AAEC/E191

AAEC/E191

13

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

COMPUTER-CONTROLLED DIFFRACTOMETER

by

P.J. ELLIS
A.W. PRYOR

June 1968

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

COMPUTER-CONTROLLED DIFFRACTOMETER

by

P. J. ELLIS

A. W. PRYOR

ABSTRACT

A computer-controlled neutron diffractometer is described for the collection of Bragg intensities from a single crystal. The computer is a PDP-8 with a store of 4096 12-bit words. The diffractometer is a "4-circle" instrument with the axes driven by pulsed stepping-motors, while axis movement is checked by light-beam markers. The programs include procedures for centring reflections, rotating about the scattering vector, calculating the setting angles in the "bisecting" and "parallel" positions, scanning over diffraction peaks, measuring backgrounds and generating ranges of Miller indices. Two similar diffractometers may be operated simultaneously.

CONTENTS

(continued)

CONTENTS (continued)

(continued)

CONTENTS (continued)

1. INTRODUCTION

The study of crystal structures by neutron diffraction has been well-established at Lucas Heights for several years. Semi-automatic equipment has been developed and used for numerous projects. In this report we describe the development of computer-controlled instruments which will enable advance in this field of research to continue in a way which will be worthy of the sound basis previously established.

The guiding principle in this project has been economy. We wanted to enable the researchers to collect as much data as possible with a minimum investment of effort. We wanted to minimise the labour of data-handling and the computer-time involved in data-processing. Above all we wanted to maximise the usage of the neutron beams: to ensure that, whenever the reactor was on, the instruments were collecting data in the most efficient way possible. And we wanted to do this as cheaply as we could. A PDP-8 computer was available as the basis of the system. We tried to make the drive systems as cheaply as they could be made, and to economise as much as possible in writing the programs so as to get maximum value from the controlling ability of the computer.

To follow the details of this report it will be necessary for the reader to have some familiarity with the programming system and operating system of the PDP-8 computer.

2. GENERAL DESCRIPTION

2.1 Diffraction from a Crystal

A diffractometer is an instrument for observing the intensity of beams diffracted from a crystal bathed in radiation of some definite wavelength, λ. (In our case neutrons, with a wavelength in the range 0.8 to 2.0 Å). These intensities are commonly observed for somewhere between 100 and 2000 different diffracted beams, and the object of the whole experiment is to determine the disposition of atoms in the unit cell of the crystal. However we are concerned here only with the observation of the intensities, not with their purpose.

A crystal is described by its unit cell, defined by three vectors, $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$. The theory is conveniently formulated in terms of the reciprocal cell, defined by vectors, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ such that

$$\mathbf{a}_i \cdot \mathbf{b}_j = \delta_{ij} \quad .$$

Let $\mathbf{h}$ be a general vector in the lattice of reciprocal cells,

that is,

$$\mathbf{h} = h_1 \mathbf{b}_1 + h_2 \mathbf{b}_2 + h_3 \mathbf{b}_3 \quad .$$

Then if we define $\mathbf{k}_0$ as a vector of magnitude $\frac{2\pi}{\lambda}$ in the direction of the incident beam, and $\mathbf{k}'$ as a vector of magnitude $\frac{2\pi}{\lambda}$ in the direction of the diffracted beam, and if we define the scattering vector $\mathbf{Q} = \mathbf{k}_0 - \mathbf{k}'$, then Bragg's law for diffraction may be conveniently formulated:

$$\mathbf{Q} = \mathbf{h} \quad . \qquad (1)$$

This defines the necessary condition for the observation of a diffracted beam. The various diffracted beams are traditionally labelled by the components $h_1\ h_2\ h_3$ of $\mathbf{h}$. (These are the same as the Miller indices, usually symbolised by hkl, of the reflecting plane in the crystal).

Equation 1 shows that three conditions must be satisfied for diffraction to occur. The crystal must be correctly oriented so that $\mathbf{Q}$ and $\mathbf{h}$ are collinear. This requires two angular rotations which will depend only on the unit cell of the crystal and the Miller indices. Then the angle of reflection θ must be chosen so that $|Q| = |h|$. Note that $|Q| = 4\pi \sin \theta/\lambda$. The intensity of the diffracted beam is observed by rotating the crystal through a small angular range about the correct θ.

2.2 The Four-Circle Diffractometer and the Procedures in Measurement

In neutron diffraction, 3-dimensional intensity data are collected on a 4-circle diffractometer, in which the counter is set at an angle of 2θ, in the horizontal plane, and the crystal is oriented in three Eulerian angles ω, χ, ϕ. These angles are defined schematically in Figure 1. It is essential that the reader of this report be able to envisage clearly the mechanical arrangement and geometrical definitions of the four angles. (When the angle ω is independent of 2θ, as it is in the instrument we are describing, then the instrument is described as a 4-circle diffractometer. When the angle ω is mechanically constrained to be half the counter angle then the instrument is called a 3-circle diffractometer).

Since four degrees of freedom exist (the angles 2θ, ω, χ, ϕ) to satisfy the three diffraction conditions summarised by Equation 1, there is one excess degree of freedom. This permits rotation of the crystal about the scattering vector **Q** by an angle ψ, referred to as the azimuth angle. Note that there is no mechanical shaft for this ψ-rotation; it is achieved by complicated simultaneous adjustments to ω, χ and ϕ. (Rotation of the azimuth angle is not possible in a 3-circle diffractometer in which $\omega = \theta$. Four-circle diffractometers are quite often used in this way, known as the bisecting condition because the χ-frame bisects the angle between $\mathbf{k}_0$ and $\mathbf{k}'$, so much so, that many authors, particularly Busing and Levy (1967), define ω as the rotation from the bisecting position, that is, our ω = Busing and Levy's $\omega + \theta$. In this we have followed normal English practice, cf. Arndt and Willis (1966). We also use the notation $\omega = \theta + \nu$.

In the past the method of working with the A.A.E.C. neutron diffraction equipment has been this: the crystal is mounted on small goniometer arcs and aligned so that some simple and convenient crystal axis is coincident with the ϕ axis. The zero of the angle of ϕ, with respect to some co-ordinate axis in the crystal, is determined by finding and observing a few reflections. With this information about the orientation of the crystal, and knowing the dimensions of the unit cell, one can calculate the four angles 2θ, ω, χ, ϕ for the reflections, and the values of the azimuth angle ψ, which are to be observed. This computation is done on the site computer (currently an IBM 360–50H). To collect data the four angles are set by hand, using the optical scales on the instrument, setting ω and 2θ a degree or two before the peak ready for the scan. The scan to determine the integrated intensity of the diffracted peak is then started. The scan is controlled by the Series 150 control equipment. For each step, counts are received while the monitor scale fills up, the accumulated count is punched on paper tape, and the values of 2θ and ω are incremented. The total time taken to observe one peak is usually somewhere between 15 and 100 minutes. The paper tapes are then processed on the site computer to edit out obvious errors, estimate background level, and finally produce a figure for the integrated intensity.

In the computer-controlled instrument described here, the procedure is: the crystal is mounted on a stalk, not on goniometer arcs, with its alignment known to within a few degrees. Using the scanning and searching routines a few selected reflections of known indices are found and then, under machine control, they are accurately centred in the counter. This angle data is then refined on the site computer to provide a convenient specification of the exact orientation and of the unit cell of the crystal. This information, in the form of a single 3 x 3 matrix, is fed back into the controlling computer which can then proceed, without further attention, to select reflections, calculate and set the four angles, and measure reflection intensities.

3. METHODS OF ANGLE CALCULATION

3.1 General

We have followed closely the methods of Busing and Levy (1967) to which reference should be made for the explanation and development of the formulae. Here we merely define names and list formulae. The procedures are based on a description of the scattering vector in different co-ordinate systems related to one another by transformation matrices.

(i) Reciprocal Lattice of the Crystal

The scattering vector is first expressed in terms of the reciprocal lattice. In this system it is called **h** and has components $h_1\ h_2\ h_3$ (Miller indices).

(ii) Cartesian System

The scattering vector may also be expressed in terms of Cartesian axes in the crystal. For

this system it is called $\mathbf{h}_C = \mathbf{B}\,\mathbf{h}$ and has components $h_{c1}\ h_{c2}\ h_{c3}$. $\mathbf{B}$ is the Cartesian transformation matrix and

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 \cos\beta_3 & b_3 \cos\beta_2 \\ o & b_2 \sin\beta_3 & -b_3 \sin\beta_2 \cos\alpha_1 \\ o & o & 1/a_3 \end{bmatrix}$$

where the a's and α's and the b's and β's, are the direct and reciprocal lattice parameters, respectively. This expression for $\mathbf{B}$ is evaluated not in the PDP-8 computer but in the site computer.

(iii) ϕ-System

Vectors may also be expressed in terms of a set of Cartesian axes attached to the ϕ-shaft of the diffractometer. When all shafts are at zero the y-axis of this system lies along the incident beam, the z-axis is vertical and the x-axis completes a right-handed system. In this system our vector is called $\mathbf{h}_\phi = \mathbf{U}\,\mathbf{h}_C$ and has components $h_{\phi 1}\ h_{\phi 2}\ h_{\phi 3}$. The matrix $\mathbf{U}$ is called the orientation matrix. It describes the orientation of the crystal on its mounting and involves 3 independent parameters. It is determined by centring identifiable reflections and refining the angle observations on the site computer.

Note, as a general proposition, that in the calculation of setting angles, there are nine unknowns to be specified. One could think of these as: the three sides and the three angles of the unit cell, plus three angles specifying the orientation of the crystal. But we want to reduce the amount of calculation to be done on the PDP-8 computer as much as possible so the nine quantities fed in are the elements of the $\mathbf{UB}$ matrix, which contain the information in a pre-digested form, permitting its assimilation with as little arithmetic as possible. The major arithmetic steps are to evaluate $\mathbf{h}_\phi = \mathbf{UB}\,\mathbf{h}$ and to calculate the angles from the elements of $\mathbf{h}_\phi$.

3.2 Calculation of θ

For all systems of working we have:

$$\sin\theta = \frac{\lambda}{2}\left(h_{C1}^2 + h_{C2}^2 + h_{C3}^2\right)^{\frac{1}{2}}. \qquad (2)$$

Since the U-transformation is unitary one may use the components of $\mathbf{h}_\phi$ in this calculation.

3.3 The Bisecting Position

Let us say that the expression atan (y,x) defines an angle arctan (y/x) in the quadrant where the signs of the sin and cos are those of y and x respectively. Then the formulae for the bisecting position are:

$$\left.\begin{aligned} \omega &= \theta \quad (\text{that is, } \nu = 0) \\ \chi &= \text{atan}\left[h_{\phi 3},\ (h_{\phi 1}^2 + h_{\phi 2}^2)^{\frac{1}{2}}\right] \\ \phi &= \text{atan}\left[h_{\phi 2},\ h_{\phi 1}\right] \end{aligned}\right\}. \qquad (3)$$

3.4 The Parallel Position

In this position χ is fixed at 90° and the value of the azimuth angle is 90° from that of the bisecting position. If χ_B and ϕ_B are the angles for the bisecting position, then, for the parallel position,

$$\left.\begin{aligned} \omega &= \theta + \chi_B - 90 \\ \phi &= \phi_B + 90 \\ \chi &= 90 \end{aligned}\right\}. \qquad (4)$$

3.5 Calculations for a Specified Value of Azimuth (ψ)

It is necessary first to define a zero for the angle ψ. Many authors (cf. Arndt and Willis 1966) adopt the convention that $\psi = 0$ at the bisecting position. Busing and Levy (1967) object to this because it makes the definition of ψ depend on the orientation of the crystal on its mount. They therefore introduce a zero reflection $\mathbf{h}_0$ which is to lie in the horizontal plane, on the same side as the diffracted beam, when $\psi = 0$. Let $\mathbf{h}_\phi = \mathbf{U}\,\mathbf{B}\,\mathbf{h}$, and $\mathbf{h}_{0\phi} = \mathbf{U}\,\mathbf{B}\,\mathbf{h}_0$, and construct a matrix $\mathbf{T}_\phi$ with columns, $\mathbf{t}_{1\phi}$, $\mathbf{t}_{2\phi}$, $\mathbf{t}_{3\phi}$. The $\mathbf{t}$'s are a right-handed set of unit vectors such that $\mathbf{t}_{1\phi}$ is parallel to $\mathbf{h}_\phi$, $\mathbf{t}_{2\phi}$ is in the plane of $\mathbf{h}_\phi$ and $\mathbf{h}_{0\phi}$, and $\mathbf{t}_{3\phi}$ completes a right-handed orthogonal set.

Consider now the matrix which will transfer from the laboratory system (that is, the ϕ-system with all axes at zero) to the ν-system, that is, will apply the ϕ, χ, ν rotations about the appropriate axes. An azimuth rotation is simply a rotation by ψ about the x-axis of the ν-system. Suppose the matrix incorporating the three rotations ϕ, χ, ν is $\mathbf{R}$. Its terms are:

$$\left.\begin{aligned}
R_{11} &= \cos\nu\cos\chi\cos\phi - \sin\nu\sin\phi \\
R_{12} &= \cos\nu\cos\chi\sin\phi + \sin\nu\cos\phi \\
R_{13} &= \cos\nu\sin\chi \\
R_{21} &= -\sin\nu\cos\chi\cos\phi - \cos\nu\sin\phi \\
R_{22} &= -\sin\nu\cos\chi\sin\phi + \cos\nu\sin\phi \\
R_{23} &= -\sin\nu\sin\chi \\
R_{31} &= -\sin\chi\cos\phi \\
R_{32} &= -\sin\chi\sin\phi \\
R_{33} &= \cos\chi
\end{aligned}\right\} . \qquad (5)$$

Let Ψ be the matrix which rotates by ψ about an x-axis. Then

$$\mathbf{R} = \Psi\,\mathbf{T}_\phi \quad ,$$

and, from the above expressions for $\mathbf{R}$ one can select the following equations as the most appropriate for the calculation of ν, χ, ϕ:

$$\left.\begin{aligned}
\nu &= \text{atan}\,(R_{13}, -R_{23}) \\
\chi &= \text{atan}\left[(R_{31}^2 + R_{32}^2)^{\frac{1}{2}}, R_{33}\right] \\
\phi &= \text{atan}\,(-R_{32}, -R_{31}) .
\end{aligned}\right\} . \qquad (6)$$

3.6 Azimuth Rotation when **UB** is Unknown

An alternative procedure is to forget that there is a crystal at all, and to consider simply the problem of rotating the sample about the scattering vector, starting from a given position (in practice this starting position may have been found by centring some reflection).

The calculations for this case are simple in principle. From the given ν, χ, ϕ the matrix $\mathbf{R}$ is calculated (Equations 5), it is rotated by pre-multiplication by Ψ, and from this new R-matrix the new angles ν, χ, ϕ are obtained by the inverse formulae (Equations 6).

A brief set of formulae may be derived for an infinitesimal rotation, $\delta\psi$. These are:

$$\left.\begin{aligned} \delta\nu &= \frac{-\cos\nu\cos\chi}{\sin\chi}\,.\,\delta\psi \\ \delta\chi &= -\sin\nu\,.\,\delta\psi \\ \delta\phi &= \frac{\cos\nu}{\sin\chi}\,\delta\psi \end{aligned}\right\}\,. \qquad (7)$$

3.7 The Programs for Angle Calculations

Before starting angle calculations it is necessary to feed in the wavelength, by the instruction /RW (Read Wavelength), and the nine components of the UB matrix, by the instruction /UB.

Calculations in the bisecting position are handled by the subroutine BIANG. First BIANG calls the routine COMTH to calculate θ. COMTH calls the routine HPHI to do the calculation:

$$\mathbf{h}_\phi = \mathbf{UB}\,\mathbf{h} \quad ,$$

and to evaluate the length of $\mathbf{h}_\phi$ from which θ is calculated (Equation 2). The subroutine BIANG then puts $\omega = \theta$ and $2\theta = 2.\theta$ and the calculation of χ and ϕ is completed in accordance with Equations 3.

The instruction /CA (Calculate Angles), followed by a set of indices, causes the angles for that reflection to be calculated for the bisecting position and typed. If one then wishes to drive to that position one must type /DA (Drive to Angles) to call the motor routines.

For the parallel position, subroutine BIANG is called and is followed by statements incorporating Equations 4.

For general azimuth positions the possibilities depend on what can be fitted into the computer. The calculations for a specified ψ, described above in Section 3.5 were completely programmed for the PDP-8. They required 150 locations, plus about 60 locations for accepting and storing data, and for managing the details of ψ-scans and data-collection runs at specified ψ's. They can be loaded any time they are required, but they are not included in the programs attached to this report.

When two diffractometers are being operated from the computer there is not sufficient room in-core for these programs, and they have been replaced by a simple azimuth-rotation program based on Equations 7. The program is entered by an azimuth scan instruction /AS which increments the azimuth, counts and prints. Errors will accumulate as the scan proceeds (see Table 1). This table was calculated on the IBM 360-50H using the procedures outlined in Section 3.6. Note that this incremental procedure is more accurate if one starts with a fairly high value of χ, and near the bisecting position. To increase accuracy, the calculation is done in steps of 0.1 degree, the increment $\delta\psi$ being specified as an integral number of these steps + ve or – ve. If the errors become excessive the reflection can be recentred. If a complete data-collection scan is required one would stop the scan by typing /OF and measure the reflection intensity by typing /MR. The program is brief and permits most operations of interest. If the **UB** matrix is known then a FORTRAN program exists which will accept the experimental values of ν, χ and ϕ (obtained by /PA) and compute the value of ψ, if some zero reflection $\mathbf{h}_o$ is specified.

4. PROCEDURES FOR DETERMINING THE CRYSTAL ORIENTATION

4.1 Overall System

As explained in Section 1 the calculation of angles is based on the concept of a general orientation, defined by an orientation matrix. One finds, and accurately centres in the counter, several reflections of known indices. These data are then refined, externally, to provide the orientation matrix. We are concerned here with the programs which search for, and then centre, the reflections, and with the FORTRAN programs which refine these observations.

4.2 Searching Routines

We assume that one knows: (i) the unit cell of the crystal, (ii) its orientation on the mounting pin to within a few degrees, and (iii) enough about the structure to be able to select a few moderately strong reflections.

A small increment, or step, in the position of an axis is achieved by the subroutine STEP which requires two integers, NANG, which specifies the axis, and XANG, which specifies the number of two-hundredths of a degree.

NANG = 1 for 2θ

= 2 for ω

= 3 for χ

= 4 for ϕ

= 0 for a step of XANG in ω and of 2.XANG in 2θ.

Subroutine STEP calls subroutine ADNANG to sort out the actual address from NANG. It then floats XANG, multiplies it by 1/200, adds it to the angle stored, and calls MOTST (Motor Start).

To get to the neighbourhood to start searching, the /DM (Drive Motor) instruction is used. This requires input of NANG to find the axis address in ADNANG and then the desired angle. (In this case NANG = 0 will be equivalent to NANG = 4).

To increment angles the /IM (Increment Motor) instruction is used. Input of NANG and XANG is requested and STEP is called. For example, /IM 2: – 100: would move ω back 0.5 degree. When moving manually in this fashion one may make spot observations of intensity by the /TC (Take Count) instruction, which calls subroutine TKCNT (cf. Section 5) to take a single count, and types it.

To search automatically one uses the instruction /SS (Start Scan) followed by NANG and XANG. This initiates a repetitive call of STEP and TKCNT. If /TN (Typewriter On) is typed the counts are typed in lines of 8; with /TF (typewriter Off) the counts are not typed, and one would have to look at the ratemeter dial to see if there is anything of interest going on. Counting and stepping proceeds until the scan is cancelled by the instruction /OF (Off), or until a limiting angle is reached.

4.3 Centring Routines

When a diffracted beam is entering the detector, in some fashion, then the command /CR (Centre Reflection) will adjust the orienting angles and the counter angle so as to maximize the reflection in the centre of the detector aperture. The program requests a single integer to specify whether the χ-axis or the ϕ-axis is to be varied in order to centre the reflection vertically in the detector aperture (3 for χ, 4 for ϕ). The choice is made by a common-sense estimate of which motion will be most effective, for example, if ω is near the bisecting position then one would choose χ.

The steps in the centring procedure are:

(i) Move ω back 0.5° and then scan ω in 10 steps of 0.1°; drive to the maximum and evaluate LIMIT = half the maximum count.

(ii) Call subroutine CONV (see below) to trim χ or ϕ so as to centre the beam accurately in the detector aperture in the vertical direction.

(iii) Call CONV to maximize ω accurately.

(iv) Call CONV to trim 2θ so as to centre the beam accurately in the detector-aperture in the horizontal direction.

CONV is a general centring subroutine which seeks the half-maximum (evaluated in step (i) above, and stored in LIMIT) on either side, and then drives midway between. It starts stepping the axis, in steps of 32/200 degree, taking counts. When the count becomes less than LIMIT it takes one step back and the step interval is reduced to 16/200 degree before proceeding. This process continues until the step-interval is 2/200 degree.

The axis then steps off to converge on the other half-maximum, and finally drives midway.

As in the scanning program with /TN the counts are typed in lines of 8, starting a fresh line for each of the above procedures. With /TF this does not happen, and the four final angles only are typed. These observations then provide the Input to the FORTRAN program CCD/2 which refines them to produce the matrix **UB** on which the angle calculations are based.

4.4 The FORTRAN Refinement Program CCD/2

4.4.1 Theory

The orientation matrix **U**, is a 3 x 3 orthogonal unitary matrix which specifies the orientation of the crystal, that is, it transforms from the Cartesian axes of the crystal to a Cartesian system attached to the ϕ-shaft. If the cell parameters of the crystal are known, then the matrix **U** can be obtained from a knowledge of the angles ω, χ, ϕ which will centre two reflections of known indices in the counter aperture. From the three angles ω, χ, ϕ one can define $\mathbf{u}_\phi$, the description in the ϕ-axis system of a unit vector in the direction of the scattering vector, by the equation:

$$\mathbf{u}_\phi = \begin{Bmatrix} \cos\omega \cos\chi \cos\phi - \sin\omega \sin\phi \\ \cos\omega \cos\chi \sin\phi + \sin\omega \cos\phi \\ \cos\omega \sin\chi \end{Bmatrix} .$$

Let us call the two reflections which we use to define **U** the Primary Orienting Reflection, $\mathbf{h}_1$, and the Secondary Orienting Reflection, $\mathbf{h}_2$. These two reflections contain between them six parameters, while there are only three parameters concealed in **U**. Problems involved in the accuracy and consistency of experimental measurement intrude on our definition. To avoid this difficulty we decide that the Primary Orienting Reflection is to determine the direction of a vector in the crystal, with respect to the ϕ-axis, while the Secondary Orienting Reflection is to determine a rotation about this vector. In terms of our definitions we say that $\mathbf{u}_{1\phi}$ and $\mathbf{h}_1$ are to be parallel, while $\mathbf{h}_2$ is to lie in the plane of $\mathbf{u}_{1\phi}$ and $\mathbf{u}_{2\phi}$.

The actual calculation of **U** proceeds as follows:

Define a right-handed, orthogonal, unit-vector triple, $\mathbf{t}_{1C}$, $\mathbf{t}_{2C}$, $\mathbf{t}_{3C}$, such that $\mathbf{t}_{1C}$ is parallel to $\mathbf{h}_{1C}$, and $\mathbf{t}_{2C}$ lies in the plane of $\mathbf{h}_{1C}$ and $\mathbf{h}_{2C}$. Define another such triple, in the ϕ-axis system, based on $\mathbf{u}_{1\phi}$ and $\mathbf{u}_{2\phi}$. Suppose that 3 x 3 matrices $\mathbf{T}_C$ and $\mathbf{T}_\phi$ are formed using these two sets as columns. Then **U** must satisfy the equation $\mathbf{T}_\phi = \mathbf{U}\,\mathbf{T}_C$, that is,

$$\mathbf{U} = \mathbf{T}_\phi \tilde{\mathbf{T}}_C \quad .$$

From two reflections, **U** can be determined if the Cartesian Transformation Matrix, **B**, is known. The determination is unique: there are no free parameters. One would prefer, for certainty, to over-determine the problem, by observing more than the bare two reflections, and to refine the data.

Quite commonly the cell parameters of the crystal are not accurately known, or, perhaps the wavelength, λ, may not be accurately known, and, to check the instrument, one may wish to refine observations to determine the instrumental zeros of the 2θ, ω and χ shafts. Therefore the program CCD/2 will accept the angular information on up to 10 centred reflections and refine it. This program finally produces values of the instrument zeros and of the matrix **UB**, which can be fed back into the PDP computer so that data collection can proceed.

The input data consists of the values of 2θ, ω, χ and ϕ for several centred reflections of known indices. Each reflection contributes 3 pieces of information: 2θ, ω, and either χ or ϕ, depending on whether χ or ϕ was used to centre the reflection vertically.

The possible free parameters are: (i) the wavelength; (ii) the instrument zeros of 2θ, ω and χ; (iii) the cell parameters; (iv) three of the six angles, ω_1, χ_1, ϕ_1 and ω_2, χ_2, ϕ_2 of the Primary and Secondary Orienting Reflections. Of course one cannot refine for λ and cell parameters simultaneously. Note that the orientation matrix, **U**, is not dealt with as itself; it is approached by means of the two orienting reflections.

