Appendix 4, processes the information. In addition to reading pickup coil voltages at regular time intervals and storing the data in an array called VOLTS, the PULRDG program also has the computer store the data describing the coil location, sample properties, and calibration data; move the probe to the proper location; etc. PULFIT prompts the operator to choose a type of polynomial, then makes the least-squares fit of the polynomial formed from readings on known standards to the actual values of the selected property and prints out the quality of the fit of the property values calculated from the polynomial, as well as the optimized coefficients in the polynomial. This procedure can be repeated as often as desired. The polynomial that produces the most satisfactory fit to the known properties can then be used to determine the value of that property from readings taken on a sample whose properties are not known. GRAPUL and GRAPUF, both listed in Appendix 5, are FORTRAN programs to graph the raw readings or the calculated property values on the printer of the IBM System 9000 computer. GPUL, listed in Appendix 6, is a microcomputer assembly language program that enables the NDT COMP9B microcomputer[8] to communicate with the peripheral equipment, such as the pulser, the probe positioner, and the pulse amplitude module.

ACTUAL OPTIMIZED DESIGN

A great advantage of this pulsed system compared with conventional eddy-current systems is that relatively little optimization is necessary, since the pulse already contains many frequency components and, therefore, much of the information used in multiple-frequency systems. Also, the hardware changes required with fixed-frequency systems are much more difficult and expensive than the simple software changes needed to change the times of measuring the output voltage magnitude of the pickup.

The main feature to be optimized is the design of the saturating coil to produce the maximum saturating field at the desired location with the least amount of power. Design studies such as that in Fig. 4 show the coil shape that will produce the maximum magnetic field at the outside of the tubing, opposite the middle of the saturating coil (where the pickup coil is located, as discussed in the next paragraph).

Other studies have shown that a small pickup coil wound around the outside of the saturating coil, as shown in Figs. 6 and 7, is much more sensitive in measuring tube thickness than a thin, disklike pickup coil located between the halves of a split saturating coil, as shown in Figs. 8 and 9. Figure 10 compares the peak pulse powers required to saturate and penetrate the tube wall thickness with the two different coil arrangements. The coil design with the solid driver and circumferential pickup requires much less peak power to penetrate the tube wall thickness. This is very important, as it reduces the heating of the coil or alternatively permits a faster pulsing rate. The improved performance is mainly a result of the increased density of turns in the driver coil, particularly in the middle, where the measurements are made. Some of the improvement also comes from the pickup turns being closer to the tube and the flaw, on the average (increased fill factor for the pickup making it more sensitive to the flaw). This type of coil is also much easier to construct.

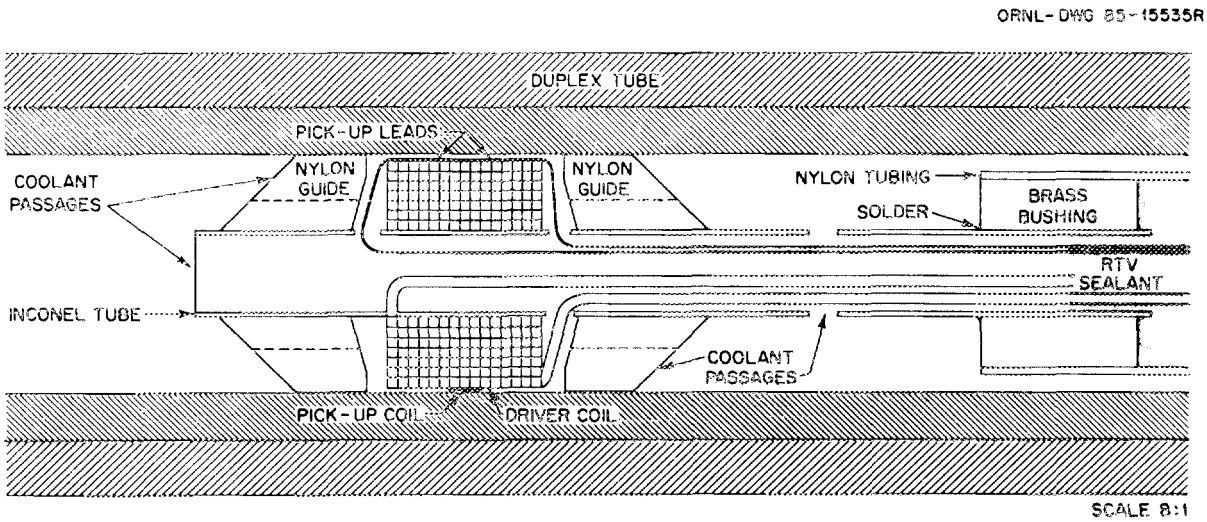


Fig. 6. Longitudinal section of the probe design with the pickup coil wound outside the middle of the solid saturating coil.

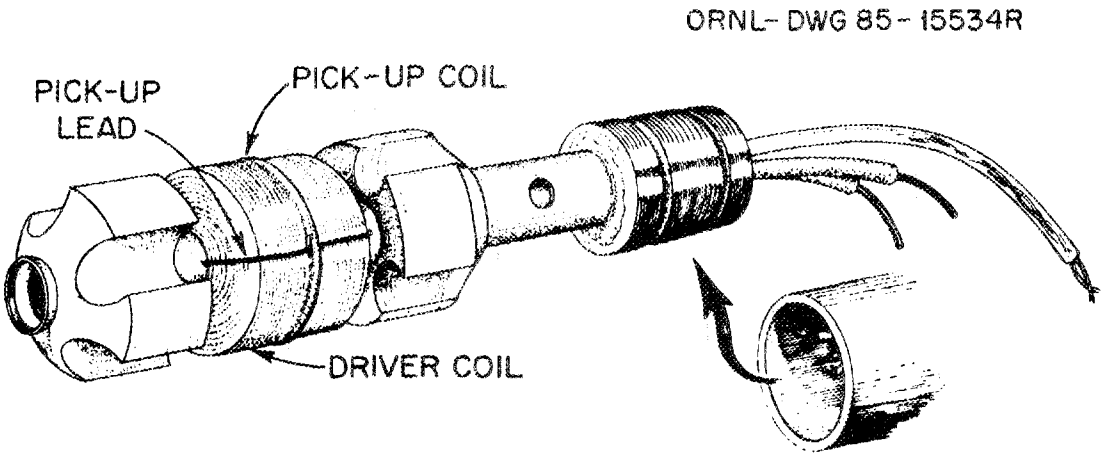


Fig. 7. Perspective view of the probe design with the pickup coil wound outside the middle of the solid saturating coil.

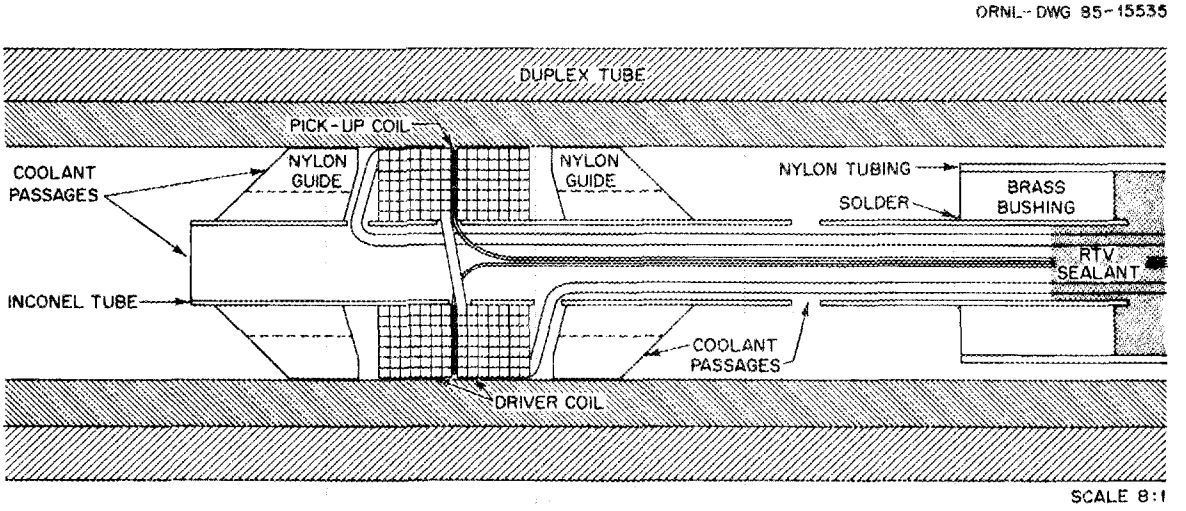


Fig. 8. Longitudinal section of the probe design with the pickup coil sandwiched between two halves of a split saturating coil.

PERFORMANCE OF THE SYSTEM

Figure 11 shows the various components of the system. An IBM System 9000 minicomputer is on the right, and pulser components are set up on the left, with the power supply on the bottom and the capacitor and switching circuit board on top. The capacitors are on the far side of the chassis, and the power transistors are in the black row on the front. The power supply and circuit board are shown mounted in a cabinet in Fig. 12.

ORNL-DWG 85-15534

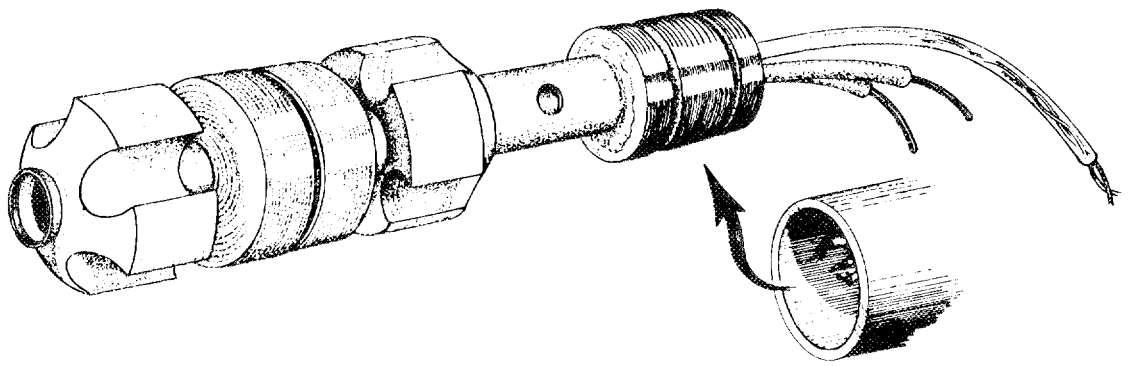


Fig. 9. Perspective view of the probe design with the pickup coil sandwiched between two halves of a split saturating coil.

ORNL-DWG 86C-7164R

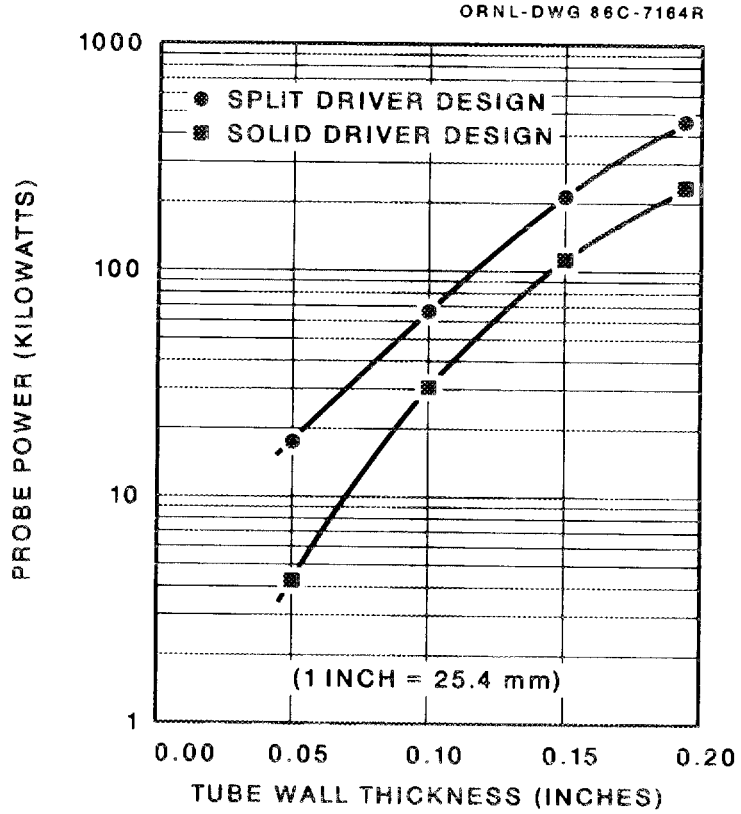


Fig. 10. Peak pulse power required to saturate various wall thicknesses of 2.25Cr-1Mo tubing.

ORNL-PHOTO 0233-85

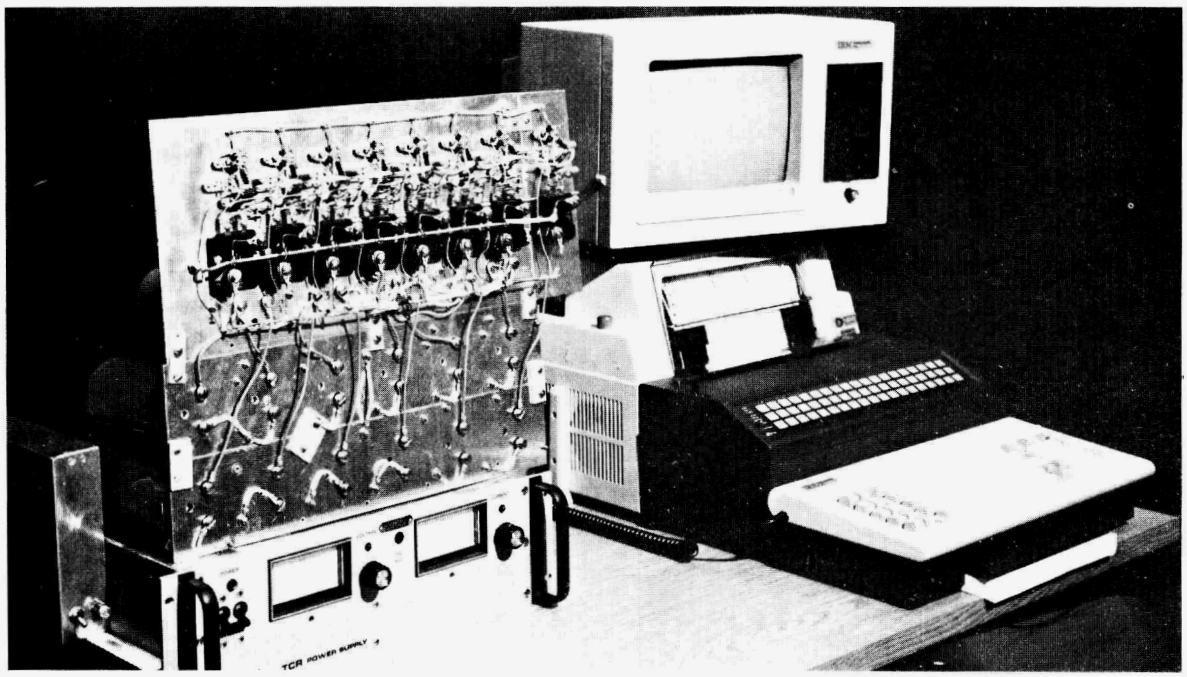


Fig. 11. Pulser components and IBM System 9000 computer.

YP1216

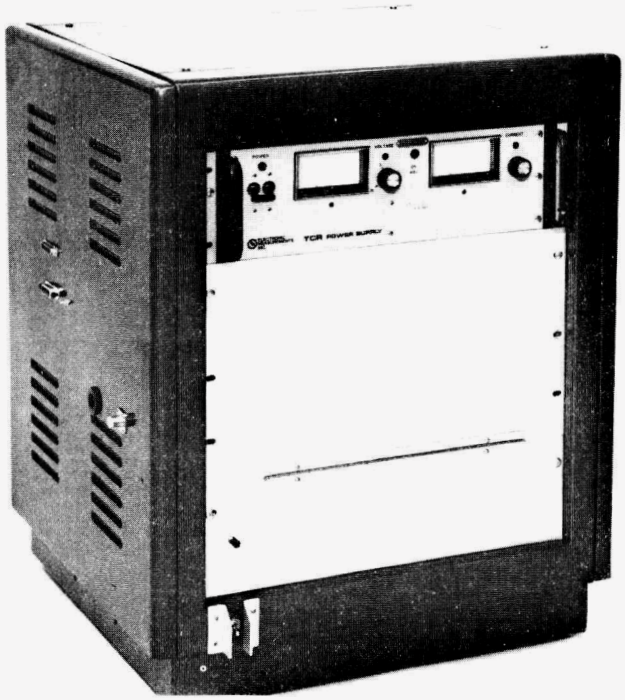


Fig. 12. Power pulser unit.

Figures 13 and 14 show a pulse saturating probe of the type with the pickup coil sandwiched between the split halves of the driver coil. Figure 13 also shows the pulser power supply and the machined standard tube, which has five different wall thicknesses and machined flat-bottom holes on the outside, as well as machined copper slip-on rings that are used to tell if the probe can "see through" the tube and is therefore saturating the metal.

ORNL-PHOTO 0234-85

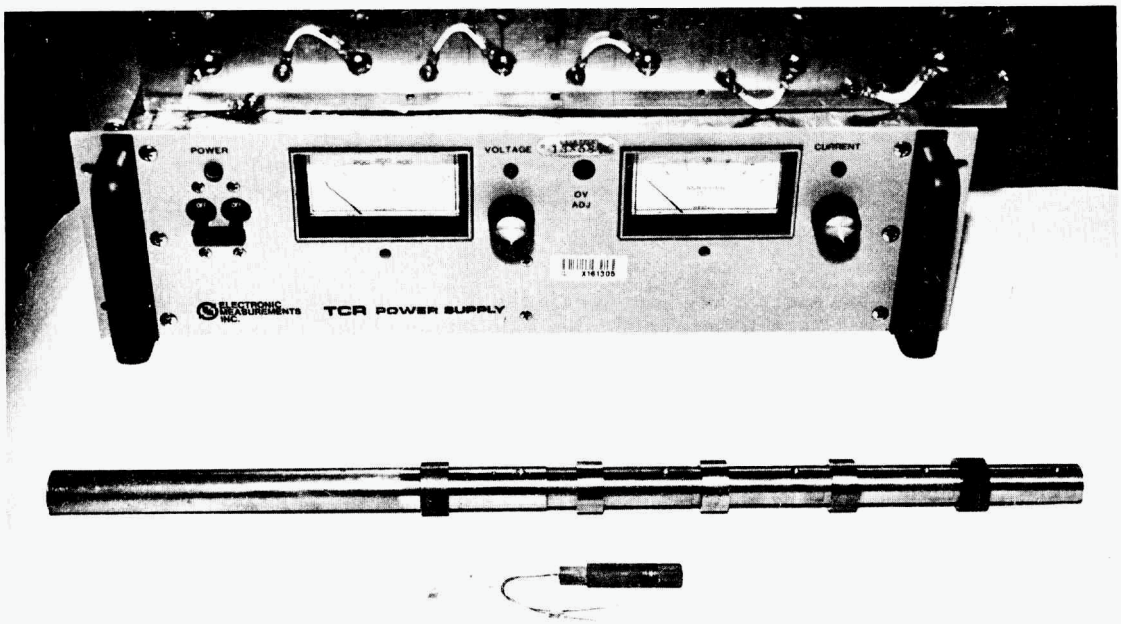


Fig. 13. Pulse saturation probe in front of machined standard tube and pulser power supply.

The system was tested and calibrated on machined standards in the laboratory. Some results are shown in Fig. 15 for thickness measurements in 2.25Cr-1Mo tubing. The tubing is ferromagnetic, with a nominal wall thickness of 4.75 mm (0.187 in.). Various flaws were machined on the outside of the tube. The machined wall thickness variations are easily distinguished, as are 0.48- and 1.2-mm (0.019- and 0.047-in.) circumferential notches and a circumferential band 6 mm wide and 0.48 mm deep (0.25 and 0.019 in.). An outside-diameter flat-bottom hole 1.6 mm in both diameter and depth could also be detected, but was barely above noise. Inside-diameter flaws are, of course, easily detected.

ORNL-PHOTO 0235-85

Fig. 14. Close-up of probe with split driver coil near machined standard tube.

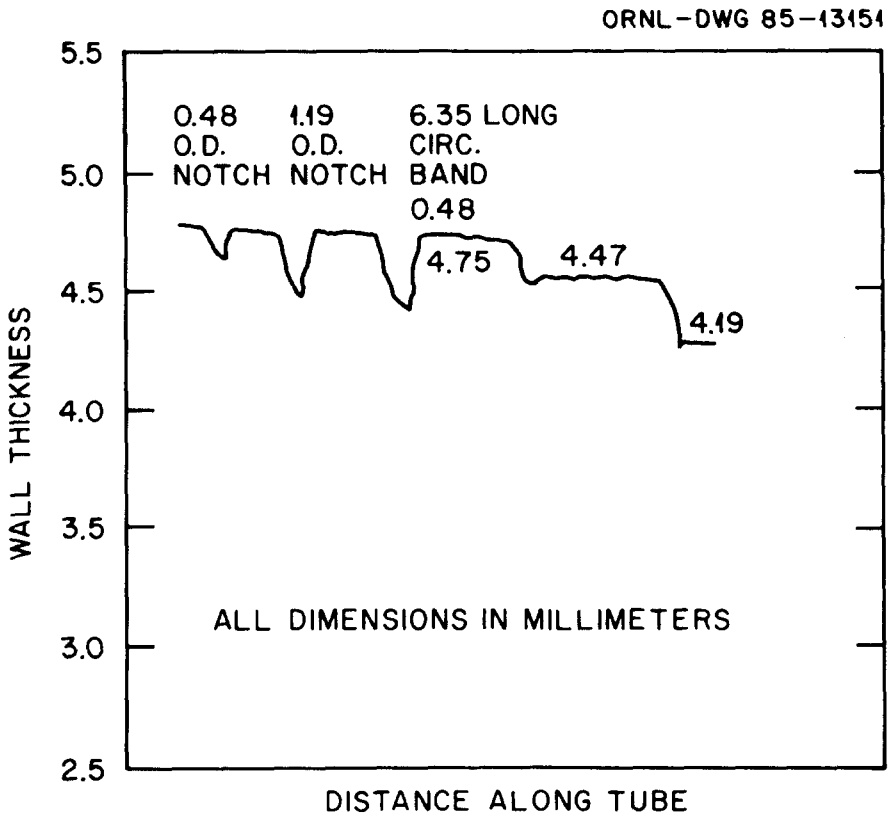


Fig. 15. Detection of wall thickness and outside diameter flaws in 2.25Cr-1Mo tubing.

The thickness fit was run with the split-driver probe design at a peak power level of 286 kW on a thickness calibration standard. The standard deviation of the thickness fit was 0.043 mm (0.0017 in.), and the "drift" was 0.019 mm (0.00074 in.). The drift is a worst-case error estimation, in which typical measurement errors are assumed to produce errors all adding in the same direction. The power level allowed complete penetration of the wall without producing a large signal from a copper ring around the outside of the tube.

A new water-cooled probe has been constructed and tested in the laboratory. New coils with Al_2O_3 insulation on square aluminum wire were also designed and constructed. The Al_2O_3 insulation is two orders of magnitude better at conducting the heat away from the interior of the coil than the insulation on the square copper wire. This should lead to better probe cooling, but it has not been tested.

FIELD DEMONSTRATION

To further demonstrate the performance of the system, a field demonstration was conducted at W-NCD in Pensacola, Florida. The ORNL mobile laboratory containing the pulse system equipment was transported to Pensacola on September 22, 1985. On September 23, the equipment was set up at the water tubesheet of the Westinghouse breeder steam generator. The equipment was shown to work properly, and each probe was tested on the laboratory demonstration standard. This demonstration was performed at the relatively slow rate of 1 pulse every 30 s, because no cooling was available on this side of the steam generator. However, prior to conducting an in-service inspection of selected steam generator tubes, an air regulator connected to a supply of compressed air was installed on the water side of the steam generator and hooked up to the tube to be inspected.

A 25-m (81-ft) probe was then inserted into tube 19 on the water side of the steam generator, and the air pressure was adjusted to 30 psi on the steam side. Although the air flow through this tube seemed weak at the time, the run was started. However, after 63 cm (25 in.) of tube had been inspected, the inner lead of the saturating coil burned out, because of lack of cooling, probably caused by the reduction in air flow/cooling resulting from 23 m (75 ft) of cable in the tube. The system had previously been tested only on short lengths of tubing. To avoid this type of problem, we recommend that future systems should incorporate a gage to measure coolant flow.

One of the two remaining probes [each of which was 14 m (45 ft) long] was then inserted, and the scan continued. The air flow at the same pressure was much greater with this shorter cable. The scan of tube 19 was completed; and tube 23 was also scanned, with only minor problems. Water (used in an earlier test of the probe) had leaked into the connector, causing one of the time channels to read incorrectly, and a computer cable that was improperly latched to the connector came loose, interrupting the scan.

At the exit from tube 19, in the tubesheet regions where the tube is expanded, signals indicating a change in wall thickness were observed. However, since the system was only calibrated for free tube regions, no conclusion could be drawn at that time.

After the inspection was completed, Westinghouse management was briefed on the development of the equipment, the tests performed, the problems encountered, and the solutions achieved. It was noted that the tests at Westinghouse were run at a slow speed (20 h per tube for 4 s per pulse), although laboratory tests with the probes indicated that faster speeds are possible. New probes with better cooling had already been designed to allow a tube to be inspected in 15 min, but these were not ready in time for this field demonstration. The probes were fabricated, but they were neither calibrated nor tested on standards. Generally such new probes need small design modifications after initial testing to achieve optimum performance.

It was recommended that improvements be made in the data recording and display. A small cyclic variation in the readings was produced by the slow mechanical printer. (After the trip, changes were made in software to allow plotting on a fast device and the variations disappeared.) The computer system has been tested at the rate of 20 samples per second without making permanent display and storage records. The data reading and display hardware for the field system should be upgraded to allow rapid display and storage when capital funds are available.

Although not part of the original development, testing of the pulsed system at Westinghouse, Pensacola, revealed sensitivity to the tube in the tubesheet (the expanded regions of the tube); therefore, calibrations should be performed and sensitivity levels should be determined for these regions also.

In general, the pulsed saturation eddy-current system was successfully demonstrated by this field test. Much valuable experience was gained, and ideas for improvements can be incorporated into future inspections.

CONCLUSIONS AND RECOMMENDATIONS

A pulsed magnetic saturation system has been developed that is capable of making accurate measurements through the wall of thick ferromagnetic tubing. The system is relatively simple electronically, compared with multiple-frequency phase-sensitive systems, and can be adapted to a wide variety of inspection problems.

Probe design is very important, but other test design problems are generally less critical than for multiple-fixed-frequency systems, since the pulse inherently contains a wide band of frequency components.

Inspection speed is slowed by the need to keep the average power to the probe within safe limits, but the ability to inspect through thick ferromagnetic samples is unequaled by conventional systems, except for specimens that can be located between the pole pieces of a large and powerful magnet.

Although the present system works extremely well, there are still areas in which further work might prove beneficial. These include the design of probes for various applications, the driving pulser, the multichannel data recorder, computer data handling, etc. At present, inspection speed is not limited by the computer but by the power handling capability of the probe. It is likely that further development could include use of materials with improved thermal properties, as well as increased flow of coolant.

ACKNOWLEDGMENTS

The authors wish to express their thanks to G. W. Scott for his preliminary investigations, to L. D. Chitwood and E. V. Davis for constructing the equipment, and to J. L. Bishop and D. L. Northern for making the manuscript legible.

REFERENCES

1. K. V. Cook, R. A. Cunningham, Jr., and W. A. Simpson, Jr., *Ultrasonic Techniques for Bore-Side Inspection of Tube-to-Tubesheet Welds*, ORNL/TM-7224, May 1980.

2. K. V. Cook and R. A. Cunningham, Jr., *Boreside Ultrasonic In-Service Inspection of Double-Wall Tubing (Status, Including Application to the Few Tube Model)*, ORNL/TM-8569, July 1983.

3. G. W. Scott, *Feasibility of Boreside Electromagnetic Inspection of 2 1/4 Cr-1 Mo Steel Tubing*, ORNL/TM-8706, June 1983.

4. G. W. Scott and L. D. Chitwood, *Eddy-Current In-Service Inspection of the Westinghouse Intermediate "A" Few Tube Steam Generator Model*, ORNL/TM-8615, June 1983.

5. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations*, ORNL/TM-6858, October 1979.

6. W. E. Deeds and C. V. Dodd, *Multiple Property Variations in Coaxial Cylindrical Conductors Determined with Multiple-Frequency Eddy Currents*, ORNL/NUREG/TM-335, November 1979.

7. C. V. Dodd, G. W. Scott, J. H. Smith, W. E. Deeds, L. D. Chitwood, and R. W. McClung, *Development of Multifrequency Eddy-Current Techniques for In-Service Inspection of 2 1/4 Cr-1 Mo Steam Generator Tubing for CRBR*, ORNL/TM-8920, February 1984.

8. C. V. Dodd and R. F. Cowan, *The NDT-COMP9 Microcomputer*, ORNL/NUREG/TM-390, September 1980.

Appendix 1

PROGRAM PULSE

PULSE is a FORTRAN program to evaluate the defect response of a reflection-type coil above a multilayer conductor with (N9 - 1) layers (up to 9). In the example shown, N9 = 3 for a single layer of iron 5.08 mm thick (0.200 in.) of resistivity RHO = 6.07 μΩ-cm, with air on both sides. A total of NFT = 80 frequencies, at integral multiples of F = 100 Hz, are used to synthesize a square-wave pulse. The program calculated the magnitude (TMAG) and phase (PHASE) of the pickup output voltage, as well as the coil impedances and the defect contributions to the impedance (GH and IJ) and the magnitudes and phases (TRGFMG and PHADF) of the output voltage at NCOUNT = 5 time intervals after the start of the pulse. There are interactive design sections for changing the coil and attenuator designs and evaluating the probable drifts, or circuit sensitivity to small random errors.