The procedures for calculating the angles in the refinement program are:

Calculation of 2θ: This is the simplest calculation and uses the formula of Section 3.2, Equation 2. This is referred to by Busing and Levy as a "Type 1" observation.

Calculation of ω: The theory of this calculation is given by Busing and Levy as a "Type 5" observation.

Calculation of χ or ϕ: The theory of the χ calculation is given by Busing and Levy as a "Type 3" observation, and that of ϕ, as a "Type 4" observation.

4.4.2 The program

When procedures for calculating the observed quantities are laid down, starting from trial parameters, the procedure for refining the parameters is well established. We employ the method of full-matrix-least-mean-squares. The derivatives of the observations with respect to the parameters are calculated numerically.

The details of the input and output to CCD/2 are given in Appendix 2, together with a complete program listing. Operational experience has shown that the major problems are: first, to name the reflections correctly; second, to obtain accurate values of χ, particularly at low θ's. We have sometimes found it convenient to observe a couple of low angle reflections and derive an approximate **UB**, and then to leave the machine running unattended to centre, slowly, 5 to 10 high angle reflections (that is, type a feed tape with /CA h_1: h_2: h_3 : /DA /CR 3: , repeated). A number of high angle reflections like this are usually capable of providing a **UB** matrix which will bring up the reflections within $\pm 0.1^\circ$ of the calculated position in ω.

The instrument zeros have been determined, and adjusted out, by centring a number of reflections at positive and negative values of 2θ.

4.4.3 General note

The procedures for calculating the diffractometer angles starting from the matrix UB, which are described in Section 3, are elegant and brief. And the experimental advantages of working in a general orientation, without goniometer arcs, are considerable. Nor is it difficult to centre reflections accurately. The major difficulty in doing things this way in any diffractometer, computer-controlled or not, is the computational problem of deriving the orientation matrix from the observations of centred reflections. Without the solution of this problem by Busing and Levy this whole procedure for calculating angles and mounting crystals could not be used. One could say that the program CCD/2 is the key to the whole system.

5. PROCEDURES FOR DATA COLLECTION

5.1 The Subroutine STCT (Start Count)

Counts from the detector counting channel and the monitor counting channel are recorded in external scalers in the interface, the detector scaler having a capacity of 2^{23} and the monitor scaler 2^8.

The subroutine STCT sets the hardware gates required for counting and then sets a program count flag. The subroutine then waits until the program count flag is reset, whereupon the external detector scaler is transferred to the Floating-Point Accumulator, normalised, and stored in the floating location COUNT.

The type and duration of one preset count is established by the instruction /SM (Set Monitor) given before any counting instruction. /SM requires two integers as data, the first is 1 for real time pulses of 1000 hertz, and 0 for monitor channel pulses; the second is the multiplying factor CT. The factor CT is the number of times the monitor overflow will be recorded to give one preset count cycle. The monitor overflow initiates an interrupt which is recorded and serviced in the MONDET routine. During DCOLL, CT is multiplied by the G-factor when the peak is scanned (cf. Section 5.3).

5.2 The Subroutine TKCNT (Take Count)

The procedure for taking a count is designed to avoid spurious counts due to unwanted interference. The interference we are concerned with here is the odd burst of electrical interference which may inject, say, 10 – 100 spurious pulses during a count. Fortunately it is rare, but precautions are taken because it could cause confusion in a routine like CENTER which is searching for certain intensity levels; and, in an intensity scan, we have, by our adopted procedures, discarded any possibility of even suspecting its presence.

Four separate preset counts are taken and intercompared to detect interference. The program is satisfied only when it has received four statistically consistent counts, within ± 4 standard deviations. If one of the preset counts is significantly different from the accumulated average, then four new counts are taken. The factor of 4 is included in the setting of the monitor: when CT = 1 then 1024 monitor counts, or approximately 1 second if the real time clock is selected, is the duration of the count.

5.3 Measuring the Intensity of a Reflection

The subroutine to measure intensity is OBSCUR. This is entered by a JMS in the mainline programs which collect data, immediately after the angle calculations. Most of the subroutine is taken up with checking whether the angle settings would make the reflection "obscure". The procedure is described in detail in Section 5.4. If the reflection is obscure the subroutine types OBSCURE, the indices, and the calculated angles, and immediately returns.

If the reflection is visible then the subroutine OBSCUR calls the measuring routine DCOLL which drives to the calculated angles and types and punches the quantities: h, k, l, 2θ, ω, χ, ϕ before any actual counting. (Note that data from DCOLL is the only data recorded on perforated paper tape.)

Having driven to the calculated angles the peak intensity for that reflection is being received. This may be quite high (several hundred counts per second) or, if the reflection is accidentally weak or absent, it may only be background level (say 0.5 – 5.0 counts/second).

The total time to be spent in scanning the peak is preset by two instructions: /SM, which sets the time for a count (see above), and /NS (Number of Steps), which is used to feed in an integer, NSTP, setting the number of two-hundredths of a degree of ω in the scan over the peak. The computer scans, in an ω–2θ mode, over the calculated peak position, and it is essential to determine the half-height positions in this scan as accurately as possible in the allotted time. One can then check later that the scan was symmetric, and of sufficient amplitude to include the peak completely. If the peak is strong the half-heights can be located accurately; if it is weak, less accurately; and if it is very weak or absent, not at all. The program is designed to determine them as well as possible in the allowed time.

At the calculated peak position, single counts are received until a total of greater than 200 has been received. Say it takes G single counts to do this. If the peak is strong G may be 1 only, but if it is weak G could be quite large. It is, however, constrained to be no more than 32. The idea of the quantity 200 is that it is high enough to make moderately good sense, statistically, when it comes to judging half-heights. Henceforth in the scan, the steps in ω are G/200 degree, and the time spent at each is G times the "single count" time. (This is achieved by increasing the monitor scale factor, CT, described in Section 5.1).

The next task, having sampled the peak intensity and set the scale, is to measure background. This is measured, half at 3° of ω on either side of the peak. First we move $+3^\circ$ of ω and $+6^\circ$ of 2θ and take one G-count. On the basis of this we decide the background counting time, on the statistical basis that:

Time spent on background = Time spent on signal/ $r^{\frac{1}{2}}$,

where r = peak signal rate/ background rate.

In detail, the number of G-steps in the scan, is:

GS = NSTP/G (truncated to an integer),

and the number of G-counts in background

= 2 (GB + 1) ,

where GB = GS/$2r^{\frac{1}{2}}$ (truncated to an integer).

So, if r = 1 (that is, absent or very weak signal) the time spent on background equals the time spent on signal, but, for strong signals, it is very much less. On the basis of the first background count we also calculate the half maximum height.

Having counted the background, and adjusted it in the ratio of the times, we proceed at last to scan over the peak and accumulate the signal counts. The subroutines TKCNT and STEP (with NANG = 0 and XANG = G) are called GS times. STEP is also used to achieve the shifts for background counting described above. In the course of the scan the step number at which the count goes above the half maximum height for the first time, and at which the count goes below the half maximum height for the last time, are recorded.

In the course of the scan over the peak the instruction /TN (Typewriter On) will cause the individual counts to be typed in lines of eight. Remember however that if, say, NSTP = 400 (that is, a 2° sweep in ω) and it was an intense peak, so that G = 1, there would be 400 numbers typed. The instruction /TF (Typewriter Off) cancels the typing of these individual counts but not the type-out of the final data described next.

At the end of the scan we print and punch:

G – the number of two-hundredths of a degree per step

GS – the number of G-steps in the scan

NH_1 – the position of the first half-height

NH_2 – the position of the second half-height

ACCNT – the total count over the peak

BGC – the adjusted background.

These quantities are processed in the FORTRAN program CCD/4 described in Appendix 3.

5.4 "Obscure" Checks

There are three ways in which the reflection can be inaccessible, or obscure:

(i) The value of 2θ may be higher than the maximum allowed. On the diffractometer 2θ max = 140° so, to allow for the procedure of subroutine DCOLL, there is an "obscure" return if 2θ calc > 134°.

(ii) The incoming or outgoing beam may be obstructed by the χ-frame (that is, the fixed circular frame carrying the track for the χ motion). These conditions are:

$$103 > \omega > 77$$

$$-77 > \omega - 2\theta > -103 ,$$

(iii) Even if the χ-frame does not obstruct the beams, if the value of χ is outside $\pm 15^\circ$ it is possible that the ϕ-frame (that is, the squarish frame which carries the bearings and motors for the ϕ motion) may interfere. These conditions are:

$$\text{if } \chi > 15 \text{ then } 148 > \omega - 2\theta > 32$$

$$\text{and } -32 > \omega > -148 \ ;$$

$$\text{if } \chi < -15 \text{ then } 148 > \omega > 32$$

$$\text{and } -32 > \omega - 2\theta > -148 \ .$$

If conditions (i) or (ii) occur there is nothing to be done; the reflection is inaccessible (at that value of the azimuth ψ, in any case) and an "obscure" return results. If condition (iii) occurs then the program generates the alternate setting (-h, -k, -l) at angles 2θ, ω, $-\chi$, $\phi + \pi$, and proceeds to test them. Note that, for the bisecting position, obscuring should never occur, except, of course, for $2\theta > 134^\circ$.

In the OBSCUR program the angles of interest, 2θ, ω, χ and α ($= \omega - 2\theta$) are first "fixed" and reduced to the ± 180 range, and then checked in a special subroutine CHEKK.

5.5 Overall Procedures

There are three ways of working:

(i) Individual sets of Miller indices may be fed in, either by tape or at the keyboard, and the diffractometer will then proceed to calculate the angles, for the bisecting position, and measure the intensities. This mode of operation is initiated by the instruction /IR (Individual Reflections). This enters the routine IRR where the routines to read the indices, calculate the angles, and measure the intensities, are called. If /IP (Individual Parallel) is typed, the action is the same except that the angles are calculated for the parallel position.

(ii) The program can also generate its own sets of indices, and then proceed to calculate the angles, for the bisecting position, and measure intensities. This mode is initiated by /CD (Collect Data) which enters the program at address CDR. This routine first requests 8 integers: H min, H max, K min, K max, L min, L max, Nsymm and Limit. The program then proceeds to generate sets of Miller indices subject to the requirements that:

$$H_{max} \geqslant H \geqslant H_{min}$$

$$K_{max} \geqslant K \geqslant K_{min}$$

$$L_{max} \geqslant L \geqslant L_{min}$$

$$H^2 + K^2 + L^2 = \text{LIMIT} \ .$$

The usual purpose of H min, H max, etc., is to specify the quadrants in which data are to be taken. They should be considered usually as having one of three values: - large number, o, + large number. But the time taken to work out the selecting arithmetic is quite appreciable so the large number shouldn't be larger than needed, for example 10 or 20 would be reasonable. One could, of course, collect data in slabs of reciprocal space if desired.

The indices are selected moving outwards in reciprocal space to successively higher values of $\text{LIMIT} = H^2 + K^2 + L^2$, starting from the value of LIMIT which is fed in.

The value of Nsymm specifies certain classes of systematic absences, not an extensive set because there was not room to do that, but a few simple ones (those which suit binary arithmetic):

Nsymm = 1	Primitive
= 2	K + L even
= 3	H + L even
= 4	H + K even
= 5	FCC, all even or all odd
= 6	BCC, H + K + L even
= 7	L even.

The program, after selecting the sets of indices, calls the angle calculation and measuring routines. When the value of LIMIT has been incremented so high that no accessible reflection can be found, the program ends in the waiting loop.

(iii) The third way of collecting data is to centre some reflection and then type /MR (Measure Reflection). This is used in setting up and for measuring the intensity after a ψ-rotation.

5.6 Insertion of Standards

The first two of the above methods of gathering data insert a standard reflection at intervals. The preliminary read-in of the data specifying the standard is initiated by /IS (Insert Standard) which enters the program to read in 5 integers; the indices of the standard reflection, the frequency at which it is to be inserted, and the initial value of the counter, that is, if the N^{th} reflection after starting is to be a standard, this fifth number is $-N$, and after that, standard reflections will be inserted at the specified frequency.

The actual insertion of the standard is achieved by a statement in each of the three data-collection sub-programs described above: JMS I LSTD. This calls the subroutine STD which increments and tests the counter. If no standard is needed it returns immediately; if it is time for a standard, the standard indices are copied into the normal location for the indices, the axis positions are checked, and the angle calculation and intensity-measuring routines are called; the counter is then reset. The standard is always measured in the bisecting position.

6. OPERATING PROCEDURES

6.1 General

This section gives an account of operations that are to be performed by the experimenter or operator. A more detailed description of what occurs when these procedures are followed is given in the main sections of this report.

Before the diffractometer can be operated from the teletype (ASR33) console the program must first be loaded into the computer. To do this we use an initial loader which can load an RIM tape (see Appendix 4) containing a BINARY format loader. Once the binary loader is in core the rest of the program is loaded using binary format. For details of these formats see the PDP-8 Programmer's Guide and Handbook. The binary format tape contains the complete main program with error checking in the form of computed checksums, etc.

Usual operation of the PDP-8 computer requires loading of two tapes, one RIM and the other BINARY. For this installation the loading programs have been rewritten so that one tape only is necessary, the first phase being the loading of the binary loader in RIM format which, when complete, transfers control to the start of the binary loader which proceeds to load the rest of the tape in binary format. On completion of a correct binary loading, control is passed to the start of the main program sequence, whereupon the computer waits for the next instruction.

When instructions are typed in at the teleprinter console (or read in using the keyboard reader) the computer will respond only to valid commands that have been accepted within 15 seconds. (See Section 6.2). No erroneous command will be printed. Likewise only valid characters will be accepted and printed on the typewritten copy when data is expected. Leader-trailer code, carriage-return-line-feed, and space, are invalid characters in an instruction field, or in a data field, and will be ignored. When a paper tape is being prepared off-line for use as a control tape these symbols can be used to give a more readable copy without affecting the resulting operation from the tape.

The detailed procedures for loading the program are given in Appendix 4. If for some reason the computer has been halted by the operator, or by an error halt in the main program, the cause or causes should be rectified and the program re-started at address 200, that is, set 0200 in the Switch Register; press LOAD ADDRESS key; press START key.

6.2 Instruction Format and Operation

Instructions and data may be entered in two ways: manually, at the keyboard of the ASR33 teletype; or at the associated tape reader. When a program is running it can only be interrupted (of course!) from the keyboard. When the computer returns to waiting loop it will immediately start reading tape, if the reader is switched on, looking for a valid instruction. The two input media are indistinguishable to the computer. Instructions begin with a slash (/) so when a slash is received the computer will try to read two characters off tape to complete the instruction format, whether the slash was initiated from the keyboard or from the reader. Be careful if a data tape is in the reader and an instruction such as /TN or /OF is required; the keyboard reader must be switched off before typing the slash.

All instructions are composed of three characters: a slash, which denotes a command, followed by two characters. These two characters must be available within fifteen seconds of the receipt of the slash character, otherwise the command will be treated as invalid. After the command is checked against the current table of instructions it will be printed out, if valid, and ignored if not. When a valid instruction is received, the program, after setting and resetting the program flags required, will pass control to the address specified in the instruction table. The particular program sequence will be performed and, eventually, the computer will end up in the waiting loop, waiting for the next instruction, unless another instruction has caused the original instruction sequence to be terminated.

There are two main types of instructions, main program instructions and interrupt program instructions. Both are confirmed by being typed but only the main program instruction terminates the existing program. The interrupt instructions /TN, /TF and /OF, are used to set and reset flags which are examined by the main program to control particular operations.

6.3 Data Formats

Various main program loops require data input before performing particular steps. This data may be integer (fixed-point) or real (floating-point).

Integer data must be between −2048 and 2047, and if this limit is exceeded, a number within this range will be taken, which, in most cases, will not resemble the intended input.

Real numbers are typed in as a normal floating-point number with a decimal point. The number of digits in the number cannot exceed the significance available with 23 binary places, that is, if the number 1234.5678 were to be typed in, the last digit (8) would not be accepted because the input, disregarding the decimal point, exceeded 8,000,000. Numbers cannot be entered in E format.

All numbers, when typed and checked, must be terminated before being accepted by the computer as valid. An integer can be terminated with a colon (:) (or a decimal point (.)); a real number must be terminated with a colon (:). Note that real numbers do not need to have a decimal point.

If a number is incorrectly entered on the keyboard the entire number field (back to the last delimiter) is cleared if the "Rubout" key is pressed before typing the terminating character. Once a terminator is typed in, the only method of correcting the number is to retype the main program instruction.

6.4 Table of Valid Instructions

Note: There are two commonly used integers: NANG and XANG

NANG = 1 for the 2θ axis

= 2 for the ω axis

= 3 for the X axis

= 4 for the ϕ axis

= 0 for an $\omega - 2\theta$ motion in STEP.

XANG = the number of two hundredths of a degree to be moved

INSTRUCTION	REQUESTS INPUT TYPE AND DEFINITION	ENSUING ACTION	PRE-REQUISITES	COMMENTS ON USAGE
/?? Reinitialize	None	Clears all flags and returns to waiting loop	None	Use only in abnormal circumstances. Operations should normally be cancelled by /OF or /CP.
/TN Typewriter On	None	In SCAN, CENTER or DCOLL it causes the individual counts to be typed in lines of 8	None	This does not suppress the typing of, say, the final angles in CENTER, or the final output in DCOLL; just the individual counts are suppressed.
/TF Typewriter Off	None	Cancels /TN	None	/TN and /TF may be typed any time. DCOLL would be normally run with /TF.
/OF Off	None	Sets the program end flag	None	This is the normal way to cancel a scan started by /SS. In DCOLL it will complete the scan over the peak before stopping.
/CP Cancel Program	None	Stops the operating program and returns to waiting loop	None	This would interrupt programs like CENTER and DCOLL. It may be used to interrupt scans. It may be more usual to interrupt with another instruction.
/PA Print Angles	None	Causes the current angles to be printed out	None	Normally used when shafts are stationary. If used in the middle of programs like SCAN, CENTER or DCOLL it would interrupt the program.
/IM Increment Motor	NANG and XANG	Increments the designated axis	None	Used, in conjunction with /TC, to search manually for peaks, or to move back before starting a scan.
/DM Drive Motor	NANG and the desired angle	Drives the designated axis to the desired location	/ZA	

/CR Centre Reflection	3 if χ is to centre vertically, 4 if ϕ	Centres and maximizes the reflection by varying ω, χ or ϕ, and 2θ. Types out final angles	/SM	/TN applies .
/ZA Zero Angles	The approximate values of 2θ, ω, χ, + ϕ to within $\pm 9^\circ$	Drives all axes to -10° and then to $+10^\circ$, setting the markers on the transit through zero	None	After loading the program this should be the first instruction in order to establish axis positions .
/DA Drive to Angles	None	Drive to the values stored in the calculated angles locations	/CA	Having calculated and printed angles with /CA, this instruction will drive the axes to the calculated positions .
/TC Take Count	None	Takes and prints a single count	/SM	Used in searching for peaks .
/SM Set Monitor	Two integers: 0 for real time (10^3 Hz) and 1 for monitor; how many times 1024 pulses for preset monitor	Sets constants	None	A necessary pre-requisite for any counting .
/SS Start Scan	NANG and XANG	A scan commences on the designated axis, stepping and taking counts	/SM	/TN applies .
/RW Read Wavelength	The wavelength	Sets constants	None	
/NS Number of Steps	The number of two-hundredths degree in ω for the intensity-measuring routine DCOLL	Sets constants	None	It is intended, in routine data collection, that the scan be set as narrow as possible .
/UB	The elements of the UB matrix row-wise	Sets constants	None	
/IS Insert Standards	Five integers: the frequency of insertion, the 3 indices of the standard, and $-N$, where it is desired to insert the first standard after the Nth reflection	Defines the standard reflection for periodic insertion during systematic data collection	None	The axes positions will be checked before inserting the standard .
/IR Individual Reflections	One or many sets of 3 integers: the Miller indices of the reflections	Calculates angles for the bisecting position; checks obscuring; measures intensity; prints and punches results	/SM, /RW, /NS, /UB, /IS	The sets of indices would normally be read from a tape, on which one could vary the counting times by incorporating /SM etc. instructions. For details of the data collection procedures and of output refer to manual . (continued)

INSTRUCTION	REQUESTS INPUT TYPE AND DEFINITION	ENSUING ACTION	PRE-REQUISITES	COMMENTS ON USAGE
/CA Calculate Angles	The indices of the reflection	Calculates and prints the angles for the bisecting position	/RW, /UB	This was originally introduced for checking the calculating routines. See /DA.
/MR Measure Reflection	None	This assumes you are at the peak of a reflection and measures its intensity with a standard DCOLL scan	/SM, /NS	Would normally be used following /CR. If one is uncertain of the accuracy of the angle calculations one could find and centre each reflection and then measure it, that is, /CA, /DA, /CR, /MR.
/CD Collect Data	Eight integers: Hmax, Hmin, Kmax, Kmin, Lmax, Lmin, Limit, Nsymm	Generates sets of indices subject to: Hmax > H > Hmin Kmax > K > Kmin Lmax > L > Lmin $H^2 + K^2 + L^2$ = LIMIT Nsymm = 1 for primitive = 2 for K + L even = 3 for H + L even = 4 for H + K even = 5 for Face Centred Cubic = 6 for Body Centred Cubic = 7 for L even	As for /IR	This is the program for the unattended collection of 3-dimensional data in specified quadrants, following the same procedures as /IR. The numbers Hmax, Hmin etc. should be kept as small in magnitude as possible. With /IR and /CD the termination would be by the instruction /OF if you wish to terminate after the current reflection, or /CP if you wish to abandon the current reflection. One would resume by typing the / instruction – inserting the constants, including the current value of LIMIT, in the case of /CD.
/AS Azimuth Scan	One integer: the number of tenth degree steps required to define the ψ increment	Steps and counts in increments of ψ about the existing azimuth	/SM	This would normally be used after centring a reflection, the current value of 2θ effectively defining the azimuth. See Section 3.6 for comments on accuracy.
/PL Punch Leader	None	Punches approximately twenty inches of leader-trailer on the paper-tape perforator	None	
/PH Punch Heading	Any character required except Slash (/)	Types and punches the keyboard input	/PL	This program enables comments to be inserted on the paper-tape record, mainly for identification, date, etc. Terminate by a valid instruction.

7. INTERRUPT PROCEDURES

7.1 Philosophy

In an On-line application such as this, the interrupt procedures are vital since they provide the asynchronous control of all interactions between the computer and the external devices. The interrupt mechanism controls input-output, motor movement and checking, detector and monitor counting, and various timing requirements of the experiment.

A complete description of the method of handling a particular interrupt will be found in the PDP-8 Handbook. A brief description follows:

When a device requests attention a device-flag is set in the instrument /PDP-8 interface. This flag causes a program interrupt which transfers control of the computer program to location 0001; the contents of the program counter are stored in location 0000. By use of the instructions skip-if-device-busy (or alternatively skip-if-device-not-busy) the actual device causing the interrupt can be determined, the appropriate handling routine can be called, and the interrupt dismissed using the ION and JMP I 0 instructions. Note that the device flag must be cleared in the appropriate handling routine before dismissing the interrupt.

The computer cannot be interrupted while the interrupt is off. Interrupts are enabled by issuing an Interrupt On (ION) instruction; the interrupt remains on until cleared by an interrupt from a device, or by an Interrupt Off (IOF) instruction. The interrupt must be turned on just before leaving the interrupt routine.

If it is intended that the main program should remain unmodified when an interrupt is called, then none of the working locations used by the interrupt routines can be common to those used in the main programs. This includes subroutine entry points which contain the return point for the main line, which would be lost if the subroutine was called under interrupt. Likewise, loop counters, etc., cannot be modified by the interrupt routines unless this modification is required as a valid part of the mainline operation.

There are three main types of interrupt resulting in different methods of interrupt handling:

1. Input/Output operations which include operation of the keyboard (and keyboard reader), the teleprinter and the paper tape perforator.

2. Status-In operations which include operation of the monitor-detector preset count, the motor markers and the axis zero markers.

3. Real Time Clock operations consisting of input timing, and motor drive operation.

7.2 Interrupt Requests

On receiving a program interrupt, program control is transferred to routine INTER, which after storing the contents of the accumulator and link in SVAC and SVLK, tests the various device flags using skip (IOT) instructions. If a flag is set, control passes to the device service routine where the interrupt is dealt with. When the device has been serviced and the appropriate flag has been reset, the interrupt is dismissed via the routine TERINT which restores the contents of the accumulator and link, turns the interrupt on, and jumps indirectly via location zero in which is stored the address of the next instruction to be executed in the interrupted program.

7.3 Interrupt Service Routines

7.3.1 Clock interrupts

The clock interrupt routine (CLOCK) is used to time the operation of motors at a submultiple of the clock frequency (either 333 hertz or 250 hertz). If the overall motor flag is set, the motor drive interrupt routine MDRCI is called. The operation of this routine will be dealt with later. The routine SGTM is then called which times the input of characters for the keyboard instruction routine and for

the input character routine for data input. This is achieved by incrementing two counters SGTK (keyboard instruction) and IPCNTR (character input) until they reach zero. (N.B. The clock routines do not have any control over the preset counting time in data collection). The clock routine is dismissed via routine TERINT.

7.3.2 Teleprinter interrupts

The teleprinter interrupt service routine (TELSER) is concerned with printing characters on the teleprinter. The number of characters to be printed (NO) is examined. If this is non-zero the next character in the output buffer is printed and NO is decremented by one. If there are no more characters to be printed a program flag RDY is set which indicates that the teleprinter is available next time output is required by the main program. For more detailed operation see the section on input-output routines.

7.3.3 Paper-tape perforator interrupts

These are dealt with in exactly the same fashion as for teleprinter interrupts.

7.3.4 Keyboard and keyboard reader interrupts

The keyboard interrupt service routine (KEYSER) is concerned with sorting input characters by examining the character and also by checking program flags.

If the input character is a slash, then control passes to the SENT routine which sets the signal expected timer (SGTK), initializes the command routine and dismisses the interrupt.

If the input character was not a slash then the signal expected flag is examined and, if it is set, control passes to the SGNL routine which picks up command characters and sorts them.

If the signal expected flag is not set, the input flag is tested and, if set, the flag is replaced by the character and the interrupt is dismissed. Likewise if the input flag is not set the type flag is examined and, if set, is replaced by the character and the interrupt dismissed. If the character is not a slash, and the appropriate flags have not been set, the keyboard interrupt is ignored and the interrupt dismissed without acknowledgement.

7.3.5 Status-in interrupts

The status-in interrupts are a group of interrupts associated with the hardware counting circuitry and with the motor and axis zeros. The actual interrupts are recorded in a 12 bit register which can be read into the contents of the accumulator (RSI = Read Status-In). There is one common skip-line for the status-in register, the independent interrupts being determined by reading the register and sorting out which bit is set, clearing the appropriate bit, and then servicing the request. The interrupt is dismissed in the normal manner (JMP I DISMIS). The routine STATUS is used to sort the interrupt, the routine CLST is used to clear the interrupt flag and exit to the appropriate service routine, the addresses of which are found in the status table STAB.

The MONDET routine, which is associated with the hardware counting and is entered by the monitor overflow interrupt, keeps a tally of the number of overflows required for one preset count. If the number of overflows is insufficient then the count gates are reopened for another monitor count. If the correct number of overflows has been recorded then the program count flag is reset.

The MOTTZ, MOTOZ, MOTCZ, MOTPZ routines are associated with the zero-checking of the motors two-theta, omega, chi and phi respectively. These routines call the MARK routine to check for errors and adjust appropriately if errors are found. For further details see the motor routines.

The MOVETZ, MOVEOZ, MOVECZ, MOVEPZ routines are associated with the zero-checking of the axis movements of 2θ, ω, χ, and ϕ respectively. They call the MOVCHK routines which check the zero and make appropriate modifications if the zeros are incorrect. For further details see the motor routines.

8. MOTOR DRIVE PROCEDURES

8.1 Overall Operation

The four axes 2θ, ω, χ and ϕ are each driven by a "Slo Syn" stepping motor through 360:1 worm gears. The motors require 200 steps per revolution so one pulse will move the axis 1/200 degree. The actual positions of the axes are only accurate to about 0.05 degree because of looseness and backlash in the bearings and gears.

To ensure that a motor is moving correctly, and that the axis is actually at the position held by the computer in the current-position location, a pointer is attached to the motor shaft. This pointer intersects a light beam, directed at a photo-sensitive diode. The resultant signal from the diode causes a computer interrupt every time the motor shaft passes through zero, and, during this interrupt, the current position location is checked, and if necessary, adjusted.

The same system is used to detect the zeros of the axes, each pointer being set to signal the computer when the axis is between zero degrees and one degree. The precise axis zero is obtained using the motor zero check as well as the axis zero check.

The backlash of each axis is minimised by always moving the axis in a positive direction before coming to rest in the final desired position. For motor excursions in a negative direction the desired position is decremented by exactly one degree and, when this position has been reached, the motor is advanced one degree in the positive direction.

To allow for the inertia of the axes the motors are started using a slow start procedure. When a motor has to be moved, two slow start counters are set up to the values of -50 and -10. These counters are incremented and examined at each motor pulse cycle; 50 motor steps are ignored before movement of the motor and the motor is given a slow start of the form: one motor pulse, nine motor cycles ignored, one motor pulse, eight cycles ignored, one motor pulse, seven cycles ignored, etc. down to continuous running of the axis. The larger motions 2θ and ω always have the slow start indicator set to -2, which means that these two axes are moved on a pulse-ignore-pulse-ignore basis, their final continuous speed being half that of χ and ϕ. The long wait before moving the axes is to allow for any oscillation to die down before moving the motors in the opposite direction, and it was introduced in an attempt to improve the motion of the 2θ axis which, because of its large inertia and loose gearing, is particularly troublesome.

The slow start constants and motor speed (nominally -50, -10 and 250 hertz) may easily be altered to obtain the maximum motor accuracy and efficiency.

8.2 Motor Data

Each motor has associated with it a block of data. This data block contains all the information required by the various routines that concern the motor movement (see below). The data block is composed of 13 constants and variables; a description of these follows, in the actual order used in the block:

(a) A constant which is the address of the floating point location which holds the motor angle. This floating point number contains the desired position of the axis when movement is requested and contains the actual position of the axis when the print angles instruction is used.

(b) A flag which is set to -1 for negative movement and 0 for positive movement. This flag also serves as a reminder that the negative movement will have to allow for the final one degree positive motion mentioned above.

(c) and (d) Two locations which contain the values of the two slow start constants.

(e) The current degrees position.

(f) The current two-hundredths position.

The numbers stored in (e) and (f) are always calculated in the form ± degrees + two-hundredths where the degrees may be negative but the two hundredths is always positive and in the range 0 – 199. For example the angle −0.1 degrees is represented as −1 degree plus 180 two-hundredths. This same system is used for all fixed-point angle locations.

(g) and (h) The desired degrees and two-hundredths positions respectively.

(i) A flag which indicates whether the motor is to be moved or not. This flag will be one for movement and zero for stationary.

(j) A constant which has to be preset as the contents of the accumulator when the motor forward or motor reverse pulses are initiated by the computer. Different bits are set or reset for particular motors. A more detailed description is available under interface hardware.

(k) and (l) Two constants which contain the two's complement of the software limits of the motor movements.

(m) A constant which is the address of the motor name as used to print out errors etc.

8.3 Motor Routines

Motor operation is controlled by the computer in three phases:

Requests for Movement: When an axis is required to be moved, the MOTST routine is called. This routine is a mainline program and as such has no direct control over motor operation. It changes the desired position locations and sets the motor indicator (program flag).

Motor Movement: Motor movement is controlled by the interrupt routine MDRCI which is entered by the real-time clock routine. This routine initiates all pulses to change the respective states of the motor registers in the hardware interface.

Motor and Axis Checking: The checking of the motor and axis positions is executed under interrupt control using the MARK and MOVCHK routines. These are entered from the status-in routine and check for motor errors and make appropriate modifications to the actual degree and two-hundredths locations.

Other routines concerned with motor movement and position are the initialization routine INIM, zero check routine ZERGO, and the error checking routines contained in the supervisor.

8.4 MOTST Sub-routine

This routine is used to call the motor moving procedures. The desired angles must be in the floating locations of the respective motors. The return address will be incremented unless an error is detected, that is,

```
JMS   I   LMOTST

ERROR  RETURN

NORMAL RETURN
```

The MOTST routine calls the MOTOR routine. This routine must have four arguments, the addresses of the motor flags for 2θ, ω, χ and ϕ respectively. The MOTOR routine operates on each motor in succession performing the following steps: the motor flag is reset to prevent interrupts to the motor, which thus preserves the data-block from changes occurring under interrupt; the address of the data block is computed from the address of the motor flag and the data block is copied into the working locations. Using the SETUP routine, the floating angle is then normalized to between −180 and +180 degrees and converted to fixed point. These fixed point angles are deposited in the address specified following the calling of SETUP, in this case the desired degree and desired two hundredths

working locations. The desired degree settings are compared with the motor limits (software) and if no error is detected, the desired and actual settings are compared to determine motor direction and the overshoot indicator is set accordingly. Finally the slow start constants are set up, the motor flag is set, and the data-block copied back to its original location. If no limit errors have occurred this procedure is carried out for the four angles, the overall motor indicator is set, and the wait-for-motors routine is called. When the motors have reached these final positions, control passes to the calling program at the normal return address.

If an error is detected the message LIMIT, followed by the axis name, is typed out on the keyboard, no motor indicators are set, the wait-for-motors routine is not called, and control passes to the calling program at the error return address.

8.5 MDRCI Sub-routine

This subroutine is called under interrupt from the clock routine, the motor frequency being a submultiple of the clock frequency (normally one quarter). The subroutine is called with four arguments following the call. These are the addresses of the motor flags.

The subroutine handles each motor in succession and goes through the following procedure. The motor flag is tested and if not set then the following steps are bypassed. The slow start flag is incremented and tested and if it is non-zero, the following steps are again bypassed. The address of the data block is computed and the block transferred to the working locations using the TRANS subroutine. The slow start variables are adjusted and the desired and current motor positions compared. If these are unequal, then a forward or reverse motor pulse is initiated and the current motor locations adjusted accordingly. If the current and desired positions are equal then the overshoot indicator is tested and if set, the initial slow start is set up and the overshot degree is allowed for. If the desired position was reached without overshoot then the motor flag is cleared. In each case the data block is transferred back to its original location and the next motor is checked.

The subroutine returns with the number of motors still in motion in the accumulator; this information is deposited in the motor indicator.

8.6 MARK Sub-routine

This subroutine is called by the motor-zero interrupts. Three arguments, the current degree address, the error flag, and the serious error flag, follow directly after the calling statement.

The subroutine MARK transfers the relevant motor data to its working locations using TRANS and then proceeds as follows: if a slow start is in progress, the interrupt is dismissed. If the two-hundredths position (HTT) is within one pulse of the expected zero the error flag is set for check and the interrupt is dismissed. If the two-hundredths location is within ten pulses of the expected zero then the error flag is incremented. If the number of pulses from the expected zero is greater than ten then the serious error flag is set to one.

For both the error and serious error cases the following changes are made according to whether the present two-hundredths is > 100 and according to direction of motor travel.

(1) If FORWARD and HTT > 100 then DTT = DTT + 1, HTT = 0.

(2) If FORWARD and HTT ≤ 100 then DTT = DTT , HTT = 0.

(3) If REVERSE and HTT > 100 then DTT = DTT , HTT = 199.

(4) If REVERSE and HTT ≤ 100 then DTT = DTT −1, HTT = 199.

The resulting data is then transferred back to the original locations and the interrupt is dismissed.

The convention for the error flags is −1 for check and no errors, 0 for no check and no errors, and a positive number is the total accumulated errors since the last supervisor acknowledgement.

8.7 MOVCHK Sub-routine

This subroutine is called as a result of an axis-zero interrupt. It requires the same arguments as MARK and the MOVCHK call must immediately precede the MARK call.

The subroutine MOVCHK sets the serious error flag to one if the current degree location is non-zero and resets the current degree location. If the current degree location is correctly zero then the flag is set to -1 for check, as described above. The interrupt is then dismissed. It should be noted that the axis zero is approximately at +½ degrees so that the current degree position should be zero regardless of direction of motor travel, and the two-hundredths should remain unaltered.

8.8 Other Motor Routines

The subroutine INIM resets all motor flags and thus stops all motor travel. This routine is only called at the start of the program or by the Re-initialize instruction. It is the one instruction which will always stop the motors.

The subroutine ZERGO drives all motors to -10 degrees and then to +10 degrees. This procedure is used to ensure that the zero-marker for each axis is crossed. This routine is called by/ZA (Zero Angles). The approximate (that is $\pm 9^{\circ}$) axis positions are required as input. This procedure is essential when core has been reloaded in order to "lock in" the axes. As ZERGO may be called by errors noted by the supervisor routine (SUPER) the desired angles (floating locations) are preserved and restored before the zero sweep.

The supervisor subroutine examines the error and serious error locations for each motor, and prints the following messages:

(a) ERROR (NAME). This is printed after twenty errors have been noted.

(b) SERIOUS ERROR (NAME). This is printed after any serious errors have been noted. The supervisor will then initiate a zero-sweep after preserving locations necessary to recontinue the program.

The subroutine ANGPRT which calls the subroutine WHERE converts the current degree and current two-hundredths into a floating-point angle which is put into the floating angle location (Subroutine SUB). When the four axis positions have been converted, the positions are typed out. ANGPRT can be called as a main program instruction /PA (Print Angles).

9. INPUT-OUTPUT PROCEDURES

9.1 Philosophy

The input-output procedures control all data transfer between the user and the computer via the keyboard, keyboard reader, teleprinter, and paper tape perforator. Input to the computer consists of data to the floating point package in either integer or real (floating point) format, characters for the /PH (Punch Heading) instruction, and actual commands to request separate instruction procedures. Output from the computer consists of acknowledgement of instructions, printing of separate results for scans, etc., on the teleprinter, and recording of heading characters and data collection results on the teleprinter and the paper tape perforator.

All input-output is controlled by two distinct phases, a mainline program that sets appropriate flags and an interrupt program that behaves in the pattern governed by the flag settings. The only input data not concerned with a mainline request are the instruction codes for each command; the request for this case is initiated by the typing of the slash of the instruction format.

9.2 Keyboard and Keyboard Reader Operation

The keyboard and the keyboard reader of the ASR 33 Send-Receive teleprinter which is used for computer communication, both use the same input channel, the only difference being that the keyboard reader requires an instruction (KCC Fetch Keyboard Character) to advance the tape through the reader.

When data is expected on either the keyboard or keyboard reader the Input Character subroutine (IPCH) is called. This subroutine issues a fetch-character instruction and sets a flag to -2048, the negative number indicating that a character is expected. When a character is received the flag is set to the code of the input character, (note that this is always +ve) and the return is to the calling subroutine.

When data is expected by the command sequence (initiated by typing a slash) the two characters are required to complete the required instruction format (/ch ch). When a slash is typed a flag is set to -4095, to indicate characters expected, and a fetch-character instruction is issued (that is, they may be entered on the reader). The flag is incremented under interrupt (250 Hz) until the flag reaches zero or two characters are received. If the flag reaches zero, program control jumps to the waiting loop. If two characters are received before the instruction is counted out, they are sorted in the instruction-sorting routine; otherwise the instruction is ignored.

9.3 Teleprinter Operation

Teleprinter operation is controlled using a buffer of 16 characters. When a request to print a character is initiated by the output-character subroutine (OPCHH), the character is loaded into the next sequential (modulo 16) location in the print buffer, and the number-of-characters location is incremented by one. If the buffer is full then the routine waits until room in the buffer becomes available. The buffer is printed out sequentially (modulo 16) by the interrupt routine servicing the teleprinter, until the number of characters in the buffer is zero.

9.4 Perforated Paper Tape Punch Operation

Perforated paper tape punch operation is exactly the same as for the teleprinter and uses the same basic routine. The punch is used solely by the data-collection routine (DCOLL) and has been introduced into the whole system solely to provide a clean data record free from the interspersed comments and instructions which appear on the typed record. The programs which process this data record on the site computer are described in Appendix 3.

9.5 Input Routines

IPCH Input a Character Routine: This routine issues the fetch keyboard instruction 6032 and waits for a character. The character is masked to the low order 7 bits and the most significant bit (AC4) is set to 1.

Calling Sequence JMS I LIPCH (pointer on page 0)

Returns with character in C(AC).

9.6 Output Routines

OPCHH Output a Character Routine: This routine prints the low order 8 bits of the accumulator as an ASCII character. If the software punch flag (PFLAG) does not equal zero the character is also punched on the paper tape perforator.

Calling Sequence JMS I LOPCH (pointer on page 0)

Character in C(AC)

If PFLAG $\neq 0$ then punch low order 8 bits

Returns with C(AC) = 0.

TY2 Output Two Characters in Trimmed Code: This routine prints the two characters contained in the C(AC) in 6 bit trimmed code. (That is all ASCII codes between 240 and 337 with the two most significant bits trimmed off).

Calling Sequence Characters in C(AC)

JMS TY2

Returns with C(AC) = 0.

CRLF Print Carriage Return Line-feed: Self-explanatory

Calling Sequence JMS I LCRLF (pointer on page 0)

STRING Print Message as String of Characters: This routine prints the contents of a specified location in trimmed code using TY2 and keeps stringing characters together from sequential locations until a zero is found.

Calling Sequence JMS LSTRING (pointer on page 0)

Address of first two characters.

A typical message (for example, LIMIT) would appear as

```
LIMIT,  1411    / L I
        1511    / M I
        2440    / T Space
        0000    / Finish of Table.
```

The messages for STRING are stored in odd dispersed locations and are commonly used to fill up pages of core storage.

10. SOFTWARE PROCEDURES

10.1 General

This section describes routines not specifically referred to before. They are listed to provide future programmers with a rough idea of the procedures involved and of the particular conventions that are adhered to in most cases. Many of the routines used in the program are not mentioned at all, and when this is the case, calling sequences are given as comments in the program listing in Appendix 1. Comments in the program listing can also be used as a guide to the operation of particular routines.

Many routines are more complex than required at present. This complexity was introduced because, eventually, two or more diffractometers will be controlled from the PDP-8. This means that a lot of indirect addressing has been used, which is pointless for a single diffractometer. Similarly there are gaps in core to hold the data for future diffractometers These gaps can be filled with extra running programs if required. Further development or alteration to the basic programs for counting and motor driving, however, seems unnecessary and inadvisable.

10.2 Floating-Point Arithmetic

These programs incorporate much of standard package supplied by the manufacturer, Digital Equipment Corp. (DEC). Floating-point numbers are stored in the standard DEC three word floating format. The three words are consecutive and consist of the binary exponent (first word) and mantissa (2nd and 3rd words). The mantissa is a signed two's complement fraction which when normalized is between ½ and 1 disregarding sign. The accuracy of the numbers is of the order of six decimal places.

Operations on floating-point numbers are done using the floating-point accumulator as the main working register. This is situated in locations 44 – 46 on page zero. Other working locations and temporaries for the floating-point package cover all locations between 40 and 60 on page zero.

The floating-point package consists of three basic blocks; the floating-point interpreter, the floating-point extended functions, and the floating-point input-output routines. The floating-point interpreter in this installation has been copied from the standard PDP-8 floating-point interpreter, the original package being modified to perform sequential data modifications without exiting from the package. The floating-point extended functions and the floating-point input-output have been developed apart from the DEC programs although some of the main algorithms are the same.

Floating Interpreter

The usable instructions are:

Floating	Exit	FEXT	
Floating	Add	FADD	address
Floating	Subtract	FSUB	address
Floating	Multiply	FMPY	address
Floating	Divide	FDIV	address
Floating	Put	FPUT	address
Floating	Get	FGET	address
Floating	Increment	FINC	address.

The reader should refer to the PDP-8 Floating Point Package for a description of the first seven instructions. The floating normalize instruction FNOR of the PDP-8 package has been replaced by the floating increment instruction. The FINC instruction increments the contents of the location specified in the address field by three and is used to increment a pointer to the next floating location. The FNOR instruction is now called using the extended function breakdown of the FEXT group, and care must be taken to prevent calling FNOR from any other extended function. In most cases the floating normalize routine is called external to the package using a JMS instruction.

10.3 Floating Point Extended Functions

The possible extended functions are:

Floating Square	FSQR
Floating Square-root	FSQRT
Floating Sine	FSIN
Floating Cosine	FCOS
Floating Arctangent	FATN
Floating Normalize	FNOR
Floating Negate	FNEG
Floating Absolute	FABS
Floating Input	FINPUT
Floating Output	FOUTPU
Floating Output	FDPOUT

The FSQR and FSQRT routines are identical to the standard PDP-8 package. FSIN, FCOS return with the sin, cos of the floating accumulator (before the instruction) in the floating accumulator. To call FATN the cosine of the angle must be in floating location B and the sine must be in the floating accumulator. The floating arctan routine returns with the result adjusted for quadrant in the floating accumulator. Note that all the angles specified above are in degrees not radians. FNOR, FNEG, and FABS modify the contents of the floating accumulator. FINPUT, FOUTPU, and FDPOUT will be dealt with under floating input-output. Note that the content of the floating accumulator is lost for the two output instructions. Many of the above are called as separate subroutines using a JMS instruction.

10.4 Floating Point Input-Output Routines

These routines are concerned with the input-output of either floating point or fixed point numbers and are called by most of the main programs when any number or string of numbers is required to be printed or required as data.

(a) Input Routines

All floating-input routines call the actual input routine DECONV. This routine calls the input routine IPCH and forms a double precision fixed-point number. All input characters are ignored that are out of context or would cause an overflow of the double precision registers. If the input is a rubout the routine is re-initialized and a new number has to be typed in. The DECONV routine can only be exited by a colon (:) or by a decimal point (.), the colon or decimal point being regarded as a delimiter for integer input, and the colon only being regarded as a delimiter for floating point. Any extraneous decimal points are ignored for floating input. When the delimiter is found the number is scaled to allow for the number of places after the decimal point in FINP.

INTIN Integer Input Routine: This routine reads a single precision number entered from the keyboard or keyboard reader. The number must be in the range − 2048 to + 2047. The number should be terminated by a colon (:) although a decimal point also acts as a terminator. If a rubout is typed before the terminating character, the routine will restart the entire number. No invalid character will be printed on the typewritten copy.

Calling Sequence JMS I LINTIN (pointer on page 0)

Returns with number between − 2048 and 2047 in C(AC).

FIXIN Multiple Integer Input Routine: This routine reads N integer numbers and deposits them in sequential core locations starting at a specified address. The routine calls INTIN N times.

Calling Sequence JMS I LFIXIN (pointer on page 0)

− N

Pointer to address of first core location.

FLINT Floating Input Routine: This routine reads a floating point number with decimal point up to about 7 significant digits. The number must be terminated by a colon (:). The rubout character has the same function as for INTIN. No invalid character will be printed on the typewritten copy.

Calling Sequence (1) JMS I 4

Returns with the number in the C (FAC)

(2) JMS I 7

FINPUT

FINP Multiple Floating Input Routine: This routine calls FLINT N times depositing the numbers in sequential floating locations.

Calling Sequence JMS I LFNP (pointer on page 0)

- N

Pointer to address of first floating location.

(b) Output Routines

The main floating output routine is the VFOUT routine. This routine has two arguments following the call, these being the number of decimal places before and after the decimal point. A typical example is the routine FOUT which calls VFOUT as follows:

```
FOUT,   0                          / ENTRY
        JMS          VFOUT
        4
        3
        JMP     I    FOUT          / EXIT
```

Here 4 is the number of digits before the decimal point and 3 is the number of digits after the decimal point (F4.3). The subroutine VFOUT, after scaling the contents of the floating accumulator and testing whether the required format is possible, transfers the resultant double precision word and calls the basic double precision output routine DROP which does the actual printing of the number. The routine FOUT has output format F4.3, the routine DPOUT has output format F7.0, and routine IOUT (integer output) has an output format of F4.0. All numbers that cannot meet format specifications are replaced by an asterisk, and all numbers have a colon delimiter printed following the number.

IOUT Integer Output: This routine prints the contents of the accumulator as a single precision signed integer. Leading zeros are suppressed and the number has a colon (:) delimiter directly following it.

Calling Sequence - Number in C (AC)

JMS I LIOUT (pointer on page 0)

Returns with C(AC) = 0 .

FOUT Floating Output: This routine prints a signed floating point number with four digits before and three digits following the decimal point. Leading zeros are suppressed and the number has a colon delimiter.

Calling Sequence - (1) Number in C (FAC)

JMS I 6 (pointer on page 0)

(2) JMS I 7

FOUTPU

Returns with C(FAC) lost.

FOPT Multiple Floating Output: This routine calls FOUT N times, the numbers being obtained from the sequential floating locations specified.

Calling Sequence JMS I LFOP (pointer on page 0)

- N

Pointer to address of first floating location.

DPOUT Double Precision Output: This routine prints a signed floating point number as a double precision integer (7 places before the decimal point). Leading zeros are suppressed and the number is delimited with a colon (:).

Calling Sequence – (1) Number in C (FAC)

JMS DPOUT

(2) JMS I 7

FDPOUT

Returns with C(FAC) lost.

10.5 Calling Subroutines with Arguments

In this chapter some programming procedures are described which are unobvious extensions of standard procedures described in the manuals.

(a) Simple Subroutine

As explained in the Programmer's Guide, a subroutine is called by the JMS instruction, for example:

Mainline Program				Subroutine			
0405	4270	JMS	SUB	0470	0000	SUB,	0
0406	- - - - - - - -			- - - - - - - - - - - -			
				0490	5670		JMP I SUB .

The JMS instruction stores the return point (0406) at the location SUB (= 0470) and the statement JMP I SUB returns the program as desired. If the subroutine is on another page then it is addressed indirectly – quite a simple extension.

The calling of subroutines is, however, almost never so easy as this simple account would suggest, because, nearly always, the subroutines have to manipulate actual numbers (which we refer to generally as arguments) which have been generated in the calling program or are stored, or must be stored, elsewhere in the memory.

(b) Subroutines with Arguments

In the simplest case, where the calling program, the subroutine, and the arguments are all on the same page, there are no problems; see, for example, the subroutine ROWCOL, which performs the three row-by-column multiplications in the matrix-vector multiplication routine HPHI.