The assumed pulse shape can be changed by altering the coefficients in the Fourier series, and the resulting pulse shape is written in File 34 for later plotting.

```
      PROGRAM    PULSE
C     VERSION 1 JUNE 1984
C        CALCULATES THE SENSITIVITY OF AN EDDY-CURRENT PULSE
C        TO A DEFECT LOCATED IN ANY LAYER OF A MULTI-LAYERED MATERIAL
C
C        SEE C. V. DODD ET AL.,
C            THE ANALYSIS OF REFLECTION-TYPE COILS FOR EDDY-CURRENT
C            TESTING, ORNL-TM-4107, APRIL 1973
C
      DIMENSION T(10), U(10), RHO(10), BETA(10), MFAC(10,80),
     & V(2,2), TR(2,2), SHIFT(5), RL(5), PHASE(5,80),
     & TMAG(5,80), GAMMA1(80), TRDFMG(5,80), PHADF(5,80),
     2 GAMMA2(80), MUT(5,80), PICKUP(5,80), DRIVER(5,80), GH(5,80),
     3 NCR(8,2),IJ(5,80),VOLT(80),
     & SUMT(5),SUMTDF(5),SUNDIF(5),SINCOE(80)
      CHARACTER*4 CURNAM(4),HDVR,HDPU,HDRIV(2),HPICK(2)
      COMPLEX BETA0, BETA1, BETA, V, TR, GAMMA1, GAMMA2, MUT, GM1, GM2
      COMPLEX DRIVER, PICKUP, GH, IJ, EX, CZRO, CTMP
      REAL L2, L3, L4, L5, L6, L7, MFAC, N3, N4
      COMMON X, Z, Q1, PI
      COMMON /B1/ BETA0, BETA1
      COMMON /B2/ R1, R2, R3, R4, L3, L4, R0, R6, R7, R9, C6, C7,
     1 V0, G5, W, F, R5, N3, N4
      COMMON /B3/ MUT, DRIVER, PICKUP, A1R1, A1R2, GH, IJ
      COMMON /B4/ GAGE, XIN, XOUT, XLEN, TURNS, N1A, J1, PERLAY, XLAY
      COMMON /B5/ V5, A1, D0
      DATA LOU/6/,LOK/0/,L1K/0/,TWOPI,RAD /6.28318531, 57.2957795/
      DATA CURNAM/'PUL ','OUT ','DEF ','DIF '/,HDVR,HDPU/'DVR','P-U'/
      DATA HDRIV, HPICK /'DRIV','ER  ','PICK','UP  ' /
      DATA RHO/1.E10,6.07,1.E10,7*0./
      DATA U/3*1.,7*0./
      DATA T/1.E10,0.200,1.,7*0./
      DATA CZRO/(0.,0.)/
      OPEN(LOU,FILE='*PR')
      OPEN(34,FILE='PLOT.DAT',STATUS='NEW')
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PULSE    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
      TIM1=(FLOAT(IHR)*60.+FLOAT(IMN))*60.+FLOAT(ISE)+FLOAT(IFR)/32768.
C
C REFER TO FIGURES 2 AND 4, ORNL-TM-4107
C
      JKL = 0
      R5 = .40
      R1 = .75
      R2 = 1.25
      L3 = .4
      R3 = .35
      R4 = .7
      L4 = .0667
      L5 = 0.
      L6 = .05
      R6 = 14.2
      R7 = 165
      N3 = 414.
      N4 = 450.
      R0 = 1.E6
      R9 = 1.E6
      C6 = 2.E-12
```

```
      C7 = 0.47E-12
      V0 = 10.
      C5 = 1000.
      F = 1.E2
      L2 = .0125
C
C     NFT IS THE NUMBER OF FREQUENCIES,
C     AND F IS THE FUNDAMENTAL FREQUENCY.
C     N9 IS THE TOTAL NUMBER OF REGIONS INCLUDING THE COIL REGION
C     N8 IS THE REGION CONTAINING THE DEFECT
C
      NFT = 80
      N9 = 3
      N8 = 2
      IF (N9 .LT. 2 .OR. N8 .GE. N9) GO TO 1300
      L7 = L4 + L5
      L7 = L3 - L7 - L7
      R8 = 1.
      Z7 = .005
      V5 = 6.54497E-5
      A1 = 1.
C     SET SINE & COSINE COEFFICIENTS FOR A SQUARE WAVE
      DO 3 N=1,NFT
      SINCOE(N)=6.366197723/FLOAT(2*N-1)
    3 CONTINUE
    5 W = TWOPI*F
      IF (JKL .EQ. 0) WRITE(LOU,10)
 10   FORMAT ('  N', 13X, 'THICK.(IN)     R(M-OHM CM)    M,SIGMA',
     1  9X, 'U'/1X)
C
C     CALCULATES THE VALUE OF RBAR**2*W*PERM*COND (MFAC(I,M))
C     FOR ALL LAYERS(I) AND FREQUENCIES(M).
C
      CON=.50939792*F*R5*R5
      DO 50 I=1,N9
      SUM = 0.
      VCON = 0.
      IF (RHO(I) .LE. 1.E9) VCON = CON*U(I)/RHO(I)
      DO 35 M=1,NFT
      MFAC(I,M)=VCON*FLOAT(2*M-1)
   35 CONTINUE
      IF(JKL.EQ.0)WRITE(LOU,40)I,T(I),RHO(I),MFAC(I,1),U(I)
   50 CONTINUE
      T(N9) = 0.
      D0 = 6.0805E-2*F*R5*R5/RHO(N8)
      IF (JKL .NE. 0) GO TO 105
      WRITE(LOU,470)
      I1 = 1
      I2 = 2
      WRITE(LOU,60) I1, R1, I2, R2, HDRIV, L3
      I1 = 3
      I2 = 4
      WRITE(LOU,60) I1, R3, I2, R4, HPICK, L4
      WRITE(LOU,80) R5, NFT, F
      WRITE(LOU,90) L5
      WRITE(LOU,100) L6, L2
      R85 = R8*R5
      WRITE(LOU,103) R85, Z7, N8
C
C     SET FIRST INTERVAL AND STEP SIZE FOR NUMERICAL INTEGRATION
```

```
C
  105 S1 = .01
      S2 = 5.
      B1 = 0.
      B2 = S2
C
C     INITIALIZE THE SUMS OF THE INTEGRAND TO ZERO
C
      DO 107 M=1,NFT
      DO 107 I=1,5
      MUT(I,M) = CZRO
      GH(I,M) = CZRO
      DRIVER(I,M) = CZRO
      IJ(I,M) = CZRO
  107 PICKUP(I,M) = CZRO
      AIR1 = 0.
      AIR2 = 0.
  110 I1 = (B2 - B1)/S1
C
C        INTEGRATE WITH THE RECTANGLE RULE, EVALUATING AT THE MIDPOINT
C     OF EACH STEP.THE FUNCTIONS EVALUATED ARE CONTAINED IN EQ(9),
C     EQ(17), AND EQ(21) IN THE REFLECTION COIL REPORT.
C
      X = B1 - .5*S1
      DO 390 IDX=1,I1
      X = X + S1
C     CALCULATION OF EXPONENTIAL FUNCTIONS
      TEST = X*L3
      W3 = 0.
      IF (TEST .LT. 30.) W3 = EXP(-TEST)
      W8 = 1. - W3
      TEST = X*L4
      W4 = 0.
      IF (TEST .LT. 30.) W4 = EXP(-TEST)
      W9 = 1. - W4
      TEST = X*L7
      W7 = 0.
      IF (TEST .LT. 30.) W7 = EXP(-TEST)
      W0 = 1. - W7*W4
      TEST = X*L5
      W5 = 0.
      IF (TEST .LT. 85.) W5 = EXP(-TEST)
C     CALCULATION OF XJ1(X) INTEGRALS
      Z = X*R2
      Q1 = R2
      CALL BESSEL(VAL2)
      Z = X*R1
      Q1 = R1
      CALL BESSEL(VAL1)
      Z = X*R4
      Q1 = R4
      CALL BESSEL(VAL4)
      Z = X*R3
      Q1 = R3
      CALL BESSEL(VAL3)
      RJ1 = BSJ1(X*R8)
      D21 = VAL2 - VAL1
      D43 = VAL4 - VAL3
      S6 = S1*D21
      S7 = S1*D43
```

```
      S3 = S6*D43
      S4 = S6*D21
      S6 = S6*RJ1
      S5 = S7*D43
      W6 = 0.
      TEST = X*L6
      IF (TEST .GT. 20.) GO TO 385
      W6 = EXP(-TEST)
      S7 = S7*RJ1
      T1 = W6*W9*W0
      A9 = T1*S7
      T2 = W6*W8
      A8 = T2*S6
      T1 = T1*W5
      A5 = T1*T2*S3
      A6 = T2*T2*S4
      A7 = T1*T1*S5
C
C          CALCULATION OF MATRICES
C
      IF (X .GT. 40.) GO TO 385
      XX=X*X
      XXXX=XX*XX
      NMIN = N9
      DO 310 MF=1,NFT
      NR=NMIN
      QSUM = 0.
  180 BETA(NR) = CMPLX(X, 0.)
      IF(MFAC(NR,MF).EQ.0.)GO TO 190
      X1=SQRT(0.5*(XX+SQRT(XXXX+MFAC(NR,MF)*MFAC(NR,MF))))/U(NR)
      Y1=MFAC(NR,MF)/(2.*X1*U(NR)*U(NR))
      BETA(NR)=CMPLX(X1,Y1)
  190 QSUM = QSUM + REAL(BETA(NR))*T(NR)/R5
C     WRITE(LOU,*)X,MF,NR,BETA(NR),MFAC(NR,MF)
      IF (QSUM .GT. 30. OR. NR .EQ. 1) GO TO 210
      NR = NR - 1
      GO TO 180
  210 N7 = NR
      NMIN=NR
      V(1,2) = BETA(NR+1) - BETA(NR)
      V(2,2) = BETA(NR+1) + BETA(NR)
      BETA0 = BETA(NR+1)
      N72 = N7 + 2
      IF (N72 .GT. N8) GO TO 250
      DO 240 NR=N72,N8
      BETA1 = BETA(NR)
      NN = NR
      CALL XFORM(NN, TR, R5, U, T)
      JUMP = 1
      CALL MATRIX(JUMP, V, TR, BETA1)
  240 BETA0 = BETA1
  250 IF (N8 .LT. N7 + 1) GO TO 270
      NR = N8
      EX = CEXP(BETA(NR)*(U(NR)*(T(NR) - Z7)/R5))
      V(1,1) = V(1,2)/EX + V(2,2)*EX
      IF (N8 .LT. N7 + 2) GO TO 290
      N7 = N8 - 1
      GO TO 290
  270 IF (N8 .LT. N7) GO TO 280
      EX = CEXP(BETA(NR)*U(NR)*Z7/R5)
```

```
      V(1,1) = (BETA(NR+1) + BETA(NR+1))/EX
      GO TO 290
  280 V(1,1) = CZRO
  290 N72 = N7 + 2
      IF (N72 .GT. N9) GO TO 302
      DO 300 NR=N72,N9
      NN = NR
      BETA1 = BETA(NR)
      CALL XFORM(NN, TR, R5, U, T)
      JUMP = 0
      CALL MATRIX(JUMP, V, TR, BETA1)
  300 BETA0 = BETA1
  302 GAMMA1(MF) = V(1,2)/V(2,2)
  310 GAMMA2(MF) = V(1,1)/V(2,2)
      RL(1) = 1.
      TEST = X*L2
      W2 = 0.
      IF (TEST .LT. 30.) W2 = EXP(-TEST)
      SUM = TEST
      DO 370 K=2,5
      SUM = SUM + TEST
      RL(K) = 0.
      IF (SUM .LT. 30.) RL(K) = W2*RL(K-1)
  370 CONTINUE
      DO 380 M=1,NFT
      GM1 = GAMMA1(M)
      GM2 = GAMMA2(M)
      DO 375 K=1,5
      CTMP = GM1*RL(K)*RL(K)
      MUT(K,M) = MUT(K,M) + CTMP*A5
      DRIVER(K,M) = DRIVER(K,M) + CTMP*A6
      PICKUP(K,M) = PICKUP(K,M) + CTMP*A7
C
C        GH AND IJ ARE CORRECTIONS TO THE DRIVER AND PICKUP
C     IMPEDANCES DUE TO THE DEFECT
C
      CTMP = GM2*RL(K)
      GH(K,M) = GH(K,M) + CTMP*A8
      IJ(K,M) = IJ(K,M) + CTMP*A9
  375 CONTINUE
  380 CONTINUE
C     CALCULATION OF AIR VALUE
  385 AIR1 = AIR1 + (S4 + S4)*(X*L3 - W8)
      TQ = X*L4 - W9
      AIR2 = AIR2 + (S5 + S5)*(TQ + TQ - W7*W9**2)
  390 CONTINUE
C
C         END OF INTEGRATION LOOP
C
      B1 = B2
      B2 = B2 + S2
      S1 = .05
      IF (X .LT. 9.) GO TO 110
      S1 = .1
      IF (X .LT. 29.) GO TO 110
      S1 = .2
      IF (X .LT. 39.) GO TO 110
      S1 = .5
      IF (X .LT. 79.) GO TO 110
C
```

```
C       INTEGRATION IS COMPLETE.
C
        CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
        TIM2=FLOAT(IHR)*3600.+FLOAT(IMN)*60.+FLOAT(ISE)+FLOAT(IFR)/32768.
        TIME=TIM2-TIM1
        WRITE(LOU,391)TIME
  391   FORMAT(' ELAPSED TIME (SEC) ',F12.4)
C
C       COMPUTE AND PRINT OUTPUT QUANTITIES
C       FOR VARIOUS COIL AND ATTENUATOR PARAMETERS.
C
 395    T1 = N3/((R2-R1)*L3)
        T2 = N4/((R4-R3)*L4)
        DO 519 M=1,NFT
        MF=2*M-1
        W=MF*TWOPI*F
        Q0=6.30047520E-7*F*MF*R5
        QDFIND = 0.
C
C       MAGNITUDE AND PHASE OF OUTPUT VOLTAGE & COIL IMPEDANCE
C       ARE CALCULATED FOR NO DEFECT PRESENT.
C
        CALL CIRCT (TMAG, PHASE, Q0, T1, T2, QDFIND, M)
        IF(M.EQ.1)WRITE(LOU,410) HDRIV
        Q1 = Q0*T1*T1*AIR1/W
        Q2 = (REAL(DRIVER(3,M)) + AIR1)/AIR1
        Q3 = Q0*T2*T2*AIR2/W
        Q4 = (REAL(PICKUP(3,M)) + AIR2)/AIR2
        DIND = Q1
        DNIM = Q2
        PIND = Q3
        PNIM = Q4
        IF(M.EQ.1)WRITE(LOU,420) R6, Q1, N3, C6, Q2
        IF(M.EQ.1)WRITE(LOU,410) HPICK
        IF(M.EQ.1)WRITE(LOU,420) R7, Q3, N4, C7, Q4
        IF(M.EQ.1)WRITE(LOU,440)
        IF(M.EQ.1)WRITE(LOU,450) V0, R0, C5, R9
        IF(M.EQ.1)WRITE(LOU,470)
        RL(1) = L6
        RL(2) = RL(1) + L2
        RL(3) = RL(2) + L2
        RL(4) = RL(3) + L2
        RL(5) = RL(4) + L2
        IF(M.EQ.1)WRITE(LOU,490) RL
        IF(M.EQ.1)WRITE(LOU,470)
        IF(M.EQ.1)WRITE(LOU,510) (TMAG(J,M), J=1,5)
        IF(M.EQ.1)WRITE(LOU,510) (PHASE(J,M), J=1,5)
        IF(M.EQ.1)WRITE(LOU,470)
C
C       MAGNITUDE AND PHASE OF OUTPUT VOLTAGE AND COIL IMPEDANCES
C       ARE CALCULATED FOR THE DEFECT(QDFIND) PRESENT.
C
  512   QDFIND =6.0805E-2*F*MF*R5*R5*V5*A1/RHO(N8)
        CALL CIRCT (TRDFMG, PHADF, Q0, T1, T2, QDFIND, M)
        IF (M .EQ. 1) WRITE(LOU,513) V5, A1
  513   FORMAT(1H ,'DEFECT VOL IS',1PE13.6,
       1 '  SHAPE AND ORIENTATION FACTOR',E13.6)
        IF(M.EQ.1)WRITE(LOU,470)
        IF(M.EQ.1)WRITE(LOU,510) (TRDFMG(J,M), J=1,5)
        IF(M.EQ.1)WRITE(LOU,510) (PHADF(J,M), J=1,5)
```

```
      IF(M.EQ.1)WRITE(LOU,470)
      SEN = SEN*.2
C 519 WRITE(LOK,470)
  519 CONTINUE
C
C     INTERACTIVE DESIGN CHANGE SECTION
C
  520 WRITE(LOK,530)
      READ(LIK,540)N5
      GO TO (550, 680, 760, 1020, 1300), N5
C
C     COIL DESIGN SECTION - USER IS TO ENTER 4 INTEGERS,ONE
C     FOR EACH OF 4 FACTORS.ZERO MEANS NO CHANGE TO THE INITIAL
C     DATA.
C
  550 WRITE(LOK,560) HDRIV, HPICK
      READ(LIK,*) N1A, N2A, N3A, N4A
      IF (N1A*N2A .EQ. 0) GO TO 580
      GAGE = N1A
      XIN = R1*R5
      XOUT = R2*R5
      XLEN = L3*R5
      TURNS = N2A
      N3 = N2A
      N1A = -1
      J1 = 1
      CALL GAGER(R6)
      GO TO 600
  580 IF (N1A .EQ. 0) GO TO 590
      GAGE = N1A
      XIN = R1*R5
      XOUT = R2*R5
      XLEN = L3*R5
      J1 = 1
      CALL GAGER(R6)
      N3 = TURNS
      GO TO 600
  590 IF (N2A .EQ. 0) GO TO 620
      N3 = N2A
      TURNS = N2A
      XIN = R1*R5
      XOUT = R2*R5
      XLEN = L3*R5
      J1 = 0
      CALL GAGER(R6)
  600 WRITE(LOK,610) HDRIV, TURNS, GAGE, PERLAY, XLAY, R6
  620 IF (N3A*N4A .EQ. 0) GO TO 630
      GAGE = N3A
      TURNS = N4A
      N4 = N4A
      XIN = R3*R5
      XOUT = R4*R5
      XLEN = L4*R5
      J1 = 1
      CALL GAGER(R7)
      R7 = R7 + R7
      GO TO 650
  630 IF (N3A .EQ. 0) GO TO 640
      GAGE = N3A
      XIN = R3*R5
```

```
      XOUT = R4*R5
      XLEN = L4*R5
      J1 = 1
      CALL GAGER(R7)
      R7 = R7 + R7
      N4 = TURNS
      GO TO 650
  640 IF (N4A .EQ. 0) GO TO 670
      N4 = N4A
      TURNS = N4A
      XIN = R3*R5
      XOUT = R4*R5
      XLEN = L4*R5
      J1 = 0
      CALL GAGER(R7)
      R7 = R7 + R7
  650 WRITE(LOK,610) HPICK, TURNS, GAGE, PERLAY, XLAY, R7
  670 GO TO 395
C
C     ATTENUATOR DESIGN SECTION - ACCEPT E-FIELD DATA FOR 4
C     ATTENUATOR PARAMETERS. ZERO MEANS NO CHANGE FROM INITIAL
C     DATA.
C
 680  WRITE(LOK,690) F
      READ(LIK,*) P1, P2, P3, P4
      IF (P1 .NE. 0.) R0 = P1
      IF (P2 .NE. 0.) C6 = P2
      IF (P3 .NE. 0.) R9 = P3
      IF (P4 .NE. 0.) C7 = P4
      Q7 = DIND*DNIM
      Q6 = SQRT(Q7/C6)
      Q5 = 1./(W*SQRT(Q7*C6))
      Q7 = R0/(W*Q7)
      Q10 = PIND*PNIM
      Q9 = SQRT(Q10/C7)
      Q8 = 1./(W*SQRT(Q10*C7))
      Q10 = R9/(W*Q10)
      WRITE(LOK,750) HDVR, Q5, Q6, Q7
      WRITE(LOK,750) HDPU, Q8, Q9, Q10
      GO TO 395
C
C     SECTION TO CALCULATE THE EFFECT OF DRIFTS ON THE SYSTEM.
C     THE VALUE OF THE DRIFT MUST BE TYPED IN BEFORE THE PROGRAM
C     IS EXECUTED.
C
  760 WRITE(LOK,770)
      DR1 = .01
      DR2 = .01
      DR3 = .01
      DR4 = .01
      DR5 = .01
      DR6 = .01
      DR7 = .01
      DR8 = .01
      DR9 = .01
      GO TO 790
  790 WRITE(LOK,800)
      IF (DR1 .EQ. 0.) GO TO 830
      R6 = R6*(1. + DR1)
      DR100 = 100.*DR1
```

```
      WRITE(LOK,820) DR100, HDVR
      R6 = R6/(1. + DR1)
  830 IF (DR2 .EQ. 0.) GO TO 850
      R7 = R7*(1. + DR2)
      DR100 = 100.*DR2
      WRITE(LOK,820) DR100, HDPU
      R7 = R7/(1. + DR2)
  850 IF (DR3 .EQ. 0.) GO TO 870
      C6 = C6*(1. + DR3)
      DR100 = 100.*DR3
      WRITE(LOK,860) DR100, HDVR
      C6 = C6/(1. + DR3)
  870 IF (DR4 .EQ. 0.) GO TO 890
      C7 = C7*(1. + DR4)
      DR100 = 100.*DR4
      WRITE(LOK,860) DR100, HDPU
      C7 = C7/(1. + DR4)
  890 IF (DR5 .EQ. 0.) GO TO 910
      R0 = R0*(1. +DR5)
      DR100 = 100.*DR5
      WRITE(LOK,900) DR100
  900 FORMAT (1X, F4.1, 10X, 'SERIES RES    ')
  910 IF (DR6 .EQ. 0.) GO TO 930
      R9 = R9*(1. + DR6)
      DR100 = 100.*DR6
      WRITE(LOK,920) DR100
      R9 = R9/(1. + DR6)
  930 IF (DR7 .EQ. 0.) GO TO 950
      V0 = V0*(1. + DR7)
      DR100 = 100.*DR7
      WRITE(LOK,940) DR100
      V0 = V0/(1. + DR7)
  950 IF (DR8 .EQ. 0.) GO TO 980
      F = F*(1. + DR8)
      DR100 = 100.*DR8
      WRITE(LOK,960) DR100
      GO TO 5
  970 F = F/(1. + DR8)
  980 IF (DR9 .EQ. 0.) GO TO 520
      R5 = R5*(1. + DR9)
      DR100 = 100.*DR9
      WRITE(LOK,990) DR100
  990 FORMAT (1X, F4.1, 10X, 'MEAN RADIUS   ')
      GO TO 5
 1000 R5 = R5/(1. + DR9)
       GO TO 520
 1020 WRITE(LOK,1025)

C     PULSE CALCULATIONS - THIS SECTION ALLOWS THE INPUT PULSE
C     COEFFICIENTS TO BE CHANGED. THE INPUT AND OUTPUT PULSES
C     CAN BE CALCULATED,STARTING AT ANY TIME(TSTART),WITH ANY
C     NUMBER OF STEPS(NCOUNT),OVER ANY INTERVAL(TINTV).
C
 1025 FORMAT (' 1 CHANGE COEF 2 TYPE COEF 3 PLOT PULSE  4 CON',1X,\)
      READ(LIK,*)N5
      GO TO (1030, 1041, 1050, 520), N5
 1030 WRITE(LOK,1035)
 1035 FORMAT ('  M  OLD COS COEF  SIN COEF   NEW COS COEF  SIN COEF',1X,\)
      DO 1040 M=1,NFT
      WRITE(LOK,1037) M,  SINCOE(M)
```

```
 1037 FORMAT (1X, I2, 1P2E12.5, 3X,\)
      READ(LIK,*)  SINCOE(M)
 1040 CONTINUE
      GOTO 1020
 1041 WRITE(LOK,1042)
 1042 FORMAT ('0 M  COS COEF   SINE COEF',1X,\)
      DO 1044 M=1,NFT
      WRITE(LOK,1043) M,  SINCOE(M)
 1043 FORMAT (1X, I2, 1P2E14.5)
 1044 CONTINUE
      GO TO 1020
 1050 WRITE(LOK,1051)
 1051 FORMAT (' START TIME, INTERVAL, NO. OF STEPS ,NO OF CURVES ? ',\)
      READ(LIK,*) TSTART, TINTV, NCOUNT,NCURVE
      WRITE(LOK,1052)
 1052 FORMAT(' WRITE PLOTFILE 1.PULSE 2.OUT(J) 3.DEF(J) 4.DIF(J) ',\)
      DO 1054 NC=1,NCURVE
      WRITE(LOK,1053)NC
 1053 FORMAT(' CURVE TYPE,LO VALUE FOR CURVE',I2,'?  ',\)
      READ(LIK,*)NCR(NC,1),NCR(NC,2)
 1054 CONTINUE
C
C       SECTION TO WRITE HEADINGS ON PLOT FILE NO 34
C
      REWIND 34
      WRITE(34,1055)
 1055 FORMAT(' 3,36,5.1,8.3',/,'PULSE RESPONSE OF REFLECTION COIL')
      WRITE(34,1056)RS,RHO(2),T(2)
 1056 FORMAT(' RBAR=',F5.3,' RHO=',F6.3,' THK=',F5.3)
      WRITE(34,1057)(CURNAM(NCR(NC,1)),NCR(NC,2),NC=1,NCURVE)
 1057 FORMAT(6(A4,I1,1X))
      WRITE(LOK,1057)(CURNAM(NCR(NC,1)),NCR(NC,2),NC=1,NCURVE)
      WRITE(34,1058)NCURVE
 1058 FORMAT('LINLIN',/,'TIME(SECONDS)',/
     & ,'VOLTAGE',/,'8',/,I1)
      DO 1150 NC=1,NCURVE
      NPRT=NCR(NC,1)
      LPRT=NCR(NC,2)
      DELTIM = TINTV/FLOAT(NCOUNT)
      IF(NPRT.LT.1)GO TO 1060
      NPTS=NCOUNT+1
      WRITE(34,*)NPTS
 1060 CONTINUE
 1070 FORMAT ('01NPUT PULSE   OUT,NO DEF   OUT,DEF     OUT DIFF')
      DO 1130 N=0,NCOUNT
      TIME = DELTIM*FLOAT(N) + TSTART
      PUL = 0.
      WT = TWOPI*F*TIME
C
C     SUMMATION OVER SINE SERIES TO GET THE VALUE OF THE PULSE
C     AT A GIVEN TIME.
C
      DO 1090 M=1,NFT
      WNT=WT*FLOAT(2*M-1)
      PUL=PUL+SIN(WNT)*SINCOE(M)
 1090 CONTINUE
C     LOOP OVER THE DIFFERENT LIFTOFF VALUES,WITHOUT AND WITH DEFECTS
      J=LPRT
      IF(LPRT.EQ.0)J=1
      SUMT(J)= 0.
```

```
      SUMTDF(J)= 0.
C     SUM OVER THE FOURIER COEFFICIENTS,NFT
      DO 1100 M=1,NFT
      WNT=WT*FLOAT(2*M-1)
      ARG = WNT + PHASE(J,M)
      SUMT(J)=SUMT(J)+TMAG(J,M)*SINCOE(M)*SIN(ARG)
      ARG = WNT + PHADF(J,M)
      SUMTDF(J)=SUMTDF(J)+TRDFMG(J,M)*SINCOE(M)*SIN(ARG)
 1100 CONTINUE
      SUMDIF(J)=SUMT(J)-SUMTDF(J)
C     WRITE(LOK,1120) PUL,SUMT(J),SUMTDF(J),SUMDIF(J)
      IF(NPRT.EQ.1)WRITE(34,*)TIME,PUL
      IF(NPRT.EQ.2)WRITE(34,*)TIME,SUMT(LPRT)
      IF(NPRT.EQ.3)WRITE(34,*)TIME,SUMTDF(LPRT)
      IF(NPRT.EQ.4)WRITE(34,*)TIME,SUMDIF(LPRT)
 1120 FORMAT (1X, 1P6E12.4)
C1125 IF (MOD(N + 1, 5) .EQ. 0) WRITE(LOK,470)
 1130 CONTINUE
 1150 CONTINUE
C     WRITE END DATA ON FILE 34
      WRITE(34,1310)
 1310 FORMAT(' 0.,0.,0.,0.')
      GO TO 520
C
C
C
   40 FORMAT(1X,I2,10X,3(1PE14.5),0PF6.2)
   60 FORMAT (2(' R',I1,'=', F8.5), 1X, A4, A3, 'LENGTH=', F8.5)
   80 FORMAT (' COIL MEAN RADIUS=', F8.5, ' INCHES',I3, ' FREQUENCIES',
     1 ' FUNDAMENTAL=', 1PE12.5)
   90 FORMAT (' PICKUP RECESSED', F8.5)
  100 FORMAT (' MIN LIFT-OFF=', F8.5, '    LIFT-OFF INCREMENT=', F8.5)
  103 FORMAT (' DEFECT LOCATED AT R=', F9.6, ' INCHES'/
     1  ' AND Z=', F9.6,
     1 ' INCHES BELOW THE SURFACE OF', I2, 'TH LAYER')
  410 FORMAT (1X,A4,A3,'RES    INDUCTANCE     NO TURNS', 6X,
     1 'SHUNT CAP      NOR IM PT')
  420 FORMAT (1X, 1PE12.5, E14.5, 2X, 0PF8.1, 1PE18.5, 2X, 0PF9.6)
  440 FORMAT (' DRIVING VOLT  SERIES RES     AMP GAIN      INPUT IMP')
  450 FORMAT (1X, F5.1, 9X, 1PE12.5, 0PF10.1, 1PE18.5)
  470 FORMAT (1X)
  490 FORMAT ('0MAGNITUDE AND PHASE VS. LIFTOFF'/1X, 4(F6.3,8X), F6.3)
  510 FORMAT (1X, 4(1PE12.5,2X), E12.5)
  530 FORMAT (' 1 COIL DESIGN 2 ATTEN. DESIGN. 3 DRIFT CHECK  4',
     1 ' PULSE CAL  5 QUIT '\)
  540 FORMAT (I1)
  560 FORMAT (2(1X, A4, A3, 'WIRE GAGE, TURNS'),1X,\)
  610 FORMAT (1X,A4,A3, F6.1, ' TURNS OF #', F5.1, ' WIRE', F5.1,
     1 '/LAYER', F5.1, ' LAYERS', 1PE12.5, ' OHMS')
  690 FORMAT ('0FREQUENCY =', 1PE12.3/
     1 ' DRIVER SERIES RES, SHUNT CAP, PICKUP RES,',
     2 ' SHUNT CAP'/1X)
  750 FORMAT (1X, A3, ' CKT', 1PE12.5, ' BELOW RES', E12.5,
     1 ' OPT RES', E12.5, ' RES/REACT.')
  770 FORMAT (' SYSTEM DRIFT VARIATIONS')
  800 FORMAT (' % VARIATION   PARAMETER VAR RADIANS',7X,'DEGREES',7X,
     1 '% OF RANGE')
  820 FORMAT (1X,F4.1,10X,A3, 'RES    ')
  860 FORMAT (1X, F4.1, 10X, A3, ' SHUNT CAP')
      R0 = R0/(1. + DR5)
```

```
  920 FORMAT (1X, F4.1, 10X, 'AMP INPUT RES')
  940 FORMAT (1X, F4.1, 10X, 'APPLIED VOLT.')
  960 FORMAT (1X, F4.1, 10X, 'FREQUENCY    ')
C     THE FIFTH POSSIBILITY ENDS CALCULATIONS
 1300 CONTINUE
      STOP
      END
      SUBROUTINE BESSEL (VAL)
C
C     THIS SUBROUTINE CALCULATES THE INTEGRAL OF XJ1(X)
C     FROM 0 TO Z, ACCORDING TO THE EQUATIONS IN COMPUTER
C     PROGRAMS 1970,PAGE 259.
C
      COMMON X, Z, Q1
      DATA PIO4 /.785398163/
      IF (Z .GT. 5.) GO TO 1510
      L = Z + Z + 3.
      F1 = .5*Q1*Q1*Q1
      VAL = F1/3.
      T = -.25*Z*Z
      D1 = Z.
      D2 = 5.
      E = 4.
      DO 1500 I=1,L
      F1 = F1*T/D1
      VAL = VAL + F1/D2
      D1 = D1 + E
      E = E + 2.
      D2 = D2 + 2.
 1500 CONTINUE
      GO TO 1520
 1510 T = 1./Z
      X0 = -6.4109E-5 + T*(.7034845 + T*(-.1730503 + T*(2.050931 +
     1 T*(-23.79333 + T*(109.1142 - T*188.1357)))))
      X1 = .7979095 + T*(-1.87344E-3 + T*(.4952024 + T*(-.6896196 +
     1 T*(2.105874 - T*5.817517))))
      ARG = Z - PIO4
      VAL = (1. - SQRT(Z)*(X1*COS(ARG) - X0*SIN(ARG)))/(X**3)
 1520 RETURN
      END
      SUBROUTINE XFORM (N, TR, R5, U, T)
      COMPLEX BETA0, BETA1, EX, TR, BSUM, BDIF
      DIMENSION EX(2), U(10), T(10), TR(2,2)
      COMMON /B1/ BETA0, BETA1
      EX(1) = CEXP(-BETA0*U(N-1)*T(N-1)/R5)
      EX(2) = 1./EX(1)
      BSUM = BETA1 + BETA0
      BDIF = BETA1 - BETA0
      DO 1610 I=1,2
      DO 1610 J=1,2
      K = I + J
      IF (K .EQ. 3) GO TO 1600
      TR(I,J) = BSUM*EX(J)
      GO TO 1610
 1600 TR(I,J) = BDIF*EX(J)
 1610 CONTINUE
      RETURN
      END
      SUBROUTINE MATRIX (JUMP, V, TR, BETA1)
      COMPLEX V, Q, TR, BETA1
```

```
      DIMENSION Q(2,2), V(2,2), TR(2,2)
      Q(1,1) = V(1,1)
      V(1,1) = (BETA1 + BETA1)*Q(1,1)
 1720 DO 1730 I=1,2
 1730 Q(1,2) = V(1,2)
      DO 1740 I=1,2
      V(1,2) = (0., 0.)
      DO 1740 J=1,2
 1740 V(1,2) = V(1,2) + TR(I,J)*Q(J,2)
      RETURN
      END
      SUBROUTINE CIRCT (TMAG, PHASE, Q0, T1, T2, Q, M)
C
C     THIS SUBROUTINE CALCULATES THE MAGNITUDE AND PHASE OF THE
C     OUTPUT VOLTAGE WITH OR WITHOUT THE DEFECT AT EACH FREQUENCY.
C     THE DEFECT FACTOR IS Q, WHICH IS SET TO ZERO IN THE CALLING PROGRAM
C     TO GIVE THE PROPERTIES WITHOUT THE DEFECT.
C
      COMPLEX QJ
      COMPLEX MUT(5,80), DRIVER(5,80), PICKUP(5,80), Z1, Z2, Z3, Z4, Q5,
     1 Z5, Z6, Z7, DENOM, TNUM, VOLT, GH(5,80), IJ(5,80), Q1, Q2, Q3, Q4
      REAL L3, L4, N3, N4
      DIMENSION TMAG(5,80), PHASE(5,80)
      COMMON /B2/ R1, R2, R3, R4, L3, L4, R0, R6, R7, R9, C6, C7, V0,
     1 G5, W, F, R5, N3, N4
      COMMON /B3/ MUT, DRIVER, PICKUP, AIR1, AIR2, GH, IJ
      WQ=W
      Z1 = CMPLX (WQ*R0*C6, -1.)
      Z2 = CMPLX (WQ*R9*C7, -1.)
      Z3 = CMPLX (0., -R0)
      Z4 = CMPLX (0., -R9)
      QJ = CMPLX (0., -Q)
      Q2 = Q0*T1
      Q1 = Q2*T2
      Q2 = Q2*T1*(0., 1.)
      Q3 = Q0*T2*T2*(0., 1.)
      Q4=Z1*Z2
      Q5=V0*R9*G5*(0.,-1.)
      DO 1800 J=1,5
      Z5 = Q1*(MUT(J,M) + QJ*GH(J,M)*IJ(J,M))
      Z6 = Q2*(DRIVER(J,M) + AIR1 + Q*GH(J,M)*GH(J,M))
      Z7 = Q3*(PICKUP(J,M) + AIR2 + Q*IJ(J,M)*IJ(J,M))
      DENOM = Q4*Z5*Z5 + (Z1*(Z6 + R6) + Z3)*(Z2*(Z7 + R7) + Z4)
      TNUM = Q5*Z5
      VOLT = TNUM/DENOM
      TMAG(J,M) = CABS(VOLT)
      PHASE(J,M) = ATAN2(AIMAG(VOLT), REAL(VOLT))
 1800 CONTINUE
      RETURN
      END
      SUBROUTINE SENS (Q, RAD)
      DIMENSION Q(5)
      Q2 = 0.
      DO 2000 J=1,5
      Q1 = Q(J)
      IF (ABS(Q2 ) .LT. ABS(Q1)) Q2 = Q1
 2000 CONTINUE
      Q2DEG = Q2*RAD
      WRITE(6,2010) Q2, Q2DEG
 2010 FORMAT (' MAX PHASE SHIFT=', 1PE13.6, ' RADIANS OR', E13.6,
```

```
     &  ' DEGREES')
      RETURN
      END
      SUBROUTINE GAGER(RES)
C
C     CALCULATES THE WIRE GAGE AND RESISTANCE OF THE COILS.
C
      COMMON /B4/ G, D1, D2, RLN, T5, N1A, J1, D, E
      DATA PIO12 / .261799388/
      IF (J1 .EQ. 1) GO TO 2120
      N1A = -1
      D3 = .95*SQRT((D2 - D1)*RLN/T5)
      X2 = 1.0371E-5/(D3*D3)
      Q = 40.
 2100 G = 40. + 4.34294482*(ALOG(X2) - ALOG(.9989 + .017*(.1*Q - 1.)))
      IF (ABS(Q - G) .LT. 1.E-4) GO TO 2110
      Q = G
      GO TO 2100
 2110 IG = G
      G = IG
 2120 TG = .1*G
      X2 = (.9989 + .017*(TG - 1.))*10.**(TG - 4.)
      D3 = SQRT(1.0371E-5/X2)
      IF (G .GT. 40.) GO TO 2130
      X3 = (.460655*ALOG(D3*1.E3) - .43444)*1.E-3
      GO TO 2140
 2130 X3 = .09802228*D3 + 2.56791E-5
 2140 ID = RLN/(D3 + X3)
      D = ID
      IE = (D2 - D1)/(D3 + X3)
      E = IE
      IF (N1A .EQ. -1) GO TO 2150
      T5 = D*E
 2150 RES = T5*X2*(D2 + D1)*PIO12
      RETURN
      END
      FUNCTION BSJ1(X)
C
C     CALCULATES THE BESSEL FUNCTION J1(X) ACCORDING TO
C     HART ET AL., COMPUTER APPROXIMATIONS, PP. 141-149.
C
      IF (X .GT. 8.) GO TO 100
      T = X**2
C
C       INDEX NO. 6039, P. 304
C
      BSJ1 = X*(.316259938E5 + T*(-.373650261E4 + T*(.138125342E3 +
     1 T*(-.236523840E1 + T*(.219996838E-1 + T*(-.117640503E-3 +
     2 T*(.352364574E-6 - T*.474270726E-9)))))))/(.632519877E5 +
     3 T*(.433493017E3 + T))
      GO TO 200
  100 Z = 8./X
      T = Z*Z
C
C       INDEX NO. 6742, P. 324
C
      P1 = (-.13437934E5 - .827754517E3*T)/(-.13437934E5 +
     1 T*(-.803149374E3 + T))
C
C       INDEX NO. 7142, P. 331
```

```
C
      Q1 = Z*(.179121078E3 + .177238253E2*T)/(.382124969E4 +
     1 T*(.394433467E3 + T))
      ARG = X + .785398163
      BSJ1 = (-P1*COS(ARG) + Q1*SIN(ARG))*SQRT(.636619772/X)
  200 RETURN
      END
```

Appendix 2

PROGRAM MAGSCN

MAGSCN is a FORTRAN program to calculate the radial (BR) and axial (BZ) components of the magnetic field at points on a rectangular grid in the vicinity of an air-core solenoid. The solenoid has inner radius R10, outer radius R20, length L30, and TN turns, each carrying current AMP, requiring total ohmic power PWR to be expended in the assumed copper wire. The program also calculates a quality factor QF, defined in this case to be $(BZ)^2/PWR$, the square of the axial magnetic field (proportional to the magnetic energy density) divided by the power required.