In other cases, where only a single argument is involved, it is possible to move it into the C(AC) before calling the subroutine and so to have it accessible to manipulation by the subroutine even when on another page. Subroutines CONV, CHEKK, and others, rely on this procedure.

At the next stage of complication one can carry the arguments (N.B. the actual arguments not their addresses) in the locations immediately following the JMS call. For example, the subroutine ABMBA does this. It is called by JMS ABMBA and the next two locations carry arguments. The calling statements are:

```
4515   4332   JMS   ABMBA
4516      3              /first argument
4517      6              /second argument
4518             Continue
```

and the routine is:

```
ABMBA, 0     /storing, initially,   4516
-- - - - - -
       TAD I   ABMBA /to obtain the 1st argument
       ISZ     ABMBA
- - - - - - - - - - - -
       TAD I   ABMBA  /to obtain the 2nd argument
       ISZ     ABMBA  /to obtain the return address
- - - - - - - - - - - -
       JMP I   ABMBA  /to return to 4518 .
```

This procedure is used quite often (for example, CHEKK) .

The final stage of complication is to write subroutines which may be called from anywhere in core to manipulate arguments which are stored anywhere. This is then the true equivalent of the FORTRAN statement CALL SUB (A, B, etc.). The procedure is to set aside locations immediately after the JMS calling instruction to store the addresses of the arguments involved.

Consider the subroutine UNIT which is used to normalize a vector to unity. It is called:

```
0405   4777   JMS   I  PUNIT
0406   0166   LHF
0407          Continue,
```

where LHF is an address (anywhere in core) which contains the address of the first of nine locations which hold the floating-point 3 x 1 vector. Now consider the problem of writing the subroutine UNIT. The first requirement is to transfer the contents of 0406 (containing the address of the address of the vector) to an address within the subroutine UNIT. This is done by a special address-transferring routine ADR which is called: JMS I LADR, followed by a location containing -N, where N is the number of arguments in question, followed next by N locations into which the addresses of the arguments are transferred. So the subroutine has the form:

```
4271  0006   UNIT,  0
4272  4422          JMS  I  LADR
4273  7777          -1              /one argument
4274  0000   UNA,   0
4275  1674          TAD  I  UNA
4276  3274          DCA     UNA
------              --------
                    JMP  I  UNIT.
```

ADR has transferred the contents of address 0406 into 4274 (= UNA). The statement TAD I UNA fetches the address of the vector, which is stored back in UNA. Arithmetic statements such as FGET I UNA will now fetch the first element of the vector itself. The subroutine ADR does one more thing; it increments the address stored in UNIT by N so that JMP I UNIT will return control to the proper point in the calling program. This general procedure is used in all the major arithmetic subroutines.

In the motor drive routines and the collect data routines, the arguments of the subroutine are in a fixed block, for example, motor data for a particular motor. An argument is generally transferred with the subroutine, the block address being obtainable from this by addition or subtraction of a constant. Once the block address is obtained, the subroutine COPY is used to transfer the block of data to another block which can be referred to directly by the program. . After manipulation, the data block is copied back to its original position to preserve changes made to the data. The above procedure is used extensively in the interrupt routines, which call the routine TRANS to transfer arguments from block locations to working areas. TRANS is used in this case because COPY cannot be called under interrupt since the return address for the mainline could be lost.

The main routines involved are:

COPY Copy a Block of Core to Another Block: This routine is a mainline program and is used to copy data blocks to a working area and then later to copy them back. The block consists of N words of core.

```
Calling Sequence        JMS  I  LCOP     (pointer on page zero)
                        - N
                        address to copy from      (first word)
                        address to copy to        (first word).
```

ADR Set Arguments for a Subroutine: This routine stores the arguments following a subroutine call in temporaries within the subroutine. The return location of the subroutine is also set.

```
Calling Sequence        SUB,    0                 /calling routine
                                JMS  I  LADR      (pointer on page 0)
                                - N               /N arguments in SUB
                        ARG1,   0
                                -
                                -
                                -
                        ARGN,   0 .
```

Returns with C(AC)* = 0 following last argument.

* By C(AC) we mean 'contents of the accumulator' – 12 bits, labelled individually AC0–11.

TRANS Transfer a Block of Core to Another Block: This routine is an interrupt program and duplicates the COPY routine.

Calling Sequence

```
JMS   TRANS
- N
(Address -1) to copy first from
(Address -1) to copy first to
```

Returns with C(AC) = 0.

10.6 Addressing Procedures

(a) Data Block Addresses

The addressing procedures used in the program have been adopted because of the possibility of operating two or more diffractometers from the one main program. For this reason procedures are slightly more complex than necessary for operating one single diffractometer.

If a data block is required by some routine, then two stages of indirect addressing are used, the first indirect is used to pick up the address of the block of data; this address is then stored in a temporary location. The data is then indirectly addressed using the temporary location which may be incremented up the block by either the ISZ or FINC instructions depending on whether fixed-point or floating-point data is being used. When more than one diffractometer is to be operated it is envisaged that a group of pointers will be updated by the diffractometer gaining control, and thus the data addresses can be altered more easily by the above procedure.

An example of this procedure is the floating input routine FINP which has a pointer to an address as one of its arguments. The first instructions of the routine are to get the address of the block indirectly before working up the block.

```
FINP, 0              /ENTRY
      JMS  I  LADR   /GET ARGUMENTS
         - - - - -
FADR, 0              /POINTER TO BLOCK ADDRESS
      TAD  I  FADR
      DCA     FADR   /BLOCK ADDRESS
         - - - - -
      FINC    FADR   /MOVE UP DATA-BLOCK
         - - - - -
```

(b) Preservation of the Return Address

If there is only one diffractometer being operated by the computer then the addressing procedures described above would suffice. However, if two diffractometers are being controlled then another problem arises. On certain occasions (when waiting for counts to accumulate, when waiting for motors to move to new locations, when waiting for fresh instructions, when waiting for I/O devices to operate) the current program dealing with one diffractometer may pass control to programs for the other diffractometers. Note that control cannot be passed while calculations are in progress: in this case the other diffractometer must wait until the calculations are over and the program enters one of the four waiting modes listed above.

If a subroutine involves calculations only, then no special precautions are necessary beyond those mentioned above. But if a subroutine is to be re-entrant then one must take precautions to preserve the return point. A typical example of a subroutine which involves this difficulty is the

subroutine CONV. It is basically a simple subroutine: one argument is carried into it in the C(AC) while others are on the same page. But it does, in its course, call routines which involve motor movements and are therefore, liable to transfer of control. Consequently it must be written:

```
CONV, 0
        TAD    CONV
        DCA    CONVPT
        .
        .
        .
        JMP  I  CONVPT .
```

The return address, stored in CONV by the original calling instruction, is preserved in CONVPT near the beginning and retrieved at the end. Note that, through the body of the subroutine, CONV is now available as a temporary working store and is so used. Note also that one desires to use as few such preserved locations as possible because they have to be stored somewhere and summoned up in a block when required. They constitute the "working file" of the diffractometer. Consequently they are made to serve common duties whenever possible, for example CONVPT, preserving the return pointer for CONV, is commoned with LOBSPT, preserving the return point for OBSCUR, because, for the one diffractometer, CONV and OBSCUR are entirely clear of one another.

10.7 Rules for Mainline Programs

This section contains general rules for all programs in case modification is necessary to existing routines.

(1) The contents of the accumulator C(AC) must remain cleared unless it is being used.

(2) When calling a subroutine, the C(AC) should be cleared unless an argument is being passed.

(3) When a subroutine is returning, the C(AC) should be clear unless an argument is being passed.

(4) The state of the link is not assumed to be known unless set to a particular state.

(5) Temporaries cannot be assumed to have a known value and should be set up by the routine using them. Care should be taken that the same temporary is not used by two programs at the same time. Subroutine entries are commonly used as temporaries.

(6) Temporaries used by the interrupt program must not be used by the mainline (or vice-versa) unless data is to be passed by this method.

(7) Waiting loops should not be used unless the supervisor routine is called from within the loop. A main program jump (keyboard instruction) is set up by the interrupt but is executed by the main program.

10.8 Rules for Interrupt Programs

(1) Contents of the accumulator and link are saved by the interrupt handler (INTER).

(2) Contents of the accumulator and link are restored by the interrupt dismiss routine (TERINT), called by JMP I DISMIS.

(3) If an interrupt routine is to exit to a mainline program the TERINT routine should be called. If not ION must be executed and, in this case, the previous mainline return will be lost on the next interrupt.

(4) The device flag causing the interrupt must be cleared by the service routine.

(5) The interrupt program can communicate to the mainline by setting particular indicators (program flags or variables), which can be examined by the mainline or vice-versa.

(6) If several variables are to be set together by the mainline, without interruption occurring, the IOF instruction should be executed before modifying the instructions and ION executed just after the modification. Note that a pseudo-interrupt can be generated by IOF, setting location 0 to the return address, clearing the save-AC location and jumping to the terminate-interrupt routine when the task required is performed. (See OPCHH).

11. INTERFACE HARDWARE

11.1 General

Copies of the interface drawings are available in Appendix 5. These should be referred to if a more detailed knowledge of operation is required, (for maintenance purposes, however, a set of current drawings should be obtained to avoid confusion).

By interface we mean the connection between the external devices and the computer. These devices are:

1. Paper Tape Reader
2. Paper Tape Perforator
3. Real Time Clock
4. Motor Drives
5. Motor Checks and Limits
6. Counting Hardware.

Most of these use program interrupt for their operation. When standard DEC programs are run on the computer only those not using interrupt will work unless the above interrupts are blanked off. The program required for use was the DEC high speed macro-assembler (before compiling programs on the IBM 360/50) which requires the paper tape reader and perforator to operate under interrupt. The remainder of the interrupts are disabled by a flip-flop unless enabled by an IOT pulse at program commencement. The flip-flop is disabled by power-clear pulses at computer turn-on.

6302 ENABLE Enable Interrupts

(all except Devices 01, 02, 03, 04 which are permanently enabled).

11.2 Real Time Clock

A real time clock of 1000 Hz has been included in the interface to provide basic timing needs for experiment operation. The timing pulses are used directly when the monitor has been selected to count for a preset time (see Section 11.5 Counting Hardware) and are also fed into the computer using program interrupt. The interrupts are divided down in the programs to provide motor timing and instruction timing.

The clock is controlled by a 1000 Hz microfork which has a stability well in excess of that required for this experiment. Clock pulses set a clock flag which causes a program interrupt, the corresponding input-output transfer commands being:

6301 SNC Skip if Clock Flag is Not Set

6304 CCF Clear Clock Flag.

11.3 Motor Drive Hardware

The four axes are driven using Slo-Syn stepping motors. These are driven by a conventional two-stage twisted ring counter, with driver amplifiers to handle the load of motor windings.

The motors for two-theta and omega are 50 oz. in. types (SS50-1008) and the motors for chi and phi are 20 oz. in. types (SS25-1002). All motor windings are driven by -15 volts through an appropriate dropping resistor, the current being 0.9 amp.

The motors two-theta, omega, chi and phi are all driven by the one instruction, each motor being selected by resetting a particular bit of the accumulator before executing the pulse-motor-forward instruction PMF. The same accumulator settings are used when motors are required to be moved in the reverse direction with the pulse-motor-reverse instruction PMR. The commands are:

6341	PMF	Pulse	Motor	Forward
6321	PMR	Pulse	Motor	Reverse

and the AC Settings are:

3777	Select Two-theta
5777	Select Omega
6777	Select Chi
7377	Select Phi .

Also associated with the motor drives are the motor limit switches. These consist of a bank of two microswitches set to operate about ½ degree apart. The first switch to be operated is wired to set a bit in the status register, and the second switch is wired to disable the motor drive amplifiers. The first switch would normally return the computer program to the initialize and wait sequence (INWT), which would reset the motor flags, which in turn stop the motors. If the computer program has lost control, the motors are stopped by the second switch. The torque of the motors is not great enough to cause any damage if they did run against mechanical limits; they are easily stalled.

11.4 Motor and Axis Checking Hardware

Each motor and axis has a small vane attached which intersects a light beam when the motor or axis passes through the zero check position. Chopping the light beam produces a current change in the photo-duo-diode detector which is a.c. amplified and sets a flip-flop in the status register.

The zero check position is between zero degrees and one degree for each axis and between zero and one pulse (reverse) for each of the motors. Because the motors are driven through flexible couplings the motor pulses and the motor zero detector will be out of alignment during a slow-start, and interrupts caused during the slow start sequence are ignored by the MARK subroutine.

The flexible couplings are cemented rubber pads. They turn about 5 degrees at rated motor torque and were incorporated in order to avoid wear on the worm gears due to the jumping motion of the motors.

11.5 Counting Hardware

Experiment counting is performed in two external scalers: a monitor scaler of 8 bits (÷ 256) and a 23 bit detector scaler. The detector scaler can be 'OR'-ed into the contents of the accumulator using two IOT instructions, the second generally being combined with the reset IOT.

6331	RDH	Read Detector High	(high order 11 bits)
6332	RDL	Read Detector Low	(low order 12 bits)
6334	CDS	Clear Detector Scaler.	

The most significant bit (AC0) of the accumulator remains unchanged for the RDH instruction. As the C(AC) are normally reset before executing this instruction, a positive number results which, with the low order bits (RDL), and the appropriate exponent, is already in floating format (but is seldom normalized). Note that the above IOT's assume that the C(AC) are cleared.

The monitor scaler cannot be read into the computer but it causes a program interrupt on overflow by setting the appropriate bit of the status register. The monitor scaler can be cleared using the CMS instruction.

6302 CMS Clear Monitor Scaler.

Counting is controlled by two flip-flops set from the computer. The first is set to 1 if the monitor scaler is desired to count real time clock pulses (1000 Hz) and set to 0 if the desired count is to be from the counter which monitors the incident beam intensity. The second flip-flop is set to one if a count is required. This opens the counting gates, etc. This flip-flop will be reset when the monitor has received one preset count; further flexibility of count is obtained using software.

Both flip-flops are set using settings of the accumulator, bit 11 being set for real time, bit 10 being set for count. To set these external flip-flops the following IOT's are required.

6342 CSO Clear Status Out

6344 SSO Set Status Out (from C(AC)).

11.6 Status-In

The status-in register is a set of 12 flip-flops that can be set by external events so that the computer program will have a knowledge of the state of external devices. Any flip-flop in the status register will cause a program interrupt if set. The various inputs to the status-in register are.

Bit 0. Monitor Count Overflow

1. Two-theta Motor Zero Check
2. Omega " " "
3. Chi " " "
4. Phi " " "
5. Two-theta Axis " "
6. Omega " " "
7. Chi " " "
8. Phi " " "
9. Forward Motor Limit
10. Reverse Motor Limit
11. Spare.

When a flip-flop in the status register is set and causes an interrupt, the interrupt is sorted by using the SNS instruction (Skip if no status bit is set) and then the particular bit is determined by software after reading the status word into the C(AC). This sorting of status bits is done in the routine STAB. When a particular bit is found to be set it is then reset and the appropriate routine is called.

A bit in the status register is reset by setting the like bit of the accumulator and executing the CSI (Clear Status In) instruction. If all other bits of the accumulator are cleared then there will be no change to the corresponding bits of the status register. If others of these had been set then they would cause another interrupt after the first has been cleared and the interrupt turned 'ON' again.

The IOT's used with the status register are:

6311 SNS Skip if No Status bits are set.

6312 RSI 'OR' Contents of Status In Register into C(AC).

6314 CSI Clear Status In Register (only those bits for which the appropriate accumulator bit is set).

11.7 Paper Tape Reader

Reader operation is identical, as regards software, to the standard PDP-8 High Speed Reader Option. The following IOT's are used:

6011 RSF Skip on Reader Flag

6012 RRB Read Reader Buffer. The contents of the reader buffer are transferred into AC4-AC11 and the reader flag is cleared.

6014 RFC Reader Fetch Character. The reader flag is cleared, the reader buffer is cleared, a character is loaded into the reader buffer from tape, and, when loading is complete, the reader flag is set.

The paper tape reader motor is turned on by the RFC instruction if not already on, the motor remaining on until approximately 5 seconds after the last fetch instruction.

11.8 Paper Tape Perforator

Punch operation is identical, as regards software, to the standard PDP-8 High Speed Paper Tape Perforation option. The following IOT's are used.

6021 PSF Skip on Punch Flag. That is, skip if punch is ready for the next character.

6022 PCF Clear Punch Flag and Punch Buffer.

6024 PPC Punch Character. An 8 bit character is transferred from AC4-AC11 into the punch buffer and the character is punched.

6026 PLS Punch Load Sequence – Clears Flags and Buffer, loads buffer with character and punches it.

The paper tape perforator motor is turned on by the PPC instruction if not already on, the motor remaining on until approximately 5 seconds after the last PPC instruction.

11.9 Keyboard Reader Modification

The keyboard reader as supplied with the PDP-8 is activated by the clear keyboard flag instruction KCC (6032). This means that when a program is to be operated with the interrupt 'on' the keyboard flag cannot be cleared without fetching a new character from the keyboard reader. For our system we require that the program can be run with the interrupt on but we also require that data for a future peak, scan, etc. remain in the keyboard reader until required, whereupon an appropriate instruction can be executed to get the required information.

For this purpose the unused processor IOT 6004 (A.D.C. conversion) has been rerouted so that it resets the keyboard flag without activating the keyboard paper-tape reader. Thus when a keyboard interrupt occurs the 6004 instruction is used to clear the flag and when data is required at some later time the normal clear keyboard flag instruction is executed. This re-clears the flag and activates the keyboard reader.

Normal operation with the keyboard reader and interrupt 'off', as is used for many PDP-8 library routines, is unaffected by the above modification, and maintenance programs, etc. can be used without erroneous operation occurring.

12. ACKNOWLEDGEMENTS

To anyone familiar with these topics it will be obvious that this project owes a great deal to the X-ray diffractometer constructed by Busing and his collaborators at Oak Ridge. We take pleasure in acknowledging this debt and are grateful to Dr. Busing for private discussions and for supplying his programs to us. We have re-written all the programs we used, and we have done many things in a very different fashion from the Oak Ridge group, but our borrowings from them are nevertheless of fundamental importance.

It is a pleasure also to acknowledge the assistance of R.J. Dullow who worked for many months on the de-bugging of programs and on the initial operating trials. The refinement program CCD/2 was written mainly by R.J. Dullow, and the data-handling program by the late J.R. Thompson. The installation of the instrument was carried out by R.W. Phillips.

Finally we are grateful to the senior staff of the A.A.E.C., in particular to Dr. R. Smith and Dr. J.K. Parry, for their support and encouragement of the project.

13. REFERENCES

Arndt, U.W. and Willis, B.T.M. (1966). – "Single Crystal Diffractometry", Cambridge.

Busing, W.R. and Levy, H.A. (1967). – Acta Cryst. 22, 457.

Programmer's Guide and Systems Handbook – Manufacturers of the PDP-8 Computer.

APPENDIX 1

PDP-8 PROGRAM

The Symbolic Language program, and the assembled Machine Language program, are given as at May 1968. The program is continually being modified and developed in minor ways and, in some cases, is slightly ahead of the description given in the text. The overall procedures are unchanged, but some of the working subroutines are slightly different.

A	5506	AAA	7251	ABIANG	4625	ABS	5570	ABSOL	0051
ACCNT	2046	ACMINS	6000	ACON6	5760	AC1H	0041	AC1L	0042
ADDEG	0307	ADDR	5656	ADJ	7345	ADNANG	3317	ADR	2520
ADRA	0105	ADRB	0015	ADRK	0016	ADRS	2531	ADV	7327
ADVAN	4201	AHKL	4624	ALFZ	5522	ALF1	5525	ALF2	5530
ALG	5255	ALGN	5770	ALIGN	6020	AMOUNT	6142	AMT1	6655
ANGADR	3262	ANGAD1	3457	ANGMAX	0123	ANGPRT	6173	ARG	3170
ARTN	5400	ART1	5415	ART2	5411	ART3	5440	ART4	5472
ASHIFT	4150	AST	7256	AUTO	0146	AUTO	0010	AUT1	0011
AUT2	0012	AUT3	0013	AUT4	0014	AUT5	0015	AUT6	0016
AUT7	0017	AX	0140	A1	2667	B	5511	BACK	1335
BACKGD	4152	BBB	7252	BCC	5124	BELOW	1222	BETZ	5533
BET1	5536	BET2	5541	BGC	2054	BIANG	4400	BLCK	1017
BUF	1622	BUFF	1621	BUFFST	1427	BUFFO	1640	BUFF1	1600
B1	3531	CALC	0130	CALCUL	4561	CCF	6304	CDR	5000
CDS	6334	CENTER	3400	CFLAG	4671	CHAR	0057	CHARA	1411
CHARAC	1344	CHEKK	3642	CHI	0676	CHIN	5575	CHRK	7374
CH1	2716	CLCU	6710	CLOCK	2200	CLOK	2363	CLST	2445
CMS	6302	CNEW	2070	COLON	7364	COMM	1350	COMO	1024
COMP	6375	COMPAR	3543	CONV	3457	CONVPT	0157	COPA	2507
COPB	2510	COPK	2506	COPS	2511	COPY	2503	COS	5310
COUNT	0125	COUNTA	7577	COUNTO	2051	CO1	3203	CO2	3211
CO3	3265	CR	1720	CRB	6322	CRLF	1632	CSI	6314
CSO	6342	CSOSSO	6346	CS1	5154	CS2	5161	CT	0134
CTC	4670	C27	7443	C3	5350	C5	5345	C7	5342
C9	5337	D	2350	DANG	0052	DATA	4173	DC	0742
DCCOLL	4677	DCH	7363	DCOLL	3772	DCOLLA	4042	DD	6367
DEC	7100	DECON	7070	DECONV	7007	DECTST	7064	DEG	5544
DEGD	0530	DEGR	0526	DEN	5514	DGTST	7026	DIGIT	7102
DISMIS	0003	DIVIDE	6361	DIV1	6200	DIV2	6420	DMULT	6221
DNORM	6600	DNUMBR	7103	DO	0725	DONE	6114	DP	0757
DPCSPT	7447	DPCVPT	7446	DPN	7451	DPOUT	7553	DPTK	7376
DRIVE	6175	DROP	7262	DS	2352	DSC	0744	DSO	0727
DSP	0761	DST	0712	DT	0710	DTT	3155	DT2	4222
DT3	4235	DT4	4234	DUBDIV	6472	DUBLAD	7135	DUMMY	7000
DUNORM	6564	DVX	6506	DV2	6522	DV3	6500	ENABLE	6302
END	5302	EOM	4276	ERRCNT	0040	ERROR	6374	ERROR1	6152
ERROR2	3374	EX	3600	EXIT	5742	EXIT1	6641	EXIT2	6651
EXIT5	2672	EXIT6	6171	EXP	0044	EXPONT	0044	EXP1	0050
EX1	0040	EX10	0040	E27	7243	FABS	0007	FADR	7504
FATN	0005	FCC	5126	FCOS	0004	FDPOUT	0015	FETCH	1352
FG	2062	FIA	7541	FIB	7542	FIC	7545	FIGO2	7413
FIGO1	7411	FIGO3	7421	FIN	2664	FINK	5772	FINP	7500
FINPUT	0013	FINS	7507	FIX	5547	FIXC	5565	FIXIN	7536
FIXIT	7240	FIX1	5503	FK	7503	FLAD	5716	FLAG	0061
FLAG1	6762	FLAG2	5503	FLDK	7375	FLDV	6305	FLGT	5676
FLINTP	7400	FLMY	5761	FLOAT	5354	FLOC	7523	FLPT	5705
FLSU	5737	FLT	5363	FNEG	0006	FNOR	0010	FNS	2057
FOPK	7522	FOPS	7526	FOPT	7517	FORW	3104	FOUR	3351
FOUT	7470	FOUTPT	0014	FPAC1	0052	FPNT	5600	FP360	0245
FSIN	0003	FSQR	0001	FSQRT	0002	FSTD	0363	F1	4411
F180	3635	F2	4434	F2CO	0657	F200PT	4147	G	0145
GBG	0155	GEN	7340	GETARG	5154	GETFLG	1022	GETX	1351
GO	0662	GOOF	6162	GO2	5655	GO6	6617	GSTP	0140
H	2351	HALF	3555	HALFPT	0253	HALVE	7254	HC	0743
HH	0135	HI	0055	HIMM	7365	HKL	4166	HKLINP	4164
HKLOPT	4572	HKLOUT	4722	HMAX	0041	HMH	2065	HMIN	0040
HMIN1	2073	HO	0726	HORD	0045	HORDER	0045	HP	0760
HPK	5117	HPL	5121	HS	2353	HSC	0745	HSO	0730
HSP	0762	HST	0713	HT	0711	HTEMP	7122	HTT	3156

HT1	0150	HT2	0151	HT3	0145	HUN	0527	HUND	0531
INDRCT	5664	INIM	0410	INIT	1400	INITA	1411	INPX	1347
INSERT	1364	INT	0000	INTER	2400	INTIN	7000	INWT	2000
IOUT	7453	IPCH	1005	IPCH1	1011	IPFLAG	1020	IPP	4600
IPP1	4621	IRR	4551	ITER1	6756	JMPAD	1354	JMPC	2455
JMPE	2454	JMPO	1025	JUMP	5653	JUMP1	5106	JUMP2	5654
K	0136	KC1	0740	KC2	0741	KEEP	6370	KEEPIT	0615
KEXP	4672	KEY	1233	KEYSER	1000	KMAX	0043	KMIN	0042
KO1	0723	KO2	0724	KPL	5133	KP1	0755	KP2	0756
KT1	0706	KT2	0707	K1	2346	K11	3153	K2	2347
K22	3154	K90	5517	L	0137	LA	2344	LABS	0021
LACCNT	0120	LACMIN	5307	LADNAN	0614	LADR	0022	LALIG	7261
LANGAD	3456	LANGL	0522	LANGLE	0307	LANGLO	0343	LANGPT	0071
LARG	3564	LBCC	0121	LBIANG	5164	LBP	4535	LCHI	0736
LCLK	2430	LCMA	4104	LCOP	0023	LCRLF	0035	LDBLAD	7370
LDPOP	7477	LDPSI	0124	LF	1721	LFG	0123	LFIX	0026
LFIXC	0346	LFIXIN	0031	LFLAD2	6335	LFLT	0033	LFNK	0004
LFNP	0024	LFNS	0122	LFOP	0025	LFOT	0006	LFPT	0007
LFP90	4623	LF200	3316	LGO	0063	LHF	0123	LHF1	0123
LHF2	0124	LHF3	0127	LHKL	4573	LHKLOT	4162	LHMIN	0112
LHTEMP	7372	LHV	0125	LIMIT	0047	LIMITS	3664	LIMITL	0120
LINIM	2012	LINIT	2011	LINT	0002	LINTIN	0030	LIOUT	0062
LIPCH	0107	LKEY	2425	LKEYS	1004	LK90	5353	LLLTEN	4375
LLTEMP	7373	LMANEG	0535	LMAPOS	0534	LMARK	3025	LMAX	0045
LMI	2361	LMIC	0406	LMIN	0044	LMIO	0405	LMIP	0407
LMIT	0404	LMLT10	7371	LMOTST	0027	LMOVE	3024	LMSIGN	7367
LNEG	0020	LNO	0166	LNOHOP	1050	LO	0056	LOBSCU	0036
LOESPT	0157	LOCSGL	1356	LOMG	0721	LOMGA	0034	LOMM	7366
LONLY	5134	LOOOP	0421	LOOP01	5627	LOOP1	2636	LOP	6622
LOPCH	0106	LORD	0046	LORDER	0046	LOT	0131	LPHI	0753
LPRSW	7074	LPUN	2426	LSAC	1501	LSET	0537	LSGTM	2362
LSTART	0201	LSTCT	0110	LSTD	0072	LSTEP	0065	LSTRNG	0067
LSTS	7476	LSTSG	7260	LSUPER	0032	LTEL	2427	LTEMP	7123
LTEN	7257	LTKCNT	0064	LTRANS	2356	LTTH	0704	LTYPOT	0066
LTY2	1051	LUB	0111	LVFOUT	7475	LWHERE	6773	LWVL	0126
LX	5504	LXSQR	5505	LZER	2720	LZERGO	1373	LZERO	0070
MARK	3050	MARK2	3125	MARK3	3136	MARK4	3140	MASK	7101
MASK3	5660	MASK5	5661	MASK7	5107	MCH	7170	MDRA	2247
MDRB	2337	MDRCE	2334	MDRCI	2220	MDRCX	2340	MDR1	2256
MDR2	2302	MDR3	2304	MDR4	2322	MDR5	2333	MEASUR	5140
MERROR	0540	MF200	0250	MI	0521	MIA	2354	MIC	0746
MIF	6544	MIND	0532	MINS	6373	MINS2	6371	MINUS	7075
MINUS2	6400	MIN14	2472	MIN40	3562	MIN7	7360	MIN9	7076
MIO	0731	MIP	0763	MIT	0714	MI1	2364	MI2	2365
MK0077	1673	MK0177	5662	MK177	1361	MK7757	1546	MNAME	1353
MOD	5213	MODT	0005	MONDET	4654	MOTC	0461	MOTCZ	3013
MOTOR	0416	MOTOZ	3006	MOTPZ	3020	MOTST	0400	MOTST1	0161
MOTST2	0162	MOTTZ	3001	MOT1	0434	MOT2	0471	MOT3	0476
MOT4	0504	MOT5	0507	MOT6	0510	MOT7	0514	MOVCHK	3026
MOVECZ	3012	MOVEOZ	3005	MOVEPZ	3017	MOVETZ	3000	MP1	6466
MP2	6471	MP2PT	6356	MP3	6467	MP4	6437	MP4PT	6355
MP5	6465	MP5PT	6357	MRBOUT	7073	MSGNPT	7450	MSIGN	7147
MS0077	1357	MULT	5767	MULT10	7104	MULT2	7124	MZERO	7077
M10	0074	M100	3563	M11	4550	M12	0075	M144	3161
M180	3641	M2C	1552	M24	2721	M257	1360	M276	3160
M30	7255	M310	0006	M360	0244	M4	0073	M40	1670
M5	0656	M6	2357	M62	0550	M8	0074	NAME	0536
NANG	0142	NARR	0105	NCP	0147	NCT2	0147	NEG	5303
NEGIT	6653	NEG3	2505	NEX	0153	NH1	0150	NH2	0151
NLIST	1355	NO	1620	NOGO	6125	NOHERE	6137	NOHOP	2631