A modified version has also been developed, with loops to vary the coil length and inner coil radius (the outer coil radius is fixed by the inner diameter of the tube through which the coil must go). The modified version was used to calculate the magnetic field and quality factor opposite the center of the solenoid, at the radius where the outside of the inspected tube would be. Data from this modified version were used to make the plot in Fig. 5 and can also be used to determine the actual number of ampere-turns needed to saturate the tubing all the way through, opposite the middle of the coil. Once the tube is saturated, of course, its effective permeability for eddy currents becomes like that of air, as assumed in the calculations.

Appendix 3

PROGRAM PULRDG

PULRDG is a FORTRAN program that controls all the peripheral equipment to collect and store the pulsed eddy-current data for a given test setup. It reads data through logical input device (LID) from input files (INDAT and REF.DAT) on the array of probe locations for the experiment, material properties, coil data, etc. It instructs the Modulynx controllers of the probe positioner to move the probe to a desired position, reads the actual position from digital encoders and makes any corrections if needed, triggers the pulser when everything is ready, reads the output pulse magnitude at the selected time intervals from the pulse amplitude instrument, writes the pertinent data on an output file through the logical output device (LOD), then moves the probe to the next position and repeats the process. The instructions to and the return information from the peripheral devices are carried over an IEEE-488 general purpose information bus (GPIB).

The program also takes and stores calibration readings from nominal standards. At each position, both for the sample and a standard, MSET = 3 readings are taken and averaged for each of the pulse times at each of the probe positions. All the collected data are stored on a disk file designated LOD.

```
      PROGRAM MAGSCN
C     13 SEPTEMBER 1985
C     PROGRAM CALCULATES R&Z MAGNETIC FIELD IN THE ALL REGIONS
C     STEPS THROUGH NRRT VALUES OF R AND NZZT VALUES OF Z
C     WRITES THE R COMPONENT OF MAG FIELD,BR,ON 37 AND BZ ON 39
C     ZZ PRINTED OUT IS MEASURED FROM THE CENTER OF THE COIL
      IMPLICIT REAL*6 (A-H,O-Z)
      REAL*6 L3,L30
      DIMENSION RR(3),ZF(4),S1(7),S2(7),ERR(7)
      DATA LOFL2/39/,LOFL/37/,LOU/6/,PI/3.141592653/,SF/1./
      DATA S1/.01,.02,.05,.1,.5,2.,5./
      DATA S2/1.0,2.0,5.0,10.0,50.0,200.0,1000.0/
      DATA ERR/.1,.01,.001,1.E-4,1.E-5,1.E-6,1.E-7/
      DATA RR/0.375,0.415,0.435/,NZT/4/
      DATA ZF/-.12,-.13,-.14,-.15/
      R10=0.170
      R20=0.370
      L30=0.30
C     NORMALIZATION OF DIMENSIONS
      R3=.5*(R10+R20)
      R1=R10/R3
      R2=R20/R3
      L3=L30/R3
      TN=500.0
      AMP=400.0
      AT=AMP*TN
      PWR=6.184E-6*AT*AT*R3/(R20-R10)/L30
      WRITE (LOU,5)
    5 FORMAT('1                    MAGSCN     ')
      WRITE(LOU,10)L30,R10,R20
   10 FORMAT(' COIL LENGTH',F7.4,2X,'IN RAD',F7.4,2X,'OT RAD',F7.4)
      WRITE(LOU,11)AT,PWR
   11 FORMAT(' AMP TURNS',F8.0,' POWER',F14.0,'WATTS')
      WRITE(LOU,520)
      FAC=TN*AMP*2.473695E-5/(L3*(R2-R1)*R3)
      WRITE(LOFL,2) R10,R20,L30
      WRITE(LOFL2,4) R10,R20,L30
    2 FORMAT('1,50,3.0,8.2',/,'RADIAL MAGNETIC FIELD(R1=',F5.2,',R2=',
     *F5.2,',L3=',F5.2,')',/,'LINLIN',/,'AXIAL DISTANCE FROM COIL CENT',
     *'ER (INCHES)',/,'RADIAL MAGNETIC FIELD (TESLAS)',/,'B',/,'3')
    4 FORMAT('1,49,3.0,8.2',/,'AXIAL MAGNETIC FIELD(R1=',F5.2,',R2=',
     *F5.2,',L3=',F5.2,')',/,'LINLIN',/,'AXIAL DISTANCE FROM COIL CENT',
     *'ER (INCHES)',/,'AXIAL MAGNETIC FIELD (TESLAS)',/,'B',/,'3')
      DO 1100 NRR=1,3
      R=RR(NRR)/R3
      WRITE(LOFL,*) NZT
      WRITE(LOFL2,*)NZT
      DO 1050 NZZ=1,NZT
      ZZ=L30*.5+ZF(NZZ)
      Z=ZZ+0.5*L30
      Z=Z/R3
      ZREG=Z*(L3-Z)
      BR=0.0
      BZ=0.0
      B1=0.0
      DO 400 JKL=1,7
   40 BZOLD=BZ
      B2=B1+S2(JKL)
```

```
      X=B1-0.5*S1(JKL)
      I1=0.1+(B2-B1)/S1(JKL)
      DO 90 I=1,I1
      X=X+S1(JKL)
      XR=X*R
      CALL BESSEL(XJR2,X,R2)
      CALL BESSEL(XJR1,X,R1)
      CALL BES1(XJ1R,XR)
      CALL BES0(XJ0R,XR)
      XFAC=X*(XJR2-XJR1)*S1(JKL)*FAC
      XL=X*DABS(Z-L3)
      XZ=X*DABS(Z)
      XPONXL=0.0
      XPONXZ=0.0
      IF(XL.GT.150.0)GO TO 50
      XPONXL=DEXP(-XL)
   50 IF(XZ.GT.150.0)GO TO 60
      XPONXZ=DEXP(-XZ)
C     COIL REGION 0. < Z < L
   60 XPT=XPONXZ-XPONXL
      BR=BR-XFAC*XJ1R*XPT
      IF(ZREG.GE.0.)BZ=BZ+XFAC*XJ0R*(2.0-XPONXL-XPONXZ)
C     REGION BELOW COIL,Z.LT.0.
      IF(Z.LT.0.)BZ=BZ+XFAC*XJ0R*XPT
C     REGION ABOVE THE COIL,Z.GT.L
      IF(Z.GT.L3)BZ=BZ-XFAC*XJ0R*XPT
   90 CONTINUE
      CHECK=(BZ-BZOLD)/DSQRT(BR*BR+BZ*BZ)
      B1=B2
      IF(ABS(CHECK).GT.ERR(JKL)) GO TO 40
      XX=X+S1(JKL)/2
  400 CONTINUE
      QF=BZ*BZ/PWR
  500 FORMAT(' QF=BZ*BZ/POWER=',1PE13.6,' TESLAS/WATT')
  520 FORMAT(' RAD LOC  Z LOC  R MAG FIELD  Z MAG FIELD',
     *11X,'QF',4X,'ALPHA')
  600 WRITE(LOU,700)RR(NRR),ZZ,BR,BZ,QF,XX
  700 FORMAT(1X,2F7.4,3(1PE13.5),F8.0)
      WRITE(LOFL,*)ZZ,BR
      WRITE(LOFL2,*)ZZ,BZ
 1050 CONTINUE
C     WRITE(LOU,1075)
 1075 FORMAT(1X)
 1100 CONTINUE
      WRITE(LOFL,1110)
      WRITE(LOFL2,1110)
 1110 FORMAT('0..0..0..0.')
 1000 STOP JOB
      END
      SUBROUTINE BESSEL(XJ1,X,R)
      IMPLICIT REAL*6 (A-H,O-Z)
      DATA PI04/.785398163/
      Z=X*R
      IF(Z.GT.5.0) GO TO 1090
      LS=2.0*Z+3.0
      F1=0.5*R*R*R
      XJ1=F1/3.0
      DO 1070 N=1,LS
```

```
      RN=N
      F1=-F1*.25*Z*Z/(RN*RN+RN)
 1070 XJ1=XJ1+F1/(2.0*RN+3.0)
      GO TO 1210
 1090 IF(Z.GT.30.0) GO TO 1160
      Q1=(((-108.1357/Z+109.1142)/Z-23.79333)/Z+2.050931)/Z
      Q1=((Q1-.1730503)/Z+.7034045)/Z-.064109E-3
      Q2=(((-5.817517/Z+2.105874)/Z-.6896196)/Z+.4952024)/Z
      Q2=(Q2-.187344E-2)/Z+.7979095
      XJ1=(1.0-SQRT(Z)*(Q2*COS(Z-PI04)-Q1*SIN(Z-PI04)))/(X*X*X)
      GO TO 1210
 1160 P3=1.0/(Z*Z)
      P1=Z*(-1.0+P3*(-.5546875+2.48062114*P3))
      P2=.875+P3*(-.93457031+8.98975114*P3)
      XJ1=1.0+.79788456*(P1*COS(Z-.78539816)
     8 +P2*SIN(Z-.78539816))/SQRT(Z)
      XJ1=XJ1/(X*X*X)
 1210 RETURN
      END
C
      SUBROUTINE BES0(XJ0R,R)
C      PROGRAM TO CALCULATE J0(R)
      IMPLICIT REAL*6 (A-H,O-Z)
      IF(R.GT.3.0)GO TO 50
      Y=R*R/9.0
      XJ0R=(((((.00021*Y-.0039444)*Y+.0444479)*Y-.3163866)
     1*Y+1.2656208)*Y-2.2499997)*Y+1.0
      GO TO 100
   50 Y=3/R
      F0=(((((.00014476*Y-.00072805)*Y+.00137237)*Y-.00009512)
     1*Y-.00552740)*Y-.00000077)*Y+.79788456
      ANG=(((((.00013558*Y-.00029333)*Y-.00054125)*Y+.00262573)
     1*Y-.00003954)*Y-.04166397)*Y-.78539816+R
      XJ0R=F0*COS(ANG)/SQRT(R)
  100 RETURN
      END
C
      SUBROUTINE BES1(XJ1R,R)
C      PROGRAM TO CALCULATE J1(R)
      IMPLICIT REAL*6 (A-H,O-Z)
      IF(R.GT.3.0)GO TO 50
      Y=R*R/9.0
      XJ1R=R*(((((.00001109*Y-.00031761)*Y+.00443319)*Y-.03954289)
     1*Y+.21093573)*Y-.56249985)*Y+.5)
      GO TO 100
   50 Y=3/R
      F1=(((((-.00020033*Y+.00113653)*Y-.00249511)*Y+.00017105)
     1*Y+.01659667)*Y+.00000156)*Y+.79788456
      ANG1=(((((-.00029166*Y+.00079824)*Y+.00074348)*Y-.00637879)
     1*Y+.00005650)*Y+.12499612)*Y-2.35619449+R
      XJ1R=F1*COS(ANG1)/SQRT(R)
  100 RETURN
      END
TOTAL RECORDS WRITTEN =   170/     25
EXIT
$AVR CI 4
$END  LIST
```

```
      PROGRAM PULRDG
C     20 DECEMBER 1985
C
C      THIS PROGRAM READS MAGNITUDE OF THE VOLTAGE AT DIFFERENT TIMES
C     USING THE PULSED EDDY CURRENT INSTRUMENT AND THE MECHANICAL SCANNER.
C     THE PROGRAM WILL READ THE POSITION DATA FROM A FILE,ALONG WITH THE
C     PROPERTIES.THE AMPLITUDE DATA IS STORED AFTER EACH SET OF PROPERTIES.
C     THE PROPERTY AND POSITIONING DATA IS READ FROM DEVICE LID,DISK FILE .
C
C     ICMD=INTEGER COMMAND ,DETERMINES TYPE OF PROGRAM THE MICROCOMPUTER RUNS
C     LI=LOGICAL INPUT TERMINAL(FROM OPERATOR)
C     LID=LOGICAL INPUT DEVICE(ASSIGNED TO DISK FILE,POSITION,PROPERTY DATA)
C     LOD=OUTPUT DEVICE TO STORE PULSE DATA
C     LOT=LOGICAL OUTPUT TERMINAL(TO OPERATOR)
C     LPT=LINE PRINTER(OUTPUT)TERMINAL
C     LTP=TEMPARARY DATA FILE TO HOLD INTERMEDIATE READINGS AS THEY ARE MADE
C     MAG=INDEX TO TURN ON THE SATURATING MAGNET(=0 IF NONE USED)
C     MSET=NO OF SETS OF READINGS THAT WILL BE TAKEN
C     NCH=NUMBER OF CHANNELS
C     NCL=COIL DATA FILE
C     NAXIS=NUMBER OF AXIS THAT ARE POSITIONED
C     NP=PROPERTY INDEX
C     NPROPM=MAX NUMBER OF PROPERTIES THAT MAY VARY
C     NPT=TOTAL NUMBER OF SETS OF PROPERTY VALUES FOR A COMPLETE
C         SET OF READINGS=NO SAMPLES+NO DEFECTS+NO LIFT-OFFS
C     NT=TIME INDEX
C     NTCAL=NUMBER OF TIME CALIBRATIONS
C     NTT=TOTAL NUMBER OF TIME INTERVALS
      CHARACTER INDAT*8,OTDAT*8,FNAME*12,NPROBE*6,NCABLE*6,INSTNO*6,COIL*6
      CHARACTER*4 PRONAM(6),AXISNM(3)
      CHARACTER ADUM*83,FF*1
      INTEGER*2 FUNC(7),TIMOUT(3),IKY,IS,KBF1,KBF2,IRR
C     DIMENSION PROP(NPT,NPROPM),TIM(NTT)
C     DIMENSION SUMMAG(NPT,NTT),SSMAG(NPT,NTT),SDVMAG(NPT,NTT),DIFMAG(NPT,NTT)
C     DIMENSION SUMCAL(NTT,NCAL),SSCAL(NTT,NCAL),SDVCAL(NTT,NCAL)
C     DIMENSION VOLTS(NTT),IN(NTT),RDGC(NTT,NCAL),OLRDGC(NTT,NCAL)
C     DIMENSION PRONAM(NPROPM),POST(NPT,NAXIS)
C
      DIMENSION PROP(30,6),TIM(8)
      DIMENSION SUMMAG(60,8),SSMAG(60,8),SDVMAG(60,8),DIFMAG(60,8)
      DIMENSION SUMCAL(8,4),SSCAL(8,4),SDVCAL(8,4)
      DIMENSION VOLTS(8),VOLSTD(8),RDGC(8,4),OLRDGC(8,4)
      DIMENSION POST(30,3),NOLD(3),XNEW(3),XHOM(3)
C
      DATA LI/0/,LID/9/,LOD/37/,LOT/0/,LPT/6/,LTP/38/,NCH/5/,NCL/28/,MSET/3/
      DATA NRDG/1/,NTCAL/4/,INDAT/'INPUL2'/,OTDAT/'OTPUL'/
      DATA NAXIS/1/,NOLD/3*0/,XHOM/3*0./,NPROPM/4/,MFEC/45/,MBUS/46/,MVL1/47/
      DATA FUNC/30,800,32,10,42,$8002,0/,TIMOUT/30,400,0/,NSEC/30/,NPT/10/
      FF=CHAR(12)
      OPEN(LPT,FILE='#PR')
      OPEN(NCL,FILE='REF.DAT')
C     MULTI FREQUENCY EDDY CURRENT INSTRUMENT ADDR=5 ON GPIB BUSS
C     ADDR=E? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 45
      OPEN(MFEC,FILE='#BUSE?')
      WRITE(0,*)'FILE OPENED FOR PULSED I/O OVER IEEE-488 BUS'
C     SET THE TIMEOUT ON PULSE RDGS FOR 300 MILISECONDS
      CALL SYSFUN(MFEC,TIMOUT(1),NERR)
C     MODULYNX AXIS POSITION CONTROLLER ADDR=6 ON GPIB BUSS,ADDR=F? ON
C     IBM BUSS DRIVER,FORTRAN LOGICAL UNIT = 46
      OPEN(MBUS,FILE='#BUSF?')
```

```
      CALL SYSFUN(MBUS,FUNC(1),IERR)
      WRITE(MBUS,*)'XD YE ZE'
C     WRITE(MBUS,*)'(CAH)'
C     FLUKE VOLT METER ADDRESS = 7 ON GPIB BUSS,LOGICAL UNIT = 47
      OPEN(MVL1,FILE='#BUSG?')
      WRITE(MVL1,*)'* F1 R5 S0 T4'
C
C PRINT TITLE AND DATE
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(LPT,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' PULRDG    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
   30 FORMAT(1X)
C
C     OPEN FILE FOR INSTRUMENT AND PROPERTY DATA - ASSUMED LOCATION ON DEFAULT
C          DISK
      FNAME=INDAT//'.DAT'
      OPEN(LID,FILE=FNAME,STATUS ='OLD')
C
C     OPEN FILE FOR OUTPUT TO STORE DATA TAKEN - LOCATE ON DEFUALT DISK
C
      FNAME=OTDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='NEW')
C
C     OPEN FILE FOR TEMPARARY DATA TO CONTAIN INTERMEDIATE READINGS
      OPEN(LTP,FILE='TEMP.DAT',STATUS ='NEW')
C
C INPUT DESCRIPTION OF EXPERIMENTAL APPARATUS
C
      WRITE(LOT,60)
60    FORMAT(' PROBE #:  ')
      READ(LID,70)NPROBE
70    FORMAT(A6)
      WRITE(LOT,80)
80    FORMAT(' SERIAL #:  ')
      READ(LID,*)NSER,RDVR
      WRITE(LOT,90)
90    FORMAT(' DRIVER SERIES RESISTANCE:  ')
      READ(LID,*)R0
      WRITE(LOT,100)
100   FORMAT(' DRIVER SHUNT CAP:  ')
      READ(LID,*)CAPDR
      WRITE(LOT,110)
110   FORMAT(' PICK-UP SHUNT RESISTANCE:  ')
      READ(LID,*)R9,RSER
      WRITE(LOT,120)
120   FORMAT(' PICK-UP SHUNT CAP:  ')
      READ(LID,*)CAPPU
      WRITE(LOT,130)
130   FORMAT(' CABLE I.D. #:  ')
      READ(LID,70)NCABLE
      WRITE(LOT,140)
140   FORMAT(' LENGTH OF CABLE:  ')
      READ(LID,*)CABLEL
      WRITE(LOT,150)
150   FORMAT(' CAPACITANCE OF CABLE:  ')
      READ(LID,*)CCABLE
      WRITE(LOT,160)
160   FORMAT(' EDDY CURRENT INSTRUMENT #:  ')
      READ(LID,70) INSTNO
C
```

```
C INPUT TIME VALUES
C
      WRITE(LOT,190)
190   FORMAT(' NO. OF TIME VALUES,PROP SETS,PULSE TIME,RDG TIM,PULSE VOLTS:')
      READ(LID,*)NTT,NPT,NTP,NTR,ICMD,PVOLT
      IF(ICMD.LE.3)NTP=NTR
      WRITE(LOT,200)
200   FORMAT(' INPUT EACH VALUE OF TIME SEPARATED BY A SPACE:',/)
      READ(LID,*)(TIM(NT),NT=1,NTT)
C
      READ(LID,210)(PRONAM(NPR),NPR=1,NPROPM),(AXISNM(NAX),NAX=1,NAXIS)
210   FORMAT(10(A4,2X))
C
C READ PROPERTY VALUES AND LOCATION DATA FOR THE POSITIONERS FROM LID
C
      DO 300 NP=1,NPT
      READ(LID,*)(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,NAXIS)
      WRITE(LOT,270)(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,NAXIS)
270   FORMAT(10(F7.3))
300   CONTINUE
C
      WRITE(MVL1,*)'?'
      READ(MVL1,320)PVOLT
  320 FORMAT(E11.4)
      PWR=((PVOLT-10.)/(RDVR+R0))**2*RDVR*.001
      WRITE(LOT,330)PVOLT,PWR
  330 FORMAT('PULSE VOLTAGE',F6.1,'    PULSE POWER',F6.1)
      WRITE(MVL1,*)'* F1  R5  S0  T0'
      CLOSE(MVL1)
C     THIS SECTION TAKES THE CALIBRATION READINGS.
C
700   WRITE(LOT,710)
710   FORMAT(' CALIBRATION READINGS:TYPE ANY KEY TO CONTINUE  ')
705   CALL CALMIC(RDGC,MFEC,NRDG,NTT,IER)
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 705
770   CONTINUE
C
C END OF SECTION WHICH TAKES CALIBRATION READINGS
C
C TAKE READINGS ON NOMINAL STANDARD FOR VOLSTD(I) READINGS
C
C SET ITIM1 TO CORRECT VALUE BEFORE STARTING TAKING READINGS
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      ITIME1=ITIME+NSEC
C     SET NP TO VALUE DESIRED FOR STANDARD READING
      NP=0
773   WRITE(LOT,775)
775   FORMAT(' PLACE PROBE ON NOMINAL STANDARD ,HIT C TO CONTINUE, S TO STOP')
C     USE A DELAY OF NSEC BETWEEN READINGS TO ALLOW COIL TEMP TO STABALIZE
 776  NP=NP+1
      IF(NP.GT.NPT)NP=1
      DO 778 NAX=1,NAXIS
      XNEW(NAX)=POST(NP,NAX)
 778  CONTINUE
C      MOVE PROBE TO A NEW POSITION BETWEEN EACH READING TO MINIMIZE HEATING
      CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
 780  CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 1900
```

```
      IF(IKY.EQ.67.AND.IRR.EQ.0)GO TO 790
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 780
      ITIME1=ITIME+NSEC
      CALL PRDG(VOLSTD,MFEC,NCH,NRDG,ICMD,NTP,NTR,IER)
      WRITE(LOT,950)(VOLSTD(NC),NC=1,NCH)
      GO TO 776
C ZERO ARRAYS THAT WILL CONTAIN SUMS OF THE VOLTAGE MAG READINGS,
C     SUMS OF CALIBRATION READINGS, AND SUMS OF SQUARES OF EACH
C
 790  CONTINUE
      DO 800 NT=1,NTT
      DO 800 NP=1,NPT
      SUMMAG(NP,NT)=0.
      SSMAG(NP,NT)=0.
800   CONTINUE
       DO 810 NT=1,NTT
       DO 810 NC=1,NTCAL
       SUMCAL(NT,NC)=RDGC(NT,NC)
       SSCAL(NT,NC)=RDGC(NT,NC)*RDGC(NT,NC)
 810   CONTINUE
C
C THIS SECTION TAKES THE ACTUAL MAGNITUDE DATA READINGS
C
      DO 1150 MSE=1,MSET
      DO 980 NP=1,NPT
      WRITE(LOT,900)NP,(PROP(NP,NPR),NPR=1,NPROPM),(POST(NP,NAX),NAX=1,NAXIS)
900   FORMAT(I4,10(F8.4))
      NSEC=PROP(NP,NPROPM)
C
C     POSITION SAMPLES BEFORE TAKING READINGS
C
      DO 905 NAX=1,NAXIS
      XNEW(NAX)=POST(NP,NAX)
 905  CONTINUE
      CALL POSIT(XNEW,NOLD,NAXIS,MBUS)
C     USE A DELAY OF NSEC BETWEEN READINGS TO ALLOW COIL TEMP TO STABALIZE
 908  CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 1900
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 908
      ITIME1=ITIME+NSEC
 910  CALL PRDG(VOLTS,MFEC,NCH,NRDG,ICMD,NTP,NTR,IER)
      IF(IER.NE.0)READ(0,913)ADUM
 913  FORMAT(A2)
      IF(IER.NE.0)GO TO 910
      WRITE(LOT,950)(VOLTS(NC),NC=1,NCH)
      WRITE(LTP,1760)NP,(VOLTS(NC),NC=1,NCH),(PROP(NP,NPR),NPR=1,NPROPM)
 950  FORMAT(8F8.3)
C     TEMPERARY TRANSFER AROUND TO USE WITHOUT AUTO POSITIONER
C     IRR=0
C     CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
C     IF(IRR.NE.0)GO TO 910
      DO 980 NT=1,NTT
      SUMMAG(NP,NT)=SUMMAG(NP,NT)+VOLTS(NT)
      SSMAG(NP,NT)=SSMAG(NP,NT)+VOLTS(NT)*VOLTS(NT)
980   CONTINUE
C     MAKE 2 CALIBRATOR READINGS TO ALLOW THE RELAY TIME TO POSITION
```

```
      CALL CALMIC(RDGC,MFEC,NRDG,NTT,IER)
      CALL CALMIC(RDGC,MFEC,NRDG,NTT,IER)
      DO 1130 NT=1,NTT
      DO 1130 NC=1,NTCAL
      SUMCAL(NT,NC)=SUMCAL(NT,NC)+RDGC(NT,NC)
      SSCAL(NT,NC)=SSCAL(NT,NC)+RDGC(NT,NC)*RDGC(NT,NC)
1130  CONTINUE
C     MAKE AN EXTRA READING TO GIVE THE CALIBRATOR RELAY A CHANCE TO SETTLE
C     USE A DELAY OF MSEC BETWEEN READINGS TO ALLOW COIL TEMP TO STABALIZE
1140  CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IKY.EQ.83.AND.IRR.EQ.0)GO TO 1900
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 1140
      ITIME1=ITIME+MSEC
      CALL FRDG(VOLTS,MFEC,NCH,NRDG,ICMD,NTF,NTR,IER)
1150  CONTINUE
C
C     RETURN POSITIONER TO XNEW(II)=0.0 SO THAT POSITIONS CAN BE CHECKED
C     TURN OFF MOTOR CURRENT
C
C     WRITE(MBUS,*)'<CAH>'
C
      CALL POSIT(XHOM,NOLD,NAXIS,MBUS)
      WRITE(MBUS,*)'XE YE ZE'
C
C      BEFORE STOPPING CALCULATE AVERAGES & STANDARD DEVIATIONS
C
      DO 1250 NT=1,NTT
      DO 1250 NC=1,NTCAL
      SUMCAL(NT,NC)=SUMCAL(NT,NC)/(FLOAT(MSET+1))
      SDVCAL(NT,NC)=SQRT(ABS((SSCAL(NT,NC)-SUMCAL(NT,NC)*SUMCAL(NT,NC)
     *                 *FLOAT(MSET+1))/FLOAT(MSET)))
 1250 CONTINUE
C
C     CALCULATE AVERAGES OF THE READINGS
C
      DO 1260 NT=1,NTT
      DO 1260 NP=1,NPT
      SUMMAG(NP,NT)=SUMMAG(NP,NT)/FLOAT(MSET)
      SDVMAG(NP,NT)=SQRT(ABS((SSMAG(NP,NT)-SUMMAG(NP,NT)*SUMMAG(NP,NT)
     *                 *FLOAT(MSET))/FLOAT(MSET-1)))
      IF(NP.GT.1) DIFMAG(NP-1,NT)=SUMMAG(NP,NT)-SUMMAG(NP-1,NT)
1260  CONTINUE
C
C STORE ALL INFORMATION IN DISK FILE
C
      WRITE(LOD,1270)IHR,IMN,ISE,IMO,IDA,IYR
 1270 FORMAT(6(1X,I4))
      WRITE(LOD,1275) NPROBE,NSER,R0,CAPDR,R9,CAPPU,NCABLE,
     *                 CABLEL,CCABLE,INSTNO
 1275 FORMAT(A6,1X,I5,4(E13.4),1X,A6,2(E13.4),1X,A6)
      WRITE(LOD,1277) NTT,NPT,NPROPM,NTCAL ,NTR,ICMD,PVOUT
 1277 FORMAT(6(1X,I5),1F6.0)
      WRITE(LOD,1279)(TIM(NT),NT=1,NTT)
 1279 FORMAT(8(1PE10.3))
      WRITE(LOD,1280)(PRONAM(NPR),NPR=1,NPROPM)
 1280 FORMAT(7(1X,A4))
      DO 1295 NC=1,NTCAL
      WRITE(LOD,1290)(SUMCAL(NT,NC),NT=1,NTT)
```

```
 1290 FORMAT(8(F7.4))
1295  CONTINUE
      WRITE(LOD,1290)(VOLSTD(NT),NT=1,NTT)
      DO 1300 NP=1,NPT
      WRITE(LOD,1297)(SUMMAG(NP,NT),NT=1,NTT),(PROP(NP,NPR),NPR=1,NPROPM)
C1297  FORMAT(8(F7.3),3(F5.3),F5.0)
1297  FORMAT(5(F7.3),3(F5.3),F5.0)
1300  CONTINUE
C
C END OF SECTION WHICH WRITES DIRECT ACCESS FILE
C PRINT SUMMARY OF JOB STATISTICS ON LPT
C
      WRITE(LPT,1350) NPROBE,NSER,RDVR
1350  FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5,' D.C. RES',F8.3 )
      WRITE(LPT,1360) R0,CAPDR
1360  FORMAT(' DRIVER SERIES RESISTANCE:',F10.3,5X,'DRIVER SHUNT CAP.:',
     *         E12.4)
      WRITE(LPT,1370) R9,RSER,CAPPU
1370  FORMAT(' PICK-UP  RES: SHUNT',F9.1,'SERIES',F9.1,'  SHUNT CAP.:',E12.4)
      WRITE(LPT,1380) NCABLE,CABLEL,CCABLE
1380  FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
     *          E12.4)
      WRITE(LPT,1390) INSTNO
1390  FORMAT(' EDDY CURRENT INSTRUMENT NO.:',A6)
      WRITE(LPT,1395)NTP,NTR,ICMD,PVOLT,PWR
1395  FORMAT('PULSE INTERVAL=',I2,' RDG INTERVAL=',I2,' COMMAND=',I2,
     *' PULSE VOLTS=',F5.0,' POWER(KW)',F6.1)
       WRITE(LPT,30)
C
C     PRINT CALIBRATION READINGS
C
1600  CONTINUE
CC    WRITE(LPT,1650) NTCAL
 1650  FORMAT('AVERAGES & STANDARD DEVIATIONS OF CALIBRATION RDGS:',I3,' MAGS')
CC     WRITE(LPT,1660)(TIM(NT),NT=1,NTT)
 1660  FORMAT('TIMES',8(1PE9.2))
       DO 1700 NC=1,NTCAL
CC     WRITE(LPT,1680)((SUMCAL(NT,NC)),NT=1,NTT)
 1680  FORMAT(5X,8(F9.4))
CC     WRITE(LPT,1690)((SDVCAL(NT,NC)),NT=1,NTT)
 1690  FORMAT(' S.D.',8(F9.4))
 1700  CONTINUE
CC     WRITE(LPT,30)
C
C     READINGS FROM NOMINAL SAMPLE
C
      WRITE(LPT,1745)
1745  FORMAT(' READINGS FROM NOMINAL TUBE SAMPLE')
      WRITE(LPT,1680)(VOLSTD(NT),NT=1,NTT)
C     PRINT OUT READINGS AND PROPERTIES
C
      WRITE(LPT,1750)(NT,NT=1,NTT),(PRONAM(NPR),NPR=1,NPROPM)
C1750  FORMAT('PSET',8('  MAG',I1,1X),6(1X,A4))
1750  FORMAT('PSET',5('  MAG',I1,1X),6(1X,A4))
      DO 1800 NP=1,NPT
      WRITE(LPT,1760)NP,(SUMMAG(NP,NT),NT=1,NTT),(PROP(NP,NPR),NPR=1,NPROPM)
      WRITE(LPT,1765)(SDVMAG(NP,NT),NT=1,NTT)
      IF(NP.LT.NPT) WRITE(LPT,1766)(DIFMAG(NP,NT),NT=1,NTT)
      WRITE(LPT,30)
1760  FORMAT(I3,8(F7.3),3(F5.3),F5.0)
```

```
1765  FORMAT(' SD',8(F7.3))
1766  FORMAT('DIF',8(F7.3))
1800  CONTINUE
      WRITE(LPT,1810)FF
1810  FORMAT(1A)
      GO TO 2000
C
C     RETURN POSITIONER TO XNEW(II)=0.0 SO THAT POSITIONS CAN BE CHECKED
C     TURN OFF MOTOR CURRENT
C
C     WRITE(MBUS,*)'(CAH)'
C
1900  CALL POSIT(XHOM,NOLD,NAXIS,MBUS)
      WRITE(MBUS,*)'XE YE ZE'
2000  STOP
      END
C
      SUBROUTINE POSIT(XNEW,NOLD,NAXIS,MBUS)
C
C     VERSION 6 JULY 1984
C     DETERMINES THE NEW LOCATION FOR 3 DIFFERENT POSITIONERS AND FROM
C     THE OLD LOCATION,THE NUMBER OF STEPS NEEDED TO REACH THE NEW
C     LOCATION.THEN SENDS THE PULSES OVER THE IEEE-488 BUSS TO THE CONTROLLER.
C     AFTER THE NEW POSITION IS REACHED THE PULSE COUNT IS COMPARED TO THE
C     ENCODER COUNT AND CORRECTED IF DIFFERENT.THE EDDY CURRENT READINGS WILL
C     NOT BE MADE UNTIL THE POSITION IS CORRECT.
C
      DIMENSION NOLD(3),XNEW(3),NSTEPS(3)
      CHARACTER*2 AXC(3),GC
      DATA AXC/'XM','YM','ZM'/,GC/'G '/,STEPSZ/0.00025/
      DO 20 IAXIS=1,NAXIS
      NSTP=(XNEW(IAXIS)+0.000125)/STEPSZ
      NSTEPS(IAXIS)=NSTP-NOLD(IAXIS)
      NOLD(IAXIS)=NSTP
   20 CONTINUE
      WRITE(MBUS,30)(AXC(IAXIS),NSTEPS(IAXIS),GC,IAXIS=1,NAXIS)
C     WRITE(MBUS,*)';(CFA)'
C     WRITE(MBUS,*)';'
   30 FORMAT(4(A2,I7,A2))
      RETURN
      END
      SUBROUTINE PRDG(VOLTS,MFEC,NCH,NRDG,ICMD,NTP,NTR,IER)
C     PROGRAM TO READ PULSE DATA OVER THE GPIB BUSS
      CHARACTER*1 TM(16),X,Y,CR
      DIMENSION VOLTS(NCH),IN(8)
      DATA TM/'!','"','#','$','%','&','(','*','0','5',':','@','O','W','^',' '/
      X=TM(NTP)
      Y=TM(NTR)
      CR=CHAR(13)
      VF1=5.0/4096.
      VFA=VF1/FLOAT(NRDG)
      DO 210 I=1,NCH
  210 VOLTS(I)=0.
      DO 250 JJ=1,NRDG
      WRITE(MFEC,35,ERR=400)ICMD,X,Y,CR
      READ(MFEC,70,ERR=500)(IN(I),I=1,NCH)
      DO 245 I=1,NCH
      VOLTS(I)=VFA*FLOAT(IN(I))+VOLTS(I)
  245 CONTINUE
  250 CONTINUE
```

```
   35 FORMAT(I1,A1,A1,A1)
C     WRITE(0,50)(VOLTS(I),I=1,NCH)
   50 FORMAT(1X,8F8.4)
   70 FORMAT(8I4)
      GO TO 600
  400 IER=1
      WRITE(0,*)'BUSS WRITE ERROR,RESET-HIT KEY RTN TO CONTINUE'
      GO TO 600
  500 IER=2
      WRITE(0,*)'BUSS READ ERROR,RESET-HIT KEY TO RTN CONTINUE'
  600 RETURN
      END
      SUBROUTINE CALMIC(RDGC,MFEC,NRDG,NTT,IER)
C
C     PROGRAM TO DRIVE AUTOMATIC CALIBRATOR MODULE
C     READINGS ARE TAKEN THROUGH MICROCOMPUTER
C
      DIMENSION RDGC(NTT,4),VOLTS(8)
      ICM=3
      DO 100 NEXT=1,4
      II=4+3*NEXT
      CALL PRDG(VOLTS,MFEC,NTT,NRDG,ICM,II,II,IER)
      DO 50 NT=1,NTT
      RDGC(NT,NEXT)=VOLTS(NT)
   50 CONTINUE
      WRITE(0,60)NEXT,(VOLTS(NT),NT=1,NTT)
   60 FORMAT(I2,8F8.3)
  100 CONTINUE
      WRITE(0,*)' '
      RETURN
      END
```