NOPUN	1437	NORM	5776	NORMF	6360	NSHIFT	4151	NSTP	0141
NSUB	0154	O	0000	OBCURE	5372	OBERR	3611	OBSCUR	3670
OB1	3702	OB2	3722	OB5	3754	OB6	3757	OFFPRG	1067
OMEGA	3360	OMG	0673	OMGA	2005	ONE	0076	OPCH	1554
OPCHH	1430	OPCHR	1441	OPMINS	5741	OSC1	3421	OSC2	3432
OTHER	3467	OUTGO	6116	OVER	3152	OVER1	0043	OVER2	0047
OVR	2345	OVRC	0737	OVRO	0722	OVRP	0754	OVRS	0523
OVRT	0705	P	1516	PAGENO	5663	PARG	0600	PBIANG	1370
PCHI	0116	PCNEW	0127	PER	7444	PFLAG	0132	PFSTO	0113
PH	5153	PHA	1372	PHALF	4376	PHB	4574	PHI	0701
PHIN	3364	PHKLOT	3610	PHMH	0124	PHT	0050	PI	5331
PICH	5241	PIOT	5334	PLEAD	1723	PLUS4	3176	PL307	3157
PL5	1553	PMF	6341	PMOT	2355	PMR	6321	PNO	1625
PNORM	5371	PNTR	5317	POBSCR	0036	POBSCU	0036	POMG	0115
POSN	0261	POSN1	0270	POSN2	0264	PPHI	0117	PPK	4020
PP200	1363	PRCH	7314	PRDY	1630	PREV	7377	PRINT	1632
PRSGN	7307	PRSW	7452	PRTANG	6763	PSTDC	1223	PTEN	0345
PTRANS	3151	PTTH	0114	PUNSER	1547	PZERGO	2717	P1	1542
P10	2711	P100	1671	P200	1733	P24	3560	P240	1672
P307	2366	P360	3640	P4	1371	P40	3561	QUOL	0051
RAD	5366	RAR1	6572	RAR2	6372	RDH	6331	RDL	6332
RDLCDS	6336	RDY	1623	RECHEK	3674	RELOOP	2224	REPEAT	3473
REST	6366	RETN1	3616	RETN2	6151	RETURN	6364	RGI	4120
RG10	4106	RN	0347	RN1	0356	ROND	7223	ROOTGO	6745
ROUND	0103	ROWCOL	4536	RSI	6312	SAV	5657	SAVA	1430
SAVE	1441	SAVFAC	4706	SCAL	7216	SCALE	7435	SCV1	0140
SECOND	1311	SECX	1273	SEL1	5005	SEL10	5101	SEL13	5135
SEL15	5074	SEL2	5011	SEL3	5022	SEL4	5033	SEL6	5065
SEL7	5077	SEL9	5071	SENT	1270	SERIUS	4701	SETA	0052
SETB	0333	SETTIM	4732	SETTTH	0601	SETUP	0202	SGN	6365
SGNL	1277	SGTK	1017	SGTM	1053	SHIFT	6076	SIGH	6336
SIGN	0060	SIGN1	6654	SIGTBL	2541	SIGX	1346	SINE	5200
SINLIM	3661	SLASH	1052	SL1	5041	SL2	5046	SL3	5054
SMACLA	6342	SNC	6301	SNS	6311	SNSWIT	6347	SOB2	3670
SOB3	3600	SOM	4163	SORT	1322	SORT1	1333	SPACE	7361
SPACLA	6362	SPARE	0000	SPCH	7171	SQCON1	6761	SQEND	6754
SQEND1	6751	SQROOT	6656	SQUARE	6163	SRB	6324	SSK1	0006
SSK2	0075	SSO	6344	SSR	4277	SSR1	4304	SSR2	4325
SS1	0524	SS2	0525	STAB	2456	STAK	2473	STAND	0164
START	2000	STATUS	2431	STCH	7312	STCT	4626	STD	1200
STEP	3262	STEPNO	7172	STEPPT	0156	STP6	3313	STRING	1706
STRT	7020	STRTOP	7016	STSG	7162	ST1	3262	SUB	0623
SUBFT	0052	SUBK	0105	SUBTM	0662	SUB1	0626	SUM	5105
SUPER	1026	SUPPT	0160	SUPPTT	0165	SVAC	2474	SVLK	2475
SWT1	2444	SWT2	2444	SWT3	2444	SYMM	0046	TABLE	5665
TABLE6	6545	TACNT	3367	TAG1	6147	TAG2	6150	TELE	1502
TELE5	1537	TELSER	1543	TEMMPY	1243	TEMP	0105	TEMPRY	1275
TEMPY	3660	TEMP11	1722	TEM1	2360	TEM2	2344	TEM3	3060
TEM4	3147	TEM5	3047	TEN	6774	TENPT	7445	TERINT	2476
TERMIN	1337	TEST	7047	TESTK	7315	TESTST	2436	TESTU	7277
TEST2	6143	TEST3	6144	TEST4	6145	TEST5	6146	THIR	6470
THR	0104	TIME	0133	TINC	3162	TINIT	0254	TKCNT	3200
TRANS	7560	TRANSP	1362	TSGNL	0010	TTH	0670	TTHETA	3354
TV1	4711	TV2	4714	TV3	4717	TWO	0101	TWOPI	5326
TYCF	1065	TYCN	1064	TYP	0152	TYPE	1060	TYPOPT	3333
TY1	1660	TY2	1674	UB	2013	UBR	4673	UNB	0120
UNORM	5771	VARY	3617	VFOUT	7200	WAVEL	6573	WDK	0144
WHERE	0615	WVL	2103	X	5320	XA	3634	XAF	0050
XANG	0143	XCI	0053	XFI	0054	XOM	0052	XSQR	5323
XTTH	0051	ZCH	7362	ZERGO	0276	ZERGO1	0163	ZERO	7175

```
                /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
                DUMMY=      NOP
                FSQR=       0001
                FSQRT=      0002
                FSIN=       0003
                FCOS=       0004
                FATN=       0005
                FNEG=       0006
                FINPUT=     0013
                FABS=       0007
                FNOR=       0010
                FOUTPT=     0014
                FDPOUT=     0015

                /INPUT-OUTPUT TRANSFER PULSE INSTRUCTIONS
                SNC=        6301                /SKIP IF NOT CLOCK
                CMS=        6302                /CLEAR MONITOR SCALER
                CCF=        6304                /CLEAR CLOCK FLAG
                SNS=        6311                /SKIP IF NOT STATUS-IN
                RSI=        6312                /READ STATUS REGISTER
                CSI=        6314                /CLEAR STATUS-IN FROM C(AC)
                PMR=        6321                /PULSE MOTOR REVERSE
                CRB=        6322                /CLEAR RECORDER BUFFER
                SRB=        6324                /SET RECORDER BUFFER
                RDH=        6331                /READ DETECTOR HIGH
                RDL=        6332                /READ DETECTOR LOW
                CDS=        6334                /CLEAR DETECTOR SCALER
                RDLCDS=     6336                /RDL CDS
                PMF=        6341                /PULSE MOTOR FORWARD
                CSO=        6342                /CLEAR STATUS OUT
                SSO=        6344                /SET STATUS OUT FROM C(AC)
                CSOSSO=     6346                /CSO SSO
                ENABLE=     6302
                C=          0
                SPARE=      0

                /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
                /PAGE       0
                *           0
0000 2000 INT,              START
0001 5602                   JMP  I    LINT      /SERVICE INTERRUPT
0002 2400 LINT,             INTER
0003 2476 DISMIS,           TERINT              /POINTER
0004 7400 LFNK,             7400                /FLOATING INPUT
0005 1000 MODT,             1000                /MODT ENTRY
0006 7470 LFOT,             FOUT                /FLOATING OUTPUT
0007 5600 LFPT,             5600                /FLOATING INTERPRETER
0010 0000 AUT0,             0                   /AUTO-INDEX REGISTERS
0011 0000 AUT1,             0
0012 0000 AUT2,             0
0013 0000 AUT3,             0
0014 0000 AUT4,             0
0015 0000 AUT5,             0
0016 0000 AUT6,             0
0017 0000 AUT7,             0
0020 5303 LNEG,             NEG                 /LOCATIONS OF SUBROUTINES
0021 5570 LABS,             ABS
0022 2520 LADR,             ADR
0023 2503 LCOP,             COPY
0024 7500 LFNP,             FINP
```

```
0025 7517 LFOP,     FOPT
0026 5547 LFIX,     FIX
0027 0400 LMOTST,   MOTST
0030 7000 LINTIN,   INTIN
0031 7536 LFIXIN,   FIXIN
0032 1026 LSUPER,   SUPER
0033 5354 LFLT,     FLOAT
0034 2005 LOMGA,    OMGA
0035 1632 LCRLF,    CRLF
0036 3670 LOBSCU,   OBSCUR
0037 0000           SPARE
          /LOCATIONS USED BY FLOATING POINT INTERPRETER
0040 0000 EX1,      0                       /OPERAND
0041 0000 AC1H,     0
0042 0000 AC1L,     0
0043 0000 OVER1,    0
0044 0000 EXP,      0                       /FLOATING ACCUMULATOR
0045 0000 HORD,     0
0046 0000 LORD,     0
0047 0000 OVER2,    0
0050 0000 EXP1,     0
0051 0000 QUOL,     0
0052 0000 FPAC1,    0                       /TEMPORARY FLOATING LOCATION
0053 0000           0
0054 0000           0
0055 0000 HI,       0
0056 0000 LO,       0
0057 0000 CHAR,     0
0060 0000 SIGN,     0
0061 0000 FLAG,     0
0062 7453 LIOUT,    IOUT
0063 0662 LGO,      GO
0064 3200 LTKCNT,   TKCNT
0065 3262 LSTEP,    STEP
0066 3333 LTYPOT,   TYPOPT
0067 1706 LSTRNG,   STRING
0070 7175 LZERO,    ZERO
0071 6763 LANGPT,   PRTANG
0072 1200 LSTO,     STO
0073 7774 M4,       -4
0074 7770 M10,      -10
0075 7766 M12,      -12
0076 0001 ONE,      1
0077 2000           2000
0100 0000           0
0101 0002 TWO,      2
0102 2000           2000
0103 0000 ROUND,    0
0104 0003 THR,      3
0105 0000 TEMP,     0
          M310=     LFOT
          SSK1=     M310
          SSK2=     M12
0106 1430 LOPCH,    OPCHH
0107 1005 LIPCH,    IPCH
0110 4626 LSTCT,    STCT
0111 2013 LUB,      UB
0112 2073 LHMIN,    HMIN1
0113 0363 PFSTO,    FSTO
0114 0670 PTTH,     TTH
```

```
0115 0673 PDMG,      DMG
0116 0676 PCHI,      CHI
0117 0701 PPHI,      PHI
0120 2046 LACCNT,    ACCNT
0121 2054 LBGC,      BGC
0122 2057 LFNS,      FNS
0123 2062 LFG,       FG
0124 2065 PHMH,      HMH
0125 2051 COUNT,     COUNTC
0126 2103 LWVL,      WVL
0127 2070 PCNEW,     CNEW
0130 4706 CALC,      SAVFAC
0131 0000 LOT,       0
0132 0000 PFLAG,     0
0133 0002 TIME,      2
0134 0000 CT,        0
0135 0000 HH,        0
0136 0000 K,         0
0137 0000 L,         0
0140 0000 GSTP,      0
0141 0000 NSTP,      0
0142 0000 NANG,      0
0143 0000 XANG,      0
0144 0000 WDK,       0
0145 0000 G,         0
0146 0000 AUTO,      0
0147 0000 NCT2,      0
0150 0000 NH1,       0
0151 0000 NH2,       0
0152 0000 TYP,       0
0153 0000 NEX,       0
0154 0000 NSUB,      0
0155 0000 GBG,       0
0156 0000 STEPPT,    0
0157 0000 LOBSPT,    0
0160 0000 SUPPT,     0
0161 0000 MOTST1,    0
0162 0000 MOTST2,    0
0163 0000 ZERO01,    0
0164 0000 STAND,     0
0165 0000 SUPPTT,    0
0166 1620 LNO,       NO
          LDPSI=     PHMH

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      1
          *          200
0200 5601            JMP  I    LSTART
0201 2000 LSTART,    START
          /SEPARATE ANGLE INTO DEGREES AND 200THS
          /ROUTINE TO BRING ANGLE TO BETWEEN -180 AND 180 DEGREES
          /CALLING SEQUENCE
          /          ANGLE IN C(FAC)
          /          JMS SETUP
          /          ADDRESS OF DEGREES PART
0202 0000 SETUP,     0                    /SUBROUTINE ENTRY
0203 1602            TAD  I    SETUP      /GET DEGREE ADDRESS
0204 3307            DCA       ADDEG
0205 2202            ISZ       SETUP      /SET RETURN
0206 4407            JMS  I    7
```

```
0207 6052            FPUT       SETA
0210 4245            FDIV       FP360
0211 1653            FADD  I    HALFPT
0212 1245            FADD       FP360
0213 1746            FADD  I    LFIXC      /ADD 2**22
0214 2746            FSUB  I    LFIXC      /SUB 2**22
0215 0006            FNEG
0216 1245            FADD       FP360
0217 3245            FMPY       FP360
0220 1052            FADD       SETA
0221 1245            FADD       FP360
0222 1103            FADD       ROUND
0223 6052            FPUT       SETA
0224 0000            FEXT
0225 4426            JMS   I    LFIX       /TRUNCATE TO INTEGER
0226 3333            DCA        SETB
0227 1333            TAD        SETB
0230 1244            TAD        M360
0231 3707            DCA   I    ADDEG      /PUT INTO LOCATION FOR DEGREES
0232 1333            TAD        SETB
0233 4433            JMS   I    LFLT       /FLOAT CONTENTS OF ACCUMULATOR
0234 4407            JMS   I    7
0235 2052            FSUB       SETA
0236 3250            FMPY       MF200      /-200 FLOATING
0237 0000            FEXT
0240 4426            JMS   I    LFIX       /TRUNCATE TO INTEGER
0241 2307            ISZ        ADDEG      /SET ADDRESS OF 200THS
0242 3707            DCA   I    ADDEG      /PUT 200THS INTO LOCATION
0243 5602            JMP   I    SETUP      /RETURN
0244 7230 M360,      -550
0245 0011 FP360,     11
0246 2640            2640
0247 0000            0
0250 0010 MF200,     0010
0251 4700            4700
0252 0000            0000
0253 3555 HALFPT,    HALF
          /SET CURRENT POSITIONS OF SHAFTS BY TYPING IN DIAL SETTINGS
0254 4261 TINIT,     JMS        POSN
0255 0710            DT
0256 0725            DO
0257 0742            DC
0260 0757            DP
0261 0000 POSN,      0
0262 1073            TAD        M4
0263 3276            DCA        ZERGO
0264 1661 POSN2,     TAD   I    POSN
0265 3270            DCA        POSN1
0266 4404            JMS   I    4          /INPUT ONE NUMBER
0267 4202            JMS        SETUP
0270 0000 POSN1,     0
0271 2261            ISZ        POSN
0272 2276            ISZ        ZERGO
0273 5264            JMP        POSN2
0274 4276            JMS        ZERGO
0275 5434            JMP   I    LOMGA
0276 0000 ZERGO,     0                     /ENTRY
0277 1276            TAD        ZERGO
0300 3163            DCA        ZERGO1
0301 1114            TAD        PTTH
```

```
0302 3307            DCA        LANGLE
0303 1130            TAD        CALC
0304 3310            DCA        LANGLE+1
0305 4423            JMS   I    LCOP
0306 7764            -14
0307 0670 LANGLE,    TTH
0310 4706            SAVFAC
0311 4407            JMS   I    7
0312 5745            FGET  I    PTEN
0313 0006            FNEG
0314 6514            FPUT  I    PTTH
0315 6515            FPUT  I    POMG
0316 6516            FPUT  I    PCHI
0317 6517            FPUT  I    PPHI
0320 0000            FEXT
0321 4427            JMS   I    LMOTST
0322 7000            NOP
0323 4407            JMS   I    7
0324 5745            FGET  I    PTEN
0325 6514            FPUT  I    PTTH
0326 6515            FPUT  I    POMG
0327 6516            FPUT  I    PCHI
0330 6517            FPUT  I    PPHI
0331 0000            FEXT
0332 4427            JMS   I    LMOTST
0333 0000 SETB,      0
0334 1130            TAD        CALC
0335 3342            DCA        LANGLO-1
0336 1114            TAD        PTTH
0337 3343            DCA        LANGLO
0340 4423            JMS   I    LCOP
0341 7764            -14
0342 4706            SAVFAC
0343 0670 LANGLO,    TTH
0344 5563            JMP   I    ZERGO1      / RETURN
0345 6774 PTEN,      TEN
0346 5565 LFIXC,     FIXC
          SETA=      FPAC1
          ADDEC=     LANGLE
          /PROGRAM FOR MANUAL SCAN
0347 4430 RN,        JMS   I    LINTIN
0350 3142            DCA        NANG
0351 4430            JMS   I    LINTIN
0352 3143            DCA        XANG
0353 7040            CMA
0354 3144            DCA        WDK
0355 2146            ISZ        AUTO
0356 4465 RN1,       JMS   I    LSTEP
0357 4464            JMS   I    LTKCNT      /COUNT
0360 4466            JMS   I    LTYPOT
0361 4463            JMS   I    LGO         /CHECK FLAG
0362 5356            JMP        RN1
0363 0000 FSTO,      0
0364 0000            0
0365 0000            0
0366 0000            0
0367 0000            0

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      2
```

```
          *          400
          /MOTOR STARTING ROUTINE
0400 0000 MOTST,     0                         /ENTRY
0401 1200            TAD        MOTST
0402 3161            DCA        MOTST1
0403 4216            JMS        MOTOR
0404 0714 LMIT,      MIT
0405 0731 LMIO,      MIO
0406 0746 LMIC,      MIC
0407 0763 LMIP,      MIP
          /INITIALISE MOTORS
0410 0000 INIM,      0                         /ENTRY
0411 3604            DCA   I    LMIT           /RESET MOTOR FLAGS
0412 3605            DCA   I    LMIO
0413 3606            DCA   I    LMIC
0414 3607            DCA   I    LMIP
0415 5610            JMP   I    INIM           /RETURN
          /MOTOR DRIVE SUBROUTINE
0416 0000 MOTOR,     0
0417 1073            TAD        M4
0420 3210            DCA        INIM
0421 1616 LOOOP,     TAD   I    MOTOR
0422 3234            DCA        MOT1
0423 3634            DCA   I    MOT1           /RESET MOTOR FLAG
0424 1234            TAD        MOT1           /SET UP COPY ROUTINE
0425 1074            TAD        M10            /-8
0426 3234            DCA        MOT1
0427 1234            TAD        MOT1
0430 3307            DCA        MOT5
0431 2216            ISZ        MOTOR          /SET RETURN
0432 4423            JMS   I    LCOP           /COPY ARGUMENTS
0433 7763            -15                       /FROM
0434 0000 MOT1,      0                         /VARIABLE
0435 0522            LANGL                     /TO LANGL
          /SEPARATE ANGLE INTO DEGREES AND TWO-HUNDREDTHS
0436 4407            JMS   I    7
0437 5722            FGET  I    LANGL          /PUT ANGLE INTO F.A.
0440 0000            FEXT
0441 4737            JMS   I    LSET           /GET DEGREES AND 200THS
0442 0530            DEGD
0443 1330            TAD        DEGD
0444 1334            TAD        LMAPOS
0445 7700            SMA   CLA
0446 5340            JMP        MERROR
0447 1330            TAD        DEGD
0450 1335            TAD        LMANEG
0451 7710            SPA   CLA
0452 5340            JMP        MERROR
0453 1330            TAD        DEGD           /COMPLEMENT DESIRED POSITION
0454 7041            CMA   IAC
0455 3330            DCA        DEGD
0456 1331            TAD        HUND
0457 7041            CMA   IAC
0460 3331            DCA        HUND
          /COMPARE CURRENT AND DESIRED POSITIONS
0461 1326 MOTC,      TAD        DEGR
0462 1330            TAD        DEGD
0463 7440            SZA
0464 5271            JMP        MOT2
0465 1327            TAD        HUN
```

```
0466 1331            TAD        HUND
0467 7450            SNA
0470 5304            JMP        MOT4        /CURRENT AND DESIRED AGREE
0471 7710 MOT2,      SPA  CLA
0472 5276            JMP        MOT3        /PULSE FORWARD,OVERSHOOT = 0
0473 2330            ISZ        DEGD        /PULSE REVERSE AND OVERSHOOT
0474 7000            NOP
0475 7240            CLA  CMA               /SET OVERSHOOT INDICATOR
0476 3323 MOT3,      DCA        OVRS
0477 1350            TAD        M62         /SET SLOW START
0500 3324            DCA        SS1
0501 1075            TAD        SSK2
0502 3325            DCA        SS2
0503 2332            ISZ        MIND        /SET MOTOR INDICATOR
0504 4423 MOT4,      JMS  I     LCOP        /COPY ARGUMENTS
0505 7763            -15                    /FROM
0506 0522            LANGL                  /LANGL
0507 0000 MOT5,      0                      /TO VARIABLE
0510 2210 MOT6,      ISZ        INIM
0511 5221            JMP        LOOOP
0512 2321            ISZ        MI          /SET OVERALL MOTOR INDICATOR
0513 2161            ISZ        MOTST1
0514 1321 MOT7,      TAD        MI
0515 7650            SNA  CLA
0516 5561            JMP  I     MOTST1
0517 4432            JMS  I     LSUPER
0520 5314            JMP        MOT7
0521 0000 MI,        0
0522 0000 LANGL,     0
0523 0000 OVRS,      0
0524 0000 SS1,       0
0525 0000 SS2,       0
0526 0000 DEGR,      0
0527 0000 HUN,       0
0530 0000 DEGD,      0
0531 0000 HUND,      0
0532 0000 MIND,      0
0533 0000            0
0534 0000 LMAPOS,    0
0535 0000 LMANEG,    0
0536 0000 NAME,      0
0537 0202 LSET,      SETUP
0540 1336 MERROR,    TAD        NAME
0541 3345            DCA        .+4
0542 4467            JMS  I     LSTRNG
0543 3664            LIMITS
0544 4467            JMS  I     LSTRNG
0545 0000            0
0546 4435            JMS  I     LCRLF
0547 5561            JMP  I     MOTST1
0550 7716 M62,       -62

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      3
          *          600
0600 3170 PARG,      ARG
0601 4600 SETTTH,    JMS  I     PARG
0602 3142            DCA        NANG
0603 4614            JMS  I     LADNAN
0604 3215            DCA        KEEPIT
```

```
0605 4404              JMS  I    LFNK
0606 4407              JMS  I    7
0607 6615              FPUT I    KEEPIT
0610 0000              0
0611 4427              JMS  I    LMOTST
0612 7000              NOP
0613 5434              JMP  I    LOMGA
0614 3317 LADNAN,      ADNANG
          /ROUTINE TO SET FLOATING POINT ANGLES TO CURRENT POSITIONS
          /RETURNS WITH CURRENT ANGLES IN FLOATING LOCATIONS
0615 0000 WHERE,       0                        /ENTRY
0616 4223              JMS       SUB
0617 0710              DT
0620 0725              DO
0621 0742              DC
0622 0757              DP
0623 0000 SUB,         0
0624 1073              TAD       M4
0625 3105              DCA       SUBK
0626 1623 SUB1,        TAD  I    SUB            /ADDRESS OF CURRENT DEGREE
0627 3262              DCA       SUBTM
0630 1662              TAD  I    SUBTM          /GET CURRENT DEGREE POSITION
0631 4433              JMS  I    LFLT           /FLOAT DEGREES
0632 4407              JMS  I    7
0633 6052              FPUT      SUBFT
0634 0000              FEXT
0635 2262              ISZ       SUBTM
0636 1662              TAD  I    SUBTM
0637 4433              JMS  I    LFLT           /FLOAT 200THS
0640 1262              TAD       SUBTM
0641 1256              TAD       M5
0642 3262              DCA       SUBTM
0643 1662              TAD  I    SUBTM          /ADDRESS OF FLOATING ANGLE
0644 3262              DCA       SUBTM
0645 4407              JMS  I    7
0646 4257              FDIV      F200           /DIVIDE BY 200
0647 1052              FADD      SUBFT          /ADD DEGREES
0650 6662              FPUT I    SUBTM          /RESULT INTO FLOATING ANGLE
0651 0000              FEXT
0652 2223              ISZ       SUB            /NEXT ANGLE
0653 2105              ISZ       SUBK           /TEST COUNTER
0654 5226              JMP       SUB1
0655 5615              JMP  I    WHERE          /RETURN
0656 7773 M5,          -5
0657 0010 F200,        0010
0660 3100              3100
0661 0000              0000
0662 0000 GO,          0                        /ENTRY
0663 1146              TAD       AUTO
0664 7450              SNA                      /TEST PROGRAM FLAG
0665 5434              JMP  I    LOMGA          /RETURN TO WAITING LOOP
0666 3146              DCA       AUTO
0667 5662              JMP  I    GO             /RETURN TO PROGRAM

          SUBK=        TEMP
          SUBTM=       GO
          KEEPIT=      WHERE
          SUBFT=       FPAC1
0670 0000 TTH,         0
0671 0000              0
```

```
0672 0000           0
0673 0000 OMG,      0
0674 0000           0
0675 0000           0
0676 0000 CHI,      0
0677 0000           0
0700 0000           0
0701 0000 PHI,      0
0702 0000           0
0703 0000           0
0704 0670 LTTH,     TTH                 /DATA FOR TWOTHETA
0705 0000 CVRT,     0
0706 0000 KT1,      0
0707 0000 KT2,      0
          DT=       KT2+1
          HT=       KT2+2
          *         .+2
0712 0000 DST,      0
0713 0000 HST,      0
0714 0000 MIT,      0
0715 3777           3777
0716 7564           -214
0717 0137           137
0720 3354           TTHETA
0721 0673 LOMG,     OMG                 /DATA FOR OMEGA
0722 0000 OVRO,     0
0723 0000 KO1,      0
0724 0000 KO2,      0
          DO=       KO2+1
          HO=       KO2+2
          *         .+2
0727 0000 DSO,      0
0730 0000 HSO,      0
0731 0000 MIO,      0
0732 5777           5777
0733 7470           -310
0734 0310           310
0735 3360           OMEGA
0736 0676 LCHI,     CHI                 /DATA FOR CHI
0737 0000 OVRC,     0
0740 0000 KC1,      0
0741 0000 KC2,      0
          DC=       KC2+1
          HC=       KC2+2
          *         .+2
0744 0000 DSC,      0
0745 0000 HSC,      0
0746 0000 MIC,      0
0747 6777           6777
0750 7641           -137
0751 0137           137
0752 5575           CHIN
0753 0701 LPHI,     PHI                 /DATA FOR PHI
0754 0000 OVRP,     0
0755 0000 KP1,      0
0756 0000 KP2,      0
          DP=       KP2+1
          HP=       KP2+2
          *         .+2
0761 0000 DSP,      0
```

```
0762 0000 HSP,      0
0763 0000 MIP,      0
0764 7377           7377
0765 7470           -310
0766 0310           310
0767 3364           PHIN

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     4
          *         1000
1000 6004 KEYSER,   KCF                     /CLEAR,DO NOT FETCH
1001 6034           KRS                     /READ,DO NOT FETCH
1002 4604           JMS  I    LKEYS
1003 1016           BLCK-1
1004 1233 LKEYS,    KEY
1005 0000 IPCH,     0
1006 7130           CLL  CML  RAR
1007 3220           DCA       IPFLAG
1010 6032           KCC                     /FETCH CHARACTER
1011 1220 IPCH1,    TAD       IPFLAG
1012 7540           SMA  SZA
1013 5605           JMP  I    IPCH
1014 7300           CLL  CLA
1015 4432           JMS  I    LSUPER
1016 5211           JMP       IPCH1
1017 0000 BLCK,     0                       /SIGNAL EXPECTED
1020 0000 IPFLAG,   0                       /INPUT FLAG
1021 0000           0                       /COMMAND
1022 0000 GETFLG,   0                       /GET FLAG
1023 6032           KCC                     /FETCH CHARACTER
1024 0000 COMO,     0                       /COMMAND(SORTED)
1025 0000 JMPO,     0                       /JUMP(SORTED)
          SGTK=     BLCK
1026 0000 SUPER,    0                       /SUPERVISOR ROUTINE
1027 1226           TAD       SUPER
1030 3160           DCA       SUPPT
1031 1224           TAD       COMO
1032 3651           DCA  I    LTY2
1033 3224           DCA       COMO
1034 1651           TAD  I    LTY2
1035 7650           SNA  CLA
1036 5650           JMP  I    LNOHOP
1037 4435           JMS  I    LCRLF
1040 3132           DCA       PFLAG
1041 1252           TAD       SLASH
1042 4506           JMS  I    LOPCH
1043 1651           TAD  I    LTY2
1044 4651           JMS  I    LTY2
1045 4435           JMS  I    LCRLF
1046 3163           DCA       ZERGO1
1047 5625           JMP  I    JMPO
1050 2631 LNOHOP,   NOHOP
1051 1674 LTY2,     TY2
1052 0257 SLASH,    257
          /ROUTINE TO LIMIT TIME FOR COMMANDS
1053 1217 SGTM,     TAD       SGTK
1054 7640           SZA  CLA
1055 2217           ISZ       SGTK
1056 7000           NOP
1057 5403           JMP  I    DISMIS
```

```
1060 2132 TYPE,     ISZ        PFLAG
1061 4205           JMS        IPCH
1062 4506           JMS  I     LOPCH
1063 5261           JMP        TYPE+1
1064 7001 TYON,     IAC
1065 3152 TYOF,     DCA        TYP
1066 5650           JMP  I     LNOHOP
1067 3146 OFFPRG,   DCA        AUTO
1070 5650           JMP  I     LNOHOP

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     5
          *         1200
1200 0000 STD,      0
1201 1200           TAD        STD
1202 3164           DCA        STAND
1203 1113           TAD        PFSTD       /FSTD
1204 3222           DCA        BELOW
1205 1222           TAD        BELOW
1206 1371           TAD        P4
1207 3223           DCA        PSTDC       /COUNTER
1210 2623           ISZ  I     PSTDC
1211 5600           JMP  I     STD
1212 1622           TAD  I     BELOW
1213 7041           CMA  IAC
1214 3623           DCA  I     PSTDC
1215 2222           ISZ        BELOW
1216 1372           TAD        PHA
1217 3223           DCA        PSTDC
1220 4423           JMS  I     LCOP
1221 7775           -3
1222 0000 BELOW,    0
1223 0000 PSTDC,    0
1224 4773           JMS  I     LZERGO
1225 3163           DCA        ZERGO1
1226 4770           JMS  I     PBIANG
1227 7410           SKP
1230 4436           JMS  I     POBSCU
1231 4463           JMS  I     LGO
1232 5564           JMP  I     STAND
          /INPUT ROUTINES
1233 0000 KEY,      0
1234 0361           AND        MK177
1235 1363           TAD        PP200
1236 3344           DCA        CHARAC
1237 1633           TAD  I     KEY
1240 3243           DCA        .+3
1241 4762           JMS  I     TRANSP
1242 7771           -7
1243 0000 TEMMPY,   0
1244 1345           SIGX-1
1245 1344           TAD        CHARAC      /TEST FOR SLASH
1246 1360           TAD        M257
1247 7650           SNA  CLA
1250 5270           JMP        SENT
1251 1346           TAD        SIGX        /SIGNAL EXPECTED?
1252 7640           SZA  CLA
1253 5277           JMP        SGNL
1254 1347           TAD        INPX        /TEST INPUT EXPECTED
1255 7700           SMA  CLA
```

```
1256 5262            JMP       .+4
1257 1344            TAD       CHARAC
1260 3347            DCA       INPX
1261 5337            JMP       TERMIN
1262 1351            TAD       GETX        /TEST CHARACTER EXPECTED
1263 7700            SMA  CLA
1264 5337            JMP       TERMIN
1265 1344            TAD       CHARAC
1266 3351            DCA       GETX
1267 5337            JMP       TERMIN
1270 7001 SENT,      IAC
1271 3346            DCA       SIGX        /SET SIGNAL EXPECTED
1272 3350            DCA       COMM
1273 1352 SECX,      TAD       FETCH
1274 3275            DCA       TEMPRY
1275 0000 TEMPRY,    0
1276 5337            JMP       TERMIN
1277 1350 SGNL,      TAD       COMM
1300 7640            SZA  CLA
1301 5311            JMP       SECOND
1302 1344            TAD       CHARAC
1303 0357            AND       MSOC77
1304 7106            CLL  RTL
1305 7006            RTL
1306 7006            RTL
1307 3350            DCA       COMM
1310 5273            JMP       SECX
1311 1344 SECOND,    TAD       CHARAC
1312 0357            AND       MSOC77
1313 1350            TAD       COMM
1314 3350            DCA       COMM
1315 3346            DCA       SIGX
1316 1355            TAD       NLIST
1317 3243            DCA       TEMMPY
1320 1356            TAD       LOCSGL
1321 3010            DCA       TSGNL
1322 1410 SORT,      TAD  I    TSGNL
1323 7041            CMA  IAC
1324 1350            TAD       COMM
1325 7650            SNA  CLA
1326 5333            JMP       SORT1
1327 2010            ISZ       TSGNL
1330 2243            ISZ       TEMMPY
1331 5322            JMP       SORT
1332 5337            JMP       TERMIN
1333 1410 SORT1,     TAD  I    TSGNL
1334 3354            DCA       JMPAD
1335 1350 BACK,      TAD       COMM
1336 3353            DCA       MNAME
1337 1633 TERMIN,    TAD  I    KEY
1340 3344            DCA       .+4
1341 4762            JMS  I    TRANSP
1342 7771            -7
1343 1345            SIGX-1
1344 0000 CHARAC,    0
1345 5403            JMP  I    DISMIS
1346 0000 SIGX,      0
1347 0000 INPX,      0
1350 0000 COMM,      0
1351 0000 GETX,      0
```

```
1352 0000 FETCH,    0
1353 0000 MNAME,    0
1354 0000 JMPAD,    0
1355 7742 NLIST,    -36
1356 2540 LOCSGL,   SIGTBL-1
1357 0077 MS0077,   77
1360 7521 M257,     -257
1361 0177 MK177,    177
1362 7560 TRANSP,   TRANS
1363 0200 PP200,    200
          TSGNL=    10
1364 4431 INSERT,   JMS  I    LFIXIN
1365 7773           -5
1366 0113           PFSTD
1367 5434           JMP  I    LOMGA
1370 4400 PBIANG,   BIANG
          POBSCR=   LOBSCU
1371 0004 P4,       4
1372 0135 PHA,      HH
1373 0276 LZERGO,   ZERGO

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     6
          *         1400
          /OUTPUT ROUTINES
1400 0000 INIT,     0
1401 1166           TAD       LNO
1402 3241           DCA       SAVE
1403 4211           JMS       INITA
1404 2241           ISZ       SAVE
1405 1352           TAD       M2C
1406 1352           TAD       M2C
1407 4211           JMS       INITA
1410 5600           JMP  I    INIT
1411 0000 INITA,    0
1412 1227           TAD       BUFFST
1413 3230           DCA       SAVA
1414 3641           DCA  I    SAVE      /NO=0
1415 2241           ISZ       SAVE
1416 1230           TAD       SAVA
1417 3641           DCA  I    SAVE      /BUFF=BUFFER START
1420 1641           TAD  I    SAVE
1421 2241           ISZ       SAVE
1422 3641           DCA  I    SAVE      /BUF=BUFFST
1423 2241           ISZ       SAVE
1424 2641           ISZ  I    SAVE      /RDY=1
1425 2241           ISZ       SAVE
1426 5611           JMP  I    INITA
1427 1640 BUFFST,   BUFF0
1430 0000 OPCHH,    0
1431 3211           DCA       CHARA
1432 1132           TAD       PFLAG
1433 7650           SNA  CLA
1434 5237           JMP       NOPUN
1435 1353           TAD       PL5
1436 4241           JMS       OPCHR
1437 4241 NOPUN,    JMS       OPCHR
1440 5630           JMP  I    OPCHH
1441 0000 OPCHR,    0
1442 1166           TAD       LNO
```

```
1443 3354            DCA       OPCH
1444 1754            TAD  I    OPCH        /NO
1445 1352            TAD       M20
1446 7650            SNA  CLA
1447 5244            JMP       .-3         /WAIT FOR BUFFER NOT FULL
1450 6002            IOF
1451 2754            ISZ  I    OPCH        /NO
1452 2354            ISZ       OPCH
1453 1754            TAD  I    OPCH        /POSITION IN BUFFER
1454 3200            DCA       INIT
1455 1211            TAD       CHARA
1456 3600            DCA  I    INIT
1457 1754            TAD  I    OPCH        /INCREMENT BUFF (MODULO 16)
1460 7001            IAC
1461 0346            AND       MK7757
1462 3754            DCA  I    OPCH
1463 2354            ISZ       OPCH
1464 2354            ISZ       OPCH
1465 1754            TAD  I    OPCH        /RDY
1466 7650            SNA  CLA
1467 5277            JMP       .+10
1470 7146            CLL  CMA  RTL
1471 1354            TAD       OPCH
1472 3302            DCA       TELE
1473 3701            DCA  I    LSAC
1474 1241            TAD       OPCHR
1475 3000            DCA       0
1476 5316            JMP       P
1477 6001            ION
1500 5641            JMP  I    OPCHR
1501 2474 LSAC,      SVAC
1502 0000 TELE,      0
1503 1702            TAD  I    TELE
1504 3302            DCA       TELE
1505 1702            TAD  I    TELE        /NO
1506 7640            SZA  CLA
1507 5316            JMP       P
1510 2302            ISZ       TELE
1511 2302            ISZ       TELE
1512 2302            ISZ       TELE
1513 7001            IAC
1514 3702            DCA  I    TELE
1515 5403            JMP  I    DISMIS      /NO=0
1516 7240 P,         CLA  CMA
1517 1702            TAD  I    TELE
1520 3702            DCA  I    TELE        /NO=NO-1
1521 2302            ISZ       TELE
1522 2302            ISZ       TELE
1523 1702            TAD  I    TELE        /BUF
1524 3342            DCA       P1
1525 1702            TAD  I    TELE
1526 7001            IAC
1527 0346            AND       MK7757
1530 3702            DCA  I    TELE        /BUF=BUF+1 (MODULO 16)
1531 2302            ISZ       TELE
1532 3702            DCA  I    TELE        /RDY=0
1533 2302            ISZ       TELE
1534 1702            TAD  I    TELE
1535 3337            DCA       TELE5
1536 1742            TAD  I    P1          /CHARACTER
```

```
1537 6046 TELE5,    TLS
1540 7300           CLA  CLL
1541 5403           JMP  I    DISMIS
1542 0000 P1,       0
1543 6042 TELSER,   TCF
1544 4302           JMS       TELE
1545 1620           NO
1546 7757 MK7757,   7757
1547 6022 PUNSER,   PCF
1550 4302           JMS       TELE
1551 1625           PNO
1552 7760 M20,      -20
          SAVA=     OPCHH
          SAVE=     OPCHR
1553 0005 PL5,      5
1554 0000 OPCH,     0
          CHARA=    INITA

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     7
          *         1600
          BUFF1=    .
          *         .+20
1620 0000 NO,       0
1621 0000 BUFF,     0
1622 0000 BUF,      0
1623 0000 RDY,      0
1624 6046           TLS
1625 0000 PNO,      0
1626 0000           0
1627 0000           0
1630 0000 PRDY,     0
1631 6026           PLS
          /CARRIAGE-RETURN LINEFEED SUBROUTINE
1632 0000 CRLF,     0                    /ENTRY
1633 1320           TAD       CR
1634 4506           JMS  I    LOPCH
1635 1321           TAD       LF
1636 4506           JMS  I    LOPCH
1637 5632           JMP  I    CRLF
          *         1640
          BUFF0=    .
          *         .+20
          /TYPE ONE CHARACTER FROM AC BITS 6-11
1660 0000 TY1,      0
1661 0273           AND       MK0077
1662 1270           TAD       M40
1663 7510           SPA
1664 1271           TAD       P100
1665 1272           TAD       P240
1666 4506           JMS  I    LOPCH
1667 5660           JMP  I    TY1
1670 7740 M40,      7740
1671 0100 P100,     100
1672 0240 P240,     240
1673 0077 MK0077,   77
          /TYPE TWO CHARACTERS FROM AC BITS 0-5 AND 6-11 RESPECTIVELY
1674 0000 TY2,      0
1675 3322           DCA       TEMP11
1676 1322           TAD       TEMP11
```

```
1677 7012            RTR
1700 7012            RTR
1701 7012            RTR
1702 4260            JMS       TY1
1703 1322            TAD       TEMP11
1704 4260            JMS       TY1
1705 5674            JMP   I   TY2
          /ROUTINE STRING TYPES A TABLE IN TRIMMED CODE
          /TABLE MUST FINISH WITH 0000
          /CALLING SEQUENCE
          /          JMS       STRING
          /          ADDRESS OF START OF TABLE
          /          RETURN
1706 0000 STRING,    0                       /ENTRY
1707 1706            TAD   I   STRING        /GET ADDRESS OF TABLE
1710 2306            ISZ       STRING        /SET RETURN
1711 3232            DCA       PRINT         /USED AS TEMPORARY
1712 1632            TAD   I   PRINT         /GET NEXT TWO CHARACTERS
1713 7450            SNA                     /TEST FOR ZERO
1714 5706            JMP   I   STRING        /RETURN
1715 4274            JMS       TY2           /TYPE TWO CHARACTERS
1716 2232            ISZ       PRINT         /NEXT TWO CHARACTERS
1717 5312            JMP       .-5
1720 0215 CR,        215
1721 0212 LF,        212
1722 0000 TEMP11,    0
1723 1006 PLEAD,     TAD       M310
1724 3322            DCA       TEMP11
1725 2132            ISZ       PFLAG
1726 1333            TAD       P200
1727 4506            JMS   I   LOPCH
1730 2322            ISZ       TEMP11
1731 5326            JMP       .-3
1732 5434            JMP   I   LOMGA
1733 0200 P200,      200
          PRINT=     CRLF

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      8
          *          2000
          /INITIALISE AND WAIT FOR KEYBOARD SIGNAL
2000 6002 START,     IOF
          INWT=      START
2001 4612            JMS   I   LINIM
2002 4611            JMS   I   LINIT       /INITIALISE TELEPRINTER
2003 6302            ENABLE
2004 6001            ION
2005 4435 OMGA,      JMS   I   LCRLF
2006 6032            KCC                   /GET PAPER TAPE READER CHAR
2007 4432            JMS   I   LSUPER
2010 5206            JMP       OMGA+1
2011 1400 LINIT,     INIT
2012 0410 LINIM,     ININ
          UB=        .
          *          .+33
          ACCNT=     .
          COUNTO=    ACCNT+3
          BGC=       COUNTO+3
          FNS=       BGC+3
          FG=        BGC+6
```

```
          HMH=      FG+3
          CNEW=     HMH+3
          HMIN1=    CNEW+3
          WVL=      HMIN1+10

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     9
          *         2200
          /CLOCK INTERRUPT ROUTINE
2200 6304 CLOCK,    CCF                   /CLEAR CLOCK FLAG
2201 2363           ISZ       CLOK        /TEST COUNTER
2202 5403           JMP  I    DISMIS
2203 1073           TAD       M4          /MOTORS STEP AT 250C/S
2204 3363           DCA       CLOK
2205 1761           TAD  I    LMI         /TEST MOTOR FLAG
2206 7650           SNA  CLA
2207 5762           JMP  I    LSGTM
2210 3761           DCA  I    LMI         /CLEAR MOTOR FLAG IF SET
          /MOTOR DRIVE INTERRUPT
2211 4220           JMS       MDRCI
2212 0714           MIT
2213 0731           MIO
2214 0746           MIC
2215 0763           MIP
2216 3761           DCA  I    LMI
2217 5762           JMP  I    LSGTM
          /CLOCK INTERRUPT MOTOR DRIVE ROUTINE
          /CALLING SEQUENCE
          /         JMS       MDRCI
          /         MOTOR FLAG TTHETA
          /         MOTOR FLAG OMEGA
          /         MOTOR FLAG CHI
          /         MOTOR FLAG PHI
          /         C(AC) RETURNS WITH SETTING FOR MOTOR INDICATOR
2220 0000 MDRCI,    0                     /ENTRY
2221 1073           TAD       M4
2222 3360           DCA       TEM1
2223 3344           DCA       TEM2
2224 1620 RELOOP,   TAD  I    MDRCI
2225 3247           DCA       MDRA
2226 2220           ISZ       MDRCI
2227 1647           TAD  I    MDRA        /TEST MOTOR FLAG
2230 7650           SNA  CLA
2231 5340           JMP       MDRCX       /FLAG NOT SET
2232 2344           ISZ       TEM2
2233 1247           TAD       MDRA
2234 1357           TAD       M6
2235 3247           DCA       MDRA
2236 2647           ISZ  I    MDRA        /TEST SLOW START
2237 5340           JMP       MDRCX
2240 7144           CLL  CMA  RAL
2241 1247           TAD       MDRA
2242 3247           DCA       MDRA
2243 1247           TAD       MDRA
2244 3337           DCA       MDRB
2245 4756           JMS  I    LTRANS      /COPY ARGUMENTS
2246 7767           -11                   /FROM
2247 0000 MDRA,     0                     /VARIABLE
2250 2344           OVR-1                 /TO OVR
2251 2347           ISZ       K2
```

```
2252 5256            JMP       MDR1
2253 1360            TAD       TEM1
2254 7130            CLL  CML  RAR        /K2 = -2 FOR 2-THETA,OMEGA
2255 3347            DCA       K2         /K2 = -1 FOR CHI,PHI
2256 1347 MDR1,      TAD       K2         /K1 = K2
2257 3346            DCA       K1
2260 1350            TAD       D          /COMPARE CURRENT,DESIRED POSN
2261 1352            TAD       DS
2262 7440            SZA
2263 5304            JMP       MDR3
2264 1351            TAD       H
2265 1353            TAD       HS
2266 7440            SZA
2267 5304            JMP       MDR3
2270 2345            ISZ       OVR
2271 5302            JMP       MDR2
2272 7240            CLA  CMA             /OVERSHOOT,RESET DESIRED POSN
2273 1352            TAD       DS
2274 3352            DCA       DS
2275 1006            TAD       SSK1       /SLOW START CONST 1
2276 3346            DCA       K1
2277 1075            TAD       SSK2       /SLOW START CONST 2
2300 3347            DCA       K2
2301 5334            JMP       MDRCE
2302 3354 MDR2,      DCA       MIA
2303 5334            JMP       MDRCE
2304 7700 MDR3,      SMA  CLA
2305 5322            JMP       MDR4
2306 1355            TAD       PMOT
2307 6341            PMF                  /PULSE MOTOR FORWARD
2310 7200            CLA
2311 2351            ISZ       H          /INCREMENT CURRENT 200THS
2312 1351            TAD       H          /IS IT LESS THAN 200
2313 1006            TAD       M310       /-200
2314 7710            SPA  CLA
2315 5334            JMP       MDRCE
2316 2350            ISZ       D          /NO  INCREMENT CURRENT DEGREE
2317 7000            NOP
2320 3351            DCA       H          /RESET 200THS POSITION
2321 5334            JMP       MDRCE
2322 1355 MDR4,      TAD       PMOT
2323 6321            PMR                  /PULSE MOTOR REVERSE
2324 7240            CLA  CMA             /DECREMENT CURRENT 200THS
2325 1351            TAD       H
2326 7500            SMA                  /LESS THAN 0 ?
2327 5333            JMP       MDR5
2330 1350            TAD       D          /YES  DECREMENT CURRENT DEG
2331 3350            DCA       D
2332 1366            TAD       P307       /SET CURRENT 200THS TO 199
2333 3351 MDR5,      DCA       H
2334 4756 MDRCE,     JMS  I    LTRANS     /COPY ARGUMENTS
2335 7767            -11                  /FROM
2336 2344            OVR-1                /OVR
2337 0000 MDRB,      0                    /TO MOTOR
2340 2360 MDRCX,     ISZ       TEM1
2341 5224            JMP       RELOOP
2342 1344            TAD       TEM2
2343 5620            JMP  I    MDRCI      /RETURN
2344 0000 LA,        0                    /STORAGE FOR DATA
2345 0000 CVR,       C                    /FROM PARTICULAR
```

```
2346 0000 K1,       0                       /MOTOR BLOCK
2347 0000 K2,       0
2350 0000 D,        0
2351 0000 H,        0
2352 0000 DS,       0
2353 0000 HS,       0
2354 0000 MIA,      0
2355 0000 PMOT,     0
2356 7560 LTRANS,   TRANS
          TEM2=     LA
2357 7772 M6,       -6
2360 0000 TEM1,     0
2361 0521 LMI,      MI
2362 1053 LSGTM,    SGTM
2363 0000 CLOK,     0
2364 0000 MI1,      0
2365 0000 MI2,      0
2366 0307 P307,     307

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     10
          *         2400
2400 3274 INTER,    DCA         SVAC        /SAVE C(AC)
2401 7204           CLA  RAL
2402 3275           DCA         SVLK        /SAVE C(L)
2403 6301           SNC                     /SKIP IF NOT CLOCK
2404 5630           JMP  I      LCLK        /CLOCK
2405 6311           SNS                     /SKIP IF NOT STATUS
2406 5231           JMP         STATUS
2407 6041           TSF                     /SKIP IF TELEPRINTER
2410 7410           SKP
2411 5627           JMP  I      LTEL        /TELSER
2412 6031           KSF                     /SKIP IF KEYBOARD
2413 7410           SKP
2414 5625           JMP  I      LKEY        /KEYSER
2415 6021           PSF                     /SKIP IF PUNCH
2416 7410           SKP
2417 5626           JMP  I      LPUN
2420 6011           RSF                     /SKIP IF READER
2421 7402           HLT                     /EXTRANEOUS INTERRUPT
2422 6012           RRB                     /IGNORE READER
2423 7200           CLA
2424 5403           JMP  I      DISMIS
2425 1000 LKEY,     KEYSER                  /INDIRECT ADDRESSES
2426 1547 LPUN,     PUNSER
2427 1543 LTEL,     TELSER
2430 2200 LCLK,     CLOCK
          /ROUTINE TESTS STATUS IN FOR INTERRUPTS
2431 1272 STATUS,   TAD         MIN14       /SET COUNTER
2432 3273           DCA         STAK
2433 1255           TAD         JMPC        /INITIALISE JUMP COMMAND
2434 3254           DCA         JMPE
2435 6312           RSI                     /READ STATUS REGISTER
2436 7004 TESTST,   RAL                     /TEST STATUS
2437 7430           SZL
2440 5245           JMP         CLST        /ONE FOUND
2441 2254           ISZ         JMPE
2442 2273           ISZ         STAK
2443 5236           JMP         TESTST
2444 7402 SWT3,     HLT                     /INCORRECT STATUS INTERRUPT
```

```
          /ROUTINE CLEARS ONE STATUS BIT
2445 7200 CLST,     CLA
2446 7004           RAL
2447 2273           ISZ       STAK
2450 5246           JMP       .-2
2451 7040           CMA
2452 6314           CSI                     /CLEAR STATUS-IN
2453 7200           CLA
2454 0000 JMPE,     0                       /VARIABLE JUMP COMMAND
2455 5656 JMPC,     JMP  I    STAB
2456 4654 STAB,     MONDET                  /MONITOR-DETECTOR PRESET
2457 3001           MOTTZ                   /2THETA MOTOR ZERO
2460 3006           MOTOZ                   /OMEGA
2461 3013           MOTCZ                   /CHI
2462 3020           MOTPZ                   /PHI
2463 3000           MOVETZ                  /2THETA TABLE ZERO
2464 3005           MOVEOZ                  /OMEGA
2465 3012           MOVECZ                  /CHI
2466 3017           MOVEPZ                  /PHI
2467 2444           SWT1
2470 2444           SWT2
2471 2444           SWT3
2472 7764 MIN14,    -14
          SWT2=     SWT3
          SWT1=     SWT3
2473 0000 STAK,     0                       /STATUS COUNTER
2474 0000 SVAC,     0                       /C(AC)
2475 0000 SVLK,     0                       /C(L)
          /SUBROUTINE TO TERMINATE INTERRUPTS
2476 1275 TERINT,   TAD       SVLK          /RESTORE C(L) & C(AC)
2477 7110           CLL  RAR
2500 1274           TAD       SVAC
2501 6001           ION                     /INTERRUPT ON
2502 5400           JMP  I    0
          /COPY FROM ONE AREA OF MEMORY TO ANOTHER
          /CALLING SEQUENCE
          /         JMS       COPY
          /         NO OF ARGUMENTS
          /FROM     ADDRESS OF TABLE
          /TO       ADDRESS OF TABLE
2503 0000 COPY,     0
2504 4422           JMS  I    LADR
2505 7775 NEG3,     -3
2506 0000 COPK,     0
2507 0000 COPA,     0
2510 0000 COPB,     0
2511 1707 COPS,     TAD  I    COPA
2512 3710           DCA  I    COPB
2513 2307           ISZ       COPA          /ADVANCE ADDRESSES
2514 2310           ISZ       COPB
2515 2306           ISZ       COPK          /TEST COUNTER
2516 5311           JMP       COPS
2517 5703           JMP  I    COPY          /RETURN
          /SET ARGUMENTS FOR A SUBROUTINE
2520 0000 ADR,      0
2521 1720           TAD  I    ADR           /SET COUNTER
2522 3016           DCA       ADRK
2523 7144           CLL  CMA  RAL
2524 1320           TAD       ADR
2525 3105           DCA       ADRA          /LOCATION OF SUBROUTINE CALL
```

```
2526 7040              CMA
2527 1505              TAD  I    ADRA
2530 3015              DCA       ADRB        /LOCATION OF ARGUMENTS
2531 2505 ADRS,        ISZ  I    ADRA        /SET EXIT FOR CALLING SUB
2532 2320              ISZ       ADR         /SET ARGUMENT STORE LOCATION
2533 1415              TAD  I    ADRB        /GET ARGUMENTS
2534 3720              DCA  I    ADR         /STORE ARGUMENT
2535 2016              ISZ       ADRK
2536 5331              JMP       ADRS
2537 2320              ISZ       ADR         /SET EXIT
2540 5720              JMP  I    ADR         /RETURN TO CALLING SUBROUTINE
           ADRA=       TEMP                  / LOCATION OF SUBROUTINE
           ADRB=       AUT5                  / LOCATION OF ARGUMENT -1
           ADRK=       AUT6
           /TWO-CHARACTER KEYBOARD SIGNAL TABLE
2541 7777 SIGTBL,      7777                  /?? REINITIALISE,KILL CURRENT
2542 2000              INWT
2543 2416              2416                  /TN TYPEWRITER ON
2544 1064              TYON
2545 2406              2406                  /TF TYPWRITER OFF
2546 1065              TYOF
2547 1706              1706                  /OF SET PROGRAM END FLAG
2550 1067              OFFPRG
2551 0320              0320                  /CP GO TO WAITING LOOP
2552 2005              OMGA
2553 2001              2001                  /PA PRINT ALL CURRENT ANGLES
2554 6173              ANGPRT
2555 1115              1115                  /IM INCREMENT MOTORS
2556 3162              TINC
2557 0415              0415                  /DM DRIVE MOTORS

           /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
           /PAGE       11
2560 0601              SETTTH
2561 2010              2010                  /PH PUNCH HEADING
2562 1060              TYPE
2563 2014              2014                  /PL PUNCH LEADER-TRAILER
2564 1723              PLEAD
2565 0322              0322                  /CR CENTRE REFLECTION
2566 3400              CENTER
2567 3201              3201                  /ZA READ INITIAL ANGLES
2570 0254              TINIT
2571 0401              0401                  /DA DRIVE TO CALCULATED ANGLES
2572 6175              DRIVE
2573 2403              2403                  /TC TAKE COUNT AND PRINT
2574 3367              TACNT
2575 2315              2315                  /SM SET MONITOR STATUS
2576 4732              SETTIM
2577 2323              2323                  /SS  START SCAN
2600 0347              RN
2601 2227              2227                  /RW READ WAVELENGTH
2602 6573              WAVEL
2603 1623              1623                  /NS READ NO OF STEPS FOR SCAN
2604 7172              STEPNO
2605 2502              2502                  /UB READ UB(ROW-WISE)
2606 4673              UBR
2607 1123              1123                  /IS INSERT STANDARDS
2610 1364              INSERT
2611 1120              1120                  / IP INDIVIDUAL PARALLEL
2612 4600              IPP
```

```
2613 0123            0123                    /AS AZIMUTH SCAN
2614 4277            SSR
2615 0304            0304                    /CD COLLECT DATA
2616 5000            CDR
2617 1122            1122                    /IR INDIVIDUAL REFLECTIONS
2620 4551            IRR
2621 0301            0301                    /CA CALCULATE ANGLES
2622 4561            CALCUL
2623 1522            1522                    /MR MEASURE REFLECTION
2624 4677            DCCOLL
2625 0000            0
2626 2005            OMGA
2627 0000            0
2630 2005            OMGA
2631 1316 NOHOP,     TAD         CH1
2632 3105            DCA         TEMP
2633 1073            TAD         M4
2634 3432            DCA   I     LSUPER
2635 3040            DCA         ERRCNT
2636 2105 LOOP1,     ISZ         TEMP
2637 2105            ISZ         TEMP
2640 1505            TAD   I     TEMP        / DEGREE ADDRESS
2641 1311            TAD         P10
2642 3267            DCA         A1
2643 1667            TAD   I     A1
2644 3267            DCA         A1          / NAME ADDRESS
2645 2105            ISZ         TEMP
2646 1321            TAD         M24
2647 1505            TAD   I     TEMP        / ERRORS
2650 7710            SPA   CLA
2651 5255            JMP         .+4
2652 3505            DCA   I     TEMP        / RESET FOR NO ERRORS,NO CHECK
2653 2105            ISZ         TEMP
2654 5264            JMP         FIN
2655 2105            ISZ         TEMP
2656 1505            TAD   I     TEMP        / SERIOUS ERROR
2657 7750            SPA   SNA   CLA         / AC>0?
2660 5272            JMP         EXIT5
2661 3505            DCA   I     TEMP        / RESET FOR NO ERRORS,NO CHECK.
2662 4467            JMS   I     LSTRNG
2663 4701            SERIUS
2664 4467 FIN,       JMS   I     LSTRNG
2665 3374            ERROR2
2666 4467            JMS   I     LSTRNG
2667 0000 A1,        0
2670 4435            JMS   I     LCRLF
2671 2040            ISZ         ERRCNT
2672 2105 EXIT5,     ISZ         TEMP
2673 2432            ISZ   I     LSUPER
2674 5236            JMP         LOOP1
2675 1040            TAD         ERRCNT
2676 7650            SNA   CLA
2677 5560            JMP   I     SUPPT       /NO ERRORS
2700 1163            TAD         ZERGO1
2701 7640            SZA   CLA
2702 5720            JMP   I     LZER        /ZEROING IN PROGRESS PRESERVE RETURN
2703 1161            TAD         MOTST1      / SAVE MOTST RETURN
2704 3162            DCA         MOTST2
2705 1160            TAD         SUPPT
2706 3165            DCA         SUPPTT
```

```
2707 4717            JMS  I    PZERGO
2710 4427            JMS  I    LMOTST
2711 0010 P10,       10
2712 3163            DCA       ZERGO1
2713 1162            TAD       MOTST2
2714 3161            DCA       MOTST1     /RESTORE RETURN
2715 5565            JMP  I    SUPPTT
2716 3000 CH1,       MOVETZ
2717 0276 PZERGO,    ZERGO
2720 0311 LZER,      LANGLE+2
2721 7754 M24,       -24
          ERRCNT=    40

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      12
          *          3000
3000 4624 MOVETZ,    JMS  I    LMOVE
3001 4625 MOTTZ,     JMS  I    LMARK
3002 0710            DT
3003 0000            0
3004 0000            0
3005 4624 MOVEOZ,    JMS  I    LMOVE
3006 4625 MOTOZ,     JMS  I    LMARK
3007 0725            DO
3010 0000            0
3011 0000            0
3012 4624 MOVECZ,    JMS  I    LMOVE
3013 4625 MOTCZ,     JMS  I    LMARK
3014 0742            DC
3015 0000            0
3016 0000            0
3017 4624 MOVEPZ,    JMS  I    LMOVE
3020 4625 MOTPZ,     JMS  I    LMARK
3021 0757            DP
3022 0000            0
3023 0000            0
3024 3026 LMOVE,     MOVCHK
3025 3050 LMARK,     MARK
3026 0000 MOVCHK,    0
3027 2226            ISZ       MOVCHK
3030 1626            TAD  I    MOVCHK
3031 3247            DCA       TEM5
3032 2226            ISZ       MOVCHK
3033 2226            ISZ       MOVCHK
3034 1626            TAD  I    MOVCHK
3035 7550            SPA  SNA
3036 7240            CLA  CMA
3037 3626            DCA  I    MOVCHK     / -1 FOR CHECK
3040 1647            TAD  I    TEM5
3041 7650            SNA  CLA
3042 5403            JMP  I    DISMIS
3043 7001            IAC
3044 3626            DCA  I    MOVCHK     / +1 FOR ERROR
3045 3647            DCA  I    TEM5
3046 5403            JMP  I    DISMIS
3047 0000 TEM5,      0
3050 0000 MARK,      0
3051 1650            TAD  I    MARK
3052 1073            TAD       M4
3053 3260            DCA       TEM3
```

```
3054 1260           TAD       TEM3
3055 3347           DCA       TEM4
3056 4751           JMS   I   PTRANS
3057 7773           -5
3060 0000 TEM3,     0
3061 3151           OVER-1
3062 1353           TAD       K11
3063 1104           TAD       THR
3064 7710           SPA   CLA
3065 5403           JMP   I   DISMIS     /SLOW START PULSE TOO CLOSE
3066 2250           ISZ       MARK
3067 1650           TAD   I   MARK
3070 7450           SNA
3071 7240           CLA   CMA
3072 3650           DCA   I   MARK
3073 1356           TAD       HTT
3074 3260           DCA       TEM3
3075 1352           TAD       OVER
3076 7700           SMA   CLA
3077 5304           JMP       FORW      /FORWARD,HTT=HTT
3100 1260           TAD       TEM3      /REVERSE,HTT=199-HTT
3101 7041           CMA   IAC
3102 1357           TAD       PL307
3103 3260           DCA       TEM3
3104 7040 FORW,     CMA
3105 1260           TAD       TEM3
3106 7750           SPA   SNA CLA
3107 5403           JMP   I   DISMIS     /OK
3110 2650           ISZ   I   MARK
3111 7410           SKP
3112 2650           ISZ   I   MARK
3113 2250           ISZ       MARK
3114 1260           TAD       TEM3
3115 1361           TAD       M144      /-100
3116 7700           SMA   CLA
3117 5325           JMP       MARK2
3120 1260           TAD       TEM3
3121 1075           TAD       M12
3122 7750           SPA   SNA CLA
3123 5340           JMP       MARK4
3124 5336           JMP       MARK3     /SERIOUS ERROR
3125 1352 MARK2,    TAD       OVER
3126 7500           SMA                 /DTT=DTT-1 REVERSE
3127 7201           CLA   IAC
3130 1355           TAD       DTT
3131 3355           DCA       DTT       /DTT=DTT+1 FORWARD
3132 1260           TAD       TEM3
3133 1360           TAD       M276
3134 7740           SMA   SZA CLA
3135 5340           JMP       MARK4
3136 7001 MARK3,    IAC
3137 3650           DCA   I   MARK
3140 1352 MARK4,    TAD       OVER
3141 7710           SPA   CLA
3142 1357           TAD       PL307     /+199
3143 3356           DCA       HTT
3144 4751           JMS   I   PTRANS
3145 7773           -5
3146 3151           OVER-1
3147 0000 TEM4,     0
```

```
3150 5403            JMP  I    DISMIS
3151 7560 PTRANS,    TRANS
3152 0000 OVER,      0
3153 0000 K11,       0
3154 0000 K22,       0
3155 0000 CTT,       0
3156 0000 HTT,       0
3157 0307 PL307,     307                    /+199
3160 7502 M276,      -276                   /-190
3161 7634 M144,      -144                   /-100