Appendix 4

PROGRAM PULFIT

PULFIT is a FORTRAN program that takes the pulse magnitude and sample properties data from files written by the PULRDG program and then uses these data to make a least-squares fitting of the polynomials of the pulse magnitudes to represent the sample properties. The process has been described in earlier reports,[1,2] and only a brief summary will be given here.

The operator is prompted to generate a polynomial for any particular property that is of interest. The polynomial has terms with the pulse magnitudes measured from standards and raised to various operator-chosen powers. Each such term in the polynomial is multiplied by a coefficient that is determined (by the subroutine ALSQS) to make the polynomial have the best least-squares fit to the actual sample properties for the standards. Then this polynomial with its optimized coefficients can be used to calculate the sample property from pulse amplitude measurements on an unknown sample. However, before this stage is reached, the operator can try different types of polynomials, with the pulse magnitudes possibly raised to different powers, or different combinations of magnitudes (i.e., measured at different times during the pulse), polynomials with different numbers of terms, etc. For each type of polynomial, the rms deviation of the fit of the calculated to the actual properties of the standards is printed, as well as a quantity called "DRIFT," which measures the standard deviation of the worst-case errors in the fit, that would be produced if typical measurement errors were all such as to make the calculated property deviate in the same direction from what it would be with perfect measurements.

When the operator has chosen a type of polynomial that produces a satisfactory fit to the standard properties with satisfactory insensitivity to measurement errors, the optimized polynomial coefficients can be printed out and stored in a disk file, the complete comparison of actual and calculated properties can be printed, another property can be fitted, and so on. The various choices are offered in line 890.

REFERENCES

1. W. E. Deeds, C. V. Dodd, and G. W. Scott, *Computer-Aided Design of Multifrequency Eddy-Current Tests for Layered Conductors with Multiple Property Variations*, ORNL/TM-6858, October 1979.

2. W. E. Deeds and C. V. Dodd, *Multiple Property Variations in Coaxial Cylindrical Conductors Determined with Multiple-Frequency Eddy Currents*, ORNL/NUREG/TM-335, November 1979.

```
      PROGRAM PULFIT
C     VERSION 23 OCTOBER 1984
C     PROGRAM TO PERFORM A LEAST SQUARES FIT TO DATA READ INTO A DISK
C     FILE BY PULRDG PROGRAM AND THEN PERFORM CONTINOUS BACK CALCULATIONS
C     USING INSTRUMENT READINGS THAT ARE MADE BETWEEN EACH DISPLAYED SET.
C
C     IRDPRM=MAXIMUM NUMBER OF COEFFICIENTS IN EXPANSION
C     LITEK=LOGICAL INPUT UNIT
C     LOTEK=LSI OUTPUT UNIT FOR PROMPTING AND DISPLAY
C     LOU=LOGICAL OUTPUT UNIT FOR PERMANENT RECORD (PRINTER)
C     NCHS=NUMBER OF DATA CHANNELS (TIME INTERVALS)
C     NT=TIME INDEX
C     NTTT=NUMBER OF TIMES
C     NP=PROPERTY INDEX
C     NPROPM=MAXIMUM NUMBER OF PROPERTIES CALCULATED(=6 NOW)
C     NPT=TOTAL NUMBER OF SETS OF PROPERTY  VALUES THAT WILL BE CONSIDERED
CFOR THE FIT.EACH SET OF PROPERTY VALUES IS ON ONE LINE IN THE DATA FILE
C     NPTT=NUMBER OF DATA LINES IN THE INPUT DATA FILE
C     NTCAL=NUMBER OF TIME CALIBRATIONS
C     NPRINT=PRINT AND TRANSFER INDEX.
C     MAG=INDEX TO TURN ON THE SATURATING MAGNET(=0 IF NONE USED)
C
      REAL L2,L4,L3,L5,L6
      CHARACTER*6 NPROBE,NCABLE,INSTNO,COIL
      CHARACTER COEDAT*8,FNAME*12,BLANKS*4,PRONAM(7)*4,AXISNM(3)*4,ADUM*82
      CHARACTER PROPTY(7)*4,STOP*4,INDAT*8,PHASW(3)*4,POLARY(16,5)*4
      CHARACTER*1 CR,TM(16)
      INTEGER*2 FUNC(7),TIMOUT(3)
      INTEGER CMD0,CMD1,CMD2,CMD3
C
C  DIMENSIONS THAT ARE CHANGED:
C **  DIMENSION READNG(NPT+1,IRDPRM+1),PRO(NPT),POLARY(IRDPRM+1,5)
C     DIMENSION PROP(1,NPROPM),TMDFT(NTT),VLDFT(NTT),JPOL(2,NTT,NPROPM)
C     DIMENSION COE(IRDPRM),COEF(IRDPRM,NPROPM),TIM(NTT)
C     DIMENSION RDG1(1,IRDPRM),NPOL(2,NTT)
C     DIMENSION JOFSET(NPROPM),JRDPR(NPROPM)
C     DIMENSION PROPTY(NPROPM),PRONAM(NPROPM)
C     DIMENSION TMAG1(1,NTT)
C     DIMENSION SUMCAL(NTT,NCAL),SDVCAL(NTT,NCAL)
C     DIMENSION RDGC(NTT,NCAL),RDG0(NTT,NCAL),COFSET(NTT),CGAIN(NTT)
C     DIMENSION VOLTS(NTT),VOLSTD(NTT),STDV(NTT),NCONV(NTT)
C     DIMENSION TOFSET(NTT),TSLOPE(NTT)
C
C  THE APPROPRIATE NUMBERS SHOULD BE INSERTED IN THE FOLLOWING
C  DIMENSION STATEMENTS, COMMENTED STATEMENTS MARKED ** MANDATORY
C
      DIMENSION READNG(16,16),PRO(36),PROP(1,7)
      DIMENSION TMDFT(8),VLDFT(8),JPOL(2,8,7),COE(15),COEF(15,7),TIM(8)
      DIMENSION RDG1(1,15),NPOL(2,8),JOFSET(7),JRDPR(7)
      DIMENSION TMAG1(1,8)
      DIMENSION RDGC(8,4),RDG0(8,4),COFSET(8),CGAIN(8)
      DIMENSION VOLTS(8),VOLSTD(8),STDV(8),NCONV(8)
C     DIMENSION TOFSET(8),TSLOPE(8)
C
C
C  DATA THAT MAY NEED TO BE CHANGED:
      DATA NPT/15/,NPRINT/0/,LOU/6/,LITEK/0/,LOTEK/0/
      DATA NCL/28/,LID/37/,LOD/38/,MFEC/45/,MBUS/46/,MAG/0/
      DATA IRDPRM/15/,IR/1/,NCH/8/,NRDG/32/,NPROPM/7/,NPROPT/1/
      DATA STOP/'STOP'/,BLANKS/'    '/,NUMIT/1/
```

```
      DATA TM/'!','"','#','$','%','&','(','*','0','5',':','@','O','W','e','~'/
      DATA CMD0/0/,CMD1/1/,CMD2/2/,CMD3/3/
      DATA NLINES/10/,INDAT/'OTP220A'/,COEDAT/'PULCOE  '/
      DATA COFSET/8*0.0/,CGAIN/8*1./
      DATA FUNC/30,400,32,10,42,$8002,0/,TIMOUT/30,6,0/
C
C PRINT TITLE AND DATE
      OPEN(LOU,FILE='*PR',ERR=990)
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
    2 FORMAT('PULFIT   TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2,4X,A8)
      WRITE(LOU,2,ERR=990)IHR,IMN,ISE,IMO,IDA,IYR,INDAT
C
   30 FORMAT(1X)
C
C     OPEN  FILE FOR INPUT DATA FROM PULROG,ASSUMED ONDEFAULT DISK
   40 FORMAT(A8)
      FNAME=INDAT//'.DAT'
   41 FORMAT(A12)
      OPEN(LID,FILE=FNAME,STATUS ='OLD',ERR=991)
C     OPEN FILE FOR OUTPUT COEFICIENT DATA STORAGE - ASSUMED  ON DEFAULT DISK
      FNAME=COEDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS ='NEW',ERR = 992)
C
C     MULTI FREQUENCY EDDY CURRENT INSTRUMENT ADDR=5 ON GPIB BUSS
C     ADDR=E? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 45
      OPEN(MFEC,FILE='#BUSE?',ERR=993)
C
C READ INITIAL INFORMATION IN DIRECT ACCESS FILE LID ON DISK
C
      READ(LID,55)IHR,IMN,ISE,IMO,IDA,IYR
      WRITE(LOTEK,55)IHR,IMN,ISE,IMO,IDA,IYR
   55 FORMAT(6(1X,I4))
      READ(LID,60) NPROBE,NSER,R0,CAPDR,R9,CAPPU,NCABLE,
     *     CABLEL,CCABLE,INSTNO
      WRITE(LOTEK,60) NPROBE,NSER,R0,CAPDR,R9,CAPPU,NCABLE,
     *     CABLEL,CCABLE,INSTNO
   60 FORMAT(A6,1X,I5,4(E13.4),1X,A6,2(E13.4),1X,A6)
      READ(LID,65) NTT,NPTT,NPROPM,NTCAL
      IF(NTT.NE.NCH)WRITE(0,*)'ERROR IN CHANNEL(NCH) AND TIME INTERVALS(NTT)'
      WRITE(LOTEK,65) NTT,NPT,NPROPM,NTCAL
   65 FORMAT(4(1X,I5))
      READ(LID,70)(TIM(NT),NT=1,NTT)
      WRITE(LOTEK,70)(TIM(NT),NT=1,NTT)
   70 FORMAT(8(1PE10.3))
      READ(LID,80)(PRONAM(NPR),NPR=1,NPROPM)
      WRITE(LOTEK,80)(PRONAM(NPR),NPR=1,NPROPM)
   80 FORMAT(7(1X,A4))
      DO 100 NC=1,NTCAL
      READ(LID,85)(RDG0(NT,NC),NT=1,NTT)
   85 FORMAT(8(F7.4))
  100 CONTINUE
C     READ NOMINAL  STANDARD VLOTAGES
C
      READ(LID,85)(VOLSTD(NT),NT=1,NTT)
      WRITE(LOTEK,85)(VOLSTD(NT),NT=1,NTT)
C
C END OF SECTION WHICH READS INITIAL DIRECT ACCESS FILE
C
C      THE DATE AND TIME THE DATA WAS TAKEN ARE PRINTED
C
```

```
      WRITE(LOU,140)IHR,IMN,ISE,IMO,IDA,IYR
  140 FORMAT(' CALIBRATION DATA TAKEN ',I2,':',I2,':',I2,'  DATE ',I2,'/'
     *,I2,'/',I2)
      WRITE(LOU,*)'DATA FILE ',INDAT,' COIL ',NPROBE,'  SERIAL NO ',NSER
  160 FORMAT(1X)
C
C     COIL DATA IS READ FROM  REF.DAT OR CIR.DAT FILE
C
C     FILE TO INSERT INTO REFLECTION OR CIRCULAR PROGRAMS FOR COIL
C     DATABASE REFERENCES
C
CTHE INPUT DATA FOR THE REFLECTION COIL IS READ FROM FILE 28
CSEE FIG.2,P.7 AND FIG.4, P.7, ORNL-TM-4107, FOR DEFINITIONS.
C
      OPEN(28,FILE='REF.DAT',STATUS ='OLD',ERR=994)
   10 READ(28,11)COIL,RBAR,R1,R2,L3,R3,R4,L4,L5,L6
     *,RDCDR,RDCPU,TNDR,TNPU
   11 FORMAT(A6,9F8.4,F10.4,F11.4,2F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)'REF COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 15
      IF(COIL.NE.NPROBE)GO TO 10
      GO TO 630
C
C     THE INPUT DATA FOR CIRCULAR COILS IS READ FROM FILE 29
C     SEE TABLE 3,PAGE 10 OF ORNL/NUREG/TM-335 FOR DEFINITIONS
C
   15 OPEN(29,FILE='CIR.DAT',STATUS ='OLD',ERR=995)
      READ(29,20)COIL,RBAR,R1,R2,XL,ZL,RLIM,RDCDR
     *,RDCPU,TNDR,TNPU,ZLDR,ZLPU,X
   20 FORMAT(A6,6F8.4,2F10.4,2F8.1,2E11.3,F8.1)
      IF(COIL.EQ.'END   ')WRITE(0,*)' CIR COIL NOT FOUND'
      IF(COIL.EQ.'END   ')GO TO 300
      IF(COIL.NE.NPROBE)GO TO 15
      GO TO 630
C
C     LEAST SQUARES DESIGN SECTION.
C
C     SELECT PROPERTY TO BE FITTED AND SET UP PROPERTY ARRAY.
C
  300 MSET=NPT
      MSET1=MSET+1
      IF(NPROPT.GT.NPROPM) GO TO 860
  310 WRITE(LOTEK,320)(NPR,PRONAM(NPR),NPR=1,NPROPM)
  320 FORMAT(' SELECT NUMBER OF THE PROPERTY TO BE FITTED:',
     */,7(I3,1X,A4),' ? ')
      READ(LITEK,*)NPROP
      IF(NPROP.GT.NPROPM)NPROP=NPROPM
  350 WRITE(LOTEK,360)
  360 FORMAT(' TYPE 1 IF THERE IS OFFSET; 0 IF NO OFFSET:',/)
      READ(LITEK,*) JOFSET(NPROPT)
      IOFSET=JOFSET(NPROPT)
      IRDPR=IOFSET
  370 WRITE(LOTEK,380)
  380 FORMAT(' SELECT THE NUMBER OF THE FUNCTION TYPE & POLYNOMIAL',
     *' DEGREE, FOR EACH TIME VALUE')
      WRITE(LOTEK,390)
  390 FORMAT(' FUNCTION TYPE:1=LINEAR 2=LOG 3=EXP 4=INV ')
  400 WRITE(LOTEK,160)
      WRITE(LOTEK,410)
  410 FORMAT(25X,'FCTN POL '/,25X'TYPE DEG ')
```

```
      DO 450 NT=1,NTT
      WRITE(LOTEK,420) TIM(NT)
  420 FORMAT(' TIM =  ',1PE12.6,' SEC  ',\)
      READ(LITEK,*)JPOL(1,NT,NPROPT),JPOL(2,NT,NPROPT)
      IRDPR=IRDPR+JPOL(2,NT,NPROPT)
      JRDPR(NPROPT)=IRDPR
      NPOL(1,NT)=JPOL(1,NT,NPROPT)
      NPOL(2,NT)=JPOL(2,NT,NPROPT)
  450 CONTINUE
      IRDPR1=IRDPR+1
      IF(IRDPRM.LT.JRDPR(NPROPT))WRITE(LOTEK,460)
      IF (IRDPRM.LT.JRDPR(NPROPT))GO TO 630
  460 FORMAT(' ERROR: # OF TERMS IN POLARY EXCEEDS DIMENSION')
      JROW=IRDPRM+1
      CALL POLPUL(POLARY,JROW,IRDPR,NPOL,NTT,IOFSET,LOTEK)
C
C      EXPAND THE RAW READINGS INTO IRDPR READINGS.
C
  470 DO 480 NT=1,NTT
C     SET THE TIME AND VOLTAGE DRIFTS TO ZERO INITIALLY
      TMDFT(NT)=0.
  480 CONTINUE
      REWIND(LID)
      DO 481 IREC=1,NLINES
      READ(LID,483)ADUM
  483 FORMAT(A2)
  481 CONTINUE
      NR=1
      DO 490 NP=1,NPTT
      READ(LID,482)(TMAG1(1,NT),NT=1,NTT),(PROP(1,NPR),NPR=1,NPROPM)
 482  FORMAT(8(F7.3),3(F5.3),F5.0)
C482  FORMAT(14(1X,F9.5))
C
C     THE PROPERTIES CAN BE SET AND MODIFIED IN THIS SECTION
C
C
C     IF THIS PROPERTY IS NOT TO BE USED TRANSFER TO 490
C
      IF(PROP(1,3).GT.0.001)GO TO 490
      IF(PROP(1,2).GT.0.001)GO TO 490
      PRO(NR)=PROP(1,NPROP)
      CALL RDGPUL(RDG1,TMAG1,NPOL,IOFSET,TMDFT,1,IRDPR1,1,NTT,1,1,1,1)
      DO 485 IRD=1,IRDPRM
      READNG(NR,IRD)=RDG1(1,IRD)
 485  CONTINUE
      NR=NR+1
  490 CONTINUE
C
C     DO THE LEAST SQUARES FIT OF THE READINGS TO THE PROPERTIES.
C
      CALL ALSQS(READNG,PRO,COE,RSOS,MSET,IRDPR,MSET1)
C
C     CALCULATE THE DIFFERENCES IN THE FIT AND THE MAXIMUM DRIFTS.
C
  500 SSDRIF=0.
      SSDIFF=0.
      IF(NPRINT.EQ.2)WRITE(LOTEK,510)PRONAM(NPROP)
      IF(NPRINT.EQ.2)WRITE(LOU,510)PRONAM(NPROP)
  510 FORMAT(' PSET',8X,A4,9X,'CAL',8X,'DIFF',7X,'DRIFT')
      REWIND(LID)
```

```
      DO 515 IREC=1,NLINES
      READ(LID,483)ADUM
  515 CONTINUE
      NR=1
      DO 570 NP=1,NPTT
      DRIFT=0.
      READ(LID,482)(TMAG1(1,NT),NT=1,NTT),(PROP(1,NPR),NPR=1,NPROPM)
C
C     THE PROPERTIES CAN BE SET AND MODIFIED IN THIS SECTION
C
C
C      TRANSFER TO 570 IF PROPERTY IS NOT THE ONE WE WANT
C
      IF(PROP(1,3).GT.0.001)GO TO 570
      IF(PROP(1,2).GT.0.001)GO TO 570
      DO 540 NT=1,NTT
C
C     ONE MAGNITUDE OR PHASE DRIFT IS SET ON AT A TIME.
C
      TMDFT(NT)=.001
      CALL RDGFUL(RDG1,TMAG1,NPOL,IOFSET,TMDFT,1,IRDPR1,1,NTT,1,1,1,1)
C
C     THE POLYNOMIAL IS CALCULATED
C
      SUM=0.
      DO 520 IR=1,IRDPR
      SUM=SUM+COE(IR)*RDG1(1,IR)
  520 CONTINUE
      DRIFT=DRIFT+ABS(READNG(NR,IRDPR1)-SUM)
      TMDFT(NT)=0.
  530 CONTINUE
  540 CONTINUE
      DIFF=PRO(NR)-READNG(NR,IRDPR1)
      SSDIFF=SSDIFF+DIFF*DIFF
      SSDRIF=SSDRIF+DRIFT*DRIFT
      IF(NPRINT.NE.2)GO TO 565
C
C      THE ENTIRE FIT IS PRINTED OUT
C
      WRITE(LOU,560)NP,PRO(NR),READNG(NR,IRDPR1),DIFF,DRIFT
      WRITE(LOTEK,560)NP,PRO(NR),READNG(NR,IRDPR1),DIFF,DRIFT
  560 FORMAT(1X,I4,4F12.5)
  565 NR=NR+1
  570 CONTINUE
      SDRIF=SQRT(SSDRIF/FLOAT(MSET))
      SDIFF=SQRT(SSDIFF/FLOAT(MSET))
      WRITE(LOU,580)PRONAM(NPROP),SDIFF,SDRIF,(NPOL(1,NT),NT=1,NTT)
      WRITE(LOU,585)IOFSET,(NPOL(2,NT),NT=1,NTT)
      WRITE(LOU,30)
      WRITE(LOTEK,580)PRONAM(NPROP),SDIFF,SDRIF
  580 FORMAT(' RMS DIF IN ',A4,'=',F10.5,2X,'DRIFT=',F10.5,3X,'FCTN'10I2)
  585 FORMAT(I2,' CONSTANT',37X,'POL ',10I2)
  590 IF (NPRINT.NE.3)GO TO 630
      IF(NPROPT.NE.1)GO TO 597
      DO 595 NC=1,NTCAL
      WRITE(LOD,*)(RDG0(JJ,NC),JJ=1,NTT)
  595 CONTINUE
      WRITE(LOD,*)(VOLSTD(NT),NT=1,NTT)
  596 CONTINUE
  597 WRITE(LOU,160)
```

```
      WRITE(LOTEK,160)
      NCOED1=NCOED+4*IRDPRM+1
      WRITE(LOD,*)IRDPR
      DO 610 I=1,IRDPR
      WRITE(LOU,600)I,COE(I),(POLARY(I,J),J=1,5)
      WRITE(LOTEK,600)I,COE(I),(POLARY(I,J),J=1,5)
  600 FORMAT(' COEF(',I2,')=',1PE15.7,4X,5A4)
      COEF(I,NPROPT)=COE(I)
      DO 610 NCO=1,4
C     WRITE(LOD,*)NCONV(NCO)
  610 CONTINUE
      NCOED=NCOED1
C
C     THE COEFFICIENT,OFFSET,NPOL AND IRDPR ARE WRITTEN ON THE DISC
C
C
      WRITE(LOD,615)PRONAM(NPROP),IRDPR,JOFSET(NPROPT)
  615 FORMAT(1X,A4,2I4)
C     WRITE(LOD,*)(COEF(IR,NPROPT),IR=1,IRDPR)
      WRITE(LOD,616)(COEF(IR,NPROPT),IR=1,IRDPR)
  616 FORMAT('       DATA C  /',1PE13.6,14(',/,C  /',1PE13.6))
      DO 620 NT=1,NTT
      WRITE(LOD,*)(NPOL(I,NT),I=1,2)
  620 CONTINUE
      PROPTY(NPROPT)=PRONAM(NPROP)
      NPROPT=NPROPT+1
C
  630 WRITE(LOTEK,640)
  640 FORMAT(' 1 FIT PROP 2 PRT ENTIRE FIT 3 PRT/SV COEF 4 CHG FCTN/POL'
     *,'TYP 5 RUN TEST',/)
      READ(LITEK,*)NPRINT
      GO TO(300,500,590,350,650),NPRINT
  650 NPROPT=NPROPT-1
      WRITE(LOU,160)
      GO TO 880
C     CALCULATES PROPERTIES FROM MAGNITUDES AND PHASES AND CONTINOUSLY
C     DISPLAYS THE VALUES ON THE CRT TERMINAL.
C
  700 CONTINUE
      IF(NPRINT.EQ.2)WRITE(LOTEK,710)(PROPTY(NPRO),NPRO=1,NPROPT)
  710 FORMAT(1X,6(8X,A4,1X))
      IF(NPRINT.EQ.3)WRITE(LOTEK,715)(II,II=1,NTT)
  715 FORMAT(8('  MAG(',I1,')'))
      WRITE(LOTEK,160)
C
C     EXPANSION OF TMAG1(1,NT)INTO READNG(1,IRDPRM)
C
  720  DO 800 NPRO=1,NPROPT
      DO 790 NT=1,NTT
      DO 780 I=1,2
      NPOL(I,NT)=JPOL(I,NT,NPRO)
  780 CONTINUE
C     WRITE(0,721)(NPOL(I,NT),I=1,6)
C 721 FORMAT(6I3)
      TMDFT(NT)=0.
  790 CONTINUE
      IOFSET=JOFSET(NPRO)
      IRDPR=JRDPR(NPRO)
      IRDPR1=IRDPR+1
      CALL RDGPUL(RDG1,TMAG1,NPOL,IOFSET,TMDFT,1,IRDPR1,1,NTT,1,1,1,1)
```

```
      PRO(NPRO)=0.
      DO 800 IR=1,IRDPR
      PRO(NPRO)=PRO(NPRO)+COEF(IR,NPRO)*RDG1(1,IR)
  800 CONTINUE
      IF(NPRINT.EQ.2)WRITE(LOTEK,805)(PRO(NPRO),NPRO=1,NPROPT)
  805 FORMAT(1X,7(F13.5))
C
C     NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
C
  825 CALL PRDG(VOLTS,MFEC,NCH,NRDG,CMD1,TM(16),IER)
C     IF(MAG.EQ.1) CALL SATMAG(0)
C     VOLTS(8)=25.*VOLTS(7)/VOLTS(8)
C     DO 830 NC=1,NCH
C     VOLTS(NC)=VOLTS(NC)+TOFSET(NC)+TSLOPE(NC)*VOLTS(8)
  830 CONTINUE
      IF(NPRINT.EQ.3)WRITE(LOTEK,820)(VOLTS(NT),NT=1,NTT)
      CALL CORRDG(VOLTS,COFSET,CGAIN,NTT)
  840 DO 845 NT=1,NTT
      TMAG1(1,NT)=VOLTS(NT)
  845 CONTINUE
  820 FORMAT(8(F8.4))
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 720
C
C     PROGRAM WILL STAY IN THIS LOOP UNTIL ANY KEY IS STRUCK.
C
      IF(NLABL+NPRINT.EQ.3)WRITE(LOU,710)(NPRO,NPRO=1,NPROPT)
      NLABL=NPRINT
      IF(NPRINT.EQ.2)WRITE(LOU,850)(PRO(NPRO),NPRO=1,NPROPT)
  850 FORMAT(8(F9.5))
      IF(NPRINT.EQ.3)WRITE(LOU,850)(VOLTS(NT),NT=1,NTT)
      GO TO 880
  860 WRITE(LOTEK,870)
  870 FORMAT(' PROP ARRAY IS FILLED.')
  880 WRITE(LOTEK,890)
  890 FORMAT(' PRINT :1.MEAS VOLT & PROPERTIES 2.CAL&DISPLAY PROPS'
     *,'3.RAW RDGS.4.RE-CALIB 5.STOP',/)
      READ(LITEK,*)NPRINT
      GO TO (1300,700,700,895,900),NPRINT
  895 CALL CALMIC(RDGC,MFEC,NRDG,NTT,CMD3,TM,IER)
      CALL RESET(RDGC,RDG0,COFSET,CGAIN,STDV,NTT)
      GO TO 880
C
C     PRINT SUMMARY OF DATA FROM DISK FILE
C
1300  WRITE(LOU,1350) NPROBE,NSER
1350  FORMAT(' PROBE NO.:',A6,5X,' SERIAL NO.:',I5)
      WRITE(LOU,1360) R0,CAPDR
1360  FORMAT(' DRIVER SERIES RESISTANCE:',F10.1,5X,'DRIVER SHUNT CAP.:',
     *       E12.4)
      WRITE(LOU,1370) R9,CAPPU
1370  FORMAT(' PICK-UP SHUNT RESISTANCE:',F10.1,5X,'PICK-UP SHUNT CAP.:'
     *        ,E12.4)
      WRITE(LOU,1380) NCABLE,CABLEL,CCABLE
1380  FORMAT(' CABLE I.D. NO.:',A6,5X,'LENGTH:',F10.1,5X,'CAP.:',
     *       E12.4)
      WRITE(LOU,1390) INSTNO
1390  FORMAT(' EDDY CURRENT INSTRUMENT NO.:',A6)
      WRITE(LOU,1410) (TIM(NT),NT=1,NTT)
1410  FORMAT('TIME:',8(1PE10.3))
```

```
C
C       PRINT CALIBRATION READINGS
C
1600    CONTINUE
        WRITE(LOU,1650) NTCAL
1650    FORMAT(' AVERAGES & STD.DEVIATIONS OF CALIBRATION READINGS:',I3,' MAGS')
        WRITE(LOU,160)
        WRITE(LOU,1660) (TIM(NT),NT=1,NTT)
1660    FORMAT(8(1PE10.3))
        DO 1700 NC=1,NTCAL
        WRITE(LOU,1680)((RDG0(NT,NC)),NT=1,NTT)
1680    FORMAT(8(F10.4))
1700    CONTINUE
C
C       PRINT OUT READINGS AND PROPERTIES
C
        WRITE(LOU,160)
        WRITE(LOU,1750)(NT,NT=1,NTT),(PRONAM(NPR),NPR=1,NPROPM)
1750    FORMAT(1X,'PSET',8('      MAG',I1),5(4X,A4,1X))
        REWIND(LID)
        DO 1755 IREC=1,NLINES
        READ(LID,483)ADUM
 1755   CONTINUE
        DO 1800 NP=1,NPTT
        READ(LID,482)(TMAG1(1,NT),NT=1,NTT),(PROP(1,NPR),NPR=1,NPROPM)
        WRITE(LOU,1760)NP,(TMAG1(1,NT),NT=1,NTT),(PROP(1,NPR),NPR=1,NPROPM)
1760    FORMAT(1X,I4,13(F9.4))
1800    CONTINUE
C
        GO TO 880
  990   WRITE(LOTEK,*)' ERROR IN OPENING PRINTER FILE-CHECK FOR OFF-LINE'
        GO TO 900
  991   WRITE(LOTEK,*)'ERROR IN OPENING INPUT DATA FILE'
        GO TO 900
  992   WRITE(LOTEK,*)'ERROR IN OPENING OUTPUT COEF DATA FILE'
        GO TO 900
  993   WRITE(LOTEK,*)'ERROR IN OPENING INSTRUMENT DATA FILE'
        GO TO 900
  994   WRITE(LOTEK,*)'ERROR IN OPENING REF DATA FILE'
        GO TO 900
  995   WRITE(LOTEK,*)'ERROR IN OPENING CIR DATA FILE'
  900   STOP
        END
        SUBROUTINE ALSQS(A,Y,B,R2,NN,MM,NA)
C
C       ALSQS IS A FORTRAN IV SUBROUTINE TO SOLVE THE LINEAR LEAST
C       SQUARES PROBLEM  NORM(AB - Y) = MIN.   CALLING SEQUENCE IS
C         CALL ALSQS(A,Y,B,R2,N,M,NA)
C       WHERE
C A     IS AN ARRAY CONTAINING THE LEAST SQUARES MATRIX.
C       UPON RETURN THE (M+1)-TH COLUMN CONTAINS THE
C       APPROXIMATING VECTOR AB.
C Y     IS THE VECTOR TO BE FIT
C B     CONTAINS UPON RETURN THE COEFFICIENTS OF THE FIT.
C R2    CONTAINS UPON RETURN THE RESIDUAL SUM OF SQUARES.
C N     IS THE NUMBER OF ROWS IN THE LEAST SQUARES MATRIX.
C M     IS THE NUMBER OF COLUMNS IN THE LEAST SQUARES
C       MATRIX.
C NA    IS THE FIRST DIMENSION OF THE ARRAY A.
C
```

```
      DIMENSION A(NA,1),Y(1),B(1)
      N = NN
      N1 = N+1
      M = MM
      M1 = M+1
      MM1 = M-1
C
C     REDUCE THE LEAST SQUARES MATRIX TO UPPER TRIANGULAR FORM
C
      DO 60 L=1,M
 SS = 0.
 DO 10 I=L,N
   10     SS = SS + A(I,L)**2
 S2 = SS
 S =SQRT(S2)
 IF (A(L,L).LT.0.) S=-S
 D = S2 + S*A(L,L)
 A(L,L) = A(L,L) + S
 IF (L.EQ.M) GO TO 50
 L1 = L+1
 DO 30 J=L1,M
    PP = 0.
    DO 20 I=L,N
   20        PP = PP + A(I,L)*A(I,J)
   30     A(N1,J) = PP/D
 DO 40 J=L1,M
    DO 40 I=L,N
   40        A(I,J) = A(I,J) - A(I,L)*A(N1,J)
   50 A(N1,L) = -S
   60 CONTINUE
C
C     REDUCE THE VECTOR Y
C
      DO 80 I=1,N
   80 A(I,M1) = Y(I)
      DO 100 L=1,M
 PP = 0.
 DO 90 I=L,N
   90     PP = PP + A(I,L)*A(I,M1)
 D = PP/(-A(L,L)*A(N1,L))
 DO 100 I=L,N
  100     A(I,M1) = A(I,M1) - D*A(I,L)
C
C     CALCULATE THE COEFFICIENT VECTOR B
C
      B(M) = A(M,M1)/A(N1,M)
      IF (M.EQ.1) GO TO 130
      DO 120 LL=1,MM1
 L = M-LL
 L1 = L+1
 PP = A(L,M1)
 DO 110 I=L1,M
  110     PP = PP - A(L,I)*B(I)
  120 B(L) = PP/A(N1,L)
C
C     CALCULATE R2
C
  130 SS = 0.
      MP1 = M+1
      DO 140 I=MP1,N
```

```
 SS = SS + A(I,M1)**2
  140 A(I,M1) = 0.
      R2 = SS
C
C     PERFORM THE BACK CALCULATIONS
C
      DO 170 LL=1,M
 L = M-LL+1
 PP = 0.
 DO 150 I=L,N
  150    PP = PP + A(I,L)*A(I,M1)
 D = PP/(-A(L,L)*A(M1,L))
 DO 160 I=L,N
  160    A(I,M1) = A(I,M1) - D*A(I,L)
  170 CONTINUE
      RETURN
      END
C
      SUBROUTINE PRDG(VOLTS,MFEC,NCH,NRDG,CMD,X,IER)
C     PROGRAM TO READ PULSE DATA OVER THE GPIB BUSS
      CHARACTER*1 X
      INTEGER CMD
      DIMENSION VOLTS(NCH),IN(8)
      VFA=5.0/(4096.*FLOAT(NRDG))
      IER=0
      DO 210 I=1,NCH
  210 VOLTS(I)=0.
      DO 250 JJ=1,NRDG
      WRITE(MFEC,35,ERR=400)CMD,X,CR
      READ(MFEC,70,ERR=500)(IN(I),I=1,NCH)
      DO 245 I=1,NCH
      VOLTS(I)=VFA*FLOAT(IN(I))+VOLTS(I)
  245 CONTINUE
  250 CONTINUE
   35 FORMAT(I1,A1,A1)
C     WRITE(0,50)(VOLTS(I),I=1,NCH)
   50 FORMAT(1X,8F8.4)
   70 FORMAT(8I4)
      GO TO 600
  400 IER=1
      WRITE(0,*)'BUSS WRITE ERROR,HIT RESET AND CONTINUE'
      GO TO 600
  500 IER=2
      WRITE(0,*)'BUSS READ ERROR,HIT RESET AND CONTINUE'
  600 RETURN
      END
      SUBROUTINE CALMIC(RDGC,MFEC,NRDG,NTT,ICM,TM,IER)
C
C     PROGRAM TO DRIVE AUTOMATIC CALIBRATOR MODULE
C     READINGS ARE TAKEN THROUGH MICROCOMPUTER
C
      CHARACTER*1 CR,X,TM(16)
      DIMENSION RDGC(NTT,4),VOLTS(8)
      DO 100 NEXT=1,4
      II=4+3*NEXT
      X=TM(II)
      CALL PRDG(VOLTS,MFEC,NTT,NRDG,ICM,X,IER)
      IF(IER.NE.0)GO TO 100
      DO 50 NT=1,NTT
      RDGC(NT,NEXT)=VOLTS(NT)
```

```
   50   CONTINUE
        WRITE(0,60)NEXT,(VOLTS(NT),NT=1,NTT)
   60   FORMAT(I2,8F9.4)
  100   CONTINUE
        WRITE(0,*)' '
        RETURN
        END
        SUBROUTINE RESET(RDGC,RDG0,COFSET,GAIN,STDV,NCH)
C
C       SUBROUTINE DOES NCH LEAST SQUARES FITS OF RDG0=COFSET+GAIN*RDGC
C       LEAST SQUARES SUBROUTINE IS STANDARD FOR Y(J)=COFSET+GAIN*X(J)
C
        DIMENSION RDGC(NCH,4),RDG0(NCH,4),COFSET(NCH),GAIN(NCH),STDV(NCH)
        DATA NRD/4/
        DO 100 NC=1,NCH
        SUMX=0.0
        SUMY=0.0
        SUMXY=0.0
        SUMXX=0.0
        SUMYY=0.0
C
C       PERFORM LEAST SQUARES FIT OVER NRD READINGS
C
        DO 50 NR=1,NRD
        SUMX=SUMX+RDGC(NC,NR)
        SUMY=SUMY+RDG0(NC,NR)
        SUMXY=SUMXY+RDGC(NC,NR)*RDG0(NC,NR)
        SUMXX=SUMXX+RDGC(NC,NR)*RDGC(NC,NR)
        SUMYY=SUMYY+RDG0(NC,NR)*RDG0(NC,NR)
   50   CONTINUE
        XBAR=SUMX/NRD
        YBAR=SUMY/NRD
        GAIN(NC)=(SUMXY-SUMX*YBAR)/(SUMXX-SUMX*XBAR)
        COFSET(NC)=YBAR-GAIN(NC)*XBAR
        STDV(NC)=SQRT(ABS((SUMYY-SUMY*YBAR-GAIN(NC)*(SUMXY-SUMX*YBAR))/
       *FLOAT(NRD-2)))
  100   CONTINUE
        WRITE(0,210)(I,I=1,NCH)
        WRITE(0,220)(COFSET(NC),NC=1,NCH)
        WRITE(0,230)(GAIN(NC),NC=1,NCH)
        WRITE(0,240)(STDV(NC),NC=1,NCH)
  210   FORMAT(5X,8('  VLT(',I1,')'))
  220   FORMAT('OFSET',8(F8.4))
  230   FORMAT('GAIN ',8(F8.4))
  240   FORMAT('STDV ',8(F8.4))
        RETURN
        END
C
        SUBROUTINE CORRDG(VOLTS,COFSET,GAIN,NCH)
        DIMENSION VOLTS(NCH),COFSET(NCH),GAIN(NCH)
        DO 100 NC=1,NCH
        VOLTS(NC)=COFSET(NC)+GAIN(NC)*VOLTS(NC)
  100   CONTINUE
        RETURN
        END
C
        SUBROUTINE POLFUL(POLARY,JROW,IRDPR,NPOL,NTIM,IOFSET,LOU)
C
C THIS SUBROUTINE CONSTRUCTS AN ARRAY FOR PRINTING
C
```

```
      CHARACTER*2 RDGTYP(2),IDEG(12),INUM(9)
      CHARACTER*4 FUNTYP(4),BLANKS,CONSTA(2),POLARY(JROW,5),ROLD(4)
      DIMENSION NPOL(2,NTIM)
      DATA  RDGTYP/' M',' T'/,CONSTA/'CONS','TANT'/
      DATA FUNTYP/' LIN',' LOG',' EXP',' INV'/
      DATA IDEG/'1 ','2 ','3 ','4 ','5 ','6 ','7 ','8 ','9 ','10','11','12'/
      DATA INUM/'1)','2)','3)','4)','5)','6)','7)','8)','9)'/
      DATA BLANKS/'    '/
C
C     BLANK OUT POLARY ARRAY
C
      DO 71 J=1,5
      DO 71 I=1,JROW
71    POLARY(I,J)=BLANKS
C
      NROW=1
      IF(IOFSET.NE.1) GO TO 70
      POLARY(NROW,1)=CONSTA(1)
      POLARY(NROW,2)=CONSTA(2)
20    FORMAT(1X,3A4)
C
70    DO 200 NT=1,NTIM
      NDEG=NPOL(2,NT)
      IF(NDEG.EQ.0) GO TO 300
      DO 400 I=1,NDEG
      NROW=NROW+1
      POLARY(NROW,1)=FUNTYP(NPOL(1,NT))
      POLARY(NROW,2)='VLT('
      POLARY(NROW,3)=INUM(NT)//IDEG(I)
400   CONTINUE
21    FORMAT(/(1X,A4,'VLT(',I2,')',A4))
300   CONTINUE
200   CONTINUE
C
C  PRINT RESULTS
C
      DO 1000 NR=1,NROW
      WRITE(LOU,20)(POLARY(NR,II),II=1,3)
1000  CONTINUE
C
      RETURN
      END
C     CALL RDGPUL(READNG,SUMT,NPOL,IOFSET,TMDFT,MSET1
C    1,IRDPR1,M,NTIM,NL,NLT,NP,NPTT)
C
C     THE POLYNOMIAL IS CALCULATED
C
C     SUM=0.
C     DO  20 IR=1,IRDPR
C     SUM=SUM+COEF(IR)*READNG(M,IR)
C  20 CONTINUE
C     DRIFT=ABS(READNG(M,IRDPR1)-SUM)
C     SSDRIF=SSDRIF+DRIFT*DRIFT
C  70 CONTINUE
C  80 CONTINUE
C     SDRIF=SQRT(SSDRIF/FLOAT(MSET))
C     WRITE(LOU,60)CKTPAR(NDRIFT),SDRIF
C  60 FORMAT(5X,A4,F15.5)
C     CKTVAL(NDRIFT+10)=CKTVAL(NDRIFT+10)/1.01
C 800 CONTINUE
```

```
C       CALL RFCIRK(TMAG,PHASE,TWOPI,RAD,TMUTRE,TMUTIM,DRIVRE
C      1,DRIVIM,PICKRE,PICKIM,DRVDR,DRVDI,PICKDR,PICKDI,NRDF,FREQ
C      2,U,WUSR,GAIN,AIR1,AIR2,NLT,NRT,NPT,NFT,NODF,NPTT,DFSIZE)
C       RETURN
C       END
        SUBROUTINE RDGPUL(READNG,SUMT,NPOL,IOFSET,TMDFT,MSET1
       1,IRDFR1,M,NTIM,NL,NLT,NP,NPTT)
C       REAL*8 READNG,RDG
        DIMENSION READNG(MSET1,IRDFR1),NPOL(2,NTIM),TMDFT(NTIM)
        DIMENSION SUMT(NLT,NPTT,NTIM)
C
C       NPOL CONTAINS A NUMBER FOR THE FUNCTION TYPE & THE POLYNOMIAL
C       DEGREE. IF IOFSET=0,NO OFFSET IS INCLUDED,=1 OFFSET IS INCLUDED
C       NPOL(2,NT) =0,THAT PARTICULAR TIME VALUE WILL NOT BE INCLUDED
C       IN THE PROPERTY CALCULATIONS.
C
        READNG(M,1)=1.
        N=1
        IF(IOFSET.EQ.1)N=2
        DO 210 NT=1,NTIM
        IF(NPOL(2,NT).EQ.0) GO TO 200
        RDG=SUMT(NL,NP,NT)+TMDFT(NT)
C
C       THE TYPE OF FUNCTION IS SELECTED
C
        IF(NPOL(1,NT).EQ.1)RDG=RDG
        IF(NPOL(1,NT).EQ.2)RDG=ALOG(RDG)
        IF(NPOL(1,NT).EQ.3)RDG=EXP(RDG)
        IF(NPOL(1,NT).EQ.4)RDG=1./RDG
C
C       THE TYPE OF POLYNOMIAL IS SELECTED
C       AND THE POLYNOMIAL VALUES ARE CONSTRUCTED.
C
        READNG(M,N)=RDG
        N=N+1
        NDEG=NPOL(2,NT)-1
        IF(NDEG.LT.1) GO TO 15
        DO 10 I=1,NDEG
        READNG(M,N)=RDG*READNG(M,N-1)
        N=N+1
   10   CONTINUE
15      CONTINUE
  200   CONTINUE
  210   CONTINUE
        RETURN
        END
```

Appendix 5

PROGRAMS GRAPUL AND GRAPUF

GRAPUL and GRAPUF are FORTRAN programs to graph the raw readings or the calculated property values on the printer of the IBM System 9000 computer. Input data are read from a file called COEDAT.DAT. The polynomial type and coefficients are read from the file and used to calculate the sample properties from pulse magnitude readings as the sample is scanned; the properties are printed on the printer and also displayed on the CRT as the probe is moved along the sample.

GRAPUF is a shorter, faster-running version of GRAPUL.

```
      PROGRAM GRAPUL
C
C     VERSION 23 SEPTEMBER 1985
C     GRAPHS THE VALUE OF THE RAW READINGS OR THE CALCULATED VALUES OF THE
C     PROPERTIES FROM THE FILE COEDAT.DAT AS THE MOTORS ARE MOVED.THE NUMBER OF
C     STEPS TO MOVE PER PULSE MUST BE INPUTTED.
C
      DIMENSION NSTP(3),LSTP(3),IVOLTS(8),IVOLTO(8),VOLTS(8),FRO(15),RDG1(1,15)
      DIMENSION COFSET(8),GAIN(8),JPOL(2,8,7),NPOL(2,8),RDG0(8,4),VOLSTD(8)
      DIMENSION COE(15),COEF(15,7),JOFSET(7),JRDPR(7),TMDET(8),TMAC1(1,8)
      DIMENSION IY(20),RAWOFS(8),RAWGIN(8)
      CHARACTER PRONAM(7)*4,COEDAT*8,FNAME*12,TFILE*14,TUBE*8
      CHARACTER*2 AXC(3),GC
      CHARACTER*1 TM(16)
      INTEGER*2 FUNC(7),FUNT(3)
      DATA FUNC/30,1200,32,10,42,$8002,0/,FUNT/2,0,0/
      DATA ICMD/1/,NTP/16/,NTR/16/,NSEC/1/,ITIME1/0/,IXX/1/,NTIM/0/
      DATA RAWOFS/0.,0.,-4.2,-1.7,-.6,-.2,0.4,0.08/
      DATA RAWGIN/0.,0.,2.0,5.0,5.0,5.0,5.0,10.0/
      DATA AXC/'XM','YM','ZM'/,GC/'G'/,COEDAT/'P37A'/
      DATA MFEC/45/,MBUS/46/,NAXIS/1/,LSTP/3*0/,NRDG/1/,COFSET/8*0/,GAIN/8*0/
      DATA LOD/38/,LOU/6/,NPD/9/,IVOLTO/8*50/,NOLD/50/,NTT/5/,NTCAL/4/,XPOS/0./
      SFAC=25.75/12200.
C     IDENTIFY TUBE AND GENERATE FILE NAME FOR TUBE DATA DISKETTE
      WRITE(0,*)' INSERT NEW DISKETTE IN 5.25 DRIVE.  ENTER TUBE NUMBER (8 ALPHA
     1           NUMERICS)'
      READ(0,20) TUBE
   20 FORMAT(A8)
      TFILE='0:'//TUBE//'.DAT'
      OPEN(NPD,FILE=TFILE,STATUS='NEW')
      FNAME=COEDAT//'.DAT'
      OPEN(LOD,FILE=FNAME,STATUS='OLD',ERR=992)
C     SET UP GRAPHICS SUBROUTINE
      OPEN(LOU,FILE='#PR')
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(LOU,2)IHR,IMN,ISE,IMO,IDA,IYR,COEDAT
    2 FORMAT(' GRAPUL     TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2,
     *'    PROBE DATA ', A8)
C     MUTI FREQUENCY EDDY CURRENT INSTRUMENT ADDR=5 ON GPIB BUSS
C     ADDR=E? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 45
      OPEN(MFEC,FILE='#BUSE?')
C     MODULYNX AXIS POSITION CONTROLLER ADDR=6 ON GPIB BUSS,ADDR=F? ON
C     IBM BUSS DRIVER,FORTRAN LOGICAL UNIT = 46
      OPEN(MBUS,FILE='#BUSF?')
      CALL SYSFUN(MBUS,FUNC(1),IERR)
      WRITE(MBUS,*)'XD YE ZE'
C     WRITE(MBUS,*)'(CAH)'
      WRITE(MBUS,*)'XA10000XB400XH1000'
      NPRO=0
      DO 595 NC=1,NTCAL
      READ(LOD,*)(RDG0(JJ,NC),JJ=1,NTT)
  595 CONTINUE
      READ(LOD,*)(VOLSTD(NT),NT=1,NTT)
      WRITE(0,*)(VOLSTD(NT),NT=1,NTT)
  596 NPRO=NPRO+1
      READ(LOD,*,END=625)IRDPR
      WRITE(0,*)IRDPR
      READ(LOD,600)PRONAM(NPRO),JRDPR(NPRO),JOFSET(NPRO)
  600 FORMAT(1X,A4,2I4)
```

```
      WRITE(0,600)PRONAM(NPRO),JRDPR(NPRO),JOFSET(NPRO)
      READ(LOD,*)(COEF(IR,NPRO),IR=1,IRDPR)
      WRITE(0,*)(COEF(IR,NPRO),IR=1,IRDPR)
      DO 620 NT=1,NTT
      READ(LOD,*)(JPOL(I,NT,NPRO),I=1,2)
      WRITE(0,*)(JPOL(I,NT,NPRO),I=1,2)
  620 CONTINUE
      GO TO 596
  625 CONTINUE
      NPROPT=NPRO-1
C  PAPER MOVES 1.0 IN. FOR 84 PULSES-1 PAPER STEP=0.011905 IN=47.62 MOTOR STEPS
C  4000 MOTOR STEPS =1 IN.
    1 WRITE(0,*)NAXIS,' SET(S) OF STEPS PER PULSE(0.0021 IN/STEP)  '
      READ(0,*)(NSTP(IAXIS),IAXIS=1,NAXIS)
      WRITE(LOU,*)'STEPS PER PULSE=',NSTP(1),' SEC PER PULSE=',NSEC
    5 WRITE(0,*)'1.RAW RDGS 2. CALCULATED PROPERTIES 3.STOP? '
      READ(0,*)ITEST
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME1=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE+NSEC
      GO TO(10,700,50),ITEST
   10 CONTINUE
      CALL POS(NSTP,NAXIS,MBUS)
      LSTP(1)=LSTP(1)-NSTP(1)
      XPOS=XPOS+SFAC*NSTP(1)
      CALL RDDG(VOLTS,MFEC,NTT,NRDG,ICMD,NTP,NTR,IER)
      WRITE(0,15)XPOS,(VOLTS(NT),NT=1,NTT)
      WRITE(NPD,15)XPOS,(VOLTS(NT),NT=3,NTT)
   15 FORMAT(F8.3,8F6.3)
      DO 30 NT=1,NTT
      IY(NT)=(RAWGIN(NT)*(RAWOFS(NT)+VOLTS(NT)))*100
   30 CONTINUE
      CALL GRAPRT(IY,IXX,NTT,LOU,NTIM)
   40 CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 40
      ITIME1=ITIME+NSEC
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 10
  100 CONTINUE
      CALL SYSFUN(LOU,FUNT(1),IRR)
      NTIM=0
      WRITE(LOU,110)(RAWGIN(NT),NT=1,NTT)
      WRITE(LOU,110)(RAWOFS(NT),NT=1,NTT)
      WRITE(LOU,120)(IY(NT),NT=1,NTT)
  110 FORMAT(8F6.3)
  120 FORMAT(8I6)
      GO TO 1
C     CALCULATED PROPERTIED FROM MAGNITUDES AND PHASES AND CONTINOUSLY
C     DISPLAYS THE VALUES ON THE CRT TERMINAL.
C
  700 CONTINUE
C
C     EXPANSION OF TMAG1(1,NT) INTO READNG(1,IRDPRM)
C
  720 DO 800 NPRO=1,NPROPT
      DO 790 NT=1,NTT
      DO 780 I=1,2
      NPOL(I,NT)=JPOL(I,NT,NPRO)
  780 CONTINUE
      TMOPT(NT)=0
```

```
  790 CONTINUE
      IOFSET=JOFSET(NPRO)
      IRDPR=JRDPR(NPRO)
      IRDPR1=IRDPR+1
      CALL RDGPUL(RDG1,TMAG1,NPOL,IOFSET,TMDFT,1,IRDPR1,1,NTT,1,1,1,1)
      PRO(NPRO)=0.
      DO 795 IR=1,IRDPR
      PRO(NPRO)=PRO(NPRO)+COEF(IR,NPRO)*RDG1(1,IR)
  795 CONTINUE
  800 CONTINUE
      IY(1)=PRO(1)*4000
      IY(2)=PRO(2)*3
      WRITE(0,15)XPOS,(PRO(NPRO),NPRO=1,NPROPT)
      CALL GRAPRT(IY,IXX,NPROPT,LOU,NTIM)
C
C     NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
C
      CALL POS(NSTP,NAXIS,MBUS)
      LSTP(1)=LSTP(1)-NSTP(1)
      XPOS=XPOS+SFAC*NSTP(1)
  825 CALL PRDG(VOLTS,MFEC,NTT,NRDG,ICMD,NTP,NTR,IER)
      WRITE(NPD,15)XPOS,(VOLTS(NT),NT=3,NTT)
C
  840 DO 845 NT=1,NTT
      TMAG1(1,NT)=VOLTS(NT)
  845 CONTINUE
  860 CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 860
      ITIME1=ITIME+NSEC
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 720
C
C     PROGRAM WILL STAY IN THIS LOOP UNTIL ANY KEY IS STRUCK.
C
      GO TO 5
   50 CONTINUE
C     WRITE(MBUS,*)'(CAH)'
C OUT FOR STEAM GENERATOR
      CALL POS(LSTP,NAXIS,MBUS)
      WRITE(MBUS,*)'XE YE ZE'
      CALL SYSFUN(LOU,FUNT(1),IRR)
      WRITE(LOU,*)' '
      GO TO 1000
  992 WRITE(0,*)'OLD COEFFICIENT FILE NOT FOUND'
      GO TO 1
 1000 STOP
      END
      SUBROUTINE PRDG(VOLTS,MFEC,NCH,NRDG,ICMD,NTP,NTR,IER)
C     PROGRAM TO READ PULSE DATA OVER THE GPIB BUSS
      CHARACTER*1 TM(16),X,Y,CR
      DIMENSION VOLTS(NCH),IN(8)
      DATA TM/'!','"','#','$','%','&','(','*','0','5',':','@','O','W','~',' '/
      X=TM(NTP)
      Y=TM(NTR)
      CR=CHAR(13)
      VF1=5.0/4096.
      VFA=VF1/FLOAT(NRDG)
      DO 210 I=1,NCH
  210 VOLTS(I)=0.
```

```
      DO 250 JJ=1,NRDG
      WRITE(MFEC,35,ERR=400)ICMD,X,Y,CR
      READ(MFEC,70,ERR=500)(IN(I),I=1,NCH)
      DO 245 I=1,NCH
      VOLTS(I)=VFA*FLOAT(IN(I))+VOLTS(I)
  245 CONTINUE
  250 CONTINUE
   35 FORMAT(I1,A1,A1,A1)
C     WRITE(0,50)(VOLTS(I),I=1,NCH)
   50 FORMAT(1X,8F8.4)
   70 FORMAT(8I4)
      GO TO 600
  400 IER=1
      WRITE(0,*)'BUSS WRITE ERROR,RESET-HIT KEY RTN TO CONTINUE'
      GO TO 600
  500 IER=2
      WRITE(0,*)'BUSS READ ERROR,RESET-HIT KEY TO RTN CONTINUE'
  600 RETURN
      END
      SUBROUTINE FOS(NSTP,NAXIS,MBUS)
      DIMENSION NSTP(3)
      CHARACTER*2 AXC(3),GC
      DATA AXC/'XM','YM','ZM'/,GC/'G '/
      WRITE(MBUS,30)(AXC(IAXIS),NSTP(IAXIS),GC,IAXIS=1,NAXIS)
      WRITE(MBUS,*)';(CFA)'
      WRITE(MBUS,*)';'
   30 FORMAT(4(A2,I7,A2))
      RETURN
      END
      SUBROUTINE RDGPUL(READNG,SUMT,NPOL,IOFSET,TMDFT,MSET1
     1,IRDPR1,M,NTIM,NL,NLT,NP,NPTT)
C     REAL*8 READNG,RDG
      DIMENSION READNG(MSET1,IRDPR1),NPOL(2,NTIM),TMDFT(NTIM)
      DIMENSION SUMT(NLT,NPTT,NTIM)
C
C     NPOL CONTAINS A NUMBER FOR THE FUNCTION TYPE & THE POLYNOMIAL
C     DEGREE. IF IOFSET=0,NO OFFSET IS INCLUDED,=1 OFFSET IS INCLUDED
C     NPOL(2,NT) =0,THAT PARTICULAR TIME VALUE WILL NOT BE INCLUDED
C     IN THE PROPERTY CALCULATIONS.
C
      READNG(M,1)=1.
      N=1
      IF(IOFSET.EQ.1)N=2
      DO 210 NT=1,NTIM
      IF(NPOL(2,NT).EQ.0) GO TO 200
      RDG=SUMT(NL,NP,NT)+TMDFT(NT)
C
C     THE TYPE OF FUNCTION IS SELECTED
C
      IF(NPOL(1,NT).EQ.1)RDG=RDG
      IF(NPOL(1,NT).EQ.2)RDG=ALOG(RDG)
      IF(NPOL(1,NT).EQ.3)RDG=EXP(RDG)
      IF(NPOL(1,NT).EQ.4)RDG=1./RDG
C
C     THE TYPE OF POLYNOMIAL IS SELECTED
C     AND THE POLYNOMIAL VALUES ARE CONSTRUCTED.
C
      READNG(M,N)=RDG
      N=N+1
      NDEG=NPOL(2,NT)-1
```

```
      IF(NDEG.LT.1) GO TO 15
      DO 10 I=1,NDEG
      READNG(M,N)=RDG*READNG(M,N-1)
      N=N+1
   10 CONTINUE
15    CONTINUE
  200 CONTINUE
  210 CONTINUE
      RETURN
      END
      SUBROUTINE GRAPRT(IY,IXX,NCUR,LOU,NTIM)
C     PLOT A GRAPH ON PRINTER,EACH CALL WILL ADVANCE 1/84 OF INCH
C     IXX IS A COUNT OF THE NUMBER OF TIMES THAT SUBROUTINE IS CALLED
C     PRINTING IS DONE EVERY 7TH TIME SUBROUTINE IS CALLED.
C     Y VALUES OF THE CURNES MUST BE SCALED FROM 1 TO 800
      INTEGER*1 GRAPH(800),GRD1(800),GRD2(800)
      INTEGER*2 FUNC(3),FUNP(3),FUNT(3),BIT(7)
      DIMENSION IBUF(10),IY(NCUR)
      DATA FUNC/6,2,0/,FUNP/2,1,0/FUNT/2,0,0/,GRD1/800*0/,GRD2/800*1/
      DATA BIT/1,2,4,8,16,32,64/,IMAX/800/,MIN/800/,IX/0/,IBUF/10*0/,NA/0/
      IF(NTIM.EQ.1)GO TO 20
      CALL SYSFUN(LOU,FUNP(1),IRR)
C     SET COLOR USING FUNC CALL,0=RED,1=GREEN,2=BLUE,3=BLACK(DEFAULT)
C     CALL SYSFUN(LOU,FUNC(1),IRR)
C     SET UP GRID
      GRD1(1)=127
      GRD2(1)=85
      DO 5,I=100,800,100
      GRD1(I)=85
      GRD2(I)=85
    5 CONTINUE
      NTIM=1
      DO 10 I=1,IMAX
      GRAPH(I)=GRD2(I)
   10 CONTINUE
   20 CONTINUE
      IX=IX+1
      DO 30 NC=1,NCUR
      IF(IY(NC).GT.799)IY(NC)=799
      IF(IY(NC).LT.1)IY(NC)=1
      GRAPH(IY(NC))=GRAPH(IY(NC))+BIT(IX)
      GRAPH(IY(NC)+1)=GRAPH(IY(NC)+1)+BIT(IX)
   30 CONTINUE
      IF(IX.LT.7)GO TO 200
      DO 40 II=1,10
   40 IBUF(II)=0
      CALL SWRITE(LOU,GRAPH(1),IMAX,IMIN,IMAX,IBYT,IRR)
      IF(IRR.NE.0)WRITE(0,*)'ERROR= ',IRR
      IX=0
      IF(MOD(IXX,84).EQ.0)GO TO 80
      DO 70 I=1,IMAX
      GRAPH(I)=GRD1(1)
   70 CONTINUE
      GO TO 100
   80 DO 90 I=1,IMAX
      GRAPH(I)=GRD2(1)
   90 CONTINUE
  100 CONTINUE
  200 IXX=IXX+1
      RETURN



      END
```

```
      PROGRAM GRAPUF
C
C     VERSION 30 SEPTEMBER 1985  FAST VERSION OF GRAPUL
C     GRAPHS THE VALUE OF THE RAW READINGS OR THE CALCULATED VALUES OF THE
C     PROPERTIES FROM THE FILE COEDAT.DAT AS THE MOTORS ARE MOVED.THE NUMBER OF
C     STEPS TO MOVE PER PULSE MUST BE INPUTTED.RESPONSE IS PLOTTED ON CRT
C     THE RAW READINGS ARE SAVED ON 0: DISK.THE RAW RDG SECTION OF THE PROGRAN
C     HAS NOT BEEN CHECKED OUT!! THE TEMPERATURE SHIFT THAT OCCURED USING THE
C     PRINTER IS NOT EVIDENT.MAX SPEED IS 18.1 FPS WITH MOTOR DRIVE,26.8 WITHOUT
C
      DIMENSION NSTP(3),LSTP(3),IVOLTS(8),IVOLTO(8),VOLTS(8),IN(8)
      DIMENSION COFSET(8),GAIN(8),VOLSTD(8),IV(7000,3)
      DIMENSION RAWOFS(8),RAWGIN(8)
      CHARACTER FNAME*12,TFILE*14,TUBE*8,XCR*1,CR*1
      CHARACTER*2 AXC(3),GC
      INTEGER*2 FUNC(7),FUNT(3)
      DATA FUNC/30,1200,32,10,42,$8002,0/,FUNT/2,0,0/,XCR/' '/
      DATA ICMD/1/,NTP/16/,NTR/16/,NSEC/0/,ITIME1/0/,IXX/1/,NTIM/0/
      DATA RAWOFS/0.,0.,-4.2,-1.7,-.6,-.2,0.4,0.08/
      DATA RAWGIN/0.,0.,2.0,5.0,5.0,5.0,5.0,10.0/
      DATA AXC/'XM','YM','ZM'/,GC/'G'/,NGR/39/
      DATA MFEC/45/,MBUS/46/,NAXIS/1/,LSTP/3*0/,NRDG/1/,COFSET/8*0/,GAIN/8*0/
      DATA LOD/38/,LOU/6/,NPD/9/,IVOLTO/8*50/,NOLD/50/,NTT/5/,NTCAL/4/,XPOS/0./
      DATA C0/ -1.706746E2/,C1/-8.333823E1/,C2/7.575977/,C3/2.980622E2/
      CR=CHAR(13)
      SFAC=25.75/12200.
      VFAC=20./4096.
C     IDENTIFY TUBE AND GENERATE FILE NAME FOR TUBE DATA DISKETTE
      WRITE(0,*)' INSERT NEW DISKETTE IN 5.25 DRIVE.  ENTER TUBE NUMBER (8 ALPHA
     1           NUMERICS)'
      READ(0,20) TUBE
   20 FORMAT(A8)
      TFILE='0:'//TUBE//'.DAT'
      OPEN(NPD,FILE=TFILE,STATUS='NEW')
      OPEN(NGR,FILE='#GR')
C     SET UP GRAPHICS SUBROUTINE
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      WRITE(0,2)IHR,IMN,ISE,IMO,IDA,IYR
    2 FORMAT(' GRAPUF    TIME ',I2,':',I2,':',I2,'  DATE ',I2,'/',I2,'/',I2)
C     PULSED EDDY CURRENT INSTRUMENT ADDR=5 ON GPIB BUSS
C     ADDR=E? ON IBM BUSS DRIVER, FORTRAN LOGICAL UNIT = 45
      OPEN(MFEC,FILE='#BUSE?')
C     MODULYNX AXIS POSITION CONTROLLER ADDR=6 ON GPIB BUSS,ADDR=F? ON
C     IBM BUSS DRIVER,FORTRAN LOGICAL UNIT = 46
      OPEN(MBUS,FILE='#BUSF?')
      CALL SYSFUN(MBUS,FUNC(1),IERR)
      WRITE(MBUS,*)'XD YE ZE'
      WRITE(MBUS,*)'(CAH)'
C     WRITE(MBUS,*)'XA10000XB400XH1000'
      XPOS=0.
  625 CONTINUE
    1 WRITE(0,*)' STEPS PER PULSE(0.0021 IN/STEP)
      READ(0,*)(NSTP(1))
      WRITE(0,*)'STEPS PER PULSE=',NSTP(1),' SEC PER PULSE=',NSEC
    5 WRITE(0,*)'1.RAW RDGS 2. CALCULATED PROPERTIES 3.STOP? '
      READ(0,*)ITEST
      CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME1=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE+NSEC
      GO TO(10,700,50),ITEST
```

```
   10 CONTINUE
      WRITE(MBUS,11)AXC(1),NSTP(1),GC
C     WRITE(MBUS,*)';(CFA)'
C     WRITE(MBUS,*)';'
   11 FORMAT(A2,I7,A2)
      XPOS=XPOS+SFAC*NSTP(1)
   15 FORMAT(F8.3,8F6.3)
      DO 30 NT=1,NTT
CCC   IY(NT)=(RAWGIN(NT)*(RAWOFS(NT)+VOLTS(NT)))*100
   30 CONTINUE
C     CALL GRAPRT(IY,IXX,NTT,LOU,NTIM)
   40 CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
      IF(ITIME.LT.ITIME1)GO TO 40
      ITIME1=ITIME+NSEC
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 10
 100  CONTINUE
      NTIM=0
      WRITE(0,110)(RAWGIN(NT),NT=1,NTT)
      WRITE(0,110)(RAWOFS(NT),NT=1,NTT)
CC    WRITE(0,120)(IY(NT),NT=1,NTT)
  110 FORMAT(8F6.3)
  120 FORMAT(8I6)
      GO TO 1
C     CALCULATED PROPERTIED FROM MAGNITUDES AND PHASES AND CONTINOUSLY
C     DISPLAYS THE VALUES ON THE CRT TERMINAL.
C
  700 CONTINUE
      CALL CLS
      CALL GRID(NGR)
      IX1=50
      NR=0
  720 IX2=IX1+1
      NR=NR+1
C
C     NEW READINGS ARE MADE FROM THE EDDY CURRENT INSTRUMENT.
C
      WRITE(MBUS,11)AXC(1),NSTP(1),GC
CC    WRITE(MBUS,*)'XM24G'
C     WRITE(MBUS,*)';(CFA)'
C     WRITE(MBUS,*)';'
      XPOS=XPOS+SFAC*NSTP(1)
      WRITE(MFEC,35,ERR=400)ICMD,XCR,XCR,CR
      READ(MFEC,70,ERR=500)(IN(I),I=1,NTT)
   35 FORMAT(I1,A1,A1,A1)
   70 FORMAT(8I4)
      IY2=C0*4.+(C1*FLOAT(IN(3))+C2*FLOAT(IN(4))+C3*FLOAT(IN(5)))*VFAC-350
CC    XY2=C0+(C1*FLOAT(IN(3))+C2*FLOAT(IN(4))+C3*FLOAT(IN(5)))*5.0/4096.
      CALL GRAPH(IX1,IY1,IX2,IY2,NGR)
CC    WRITE(0,835)XPOS,(IN(NT),NT=3,NTT),IY2,XY2
  835 FORMAT(F8.3,3I6,I5,F9.3)
      IY1=IY2
      IX1=IX2
C
  840 DO 845 NT=3,NTT
      IV(NR,NT-2)=IN(NT)
  845 CONTINUE
  860 CALL GETTIM(IYR,IMO,IDA,IHR,IMN,ISE,IFR)
      ITIME=((((30*IMO)+IDA)*24+IHR)*60+IMN)*60+ISE
```

```
      IF(ITIME.LT.ITIME1)GO TO 860
      ITIME1=ITIME+NSEC
      CALL GETKEY(IKY,IS,KBF1,KBF2,IRR)
      IF(IRR.NE.0)GO TO 720
C
C     PROGRAM WILL STAY IN THIS LOOP UNTIL ANY KEY IS STRUCK.
C
      GO TO 5
   50 CONTINUE
      WRITE(MBUS,*)'(CAH)'
      GO TO 1000
  992 WRITE(0,*)'OLD COEFFICIENT FILE NOT FOUND'
      GO TO 1
  400  WRITE(0,*)'BUSS WRITE ERROR,RESET-HIT KEY RTN TO CONTINUE'
      GO TO 860
  500  WRITE(0,*)'BUSS READ ERROR,RESET-HIT KEY TO RTN CONTINUE'
      GO TO 860
 1000 DO 1050 N=1,NR
      WRITE(NPD,1060)(IV(N,NT),NT=1,3)
 1050 CONTINUE
 1060 FORMAT(8I5)
      STOP
      END
      SUBROUTINE GRAPH(IX1,IY1,IX2,IY2,NGR)
C     TEST PROGRAM TO DO GRAPHICS IN FORTRAN
C     EXAMPLE OF LINE PLOT CALL IN BASIC  300 LINE(XOLD,YOLD,XNEW,YNEW,1)
      INTEGER*2 XYFUN(7)
      DATA XYFUN/23,0,0,27,0,0,0/
      XYFUN(2)=IX1
      XYFUN(3)=IY1
      XYFUN(5)=IX2
      XYFUN(6)=IY2
      CALL SYSFUN(NGR,XYFUN(1),IERR)
      RETURN
      END
      SUBROUTINE GRID(NGR)
      CALL GRAPH(50,50,750,50,NGR)
      CALL GRAPH(50,150,750,150,NGR)
      CALL GRAPH(50,250,750,250,NGR)
      CALL GRAPH(50,350,750,350,NGR)
      CALL GRAPH(50,450,750,450,NGR)
      CALL GRAPH(50,50,50,450,NGR)
      CALL GRAPH(150,50,150,450,NGR)
      CALL GRAPH(250,50,250,450,NGR)
      CALL GRAPH(350,50,350,450,NGR)
      CALL GRAPH(450,50,450,450,NGR)
      CALL GRAPH(550,50,550,450,NGR)
      CALL GRAPH(650,50,650,450,NGR)
      CALL GRAPH(750,50,750,450,NGR)
      RETURN
      END
```

Appendix 6

PROGRAM GPUL

GPUL is a microcomputer assembly language program that enables the NDT COMP9B microcomputer to communicate with the peripheral equipment, such as the pulser, the probe positioner, and the pulse amplitude instrument. The pulse control with the IBM System 9000 is over an IEEE-488 general purpose information bus (GPIB).

GPUL configures the COMP9B microcomputer to furnish timing instructions to the sample-and-hold (S/H) modules and read the analog-to-digital converters (ADC). Lines with a semicolon are comments; lines 16 through 24 define the port addresses; lines 26 through 31 give the GPIB and mask definitions, and so on.

Timing instructions are determined from an array called TM, set up in PULRDG, with 16 different time scales, according to the ASCII character in TM, as shown in Table A6-1. The times T1, T2, ... are times available for the corresponding S/H module to take a reading, allowing a certain amount of dead time between readings. Since the pulse is limited to about 435 μs to avoid overheating, only the faster schedules can be used if many readings per pulse are needed.

Table A6-1.

TM number	ASCII character	Time (μs)	Times available (μs)				
			T1	T2	T3	T4	...
1	!	2541.8	.	.	.	.	
.	.	.	.	.	.	.	
8	%	296	366.0	355.5			
9	&	200	268.4	257.9			
10	(	170	239.1	228.5			
11	*	140	209.9	199.2			
12	@	120	190.2	179.7			
13	O	100.5	170.8	160.1			
14	W	90.7	161.0	150.4	150.4	150.4	
15	~	71.2	141.4	130.9	130.9	130.8	
16	(space)	24.8	95.1	84.5	84.4	84.5	...

There is a number ICMD, also set up in PULRDG, which tells GPUL which command to run (for the pulser, S/H module, etc.). The various possibilities are determined in the following sections of GPUL.

ICMD name	Comment lines	Command lines	Description
CMD1 CMD2	284-288	290-328	Readings made during pulse
CMD3	330-331	333-368	Calibration only; no pulse
CMD4 CMD5	370-374	376-419	Readings made after pulse turned off
CMD6	421-422	424-462	Fast pulse and readings during pulse
CMD7	464-465	467-504	Fast pulse and readings after pulse turned off

The section following line 506 has instructions for reading the ADCs and converting to ASCII representation. The section following line 593 takes care of data summation, converts data, etc.

The NDT COMP9B port assignments for the pulsed instrument are shown below.

PORT 1		PORT 2	
Input		Output	
A0	ADC0, LSB	A0	calibrate, operate
A1	ADC1	A1	
A2	ADC2	A2	
A3	ADC3	A3	
A4	ADC4	A4	
A5	ADC5	A5	
A6	ADC6	A6	
A7	ADC7	A7	
Output		Output	
B0	ADC sel. A0, 8205	S/H1	
B1	ADC sel. A1, 8205	S/H2	
B2	ADC sel. A2, 8205	S/H3	
B3	ADC sel. E3, 8205	S/H4	
B4		S/H5	
B5	ADC start	S/H6	
B6		S/H7	
B7		S/H8	
Input		Output	
C0	in ADC 8	C0	
C1	in ADC 9	C1	
C2	in ADC 10	C2	
C3	in ADC 11, MSB	C3	
C4		C4	
C5		C5	
C6		C6	
C7		C7	pulser and calibrator

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    1

   1                                          TITLE 'GPUL PROGRAM VERSION  5 AUGUST 85'
   2                                          NAME    GPUL
   3                                          LIST   B,G,O,T
   4                                ;         NLIST   I,M,S,T
   5                                ;
   6                                ;         PROGRAM TO CONFIGURE THE COMP9B MICROCOMPUTER
   7                                ;         AS A TALKER/LISTENER ON THE IEEE488/GPIB BUSS
   8                                ;         USING AN 8291A& 2 8293'S.
   9                                ;         START PULSE,OPERATE S/H READ ADC'S OVER GPIB
  10                                ;
  11                                          ORG 0800H        START PROGRAM IN SECOND PROM.
  12                                ; SYMBOL DEFINITIONS
  13                                ;
  14                                ;  DEFINE PORT ADDRESSES
  15                                ;
  16  00CF                          PORT1     EQU 0CFH         PORT 1 CONTROL WORD ADDRESS
  17  00CC                          PORT1A    EQU 0CCH         PORT1A ADDRESS
  18  00CD                          PORT1B    EQU 0CDH         PORT1B ADDRESS
  19  00CE                          PORT1C    EQU 0CEH         PORT1C ADDRESS
  20  00D7                          PORT2     EQU 0D7H         PORT 2 CONTROL WORD ADDRESS
  21  00D4                          PORT2A    EQU 0D4H         PORT2A ADDRESS
  22  00D5                          PORT2B    EQU 0D5H         PORT2B ADDRESS
  23  00D6                          PORT2C    EQU 0D6H         PORT2C ADDRESS
  24  00EE                          APDATA    EQU 0EEH         ARITHMETIC PROCESSOR DATA PORT
  25                                ;         GPIB ADDRESS AND MASK DEFINITIONS
  26  00D8                          BUSPRT    EQU 0D8H          BASE ADDR OF GPIB BUSS
  27  00D8                          BUSIN     EQU 0D8H          DATA IN FROM BUSS ADDR
  28  00D8                          BUSOUT    EQU BUSIN         DATA OUT TO BUSS ADDR
  29  00D9                          S1        EQU 0D9H          INTERRUPT STATUS REGISTER1 ADDR
  30  00D9                          INTE1     EQU 0D9H          INTERRUPT ENABLE REGISTER 1 ADDR
  31  0002                          BOM       EQU 02            BYTE OUT INTR MASK,BYTE
  32                                ;                           SHOULD BE WRITTEN IN BUSOUT
  33  0001                          BIM       EQU 01            BYTE IN INTR MASK,BYTE SHOULD
  34                                ;                           BE READ FROM BUSIN REGISTER
  35  0010                          ENDMK     EQU 10H           END INTERRUPT MASK
  36  0080                          CPT       EQU 080H          COMMAND PASS THROUGH MASK
  37                                ;         REG #2 INTERRUPTS
  38  00DA                          INTE2    EQU 0DAH          INTERRUPT REGISTER 2 ADDR
  39                                ;         REG #4 ADDRESS MODE CONSTANTS
  40  00DC                          ADRMD     EQU 0DCH          ADDRESS MODE REGISTER ADDR
  41  0080                          TON       EQU 080H          TALK ONLY,NOT LISTEN  MODE
  42  0040                          LON       EQU 040H          LISTEN ONLY,NOT TALK MODE
  43  00C0                          TLON      EQU 0C0H          TALK AND LISTEN ONLY MODE
  44  0001                          MODE1     EQU 01H           MODE 1 ADDRESSING
  45                                ;         REG #4 (READ) ADDRESS STATUS REGISTER
  46  00DC                          ADRST     EQU ADRMD         ADDRESS STATUS REGISTER ADDR
  47  0020                          EOIST     EQU 20H           END OR IDENTIFY MASK
  48  0002                          TA        EQU 02H           TALKER ADDR OR ACT;SER POLL-TADS TACS SPAS
  49  0004                          LA        EQU 04H           LISTENER ADDRESSED OR ACTIVE-LADS OR LACS
  50  0001                          MJMN      EQU 01H           MAJOR OR MINOR TALKER/LISTENER,1=MINOR
  51                                ;         REG #5(WRITE) AUXILLARY MODE REGISTER
  52  00DD                          AUXMD     EQU 0DDH          AUXILLARY MODE REG ADDR
  53  0022                          CLKRT     EQU 022H          CLOCK SET FOR 2MHZ
  54  0003                          FNHSK     EQU 03H           FINISH HANDSHAKE AUX COMMAND
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    2

      55  0006                    SDEOI       EQU 06H           SEND END OR IDENTIFY WITH NEXT BYTE
      56  0080                    AXRA        EQU 080H          WRITE DDDDD INTO AUX REGISTER A
      57  0001                    HOHSK       EQU 01H           HOLD OFF HANDSHAKE ON ALL BYTES
      58  0002                    HOEND       EQU 02H           HOLD OFF HANHSHAKE ON END BYTE
      59  0003                    CAHCY       EQU 03H           CONTINIOUS ACCEPTOR HANDSHAKE CYCLING
      60  0004                    EDEOS       EQU 04H           END ON EOS RECEIVED DAT REG MATCHES EOS RE
      61  0008                    EOIS        EQU 08H           OUTPUT EOI ON EOS SENT
      62  0040                    EOSBC       EQU 40H           EOS FUNCTIONS AS FULL 8-BIT REG
      63  000F                    VSCMD       EQU 0FH           VALID COMMAND PASS THROUGH
      64  0007                    NVCMD       EQU 07H           INVALID COMMAND PASS THROUGH
      65  00A0                    AXRB        EQU 0A0H          AUXILLARY REG B PATTERN
      66  0001                    CPTEN       EQU 01H           COMMAND PASS THROUGH ENABLE
      67                          ;           REG #5(READ)
      68  00DD                    CPTRG       EQU 0DDH          ADDR TO READ COMMAND PASS THROUGH
      69                          ;           REG #6 (WRITE) ADDRESS 0/1 REG CONSTANTS
      70  00DE                    ADR01       EQU 0DEH          COMP 9 GPIB ADDRESSES
      71  0060                    DTDL1       EQU 060H          DISABLE MAJOR TALKER/LISTENER
      72  00E0                    DTDL2       EQU 0E0H          DISABLE MINOR TALKER/LISTENER
      73  0005                    ADRTL       EQU 05H           TALKER LISTENER ADDRESS SET TO 5
      74                          ;           REG #7 EOS-END OF SEQUENCE CHARACTER REG
      75  00DF                    EOSR        EQU 0DFH          FLAGS END OF BLOCK BY CHAR IN REG
      76                          ;
      77                          ;           GPIB MESSAGES(COMMANDS)
      78                          ;
      79                           PUBLIC       PRTSU
      80                          ;
      81                          ;
      82                          ;  MATH SUBROUTINES FOR THE COMP 9 ARE STORED AS PUBLIC.ANY ROUTINE
      83                          ;  CAN BE CALLED USING AN 'EXTRN' STATEMENT.
      84                          ;
      85                           EXTRN          ACOS,ASIN,ATAN,ATANA,BIDEC,BIDECF
      86                           EXTRN          CHSD,CHSDA,CHSF,CHSFA,CHSS,CHSSA,COS,COSA
      87                           EXTRN          DADD,DADDA,DADDB,DDIV,DDIVA,DDIVB,DECND
      88                           EXTRN          DMUL,DMULA,DMULB,DMUU,DMUUA,DMUUB,DSUB,DSUBA,DSUBB
      89                           EXTRN          EXP,EXPA,EXP10,FADD,FADDA,FADDB,FDIV,FDIVA,FDIVB
      90                           EXTRN          FIXD,FIXDA,FIXS,FIXSA,FLTD,FLTDA,FLTS,FLTSA
      91                           EXTRN          FMUL,FMULA,FMULB,FSUB,FSUBA,FSUBB,LN,LNA,LOG,LOGA
      92                           EXTRN          MDAD,POPD,POPS,PTOD,PTOS,PUPI,PWR,PWRA,PWRB
      93                           EXTRN          SADD,SADDA,SADDB,SDIV,SDIVA,SDIVB,SIN,SINA
      94                           EXTRN          SMUL,SMULA,SMULB,SMUU,SMUUA,SMUUB,SQRT,SQRTA
      95                           EXTRN          SSUB,SSUBA,SSUBB,TAN,TANA,TOS2,TOS4
      96                           EXTRN          WRT2,WRT4,XCHD,XCHS
      97                           EXTRN          CROUT,COPDT,GETCM,PRINT,PRINTF,EPRNT,FPRNT,ZERO
      98                           EXTRN          GETC,GETNM,CO,NMOUT,ECHO,GETHX,SGNON
      99                          ;
     100                          ;
     101                          ;CONSTANTS ARE SET
     102  0001                    NRDG        EQU 001H          NRDG=2**N=NUMBER OF RDGS PER CHANNEL=1
     103                          ;
     104  0005                    NCHA        EQU 05H           NCHA=NUMBER OF CHANNELS TO BE READ
     105  0030                    PRT10       EQU 003H          PRT10=OUTPUT MODE,A,B,C0-3 OUTPUT,C4-7 IN
     106  0099                    PRT11       EQU 099H          PRT11=INPUT MODE,A&C INPUT;B OUTPUT
     107                          ;
     108                          ;       RAM ADDRESS DEFINITIONS
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    3

     109                          ;
     110  3C10                    RAWDA       EQU 03C10H        RAW RDGS,4 BYTES/CHNL,MG1,PH1,MG2.. HI-LO
     111  3C24                    OLDSW       EQU RAWDA+4*NCHA  CALIBRATION SWITCH POSITION FROM CONTROLLE
     112  3C25                    ASCDAT      EQU OLDSW+1       ASCII DATA,8*NCHA+1 BYTES
     113  3C4E                    DIAD        EQU ASCDAT+8*NCHA+1  ;ADDRESS FOR INPUT DATA
     114  3C50                    NCH         EQU DIAD+2        CHANNEL NUMBER COUNT FOR ADC RDGS
     115  3C51                    CNTSH       EQU NCH+1         SAMPLE AND HOLD COUNT NUMBER
     116  3C52                    SHDAT       EQU CNTSH+1       SAMPLE AND HOLD DATA BYTE
     117  3C53                    DATIN       EQU SHDAT+1       START OF DATA READ IN FROM GPIB,32 BYTES
     118  3800                    RAMADR      EQU 3800H         ADDRESS OF RAM PROGRAM,CALLED BY CMD6
     119  1000                    INTRP       EQU 1000H         ADDRESS OF FRONT PANEL INTERUPTS
     120                          ;       ENTRY POINT FOR INTERUPTS
     121  0800  F1                            POP PSW           POP STACK SO WE WON'T HAVE OLD ADDR
     122  0801  DB D9                         IN S1             READ INPUT STATUS
     123  0803  FE 02                         CPI BOM           CHECK FOR BYTE OUT REQUEST
     124  0805  CA 1D 08                      JZ BOUT           JUMP TO OUTPUT DATA
     125  0808  FE 01                         CPI BIM           CHECK FOR BYTE IN REQUEST
     126  080A  CA 27 08                      JZ BIN            JUMP TO BYTE IN ROUTINE
     127  080D  FE 10                         CPI ENDMK         CHECK FOR END OF TRANSMISSION
     128  080F  CA 33 08                      JZ BEND           JUMP TO END OF TRANSMISSION ROUTINE
     129  0812  FE 00                         CPI 00H           COMPARE TO ZERO
     130  0814  CA 00 10                      JZ INTRP          INTERUPT WAS NOT FROM 8291.
     131  0817  CD 00 00      E               CALL NMOUT        PRINT A REGISTER
     132  081A  C3 00 00      E               JMP GETCM         RETURN TO MONITOR
     133  081D  7E                    BOUT    MOV A,M           LOAD ASCII BYTE ADDR BY HL INTO A
     134  081E  D3 D8                         OUT BUSPRT        WRITE ONTO GPIB BUSS
     135  0820  23                            INX H             ADDRESS NEXT BYTE
     136  0821  0D                            DCR C             DECREMENT COUNTER
     137  0822  CA 45 08                      JZ WREND          FINISHED TRANSMISSION
     138  0825  FB                            EI                ENABLE INTERUPT
     139  0826  76                            HLT               HALT COMPUTER UNTIL NEXT INTERUPT
     140  0827  DB D8                 BIN     IN BUSPRT         READ A BYTE IN FROM THE BUSS
     141  0829  2A 4E 3C                      LHLD DIAD         LOAD ADDRESS FOR DATA-IN TO HL
     142  082C  77                            MOV M,A           STORE DATA READ IN INTO RAM
     143  082D  23                            INX H             INCREMENT FOR NEXT BYTE
     144  082E  22 4E 3C                      SHLD DIAD         STORE NEXT ADDRESS
     145  0831  FB                            EI                ENABLE INTERUPT
     146  0832  76                            HLT               WAIT FOR NEXT INTERUPT
     147  0833  3A 53 3C             BEND    LDA DATIN         LOAD FIRST BYTE WRITTEN OVER GPIB
     148  0836  E6 0F                         ANI 0FH           DUMP HI BITS,CONVERT TO ADDRESS
     149  0838  87                            ADD A             DOUBLE VALUE,ADDRESSES ARE 2 BYTES
     150  0839  4F                            MOV C,A           LOAD INTO C
     151  083A  06 00                         MVI B,00          ZERO B,BC CONTAIN ADDR INCREMENT
     152  083C  21 85 08                      LXI H,CADDR       COMMAND ADDRESS LISTS IN HL
     153  083F  09                            DAD B             OFFSET TO STARTING ADDRESS OF PROGRAMS
     154  0840  7E                            MOV A,M           PUT LOW ORDED CMD ADDR IN A
     155  0841  23                            INX H             ADDRESS HI BYTE OF CMD ADDR
     156  0842  66                            MOV H,M           SEND TO H
     157  0843  6F                            MOV L,A           PUT LOW ORDER IN L
     158  0844  E9                            PCHL              TRANSFER COMMAND TO ADDR IN HL
     159  0845  3E 06                 WREND   MVI A,06H         SEND EOI WITH NEXT BYTE OUT
     160  0847  D3 DD                         OUT AUXMD         SEND TO AUX MODE REGISTER
     161  0849  FB                            EI                ENABLE INTERUPT
     162  084A  76                            HLT               HALT COMPUTER UNTIL NEXT INTERUPT
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    4

     163                          ;
     164                          ;GPIB PORT SET-UP
     165  084B  3E 99              PRTSU       MVI A,PRT1I       LOAD PROGRAM WORD FOR A&C INPUT,B OUTPUT
     166  084D  D3 CF                          OUT PORT1        SEND TO PORT 1
     167  084F  3E 80                          MVI A,080H        LOAD PRG WRD,A=OUTPUT,B&C=OUTPUT
     168  0851  D3 D7                          OUT PORT2         SEND TO PORT2
     169  0853  3E 02                          MVI A,02          RESET THE 8291A
     170  0855  D3 DD                          OUT AUXMD         SEND TO THE AUXULARY MODE REG
     171  0857  3E 00                          MVI A,00          ZERO A REGISTER
     172  0859  D3 D9                          OUT INTE1         DISABLE INTERUPTS TEMPORARILY
     173  085B  D3 DA                          OUT INTE2         DISABLE SECONDARY INTERUPTS
     174  085D  3E 01                          MVI A,MODE1       ENABLE MODE 1 ADDRESSING
     175  085F  D3 DC                          OUT ADRMD         SEND TO THE ADDRESS MODE REG
     176  0861  3E 22                          MVI A,CLKRT       SET THE CLOCK RATE FOR 2 MHZ
     177  0863  D3 DD                          OUT AUXMD         SEND TO THE AUX MODE REG
     178  0865  3E 05                          MVI A,ADRTL       LOAD TALKER/LISTENER ADDRESS
     179  0867  D3 DD                          OUT AUXMD         SEND TO AUXILIARY COMMAND REG
     180  0869  D3 DE                          OUT ADR01         SEND TO COMP 9 GPIB ADDRESSES
     181  086B  3E 0D                          MVI A,0DH         LOAD A CARRIAGE RETURN AS END OF MESS
     182  086D  D3 DF                          OUT EOSR          SEND TO END OF SEQUENCE REGISTER
     183  086F  3E 84                          MVI A,084H        LOAD END ON EOS RECEIVED DAT
     184  0871  D3 DD                          OUT AUXMD         SEND TO AUXULARY MODE REGISTER
     185  0873  3E 13                          MVI A,013H        ENABLE END BYTE,BYTE IN,BYTE OUT INTRPS
     186  0875  D3 D9                          OUT INTE1         ENABLE INTERUPTS
     187  0877  3E 00                          MVI A,00          LOAD ZERO INTO A
     188  0879  D3 DD                          OUT AUXMD         PUT 8291A INTO IDLE MODE
     189  087B  21 53 3C                       LXI H,DATIN       STORE ADDRESS FOR INPUT DATA INTO HL
     190  087E  22 4E 3C                       SHLD DIAD         STORE IN RAM
     191  0881  21 00 00      E               LXI H,SGNON       LOAD SIGNON MESSAGE ADDRESS
     192  0884  C9                            RET
     193                          ;
     194                          ;           TABLE OF ADDRESSES FOR COMMANDS FOR MICROCOMPUTER
     195                          ;           THE COMMAND TRANSFER ADDRESSES ARE PICKED UP FROM
     196                          ;           THIS TABLE.FIRST BYTE XMITTED BY CONTROLLER IS CMD NUMBER
     197                          ;           THE SECOND BYTE DETERMINES THE DELAY TIME BETWEEN S/H PULSES
     198  0885  B6 08              CADDR       DW CMD0           NORMAL PULSE READING
     199  0887  1C 09                          DW CMD1           FAST READINGS AND CONVERSION,ASCII XMITT
     200  0889  1D 09                          DW CMD2           FAST RDG,SEND OLD RDG WHILE PROCSNG NXT
     201  088B  6A 09                          DW CMD3           FAST CALIBRATOR READINGS
     202  088D  B2 09                          DW CMD4           FAST RDGS & CONV,ASCII XMITT,RDG PULSE ON
     203  088F  B3 09                          DW CMD5           FAST RDG ON TURN OFF,SEND OLD RDG WHILE PR
     204  0891  0B 0A                          DW CMD6           FAST RDG,NO DELAYS,RDGS WHILE PULSE IS ON
     205  0893  54 0A                          DW CMD7           FAST RDG,NO DELAYS,RDGS AFTER PULSE OFF
     206                          ;
     207                          ;           BASK              CONVERTS BINARY BYTE ADDR BY HL
     208                          ;                             TO ASCII NO ADDR BY DE FOR C BYTES
     209                          ;
     210  0895  7E                BASK        MOV A,M           LOAD BYTE ADDRESSED BY HL
     211  0896  1F                            RAR               CONVERT 4 HI BITS FIRST
     212  0897  1F                            RAR               ROTATE 4 HI BITS TO LOW POS
     213  0898  1F                            RAR
     214  0899  1F                            RAR
     215  089A  E6 0F                        ANI 0FH           LOOK AT ONLY 4 LOW BITS
     216  089C  27                            DAA               DEC ADJUST ACC
```

```
217  089D  C6 F6                    ADI 0F6H      SET CARRY FOR BYTE GT 9
218  089F  CE 3A                    ACI 03AH      ADD REST OF VALUE WITH CARRY
219  08A1  12                       STAX D        STORE IN RAM ADDR BY DE
220  08A2  13                       INX D         ADDRESS NEXT ASCII MEM LOC
221  08A3  7E                       MOV A,M       READ HEX BYTE BACK IN
222  08A4  E6 0F                    ANI 0FH       LOOK AT 4 LOW BITS
223  08A6  27                       DAA           DEC ADJUST ACC
224  08A7  C6 F6                    ADI 0F6H      SET CARRY FOR BYTE GT 9
225  08A9  CE 3A                    ACI 03AH      ADD REST OF NUMBER WITH CARRY
226  08AB  12                       STAX D        STORE IN RAM ADDR BY DE
227  08AC  13                       INX D         ADDRESS NEXT ASCII MEM LOC
228  08AD  23                       INX H         ADDRESS NEXT NEXT HEX BYTE
229  08AE  0D                       DCR C         DECREMENT COUNTER
230  08AF  C2 95 08                 JNZ BASK      GO BACK FOR NEXT BYTE IF NOT DONE
231  08B2  3E 0D                    MVI A,0DH     ADD A CARRIAGE RETURN AS THE LAST CHAR
232  08B4  12                       STAX D        STORE IT IN RAM AFTER THE DATA
233  08B5  C9                       RET           RETURN TO CALLER IF DONE
234                          ;
235                          ;      CMD0          LOOP TO TAKE READINGS AND SEND THEM OUT
236                          ;
237  08B6  3E 05             CMD0   MVI A,NCHA    SET THE NUMBER OF S&H CHANNELS IN RAM
238  08B8  32 51 3C                 STA CNTSH     STORE
239  08BB  3E 80                    MVI A,080H    SET HI ORDER BYTE HI(1000/0000)
240  08BD  D3 D6                    OUT PORT2C    SEND PULSE OUT
241  08BF  3A 54 3C                 LDA DATIN+1   LOAD VALUE OF SECOND BYTE TRANSMITTED
242  08C2  D6 30                    SUI 030H      CONVERT TO HEX
243  08C4  47                       MOV B,A       STORE IN B REG
244  08C5  0E 00                    MVI C,0H      ZERO C REG
245  08C7  C5                LOPSH  PUSH B        TEMP STORE ON STACK
246  08C8  CD 6B 0B                 CALL DELAY    DELAY BY BC
247  08CB  CD 9B 0B                 CALL SHLD     SEND SAMPLE AND HOLD HI
248  08CE  C1                       POP B         RESTORE BC DELAY VALUE
249  08CF  3A 51 3C                 LDA CNTSH     LOOK AT COUNTER VALUE
250  08D2  3D                       DCR A         DECREMENT COUNTER
251  08D3  32 51 3C                 STA CNTSH     RESTORE COUNTER
252  08D6  C2 C7 08                 JNZ LOPSH     STAY IN LOOP UNTIL ALL S/H'S ON
253  08D9  AF                       XRA A         ZERO ACCUMULATOR
254  08DA  D3 D6                    OUT PORT2C    SEND OUT TO PORT 2C
255  08DC  3E 99                    MVI A,PRT1I
256  08DE  D3 CF                    OUT PORT1     SET PORT1 FOR INPUT
257  08E0  3E B0                    MVI A,080H    STROBE ADC 1011/0000,BIT B5 HI
258  08E2  D3 CD                    OUT PORT1B
259  08E4  3E 90                    MVI A,090H    THEN STROBE 1001/0000,BIT B5 LO
260  08E6  D3 CD                    OUT PORT1B    A START COMMAND HAS BEEN SENT TO ADC'S
261  08E8  CD 63 0B                 CALL ADBUS    HANG IN LOOP UNTIN ADC READY
262  08EB  06 01                    MVI B,NRDG    LOAD THE NUMBER OF READINGS INTO B
263  08ED  0E 05                    MVI C,NCHA    LOAD THE NUMBER OF CHANNELS INTO C
264  08EF  CD 20 0B                 CALL SMDAT    SUM THE READINGS
265  08F2  3E 88                    MVI A,PRT10   SET PORT1 FOR OUTPUT
266  08F4  D3 CF                    OUT PORT1
267  08F6  CD 78 0B                 CALL HEXDEC   CONVERT HEX TO DECIMAL
268  08F9  21 10 3C                 LXI H,RAWDA   LOAD STARTING ADDRESS OF RAW DAT
269  08FC  11 25 3C                 LXI D,ASCDAT  LOAD START ADDR OF ASCII DAT
270  08FF  0E 14                    MVI C,4*NCHA  16 BYTES OF DATA TO BE CONVERTED
```

```
271  0901  CD 95 08                    CALL BASK        HEX DAT TO ASCII DAT
272  0904  21 53 3C                    LXI H,DATIN      LOAD ADDRESS OF FIRST BYTE OF INPUT DATA
273  0907  22 4E 3C                    SHLD DIAD        STORE IN RAM,RESET COUNTER FOR NEXT READ
274  090A  3E FF                       MVI A,0FFH       SET ACCUMULATOR TO 1111/1111
275  090C  D3 D5                       OUT PORT2B       TURN ALL SAMPLE & HOLD CHIPS TO SAMPLE
276  090E  32 52 3C                    STA SHDAT        RESET SAMPLE AND HOLD DATA
277  0911  21 25 3C                    LXI H,ASCDAT     LOAD ADDRESS OF DATA TO BE SENT
278  0914  0E 28                       MVI C,8*NCHA       NUMBER OF CHARACTERS TO BE SENT
279  0916  3E 13                       MVI A,013H       ENABLE READ,WRITE OR EOI
280  0918  D3 D9                       OUT INTE1        SEND TO 8291 INT ENA
281  091A  FB                          EI               ENABLE INTERUPTS ON 8080
282  091B  76                          HLT              WAIT FOR INTERUPT FROM 8291 OVER GPIB
283                             ;
284                             ;      CMD1             LOOP FOR FAST READINGS WITH NO
285                             ;                       STORAGE IN THE COMP9B MICROPUTER
286                             ;      CMD2             SAME AS CMD1 EXCEPT CMD2 TAKES NEXT
287                             ;                       READING WHILE THE PREVIOUS ONE IS
288                             ;                       BEING PROCESSED.
289                             ;
290  091C  00                   CMD1   NOP              BOTH PROGS ARE SAME EXCEPT FOR ADBUSY
291  091D  3E 01                CMD2   MVI A,01         SET FOR READ
292  091F  D3 D4                       OUT PORT2A       SET RELAY FOR READ
293  0921  CD 63 0B                    CALL ADBUS       DON'T START PULSE IF WE ARE STILL READING
294  0924  3E 05                       MVI A,NCHA       SET THE NUMBER OF S&H CHANNELS IN RAM
295  0926  32 51 3C                    STA CNTSH        STORE
296  0929  3A 54 3C                    LDA DATIN+1      LOAD VALUE OF SECOND BYTE TRANSMITTED
297  092C  D6 20                       SUI 020H         CONVERT CHR TO DELAY;"!:=LONGEST,""=SHORT
298  092E  47                          MOV B,A          STORE IN B REG
299  092F  0E 00                       MVI C,0H         ZERO C REG
300  0931  3E 80                       MVI A,080H       SET HI ORDER BYTE HI(1000/0000)
301  0933  D3 D6                       OUT PORT2C       SEND PULSE OUT
302  0935  C5                   LOPSH2 PUSH B           TEMP STORE ON STACK
303  0936  CD 6B 0B                    CALL DELAY       LEAVE PULSE ON FOR A TIME .640/BC
304  0939  CD 9B 0B                    CALL SHLD        SEND SAMPLE AND HOLD HI (28.5 USEC)
305  093C  C1                          POP B            RESTORE BC DELAY VALUE
306  093D  3A 51 3C                    LDA CNTSH        LOOK AT COUNTER VALUE
307  0940  3D                          DCR A            DECREMENT COUNTER
308  0941  32 51 3C                    STA CNTSH        RESTORE COUNTER
309  0944  C2 35 09                    JNZ LOPSH2       STAY IN LOOP UNTIL ALL S/H'S ON
310  0947  AF                          XRA A            ZERO ACCUMULATOR
311  0948  D3 D6                       OUT PORT2C       SEND OUT TO PORT2C, TURN PULSE OFF
312  094A  3E 99                       MVI A,PRT1I
313  094C  D3 CF                       OUT PORT1        SET PORT1 FOR INPUT
314                             ;      LXI B,0A0H       LOAD 4 MS DELAY IN BC FOR NOW
315                             ;      CALL DELAY       TEMPRARY DELAY TO ALLOW DATA TO SETTLE
316                             ;      THIS CAN PROBABLY BE REMOVED AND REPLACED WITH COMPUTATIONS
317  094E  3E B0                       MVI A,0B0H       STROBE ADC 1011/0000,BIT B5 HI
318  0950  D3 CD                       OUT PORT1B
319  0952  3E 90                       MVI A,090H       THEN STROBE 1001/0000,BIT B5 LO
320  0954  D3 CD                       OUT PORT1B       A START COMMAND HAS BEEN SENT TO ADC'S
321  0956  21 39 3C                    LXI H,ASCDAT+4*NCHA:LOAD HL FOR DATA COUNTER VALUE
322  0959  36 0D                       MVI M,0DH        LOAD CARRIAGE RETURN INTO MEMORY AT END OF
323  095B  2B                          DCX H            DECREMENT FOR NEXT DATA BYTE
324  095C  22 4E 3C                    SHLD DIAD        STORE DATA COUNT ADDRESS IN RAM
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    7

    325  095F  3A 53 3C                        LDA DATIN          TEST FOR CMD1 OR CMD2
    326  0962  FE 31                           CPI 031H           SEE IF CMD1 WAS TRANSMITTED
    327  0964  CC 63 0B                        CZ ADBUS           WAIT FOR RDG TO END BEFORE DATA IS READ
    328  0967  C3 9C 0A                        JMP RDGOUT         READ ADC'S,SET UP DATA FOR TRANSMITT,HALT
    329                              ;
    330                              ;         CMD3               LOOP FOR FAST CALIBRATION READINGS WITH NO
    331                              ;                            STORAGE IN THE COMP9B MICROPUTER
    332                              ;
    333  096A  3E 00                 CMD3      MVI A,00           SET FOR CALIBRATE
    334  096C  D3 D4                           OUT PORT2A         SET RELAY FOR CALIBRATE
    335  096E  CD 63 0B                        CALL ADBUS         DON'T START PULSE IF WE ARE STILL READING
    336  0971  3E 05                           MVI A,NCHA         SET THE NUMBER OF S&H CHANNELS IN RAM
    337  0973  32 51 3C                        STA CNTSH          STORE
    338  0976  3E 80                           MVI A,080H         SET HI ORDER BYTE HI(1000/0000)
    339  0978  D3 D6                           OUT PORT2C         SEND PULSE OUT
    340  097A  3A 54 3C                        LDA DATIN+1        LOAD VALUE OF SECOND BYTE TRANSMITTED
    341  097D  D6 20                           SUI 020H           CONVERT CHR TO DELAY;"!:=LONGEST,""=SHORT
    342  097F  47                              MOV B,A            STORE IN B REG
    343  0980  0E 00                           MVI C,0H           ZERO C REG
    344  0982  C5                 LOPSH3    PUSH B               TEMP STORE ON STACK
    345  0983  CD 6B 0B                        CALL DELAY         DELAY BY BC
    346  0986  CD 9B 0B                        CALL SHLD          SEND SAMPLE AND HOLD HI
    347  0989  C1                              POP B              RESTORE BC DELAY VALUE
    348  098A  3A 51 3C                        LDA CNTSH          LOOK AT COUNTER VALUE
    349  098D  3D                              DCR A              DECREMENT COUNTER
    350  098E  32 51 3C                        STA CNTSH          RESTORE COUNTER
    351  0991  C2 82 09                        JNZ LOPSH3         STAY IN LOOP UNTIL ALL S/H'S ON
    352  0994  AF                              XRA A              ZERO ACCUMULATOR
    353  0995  D3 D6                           OUT PORT2C         SEND OUT TO PORT 2C
    354  0997  3E 99                           MVI A,PRT1I
    355  0999  D3 CF                           OUT PORT1          SET PORT1 FOR INPUT
    356                              ;         LXI B,0A0H         LOAD 4 MS DELAY IN BC FOR NOW
    357                              ;         CALL DELAY         TEMPRARY DELAY TO ALLOW DATA TO SETTLE
    358                              ;         THIS CAN PROBABLY BE REMOVED AND REPLACED WITH COMPUTATIONS
    359  099B  3E B0                           MVI A,0B0H         STROBE ADC 1011/0000,BIT B5 HI
    360  099D  D3 CD                           OUT PORT1B
    361  099F  3E 90                           MVI A,090H         THEN STROBE 1001/0000,BIT B5 LO
    362  09A1  D3 CD                           OUT PORT1B         A START COMMAND HAS BEEN SENT TO ADC'S
    363  09A3  21 39 3C                        LXI H,ASCDAT+4*NCHA;LOAD HL FOR DATA COUNTER VALUE
    364  09A6  36 0D                           MVI M,0DH          LOAD CARRIAGE RETURN INTO MEMORY AT END OF
    365  09A8  2B                              DCX H              DECREMENT FOR NEXT DATA BYTE
    366  09A9  22 4E 3C                        SHLD DIAD          STORE DATA COUNT ADDRESS IN RAM
    367  09AC  CC 63 0B                        CZ ADBUS           WAIT FOR RDG TO END BEFORE DATA IS READ
    368  09AF  C3 9C 0A                        JMP RDGOUT         READ ADC'S,SET UP DATA FOR TRANSMITT,HALT
    369                              ;
    370                              ;         CMD4               FAST READING LOOP,NO STORAGE IN
    371                              ;                            COMP9B MICRO,RDGS MADE ON TURN-OFF
    372                              ;         CMD5               SAME AS CMD4 EXCEPT CMD5 TAKES NEXT
    373                              ;                            READING WHILE THE PREVIOUS ONE IS
    374                              ;                            BEING PROCESSED.
    375                              ;
    376  09B2  00                    CMD4      NOP                BOTH PROGS ARE SAME EXCEPT FOR ADBUSY
    377  09B3  3E 01                 CMD5      MVI A,01           SET FOR READ
    378  09B5  D3 D4                           OUT PORT2A         SET RELAY FOR READ
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE    8

     379  09B7  CD 63 0B                        CALL ADBUS        DON'T START PULSE IF WE ARE STILL READING
     380  09BA  3E 05                           MVI A,NCHA        SET THE NUMBER OF S&H CHANNELS IN RAM
     381  09BC  32 51 3C                        STA CNTSH         STORE
     382  09BF  3A 54 3C                        LDA DATIN+1       LOAD VALUE OF SECOND BYTE TRANSMITTED
     383  09C2  D6 20                           SUI 020H          CONVERT CHR TO DELAY;"!:=LONGEST,""=SHORT
     384  09C4  47                              MOV B,A           STORE IN B REG
     385  09C5  0E 00                           MVI C,0H          ZERO C REG
     386  09C7  3E 80                           MVI A,080H        SET HI ORDER BYTE HI(1000/0000)
     387  09C9  D3 D6                           OUT PORT2C        SEND PULSE OUT
     388  09CB  CD 6B 0B                        CALL DELAY        LEAVE PULSE ON FOR A TIME .640/BC
     389  09CE  AF                              XRA A             ZERO ACCUMULATOR
     390  09CF  D3 D6                           OUT PORT2C        SEND OUT TO PORT2C, TURN PULSE OFF
     391  09D1  3A 55 3C                        LDA DATIN+2       LOAD VALUE OF THIRD BYTE TRANSMITTED
     392  09D4  D6 20                           SUI 020H          CONVERT CHR TO DELAY;"!:=LONGEST,""=SHORT
     393  09D6  47                              MOV B,A           STORE IN B REG
     394  09D7  0E 00                           MVI C,0H          ZERO C REG
     395  09D9  C5                   LOPSH4     PUSH B            TEMP STORE ON STACK
     396  09DA  CD 6B 0B                        CALL DELAY        LEAVE PULSE ON FOR A TIME .640/BC
     397  09DD  CD 9B 0B                        CALL SHLD         SEND SAMPLE AND HOLD HI (28.5 USEC)
     398  09E0  C1                              POP B             RESTORE BC DELAY VALUE
     399  09E1  3A 51 3C                        LDA CNTSH         LOOK AT COUNTER VALUE
     400  09E4  3D                              DCR A             DECREMENT COUNTER
     401  09E5  32 51 3C                        STA CNTSH         RESTORE COUNTER
     402  09E8  C2 D9 09                        JNZ LOPSH4        STAY IN LOOP UNTIL ALL S/H'S ON
     403  09EB  3E 99                           MVI A,PRTII
     404  09ED  D3 CF                           OUT PORT1         SET PORT1 FOR INPUT
     405                            ;           LXI B,0A0H        LOAD 4 MS DELAY IN BC FOR NOW
     406                            ;           CALL DELAY        TEMPRARY DELAY TO ALLOW DATA TO SETTLE
     407                            ;           THIS CAN PROBABLY BE REMOVED AND REPLACED WITH COMPUTATIONS
     408  09EF  3E B0                           MVI A,0B0H        STROBE ADC 1011/0000,BIT B5 HI
     409  09F1  D3 CD                           OUT PORT1B
     410  09F3  3E 90                           MVI A,090H        THEN STROBE 1001/0000,BIT B5 LO
     411  09F5  D3 CD                           OUT PORT1B        A START COMMAND HAS BEEN SENT TO ADC'S
     412  09F7  21 39 3C                        LXI H,ASCDAT+4*NCHA;LOAD HL FOR DATA COUNTER VALUE
     413  09FA  36 0D                           MVI M,0DH         LOAD CARRIAGE RETURN INTO MEMORY AT END OF
     414  09FC  2B                              DCX H             DECREMENT FOR NEXT DATA BYTE
     415  09FD  22 4E 3C                        SHLD DIAD         STORE DATA COUNT ADDRESS IN RAM
     416  0A00  3A 53 3C                        LDA DATIN         TEST FOR CMD1 OR CMD2
     417  0A03  FE 34                           CPI 034H          SEE IF CMD4 WAS TRANSMITTED
     418  0A05  CC 63 0B                        CZ ADBUS          WAIT FOR RDG TO END BEFORE DATA IS READ
     419  0A09  C3 9C 0A                        JMP RDGOUT        READ ADC'S ,SET DATA UP FOR TRANSMISSION,H
     420                            ;
     421                            ;           CMD6              FAST PULSE AND READING-NO DELAYS
     422                            ;                             READINGS INSIDE PULSE,63 U-SEC
     423                            ;
     424  0A0B  3E 01                CMD6        MVI A,01          SET FOR READ
     425  0A0D  D3 D4                           OUT PORT2A        SET RELAY FOR READ
     426  0A0F  CD 63 0B                        CALL ADBUS        DON'T START PULSE IF WE ARE STILL READING
     427  0A12  3E 05                           MVI A,NCHA        SET THE NUMBER OF S&H CHANNELS IN RAM
     428  0A14  32 51 3C                        STA CNTSH         STORE
     429  0A17  06 FE                           MVI B,0FEH        STORE 1111/1110 IN B REGISTER
     430  0A19  3E 80                           MVI A,080H        SET HI ORDER BYTE HI(1000/0000)
     431  0A1B  D3 D6                           OUT PORT2C        SEND PULSE OUT
     432  0A1D  AF                              XRA A             ZERO ACCUMULATOR
```

```
 433  0A1E  80                          ADD B          1111/1110 IN A
 434  0A1F  D3 D5                       OUT PORT2B     HOLD 1ST TRACK AND HOLD
 435  0A21  87                          ADD A          1111/1100 IN A
 436  0A22  D3 D5                       OUT PORT2B     HOLD 2ND TRACK AND HOLD
 437  0A24  87                          ADD A          1111/1000 IN A
 438  0A25  D3 D5                       OUT PORT2B     HOLD 3RD TRACK AND HOLD
 439  0A27  87                          ADD A          1111/0000 IN A
 440  0A28  D3 D5                       OUT PORT2B     HOLD 4TH TRACK AND HOLD
 441  0A2A  87                          ADD A          1110/0000 IN A
 442  0A2B  D3 D5                       OUT PORT2B     HOLD 5TH TRACK AND HOLD
 443  0A2D  87                          ADD A          1100/000 IN A
 444  0A2E  D3 D5                       OUT PORT2B     HOLD 6TH TRACK AND HOLD
 445  0A30  87                          ADD A          1000/000 IN A
 446  0A31  D3 D5                       OUT PORT2B     HOLD 7TH TRACK AND HOLD
 447  0A33  87                          ADD A          0000/0000 IN A
 448  0A34  D3 D5                       OUT PORT2B     HOLD 8TH TRACK AND HOLD
 449  0A36  AF                          XRA A          ZERO ACCUMULATOR
 450  0A37  D3 D6                       OUT PORT2C     SEND OUT TO PORT2C, TURN PULSE OFF
 451  0A39  3E 99                       MVI A,PRT1I
 452  0A3B  D3 CF                       OUT PORT1      SET PORT1 FOR INPUT
 453  0A3D  3E B0                       MVI A,0B0H     STROBE ADC 1011/0000,BIT B5 HI
 454  0A3F  D3 CD                       OUT PORT1B
 455  0A41  3E 90                       MVI A,090H     THEN STROBE 1001/0000,BIT B5 LO
 456  0A43  D3 CD                       OUT PORT1B     A START COMMAND HAS BEEN SENT TO ADC'S
 457  0A45  21 39 3C                    LXI H,ASCDAT+4*NCHA;LOAD HL FOR DATA COUNTER VALUE
 458  0A48  36 0D                       MVI M,0DH      LOAD CARRIAGE RETURN INTO MEMORY AT END OF
 459  0A4A  2B                          DCX H          DECREMENT FOR NEXT DATA BYTE
 460  0A4B  22 4E 3C                    SHLD DIAD      STORE DATA COUNT ADDRESS IN RAM
 461  0A4E  CD 63 0B                    CALL ADBUS     WAIT FOR RDG TO END BEFORE ADC'S ARE READ
 462  0A51  C3 9C 0A                    JMP RDGOUT     READ ADC'S ,SET DATA UP FOR TRANSMISSION,H
 463                            ;
 464                            ;       CMD7           FAST PULSE AND READING-NO DELAYS
 465                            ;                      READINGS AFTER  PULSE,7 U-SEC EACH
 466                            ;
 467  0A54  3E 01               CMD7    MVI A,01       SET FOR READ
 468  0A56  D3 D4                       OUT PORT2A     SET RELAY FOR READ
 469  0A58  CD 63 0B                    CALL ADBUS     DON'T START PULSE IF WE ARE STILL READING
 470  0A5B  3E 05                       MVI A,NCHA     SET THE NUMBER OF S&H CHANNELS IN RAM
 471  0A5D  32 51 3C                    STA CNTSH      STORE
 472  0A60  06 FE                       MVI B,0FEH     STORE 1111/1110 IN B REGISTER
 473  0A62  3E 80                       MVI A,080H     SET HI ORDER BYTE HI(1000/0000)
 474  0A64  D3 D6                       OUT PORT2C     SEND PULSE OUT
 475  0A66  AF                          XRA A          ZERO ACCUMULATOR
 476  0A67  D3 D6                       OUT PORT2C     SEND OUT TO PORT2C, TURN PULSE OFF
 477  0A69  80                          ADD B          1111/1110 IN A
 478  0A6A  D3 D5                       OUT PORT2B     HOLD 1ST TRACK AND HOLD
 479  0A6C  87                          ADD A          1111/1100 IN A
 480  0A6D  D3 D5                       OUT PORT2B     HOLD 2ND TRACK AND HOLD
 481  0A6F  87                          ADD A          1111/1000 IN A
 482  0A70  D3 D5                       OUT PORT2B     HOLD 3RD TRACK AND HOLD
 483  0A72  87                          ADD A          1111/0000 IN A
 484  0A73  D3 D5                       OUT PORT2B     HOLD 4TH TRACK AND HOLD
 485  0A75  87                          ADD A          1110/0000 IN A
 486  0A76  D3 D5                       OUT PORT2B     HOLD 5TH TRACK AND HOLD
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE   10

     487  0A78  87                              ADD A              1100/000 IN A
     488  0A79  D3 D5                           OUT PORT2B         HOLD 6TH TRACK AND HOLD
     489  0A7B  87                              ADD A              1000/000 IN A
     490  0A7C  D3 D5                           OUT PORT2B         HOLD 7TH TRACK AND HOLD
     491  0A7E  87                              ADD A              0000/0000 IN A
     492  0A7F  D3 D5                           OUT PORT2B         HOLD 8TH TRACK AND HOLD
     493  0A81  3E 99                           MVI A,PRT1I
     494  0A83  D3 CF                           OUT PORT1          SET PORT1 FOR INPUT
     495  0A85  3E B0                           MVI A,0B0H         STROBE ADC 1011/0000,BIT B5 HI
     496  0A87  D3 CD                           OUT PORT1B
     497  0A89  3E 90                           MVI A,090H         THEN STROBE 1001/0000,BIT B5 LO
     498  0A8B  D3 CD                           OUT PORT1B         A START COMMAND HAS BEEN SENT TO ADC'S
     499  0A8D  21 39 3C                        LXI H,ASCDAT+4*NCHA;LOAD HL FOR DATA COUNTER VALUE
     500  0A90  36 0D                           MVI M,0DH          LOAD CARRIAGE RETURN INTO MEMORY AT END OF
     501  0A92  2B                              DCX H              DECREMENT FOR NEXT DATA BYTE
     502  0A93  22 4E 3C                        SHLD DIAD          STORE DATA COUNT ADDRESS IN RAM
     503  0A96  CD 63 0B                        CALL ADBUS         WAIT FOR RDG TO END BEFORE ADC'S ARE READ
     504  0A99  C3 9C 0A                        JMP RDGOUT         READ ADC'S ,SET DATA UP FOR TRANSMISSION,H
     505                            ;
     506                            ;             RDGOUT          READ EACH ADC AND CONVERT TO ASCII REP OF
     507                            ;                             SET THE SYSTEM FOR OUTPUT OVER THE BUSS
     508  0A9C  3E 05               RDGOUT       MVI A,NCHA         LOAD CHANNEL COUNT
     509  0A9E  32 50 3C                          STA NCH            STORE IN RAM,CH NO GOES FROM 0 TO 7
     510  0AA1  C6 97               ADRDLP       ADI 097H           CH SEL NO +CD'S ON 8304,SEL ON ADC
     511  0AA3  D3 CD                           OUT PORT1B         SET CH. SWITCH TO CH. NO.
     512  0AA5  DB CE                           IN PORT1C          HIGH ORDER BYTE IS BROUGHT IN.
     513  0AA7  E6 0F                           ANI 0FH            4 HIGHEST ORDER BITS ARE DUMPED
     514  0AA9  07                              RLC                ROTATE LEFT,DOUBLE VALUE,ADDR ARE 2 BYTES
     515  0AAA  5F                              MOV E,A            STORE IN E TEMPERARILY
     516  0AAB  DB CC                           IN PORT1A          LOW ORDER BYTE IS BROUGHT IN AND
     517  0AAD  E6 F0                           ANI 0F0H           DUMP LOW NIBBLE
     518  0AAF  0F                              RRC                ROTATE RIGHT 3 TIMES
     519  0AB0  0F                              RRC                RESULT IS 2ND NIBBLE DOUBLED
     520  0AB1  0F                              RRC                ADDRESSES OF VALUES ARE 2 BYTES WIDE
     521  0AB2  4F                              MOV C,A            STORE IN C
     522  0AB3  06 00                           MVI B,0            BC CONTAIN OFFSET ADDR FOR 2ND NIBBLE
     523  0AB5  21 A7 0B                        LXI H,NIB2B        LOAD BASE ADDRESS FOR 2ND NIBBLE
     524  0AB8  09                              DAD B              ADD OFFSET TO HL ADDRESS
     525  0AB9  4B                              MOV C,E            BC NOW CONTAINS OFFSET FOR 3RD NIBBLE
     526  0ABA  DB CC                           IN PORT1A          LOW ORDER BYTE IS ENTERED AGAIN
     527  0ABC  E6 0F                           ANI 0FH            DUMP 2ND NIBBLE,HI 4 BITS
     528  0ABE  27                              DAA                DECIMAL ADJUST ACCUMULATOR
     529  0ABF  86                              ADD M              ADD 2ND NIBBLE DECIMAL VALUE FROM MEMORY
     530  0AC0  27                              DAA                DECIMAL ADJUST ACCUMULATOR
     531  0AC1  5F                              MOV E,A            STORE SUM IN DE
     532  0AC2  23                              INX H              HI BYTE OF 2ND NIBBLE
     533  0AC3  3E 00                           MVI A,0            ZERO A
     534  0AC5  8E                              ADC M              ADD HI BYTE OF DEC VAL OF 2ND NIB
     535  0AC6  57                              MOV D,A            DE CONTAINS SUM
     536  0AC7  21 C7 0B                        LXI H,NIB3B        BASE ADDRESS FOR 3RD NIBBLE
     537  0ACA  09                              DAD B              ADD OFFSET POINTER FOR DEC VAL OF 3RD NIB
     538  0ACB  7B                              MOV A,E            BRING SUM OF 1ST 2 NIBS BACK IN
     539  0ACC  86                              ADD M              ADD LO BYTE OF 3RD
     540  0ACD  27                              DAA                DECIMAL ADJUST
```

```
541  0ACE  5F                    MOV E,A          STORE BACK IN DE
542  0ACF  23                    INX H            INCREMENT TO HI BYTE OF DEC VAL OF 3RD NIB
543  0AD0  7A                    MOV A,D          BRING IN 2ND BYTE OF SUM OF 1ST 2 NIBS
544  0AD1  8E                    ADC M            ADD HI BYTE OF 3RD NIB
545  0AD2  27                    DAA              DECIMAL ADJUST
546  0AD3  57                    MOV D,A          STORE SUM IN DE
547  0AD4  2A 4E 3C              LHLD DIAD        LOAD ADDRESS FOR ASCII DATA IN HL
548  0AD7  7B                    MOV A,E          LOAD LOW ORDER BYTE FROM DE
549  0AD8  E6 0F                 ANI 0FH          DUMP 4 HI BYTES OF LOW ORDER BYTE
550  0ADA  C6 30                 ADI 30H          CONVERT TO ASCII
551  0ADC  77                    MOV M,A          STORE IN RAM
552  0ADD  2B                    DCX H            ADDRESS NEXT MEMORY BYTE
553  0ADE  7B                    MOV A,E          BRING LO BYTE BACK
554  0ADF  E6 F0                 ANI 0F0H         DUMP LOW 4 BITS
555  0AE1  1F                    RAR
556  0AE2  1F                    RAR              ROTATE TO LOW 4 BIT POSITION
557  0AE3  1F                    RAR
558  0AE4  1F                    RAR
559  0AE5  C6 30                 ADI 30H          CONVERT TO ASCII
560  0AE7  77                    MOV M,A          STORE IN RAM
561  0AE8  2B                    DCX H            ADDRESS NEXT MEMORY BYTE
562  0AE9  7A                    MOV A,D          LOAD HI ORDER BYTE FROM DE
563  0AEA  E6 0F                 ANI 0FH          DUMP 4 HI BITS OF HI ORDER BYTE
564  0AEC  C6 30                 ADI 30H          CONVERT TO ASCII
565  0AEE  77                    MOV M,A          STORE IN RAM
566  0AEF  2B                    DCX H            ADDRESS NEXT MEMORY BYTE
567  0AF0  7A                    MOV A,D          BRING HI BYTE BACK
568  0AF1  E6 F0                 ANI 0F0H         DUMP LOW 4 BITS
569  0AF3  1F                    RAR
570  0AF4  1F                    RAR              ROTATE TO LOW 4 BIT POSITION
571  0AF5  1F                    RAR
572  0AF6  1F                    RAR
573  0AF7  C6 30                 ADI 30H          CONVERT TO ASCII
574  0AF9  77                    MOV M,A          STORE IN RAM
575  0AFA  2B                    DCX H            DECREMENT ASCII MEMORY COUNTER
576  0AFB  22 4E 3C              SHLD DIAD        STORE BACK FOR NEXT LOOP
577  0AFE  3A 50 3C              LDA NCH          LOAD CHANNEL COUNT FROM RAM
578  0B01  3D                    DCR A            DECREMENT CHANNEL COUNT
579  0B02  32 50 3C              STA NCH          STORE IT BACK IN RAM
580  0B05  C2 A1 0A              JNZ ADRDLP       GO BACK TO ADC READINGS IF NOT DONE
581  0B08  21 53 3C              LXI H,DATIN      RESET INPUT DATA ADDRESS FOR NEXT WRITE
582  0B0B  22 4E 3C              SHLD DIAD        STORE ADDRESS IN RAM
583  0B0E  3E FF                 MVI A,0FFH       SET ACCUMULATOR TO 1111/1111
584  0B10  D3 D5                 OUT PORT2B       TURN ALL SAMPLE & HOLD CHIPS TO SAMPLE
585  0B12  32 52 3C              STA SHDAT        RESET SAMPLE AND HOLD DATA
586  0B15  21 25 3C              LXI H,ASCDAT     LOAD ADDRESS OF DATA TO BE SENT
587  0B18  0E 14                 MVI C,4*NCHA     NUMBER OF CHARACTERS TO BE SENT
588  0B1A  3E 13                 MVI A,013H       ENABLE READ,WRITE OR EOI
589  0B1C  D3 D9                 OUT INTE1        SEND TO 8291 INT ENA
590  0B1E  FB                    EI               ENABLE INTERUPTS ON 8080
591  0B1F  76                    HLT              WAIT FOR INTERUPT FROM 8291 OVER GPIB
592                         ;
593                         ;SMDAT - DATA SUMMATION ROUTINE,STARTING WITH CHANNEL 1,READS NO OF
594                                 ;CHANNELS IN C,WITH NO OF READINGS PER CHANNEL IN B (UP TO
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE   12

     595                                        ;255).SUM OF THE DATA WILL BE IN RAWDA -4 + (4*CH. NO.)
     596                                        ;RAW DATA IS STORED AS MAG1,PH1,MAG2,PH2,MAG3,PH3;4 BYTES EACH
     597  0B20  21 10 3C            SMDAT        LXI H,RAWDA        HL LOADED WITH LAST RAW DATA ADDRESS
     598  0B23  C5                               PUSH B             COPY OF REGS B,C STORED ON STACK
     599  0B24  79                               MOV A,C            NUMBER OF CHANNELS LOADED INTO A
     600  0B25  87                               ADD A
     601  0B26  87                               ADD A              4*NO OF CHANNELS IN A
     602  0B27  4F                               MOV C,A            4*NO OF CHANNELS IN REG C
     603  0B28  AF                               XRA A              A REGISTER ZEROED
     604  0B29  CD 00 00       E                CALL ZERO          ZERO RAM FROM HL TO HL+C
     605  0B2C  C1                               POP B              NO OF CHANNELS (REG C) & TIMES/CH (B) RES
     606  0B2D  11 04 00                         LXI D,0004H        LOAD +4 INTO DE
     607  0B30  21 13 3C            SMLOP        LXI H,RAWDA+03H    HL LOADED WITH FIRST RAW DATA ADDRESS
     608  0B33  C5                               PUSH B             AND COPY STORED ON STACK AGAIN
     609  0B34  3E 05               CHLOP        MVI A,NCHA         TOTAL NUMBER OF CHANNELS IN A
     610  0B36  91                               SUB C              NCH = TOTAL NO. - CHA COUNT IN REG C
     611  0B37  CD 4B 0B                         CALL ADRDG         MAKE AD CON RDG,ANSWER ON AP TOP OF STACK
     612  0B3A  CD 00 00       E                CALL DADDA         32 BIT FIXED ADD TO DATA AT HL,HL=HL-3
     613  0B3D  CD 00 00       E                CALL TOS4          AP TOP OF STACK STORED IN RAM AT HL
     614  0B40  19                               DAD D              DE (+4) ADDED TO HL,POINTS TO NEXT ADDR
     615  0B41  0D                               DCR C              CHANNEL COUNT DECREMENTED
     616  0B42  C2 34 0B                         JNZ CHLOP          JUMP BACK TO CHANNEL LOOP IF NOT FINISHED
     617  0B45  C1                               POP B              RESTORE BC FROM STACK
     618  0B46  05                               DCR B              DECREMENT THE NUMBER OF RDGS/CHANNEL
     619  0B47  C2 30 0B                         JNZ SMLOP          GO BACK TO THE SUM LOOP IF NOT FINISHED
     620  0B4A  C9                               RET
     621                            ;ADRDG       SELECTS THE ADCONVERTOR NUMBER THAT IS IN A,READS THE
     622                            ;            12 BITS FROM THE CONVERTOR,ADDS 20 ZEROS AND RETURNS WITH
     623                            ;            A POSITIVE 32 BIT FIXED NUMBER ON THE TOP OF THE APSTACK.
     624                            ;            FOR DATEL ADC-ET12B ANALOG TO DIGITAL CONVERTOR.
     625  0B4B  C6 98               ADRDG        ADI 098H           CH SEL NO +CD'S ON 0304,SEL ON ADC
     626  0B4D  D3 CD                             OUT PORT1B         SET CH. SWITCH TO CH. NO.
     627  0B4F  DB CC                             IN PORT1A          LOW ORDER BYTE IS BROUGHT IN AND
     628  0B51  D3 EE                             OUT APDATA        STORE ON ARITHMETIC PROCESSOR STACK.
     629  0B53  DB CE                             IN PORT1C          HIGH ORDER BYTE IS BROUGHT IN.
     630  0B55  E6 0F                             ANI 0FH            4 HIGHEST ORDER BITS ARE DUMPED
     631  0B57  D3 EE                             OUT APDATA        STORE ON ARITHMETIC PROCESSOR STACK.
     632  0B59  AF                               XRA A              ZERO ACCUMULATOR
     633  0B5A  D3 EE                             OUT APDATA        STORE ON ARITHMETIC PROCESSOR STACK.
     634  0B5C  D3 EE                             OUT APDATA        STORE ON ARITHMETIC PROCESSOR STACK.
     635  0B5E  3E 90                             MVI A,090H        LEAVE START LO,DESELECT ADCONVERTORS
     636  0B60  D3 CD                             OUT PORT1B        SEND TO PORT1B
     637  0B62  C9                               RET                RETURNS WITH POS. ANSWER ON AP STACK.
     638                            ;
     639                            ;            ADBUS             HANG HERE UNTIL ALL BUSY LINES OF THE DATEL
     640                            ;                            ADC-ET12B ANALOG TO DIGITAL CONVERTORS GOES LOW
     641                            ;                            DATA FROM THE LAST CONVERSION CAN BE READ FOR
     642                            ;                            THE NEXT 9 MILLISECONDS.
     643                            ;
     644  0B63  DB CE               ADBUS        IN PORT1C          CHECK PORT1C,BIT4 FOR BUSY (HI)
     645  0B65  E6 10                             ANI 010H           DUMP ALL BITS EXCEPT 1C4,/0001/0000/
     646  0B67  C2 63 0B                         JNZ ADBUS          STAY IN LOOP UNTIL BUSY LINE LOW
     647  0B6A  C9                               RET                START BIT WILL GO LO WHEN DATA CH READ
     648                            ;DELAY         - GENERATES A TIME DELAY.DEPENDING ON THE CONSTANT LOADED
```

```
649                                 ;                INTO BC. # CYCLES = 20*FFFF/(BC)  ABOUT 10 USEC/LOOP
650                                 ;            WITH 18.432 MHZ CLOCK,TIME=.640/BC SEC OR BC=.640/TIME(SEC)
651  0B6B  79                       DELAY        MOV A,C          MOVE CONTENTS OF C REG TO A
652  0B6C  B0                                    ADD B            CHECK REG BC FOR ZERO
653  0B6D  C8                                    RZ               RETURN IF BC ZERO
654  0B6E  E5                                    PUSH H           SAVE THE CONTENTS OF HL AND
655  0B6F  21 00 00                              LXI H,0H         ZERO HL
656  0B72  09                       LOPSM        DAD B            ADD BC TO THE CONTENTS OF HL
657  0B73  D2 72 0B                              JNC LOPSM        STAY IN LOOP UNTIL HL OVERFLOWS.
658  0B76  E1                                    POP H            THEIR ORIGINAL VALUES
659  0B77  C9                                    RET              AND RETURN.
660                                 ;
661                                 ;            HEXDEC           PROGRAM CONVERTS FIXED HEX TO DECIMAL,12 B
662                                 ;            THIS IS NOT NEEDED IF THE HOST COMPUTER CAN READ BINARY
663                                 ;            THE OLD HEX DATA IS OVER WRITTEN WITH BCD DATA
664                                 ;
665  0B78  21 13 3C                 HEXDEC       LXI H,RAWDA+3    START WITH LO BYTE OF 1ST RAW DATA WRD
666  0B7B  0E 05                                 MVI C,NCHA       LOAD THE NUMBER OF CHANNELS TO BE CONVERTE
667  0B7D  C5                       CONLOP       PUSH B           SAVE A COPY OF THE CHANNEL COUNTER
668  0B7E  E5                                    PUSH H           SAVE A COPY OF HL
669  0B7F  CD 00 00      E                       CALL WRT4        WRITE THE NUMBER TO BE CONVERTED ON THE AP
670  0B82  16 04                                 MVI D,04         LOAD NUMBER OF BYTES TO BE CONVERTED
671  0B84  CD 00 00      E                       CALL BIDEC       CONVERT THE BINARY NO TO DECIMAL
672  0B87  E1                                    POP H            GET THE ADDRESS OF THE CONVERTED NO BACK
673  0B88  11 0A 3C                              LXI D,3C0AH      ADDRESS OF THE DECIMAL NUMBER
674  0B8B  1A                                    LDAX D           LOAD LO BYTE OF DECIMAL NUMBER
675  0B8C  77                                    MOV M,A          STORE IT OVER THE HEX NUMBER
676  0B8D  2B                                    DCX H            NEXT BYTE OF HEX NO
677  0B8E  13                                    INX D            NEXT BYTE OF DEC NO
678  0B8F  1A                                    LDAX D           LOAD NEXT BYTE OF DEC NO INTO A
679  0B90  77                                    MOV M,A          SEND IT TO MEMORY
680                                 ;            REPEAT TWICE MORE FOR LARGER NUMBERS
681  0B91  11 05 00                              LXI D,05         LOAD OFFSET ADDRESS FOR HL INTO DE
682  0B94  19                                    DAD D            HL ADDRESSES LO ORDER BYTE FOR NEXT CHANNE
683  0B95  C1                                    POP B            GET COUNTER BACK FROM STACK
684  0B96  0D                                    DCR C            DECREMENT THE COUNTER
685  0B97  C2 7D 0B                              JNZ CONLOP       LOOP BACK IF NOT DONE
686  0B9A  C9                                    RET              RETURN IF FINISHED
687                                 ;
688                                 ;            SHLD             PROGRAM TO RUN THE SAMPLE
689                                 ;                             AND HOLD CIRCUITS (28.5 USECS)
690                                 ;
691  0B9B  3A 52 3C                 SHLD         LDA SHDAT        LOAD SAMPLE AND HOLD DATA WORD
692  0B9E  E6 7F                                 ANI 07FH         SET HIGEST BIT TO ZERO,KEEP OTHERS SAME
693  0BA0  07                                    RLC              ROTATE ACCUMULATOR LEFT
694  0BA1  32 52 3C                              STA SHDAT        STORE FOR NEXT TIME
695  0BA4  D3 D5                                 OUT PORT2B       TURN NEXT S&H LOW,SWITCH TO HOLD
696  0BA6  C9                                    RET              GO BACK TO CALLER
697                                 ;
698  0BA7  00 00 16 00              NIB2B        DB 0,0,16H,0,32H,0,48H,0,64H,0,080H,0,096H,0,12H,1
699  0BAB  32 00 48 00
700  0BAF  64 00 80 00
701  0BB3  96 00 12 01
702  0BB7  28 01 44 01                           DB 28H,1,44H,1,60H,1,76H,1,092H,1,8,2,24H,2,40H,2
```

```
ERR LINE  ADDR  B1 B2 B3 B4        GPUL PROGRAM VERSION  5 AUGUST 85                          PAGE   14

    703  0BBB  60 01 76 01
    704  0BBF  92 01 08 02
    705  0BC3  24 02 40 02
    706  0BC7  00 00 56 02        NIB3B       DB 0,0,56H,2,12H,5,68H,7,24H,10H,080H,12H
    707  0BCB  12 05 68 07
    708  0BCF  24 10 80 12
    709  0BD3  36 15 92 17                    DB 36H,15H,092H,17H,048H,20H,4,23H,60H,25H
    710  0BD7  48 20 04 23
    711  0BDB  60 25
    712  0BDD  16 28 72 30                    DB 16H,28H,72H,30H,28H,33H,084H,35H,40H,38H
    713  0BE1  28 33 84 35
    714  0BE5  40 38
    715  0BE7                                 END

  ASSEMBLER ERRORS =    0
```

Appendix 7

CIRCUIT DIAGRAMS

The NDT COMP9B is a microcomputer developed at ORNL[1] around an Intel 8080 microprocessor. The schematic diagrams for the processor and memory boards are shown in Figs. A7-1 and A7-2. The microcomputer can control all the peripheral equipment, using instructions stored in the EPROM chips, or can act as an intermediary between another computer and the equipment. It instructs the pulser when to fire, controls the times at which the pulse amplitude measurements are made, and records the data; it can also control the probe positioner.

There are a number of the pulse switching circuit modules in parallel to handle the very large peak currents needed. Figure A7-3 shows a schematic diagram of one pulser module, and Fig. A7-4 shows a schematic diagram of the circuit to measure the pulse amplitude. There are eight sample-and-hold circuits that can measure the pulse amplitude from the pickup coil at as many as eight times, according to software-determined computer instructions. Actually, we have found that five different read times during a pulse seem to be sufficient at present. The analog-to-digital converter, shown schematically in Fig. A7-5, changes the sample-and-hold voltages to digital form for use by the computer.

REFERENCE

1. C. V. Dodd and R. F. Cowan, *The NDT-COMP9 Microcomputer*, ORNL/NUREG/TM-390, September 1980.

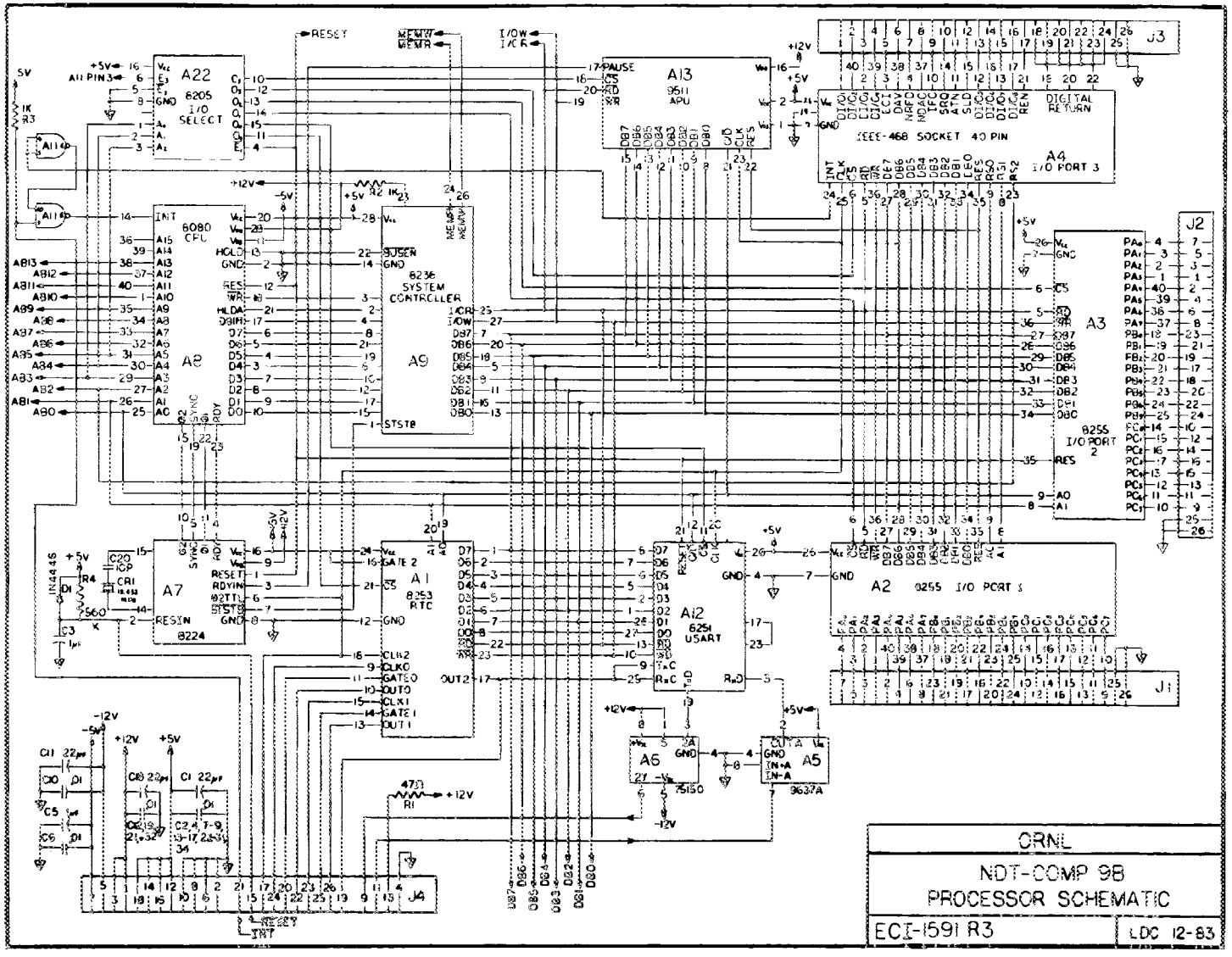


Fig. A7-1. Schematic diagram of NDT-COMP9B microcomputer processor board.

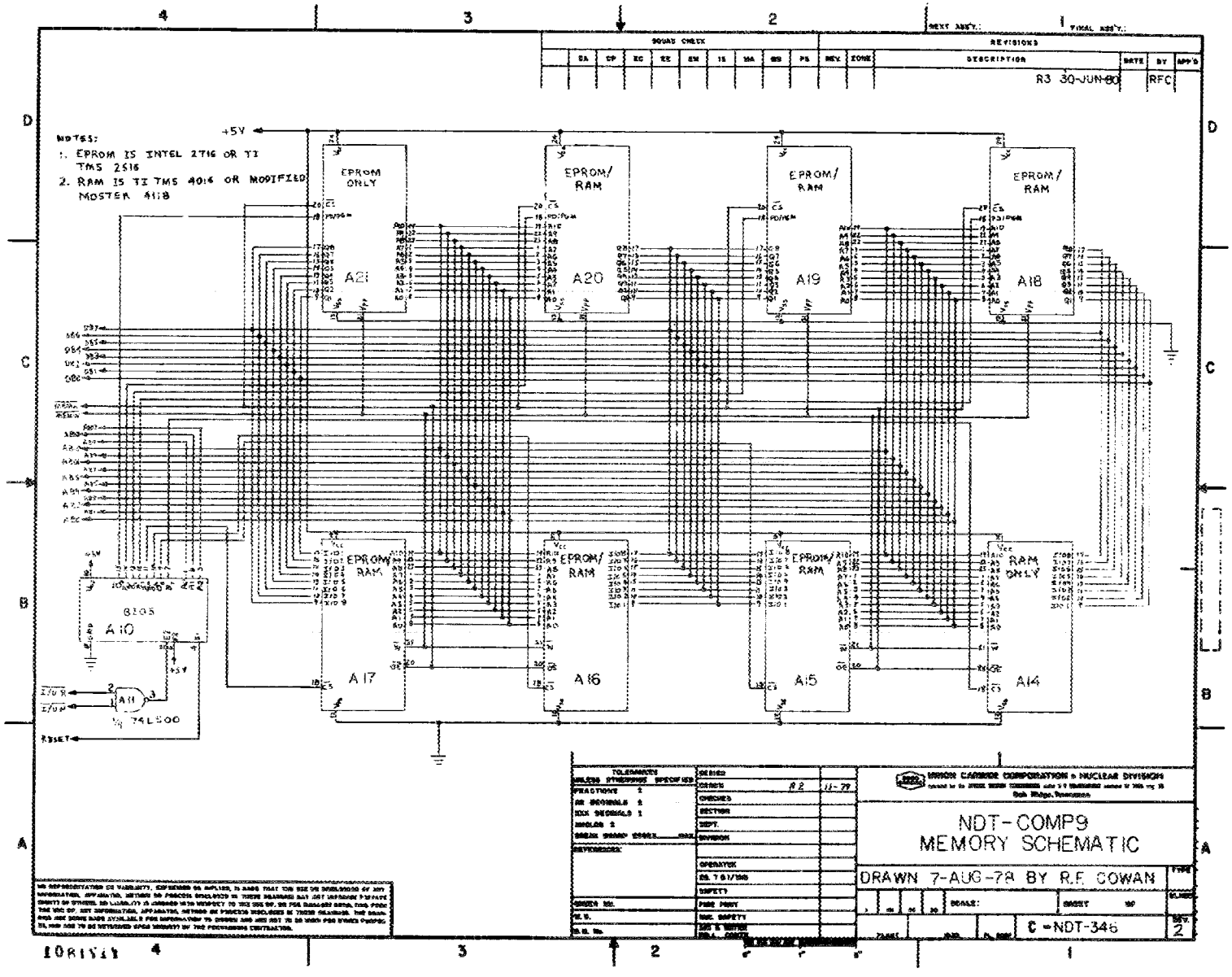


Fig. A7-2. Schematic diagram of NDT-COMP9B microcomputer memory board.

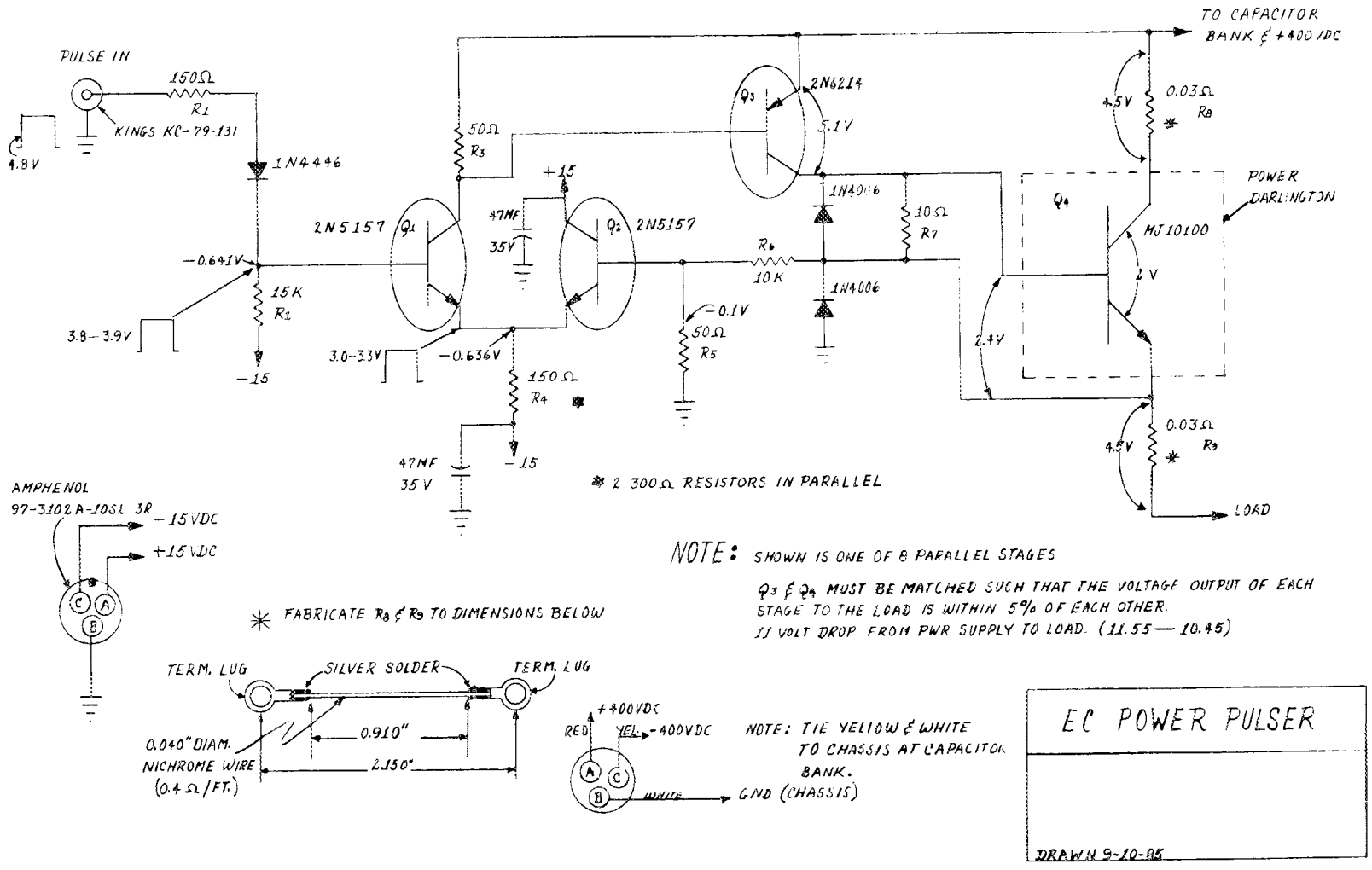


Fig. A7-3. Schematic diagram of a power-pulse switching circuit.

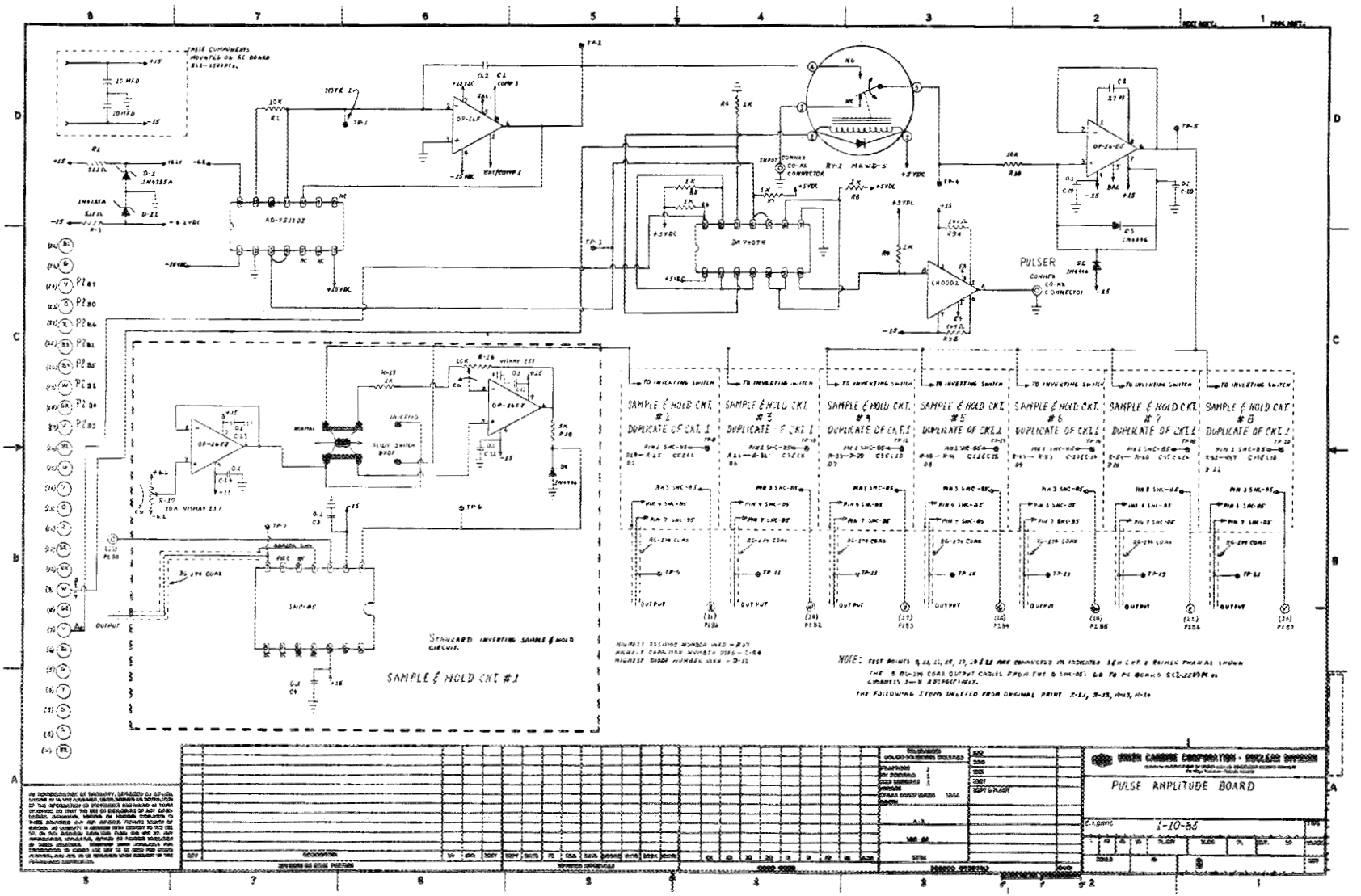


Fig. A7-4. Schematic diagram of the circuit to measure the pickup pulse amplitude at computer-controlled times.

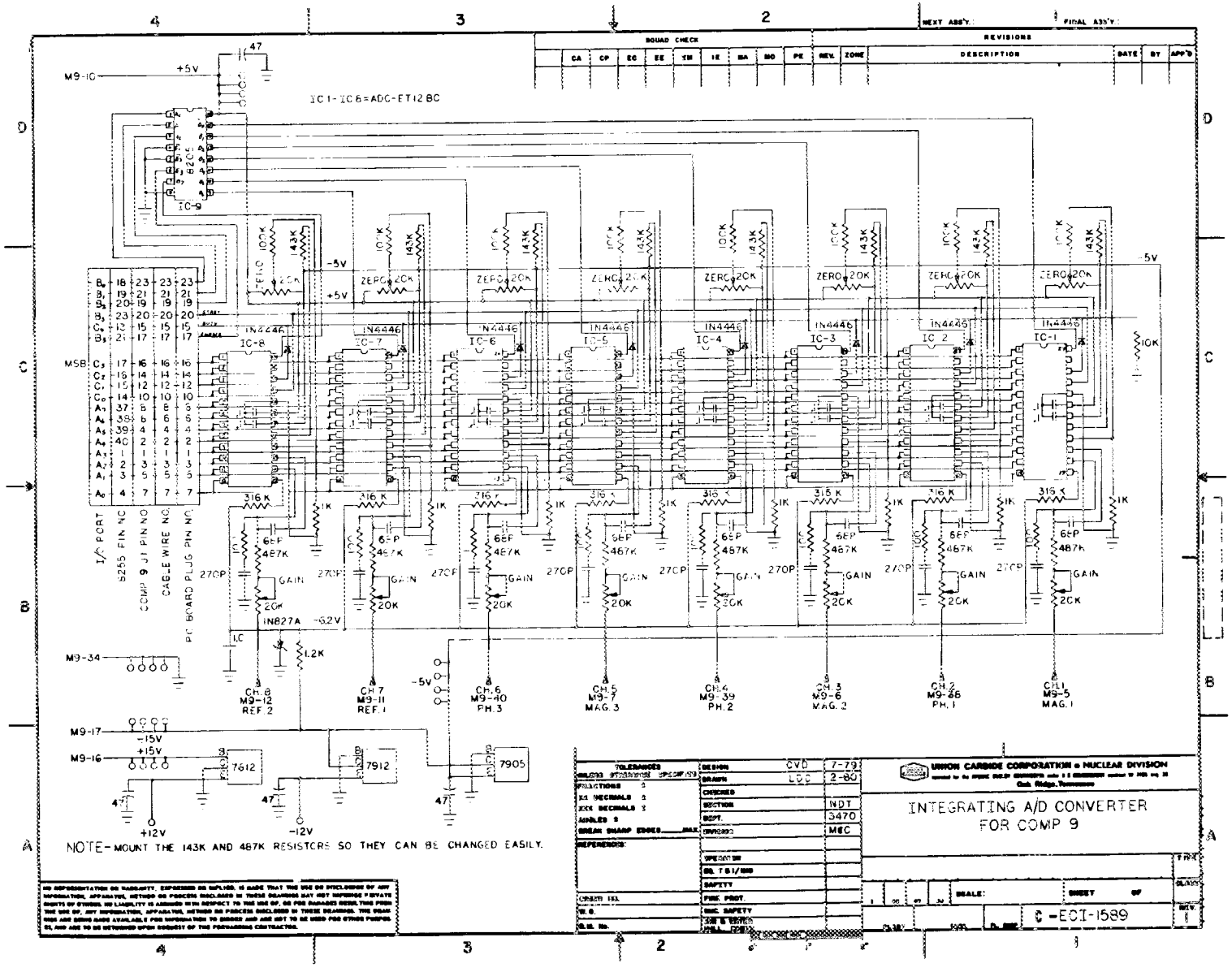


Fig. A7-5. Schematic diagram of the analog-to-digital converter circuit.

ORNL-6305
Distribution
Categories UC-79Th,
-Tk, -Tr

INTERNAL DISTRIBUTION

1-2. Central Research Library
3. Document Reference Section
4-5. Laboratory Records Department
6. Laboratory Records, ORNL RC
7. ORNL Patent Section
8-12. W. E. Deeds
13-17. C. V. Dodd
18-22. R. W. McClung
23. R. K. Nanstad
24. P. L. Rittenhouse
25. G. M. Slaughter
26. J. H. Smith
27-29. P. T. Thornton
30. J. R. Weir
31. R. J. Charles (Consultant)
32. G. Y. Chin (Consultant)
33. H. E. Cook (Consultant)
34. Alan Lawley (Consultant)
35. W. D. Nix (Consultant)
36. J. C. Williams (Consultant)

EXTERNAL DISTRIBUTION

37. ARGONNE NATIONAL LABORATORY, 9700 S. Cass Avenue, Argonne, IL 60439

O. K. Chopra

38. BABCOCK AND WILCOX COMPANY, Fossil Power Generation Division, 20 South Van Buren Avenue, Barberton, OH 44203

M. Gold

39. BURNS AND ROE, INC., 700 Kinderkamack Road, Oradell, NJ 07469

C. S. Ehrman

40. COMBUSTION ENGINEERING, INC., 911 W. Main Street, Chattanooga, TN 37402

D. A. Canonico

41-42. COMBUSTION ENGINEERING, INC., 1000 Prospect Hill Road, Windsor, CT 06095

P. E. C. Bryant
A. L. Gaines

43. ELECTRIC POWER RESEARCH INSTITUTE, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303

R. I. Jaffee

44. ELECTRIC POWER RESEARCH INSTITUTE-CONSOLIDATED MANAGEMENT ORGANIZATION, Suite 220, One Energy Center, Naperville, IL 60566

D. R. Riley

45. FOSTER WHEELER DEVELOPMENT CORPORATION, 12 Peach Tree Hill Road, Livinston, NJ 07039

W. R. Apblett

46. GA TECHNOLOGIES, INC., P.O. Box 86508, San Diego, CA 92138

D. I. Roberts

47–48. GENERAL ELECTRIC COMPANY, Nuclear Systems Technology Operations, 310 DeGuigne Drive, P.O. Box 3508, Sunnyvale, CA 94088

P. J. Ring
P. Roy

49. HANFORD ENGINEERING DEVELOPMENT LABORATORY, P.O. Box 1970, Richland, WA 99352

L. D. Blackburn

50–51. ROCKWELL INTERNATIONAL, Rocketdyne Division, 6633 Canoga Avenue, Canoga Park, CA 91304

T. L. Anderson
W. T. Lee

52. ROCKWELL INTERNATIONAL, Energy Technology Engineering Group, P.O. Box 1449, Canoga Park, CA 91304

H. C. Wieseneck

53. WESTINGHOUSE ELECTRIC CORPORATION, Advanced Energy Systems Division, P.O. Box 10864, Pittsburgh, PA 15236

R. W. Buckman

54–57. DOE, Washington, DC 20545

Office of Fusion Energy
T. C. Reuther

Office of Reactor Systems Development and Technology
C. C. Bigelow
R. J. Neuhold
B. J. Rock

58. DOE, OAK RIDGE OPERATIONS OFFICE, P.O. Box E, Oak Ridge, TN 37831

Office of Assistant Manager for Energy Research and Development

59–167. DOE, TECHNICAL INFORMATION CENTER, Office of Information Services P.O. Box 62, Oak Ridge, TN 37831

For distribution as shown in DOE/TIC-4500, Distribution Categories UC-79Th (Structural Materials and Design Engineering), UC-79Tk (Components), and UC-79Tr (Structural and Component Materials Development).