3162 4370 TINC,      JMS       ARG
3163 3142            DCA       NANG
3164 4430            JMS  I    LINTIN
3165 3143            DCA       XANG
3166 4465            JMS  I    LSTEP
3167 5434            JMP  I    LOMGA
3170 0000 ARG,       0
3171 4430            JMS  I    LINTIN
3172 0104            AND       THR
3173 7450            SNA
3174 1376            TAD       PLUS4
3175 5770            JMP  I    ARG
3176 0004 PLUS4,     4

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      13
          *          3200
          /SUBROUTINE FOR RECEIVING COUNT
3200 0000 TKCNT,     0
3201 1200            TAD       TKCNT
3202 3156            DCA       STEPPT
3203 4510 C01,       JMS  I    LSTCT
3204 4407            JMS  I    7
3205 6527            FPUT I    PCNEW
3206 0000            FEXT
3207 7001            IAC
3210 3154            DCA       NSUB
3211 4510 C02,       JMS  I    LSTCT
3212 4407            JMS  I    7
3213 1101            FADD      TWO
3214 0002            FSQRT
3215 3351            FMPY      FOUR
3216 6052            FPUT      DANG        /TEMP USE
3217 0006            FNEG
3220 1525            FADD I    COUNT       /LOWER LIMIT
3221 2527            FSUB I    PCNEW
3222 0000            FEXT
3223 1045            TAD       HORD
3224 7700            SMA  CLA
3225 5203            JMP       C01         /COUNT TOO HIGH, START AGAIN
3226 4407            JMS  I    7
3227 5525            FGET I    COUNT
3230 1052            FADD      DANG        /UPPER LIMIT
3231 2527            FSUB I    PCNEW
3232 0000            FEXT
3233 1045            TAD       HORD
3234 7710            SPA  CLA
3235 5203            JMP       C01         /PREVIOUS COUNTS TOO HIGH
3236 2154            ISZ       NSUB
3237 1154            TAD       NSUB
```

```
3240 4433            JMS  I    LFLT
3241 4407            JMS  I    7
3242 6052            FPUT      DANG        /TEMP USE
3243 5525            FGET I    COUNT       /CALCULATE NEW AVERAGE
3244 2527            FSUB I    PCNEW
3245 4052            FDIV      DANG
3246 1527            FADD I    PCNEW
3247 6527            FPUT I    PCNEW
3250 0000            FEXT
3251 1154            TAD       NSUB
3252 1073            TAD       M4
3253 7710            SPA  CLA
3254 5211            JMP       CO2
3255 4407            JMS  I    7
3256 3351            FMPY      FOUR
3257 6525            FPUT I    COUNT
3260 0000            FEXT
3261 5556            JMP  I    STEPPT
3262 0000 STEP,      0
3263 1262            TAD       STEP
3264 3156            DCA       STEPPT
3265 1143 CO3,       TAD       XANG        /NO OF TWO-HUNDREDTHS PER STEP
3266 4433            JMS  I    LFLT
3267 4407            JMS  I    7
3270 4716            FDIV I    LF200
3271 6052            FPUT      DANG
3272 0000            FEXT
3273 4317            JMS       ADNANG      /GET POINTER TO ANGLE
3274 3262            DCA       ANGADR      /POINTER TO ANGLE
3275 4407            JMS  I    7
3276 5662            FGET I    ANGADR      /INCREMENT ANGLE
3277 1052            FADD      DANG
3300 6662            FPUT I    ANGADR
3301 0000            FEXT
3302 1142            TAD       NANG
3303 7640            SZA  CLA
3304 5313            JMP       STP6
3305 4407            JMS  I    7           /NANG=0,STEP 2-THETA TWICE
3306 5052            FGET      DANG
3307 1052            FADD      DANG
3310 1514            FADD I    PTTH
3311 6514            FPUT I    PTTH
3312 0000            FEXT
3313 4427 STP6,      JMS  I    LMOTST      /MOVE TO NEXT POSITION
3314 7000            NOP
3315 5556            JMP  I    STEPPT
3316 0657 LF200,     F200
3317 0000 ADNANG,    0
3320 1142            TAD       NANG
3321 7450            SNA
3322 1101            TAD       TWO         /OMEGA = 0 OR 2
3323 7041            CMA  IAC
3324 3262            DCA       ST1
3325 7146            CLL  CMA  RTL         /-3
3326 1104            TAD       THR
3327 2262            ISZ       ST1
3330 5326            JMP       .-2
3331 1114            TAD       PTTH
3332 5717            JMP  I    ADNANG
          /SUBROUTINE FOR TYPING LINES
```

```
3333 0000 TYPOPT,   0
3334 1152           TAD       TYP
3335 7650           SNA  CLA
3336 5733           JMP  I    TYPOPT     /RETURN IF FLAG=0
3337 2144           ISZ       WDK
3340 5344           JMP       .+4
3341 1074           TAD       M10        /-8
3342 3144           DCA       WDK
3343 4435           JMS  I    LCRLF
3344 4407           JMS  I    7
3345 5525           FGET I    COUNT
3346 0015           FDPOUT
3347 0000           FEXT
3350 5733           JMP  I    TYPOPT
3351 0003 FOUR,     0003
3352 2000           2000
3353 0000           0000
3354 6224 TTHETA,   6224
3355 1005           1005
3356 2401           2401
3357 0000           0
3360 1715 OMEGA,    1715
3361 0507           0507
3362 0140           0140
3363 0000           0
3364 2010 PHIN,     2010
3365 1140           1140
3366 0000           0
          ANGADR=   STEP
          ST1=      ANGADR
          DANG=     FPAC1
3367 4464 TACNT,    JMS  I    LTKCNT
3370 4407           JMS  I    7
3371 0015           FDPOUT
3372 0000           FEXT
3373 5434           JMP  I    LOMGA
3374 0522 ERROR2,   0522
3375 2217           2217
3376 2240           2240
3377 0000           0

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     14
          *         3400
          /PROGRAM BLOCK FOR CENTERING A REFLECTION
3400 4764 CENTER,   JMS  I    LARG       / 3 FOR CHI, 4 FOR PHI
3401 3147           DCA       NCP
          /ROUTINE TO PERFORM COARSE OMEGA SCAN TO LOCATE PEAK
          /STEPSIZE FOR COARSE SCAN = 0.1 DEG
3402 1101           TAD       TWO
3403 3142           DCA       NANG
3404 1363           TAD       M100
3405 3143           DCA       XANG
3406 4465           JMS  I    LSTEP
3407 1360           TAD       P24        /20
3410 3143           DCA       XANG
3411 4407           JMS  I    7
3412 5470           FGET I    LZERO
3413 6520           FPUT I    LIMIT1
3414 0000           FEXT
```

```
3415 7040            CMA
3416 3144            DCA       WDK
3417 1075            TAD       M12         /-10
3420 3140            DCA       SCV1
3421 4464 OSC1,      JMS   I   LTKCNT      /STEP
3422 4343            JMS       COMPAR      /TYPOPT, COUNT = LIMIT ?
3423 5232            JMP       OSC2        /COUNT < LIMIT
3424 4407            JMS   I   7           /COUNT > LIMIT
3425 5515            FGET  I   POMG
3426 6523            FPUT  I   ANGMAX
3427 5525            FGET  I   COUNT
3430 6520            FPUT  I   LIMIT1
3431 0000            FEXT
3432 4465 OSC2,      JMS   I   LSTEP
3433 2140            ISZ       SCV1
3434 5221            JMP       OSC1
3435 4407            JMS   I   7
3436 5520            FGET  I   LIMIT1
3437 3355            FMPY      HALF
3440 6520            FPUT  I   LIMIT1
3441 5523            FGET  I   ANGMAX
3442 6515            FPUT  I   POMG
3443 0000            FEXT
3444 4427            JMS   I   LMOTST      /MOVE TO MAX
3445 7000            NOP
3446 1147            TAD       NCP
3447 4257            JMS       CONV        /TRIM CHI OR PHI
3450 1101            TAD       TWO
3451 4257            JMS       CONV        /TRIM OMEGA USING CONVERGE
3452 7001            IAC
3453 4257            JMS       CONV        /TRIM 2 THETA USING CONVERGE
3454 4471            JMS   I   LANGPT      /PRINT ANGLES
3455 5434            JMP   I   LOMGA
          NCP=       NCT2
          SCV1=      GSTP
          LIMIT1=    LACCNT
          ANGMAX=    LFG
3456 3317 LANGAD,    ADNANG
          /CONV TRIMS ANGLE TO BISECTION OF HALF-MAXIMUM HEIGHTS
          /CALLING SEQUENCE
          /          JMS       CONV
          /          VALUE FOR NANG IN C(AC)
          /          VALUE FOR HALF-MAXIMUM IN LIMIT
3457 0000 CONV,      0
3460 3142            DCA       NANG
3461 1257            TAD       CONV
3462 3157            DCA       CONVPT
3463 1362            TAD       MIN40       /-32
3464 3143            DCA       XANG
3465 7040            CMA
3466 3140            DCA       SCV1
3467 4656 OTHER,     JMS   I   LANGAD
3470 3257            DCA       ANGAD1
3471 7040            CMA
3472 3144            DCA       WDK
3473 4465 REPEAT,    JMS   I   LSTEP
3474 4464            JMS   I   LTKCNT      /RESTORE POINTER TO ANGLE
3475 4656            JMS   I   LANGAD
3476 3257            DCA       ANGAD1
3477 4343            JMS       COMPAR      /TYPOPT, COUNT = LIMIT ?
```

```
3500 7410            SKP                        /COUNT LESS THAN LIMIT
3501 5273            JMP        REPEAT          /COUNT GREATER THAN LIMIT
3502 1143            TAD        XANG
3503 7041            CMA  IAC
3504 3143            DCA        XANG
3505 4465            JMS  I     LSTEP
3506 1143            TAD        XANG            /XANG=-XANG/2
3507 7141            CLL  CMA   IAC
3510 7510            SPA
3511 7020            CML
3512 7010            RAR
3513 3143            DCA        XANG
3514 1143            TAD        XANG
3515 0104            AND        THR
3516 7650            SNA  CLA
3517 5273            JMP        REPEAT
3520 2140            ISZ        SCV1
3521 5331            JMP        B1              /2ND
3522 4407            JMS  I     7               /1ST,FIND 2ND
3523 5657            FGET I     ANGAD1
3524 6523            FPUT I     ANGMAX          /SAVE IST HALF-HEIGHT ANGLE
3525 0000            FEXT
3526 1361            TAD        P4C             /+32
3527 3143            DCA        XANG
3530 5267            JMP        OTHER
3531 4407 B1,        JMS  I     7               /PEAK = (ANGLE + ANGMAX)/2
3532 5657            FGET I     ANGAD1
3533 1523            FADD I     ANGMAX
3534 3355            FMPY       HALF
3535 6657            FPUT I     ANGAD1
3536 0000            FEXT
3537 4435            JMS  I     LCRLF
3540 4427            JMS  I     LMOTST          /MOVE TO PEAK
3541 5434            JMP  I     LOMGA
3542 5557            JMP  I     CONVPT
          CONVPT=    LOBSPT
          ANGAD1=    CONV
          /ROUTINE TO TYPE OUT AND COMPARE COUNT AND LIMIT
          /CALLING SEQUENCE
          /          JMS COMPAR
3543 0000 COMPAR,    0
3544 4466            JMS  I     LTYPOT
3545 4407            JMS  I     7
3546 5525            FGET I     COUNT
3547 2520            FSUB I     LIMIT1
3550 0000            FEXT
3551 1045            TAD        45              /HORD
3552 7700            SMA  CLA
3553 2343            ISZ        COMPAR
3554 5743            JMP  I     COMPAR
3555 0000 HALF,      0
3556 2000            2000
3557 0000            0
3560 0024 P24,       24
3561 0040 P40,       40
3562 7740 MIN40,     -40
3563 7634 M100,      -144
3564 3170 LARG,      ARG
```

```
          /PAGE     15
          *         3600
3600 0000 EX,       0
3601 7510           SPA
3602 7041           CMA  IAC
3603 4242           JMS       CHEKK
3604 0115           115                    /+77
3605 0147           147                    /+103
3606 5211           JMP       OBERR        /OMEGA BETWEEN 103 AND 77
3607 5600           JMP  I    EX
3610 4722 PHKLOT,   HKLOUT
3611 4435 OBERR,    JMS  I    LCRLF
3612 4467           JMS  I    LSTRNG
3613 5372           OBCURE
3614 4610           JMS  I    PHKLOT
3615 4471           JMS  I    LANGPT
3616 5557 RETN1,    JMP  I    LOBSPT
3617 1105 VARY,     TAD       NARR
3620 7640           SZA  CLA
3621 5211           JMP       OBERR
3622 2105           ISZ       NARR
3623 4407           JMS  I    7            /TRY OMG,-CHI,PHI+PI
3624 5516           FGET I    PCHI
3625 0006           FNEG
3626 6516           FPUT I    PCHI
3627 5235           FGET      F180
3630 1517           FADD I    PPHI
3631 6517           FPUT I    PPHI
3632 0000           FEXT
3633 5274           JMP       RECHEK
3634 0051 XA,       XTTH
3635 0010 F180,     0010
3636 2640           2640
3637 0000           0000
3640 0550 P360,     550                    /+360
3641 7514 M180,     -264                   /-180
          /ROUTINE TO CHECK RANGE OF ANGLE
          /CALLING SEQUENCE
          /         JMS       CHEKK
          /         ANGLE IN C(AC)
          /         MINIMUM VALUE FOR THE ANGLE
          /         MAXIMUM VALUE FOR THE ANGLE
          /         RETURN WITHIN RANGE
          /         RETURN OUT OF RANGE
3642 0000 CHEKK,    0
3643 7041           CMA  IAC
3644 3260           DCA       TEMPY
3645 1642           TAD  I    CHEKK
3646 2242           ISZ       CHEKK
3647 1260           TAD       TEMPY
3650 7700           SMA  CLA
3651 5255           JMP       .+4
3652 1642           TAD  I    CHEKK
3653 1260           TAD       TEMPY
3654 7710           SPA  CLA
3655 2242           ISZ       CHEKK
3656 2242           ISZ       CHEKK
3657 5642           JMP  I    CHEKK
3660 0000 TEMPY,    0
3661 4023 SINLIM,   4023
```

```
3662 1116              1116
3663 0540              540
3664 1411 LIMITS,      1411
3665 1511              1511
3666 2440              2440
3667 0000              0
          /ROUTINE TO CHECK WHETHER CALCULATED ANGLES ARE OBSCURE
          /IF ANGLES OK THEN MEASURE REFLECTION
3670 0000 OBSCUR,      0
3671 1270              TAD       OBSCUR
3672 3157              DCA       LOBSPT
3673 3105              DCA       NARR
3674 1073 RECHEK,      TAD       M4
3675 3050              DCA       XAF
3676 1234              TAD       XA
3677 3270              DCA       SOB2
3700 1114              TAD       PTTH        /ADDRESS OF CALC ANGLES
3701 3200              DCA       SOB3        /FIX CALC ANGLES
3702 4407 OB1,         JMS   I   7
3703 560C              FGET  I   SOB3
3704 7200              FINC      SOB3
3705 0000              FEXT
3706 4426              JMS   I   LFIX
3707 3670              DCA   I   SOB2
3710 2270              ISZ       SOB2
3711 2050              ISZ       XAF
3712 5302              JMP       OB1
3713 1051              TAD       XTTH        /CHECK 2-THETA
3714 4242              JMS       CHEKK
3715 7654              -124
3716 0206              206
3717 7410              SKP
3720 5211              JMP       OBERR       /2-THETA 0>TTH>140
3721 1051              TAD       XTTH        /EVALUATE ALFA=OM-2TH AND
3722 7041 OB2,         CMA   IAC               /BRING INTO RANGE +180 TO -180
3723 1052              TAD       XOM
3724 1241              TAD       M180
3725 7500              SMA
3726 5332              JMP       .+4
3727 1240              TAD       P360
3730 7510              SPA
3731 1240              TAD       P360        /LESS THAN 180,ADD 360
3732 1241              TAD       M180
3733 3050              DCA       XAF
3734 1050              TAD       XAF
3735 4200              JMS       EX
3736 1052              TAD       XOM
3737 4200              JMS       EX
3740 1053              TAD       XCI
3741 4242              JMS       CHEKK
3742 7761              -17                   /-15
3743 0017              17                    /+15
3744 5372              JMP       DCOLL       /-15<CHI<15
3745 1053              TAD       XCI
3746 7710              SPA   CLA
3747 5354              JMP       OB5         /CHI <-15
3750 1052              TAD       XOM         /CHI > 15
3751 3200              DCA       SOB3
3752 1050              TAD       XAF
3753 5357              JMP       OB6
```

```
3754 1050 CB5,    TAD       XAF
3755 3200         DCA       SOB3
3756 1052         TAD       XOM
3757 3270 CB6,    DCA       SOB2
3760 1200         TAD       SOB3
3761 4242         JMS       CHEKK
3762 0040         40                  /+32
3763 0224         224                 /+148
3764 5217         JMP       VARY      /TRY ALTERNATIVE ARRANGEMENT
3765 1270         TAD       SOB2
3766 4242         JMS       CHEKK
3767 7554         -224                /-148
3770 7740         -40                 /-32
3771 5217         JMP       VARY
          SOB2=   OBSCUR
          SOB3=   EX
          NARR=   TEMP
          XAF=    50
          XTTH=   51
          XOM=    XTTH+1
          XCI=    XOM+1
          XFI=    XCI+1

          /ND6E2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE   16
          /ROUTINE DCOLL SCANS A PEAK
3772 4427 DCOLL,  JMS  I    LMOTST
3773 7000         NOP
3774 1134         TAD       CT
3775 3131         DCA       LOT
3776 3150         DCA       NH1
3777 3142         DCA       NANG
4000 2132         ISZ       PFLAG
4001 1363         TAD       SOM
4002 4506         JMS  I    LOPCH
4003 4435         JMS  I    LCRLF
4004 4762         JMS  I    LHKLOT
4005 4471         JMS  I    LANGPT
4006 3132         DCA       PFLAG
4007 1141         TAD       NSTP
4010 4433         JMS  I    LFLT
4011 4407         JMS  I    7
4012 6522         FPUT I    LFNS      /FIXED NSTEP
4013 5470         FGET I    LZERO
4014 6520         FPUT I    LACCNT    /CLEAR ACCNT
4015 6521         FPUT I    LBGC
4016 0000         FEXT
4017 3145         DCA       G
          /FIND NO (G) OF COUNTS ON PEAK REQUIRED TO REACH 200
4020 2145 PPK,    ISZ       G
4021 1150         TAD       NH1
4022 1134         TAD       CT
4023 3150         DCA       NH1
4024 4464         JMS  I    LTKCNT
4025 4407         JMS  I    7
4026 1520         FADD I    LACCNT
4027 6520         FPUT I    LACCNT
4030 2747         FSUB I    F200PT
4031 0000         FEXT
```

```
4032 1145            TAD       G
4033 0304            AND       LCMA
4034 7440            SZA
4035 5242            JMP       DCOLLA
4036 1045            TAD       45
4037 7710            SPA  CLA
4040 5220            JMP       PPK
4041 1145            TAD       G
4042 4433 DCOLLA,    JMS  I    LFLT
4043 1150            TAD       NH1
4044 3134            DCA       CT          /SET MONITOR SCALE-FACTOR
        /CALCULATE NO OF GROUPS OF G IN NSTP AND FIX. RESULT IN GSTP
4045 4407            JMS  I    7
4046 6523            FPUT I    LFG
4047 5522            FGET I    LFNS
4050 4523            FDIV I    LFG
4051 0000            FEXT
4052 4426            JMS  I    LFIX
4053 3140            DCA       GSTP
4054 1350            TAD       ASHIFT
4055 3143            DCA       XANG
4056 4465            JMS  I    LSTEP       /SHIFT TO BACKGROUND
4057 4464            JMS  I    LTKCNT
4060 1140            TAD       GSTP
4061 4433            JMS  I    LFLT
4062 4407            JMS  I    7
4063 6522            FPUT I    LFNS        /GSTP NOW FLOATED
4064 5520            FGET I    LACCNT
4065 1525            FADD I    COUNT
4066 4101            FDIV      TWO
4067 6524            FPUT I    PHMH        /HALF MAXIMUM HEIGHT
4070 5525            FGET I    COUNT       /CONTAINING G BG COUNTS
4071 1101            FADD      TWO
4072 4520            FDIV I    LACCNT      /GROUP COUNT ON PEAK
4073 0002            FSQRT                 /(BG/PEAK)**1/2
4074 3522            FMPY I    LFNS
4075 4101            FDIV      TWO
4076 0000            FEXT
4077 4426            JMS  I    LFIX        /ESTIMATE OF 1/2 BKGND GROUPS
4100 3155            DCA       GBG
4101 1155            TAD       GBG
4102 7041            CMA  IAC
4103 3147            DCA       NCT2
4104 7040 LCMA,      CMA
4105 3144            DCA       WDK
4106 4464 RG10,      JMS  I    LTKCNT
4107 4352            JMS       BACKGD
4110 2147            ISZ       NCT2
4111 5306            JMP       RG10
4112 1351            TAD       NSHIFT
4113 3143            DCA       XANG
4114 4465            JMS  I    LSTEP       /SHIFT TO OTHER SIDE
4115 1155            TAD       GBG
4116 7041            CMA  IAC
4117 3147            DCA       NCT2
4120 4464 RGI,       JMS  I    LTKCNT
4121 4352            JMS       BACKGD
4122 2147            ISZ       NCT2
4123 5320            JMP       RGI
4124 1155            TAD       GBG
```

```
4125 4433            JMS  I    LFLT
4126 4407            JMS  I    7
4127 3101            FMPY      TWO
4130 6525            FPUT I    COUNT      /TOTAL BACKGROUND GROUPS
4131 5521            FGET I    LBGC
4132 3522            FMPY I    LFNS
4133 4525            FDIV I    COUNT
4134 6521            FPUT I    LBGC       /ADJUSTED BACKGROUD COUNT
4135 5470            FGET I    LZERO
4136 6520            FPUT I    LACCNT
4137 0000            FEXT
4140 1141            TAD       NSTP
4141 7110            CLL  RAR
4142 7041            CMA  IAC
4143 1350            TAD       ASHIFT
4144 3143            DCA       XANG
4145 4465            JMS  I    LSTEP
4146 5373            JMP       DATA
4147 0657 F200PT,    F200
4150 1130 ASHIFT,    1130                 /600 STEPS I.E. 3 DEGREES
4151 5520 NSHIFT,    5520                 /-1200 I.E. -6 DEGREES
4152 0000 BACKGD,    0
4153 4466            JMS  I    LTYPOT
4154 4407            JMS  I    7
4155 5525            FGET I    COUNT
4156 1521            FADD I    LBGC
4157 6521            FPUT I    LBGC
4160 0000            FEXT
4161 5752            JMP  I    BACKGD
4162 4722 LHKLOT,    HKLOUT
4163 0201 SOM,       201
4164 4366 HKLINP,    JMS       HKL
4165 5434            JMP  I    LOMGA
4166 0000 HKL,       0                    /READ H,K,L
4167 4431            JMS  I    LFIXIN
4170 7775            -3
4171 5153            PH
4172 5766            JMP  I    HKL        /RETURN

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      17
4173 3150 DATA,      DCA       NH1
4174 3147            DCA       NCT2
4175 1145            TAD       G
4176 3143            DCA       XANG
4177 7040            CMA
4200 3144            DCA       WDK
4201 4464 ADVAN,     JMS  I    LTKCNT
4202 4465            JMS  I    LSTEP
4203 4466            JMS  I    LTYPOT
4204 2147            ISZ       NCT2
4205 1150            TAD       NH1
4206 7640            SZA  CLA               /FOUND 1ST HALF HEIGHT
4207 5222            JMP       DT2          /YES,LOOK FOR SECOND
4210 4407            JMS  I    7
4211 5525            FGET I    COUNT
4212 2524            FSUB I    PHMH
4213 0000            FEXT
4214 1045            TAD       45           /HORD
4215 7710            SPA  CLA               /IS THIS FIRST HALF HEIGHT?
```

```
4216 5235            JMP       DT3        /NO
4217 1147            TAD       NCT2
4220 3150            DCA       NH1
4221 5235            JMP       DT3
4222 4407 DT2,       JMS   I   7
4223 5524            FGET  I   PHMH
4224 2525            FSUB  I   COUNT
4225 0000            FEXT
4226 1045            TAD       45         /HORD
4227 7710            SPA   CLA              /IS THIS SECOND HALF HEIGHT?
4230 5234            JMP       DT4        /NO
4231 1151            TAD       NH2
4232 7450            SNA
4233 1147            TAD       NCT2
4234 3151 DT4,       DCA       NH2
4235 4407 DT3,       JMS   I   7
4236 5520            FGET  I   LACCNT
4237 1525            FADD  I   COUNT
4240 6520            FPUT  I   LACCNT
4241 0000            FEXT
4242 1140            TAD       GSTP
4243 7041            CMA   IAC
4244 1147            TAD       NCT2
4245 7710            SPA   CLA
4246 5201            JMP       ADVAN      /NO
4247 2132            ISZ       PFLAG
4250 1145            TAD       G
4251 4462            JMS   I   LIOUT
4252 1140            TAD       GSTP
4253 4462            JMS   I   LIOUT
4254 1150            TAD       NH1
4255 4462            JMS   I   LIOUT
4256 1151            TAD       NH2
4257 4462            JMS   I   LIOUT
4260 4435            JMS   I   LCRLF
4261 4407            JMS   I   7
4262 5520            FGET  I   LACCNT
4263 0015            FDPOUT
4264 5521            FGET  I   LBGC
4265 0015            FDPOUT
4266 0000            FEXT
4267 4435            JMS   I   LCRLF
4270 1276            TAD       EOM
4271 4506            JMS   I   LOPCH
4272 3132            DCA       PFLAG
4273 1131            TAD       LOT
4274 3134            DCA       CT
4275 5557            JMP   I   LOBSPT
4276 0203 EOM,       203
          /AZIMUTH SCAN ROUTINE CALLED BY /AS
          /NO OF TENTH-DEGREE STEPS REQUIRED
4277 2146 SSR,       ISZ       AUTO
4300 4430            JMS   I   LINTIN
4301 3150            DCA       NH1
4302 7040            CMA
4303 3144            DCA       WDK
4304 4427 SSR1,      JMS   I   LMOTST
4305 5434            JMP   I   LOMGA
4306 4464            JMS   I   LTKCNT
4307 4466            JMS   I   LTYPOT
```

```
4310 1150            TAD        NH1
4311 7710            SPA  CLA
4312 7144            CLL  CMA   RAL
4313 7001            IAC
4314 4433            JMS  I     LFLT
4315 4407            JMS  I     7
4316 4775            FDIV I     LLLTEN
4317 6524            FPUT I     LDPSI
4320 0000            FEXT
4321 1150            TAD        NH1
4322 7500            SMA
4323 7041            CMA  IAC
4324 3151            DCA        NH2
4325 4407 SSR2,      JMS  I     7
4326 5514            FGET I     PTTH
4327 3776            FMPY I     PHALF
4330 6521            FPUT I     LBGC
4331 0006            FNEG
4332 1515            FADD I     POMG
4333 6515            FPUT I     POMG
4334 5516            FGET I     PCHI
4335 0003            FSIN
4336 6525            FPUT I     COUNT
4337 5515            FGET I     POMG
4340 0004            FCOS
4341 6520            FPUT I     LACCNT
4342 4525            FDIV I     COUNT
4343 3524            FMPY I     LDPSI
4344 1517            FADD I     PPHI
4345 6517            FPUT I     PPHI       /PHI=PHI+(COS(W)*DPSI/SIN(C))
4346 5516            FGET I     PCHI
4347 0004            FCOS
4350 4525            FDIV I     COUNT      /SIN CHI
4351 3520            FMPY I     LACCNT     /COS OMEGA
4352 3524            FMPY I     LDPSI
4353 6525            FPUT I     COUNT      /-DELTA OMG
4354 1515            FADD I     POMG
4355 6515            FPUT I     POMG       /OMEGA=OMEGA+DELTA OMEGA
4356 5525            FGET I     COUNT      /DELTA OMEGA
4357 3776            FMPY I     PHALF
4360 1515            FADD I     POMG
4361 0003            FSIN
4362 3524            FMPY I     LDPSI
4363 1516            FADD I     PCHI
4364 6516            FPUT I     PCHI       /CHI=CHI+DELTA CHI
4365 5515            FGET I     POMG
4366 1521            FADD I     LBGC
4367 6515            FPUT I     POMG
4370 0000            FEXT
4371 2151            ISZ        NH2
4372 5325            JMP        SSR2
4373 4463            JMS  I     LGO
4374 5304            JMP        SSR1
4375 6774 LLLTEN,    TEN
4376 3555 PHALF,     HALF

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      18
          *          4400
          /ROUTINE TO CALCULATE ANGLES FOR BISECTING POSITION
```

```
          /VALUES FOR H,K,L AND UB MUST BE SET PRIOR TO CALL
          /CALLING SEQUENCE
          /         JMS       BIANG
          /         ERROR RETURN   I.E. SIN(THETA)>1
          /         NORMAL RETURN
4400 0000 BIANG,    0
          /ROUTINE TO FLOAT H,K,L;CALCULATE HF=UB*H AND STORE HF
4401 7146           CLL  CMA  RTL
4402 3151           DCA       HT2
4403 1125           TAD       LHV
4404 3140           DCA       AX
4405 1140           TAD       AX
4406 3145           DCA       HT3
4407 1374           TAD       PHB
4410 3150           DCA       HT1
4411 1550 F1,       TAD  I    HT1        /FLOAT AND STORE H,K,L
4412 4433           JMS  I    LFLT
4413 4407           JMS  I    7
4414 6545           FPUT I    HT3
4415 7145           FINC      HT3
4416 0000           FEXT
4417 2150           ISZ       HT1
4420 2151           ISZ       HT2
4421 5211           JMP       F1
4422 1123           TAD       LHF
4423 3150           DCA       HT1
4424 1111           TAD       LUB
4425 3151           DCA       HT2
4426 7146           CLL  CMA  RTL        /-3
4427 3145           DCA       HT3
4430 4407           JMS  I    7
4431 5470           FGET I    LZERO
4432 6520           FPUT I    UNB
4433 0000           FEXT
4434 4407 F2,       JMS  I    7
4435 5470           FGET I    LZERO
4436 6550           FPUT I    HT1
4437 0000           FEXT
4440 4336           JMS       ROWCOL
4441 4336           JMS       ROWCOL
4442 4336           JMS       ROWCOL
4443 4407           JMS  I    7
4444 0001           FSQR
4445 1520           FADD I    UNB
4446 6520           FPUT I    UNB
4447 7150           FINC      HT1
4450 0000           FEXT
4451 1350           TAD       M11        /-9
4452 1140           TAD       AX
4453 3140           DCA       AX
4454 2145           ISZ       HT3
4455 5234           JMP       F2
4456 4407           JMS  I    7
4457 0002           FSQRT                /LENGTH OF VECTOR
4460 3526           FMPY I    LWVL       /WAVELENGTH
4461 4101           FDIV      TWO
4462 6515           FPUT I    POMG       /SIN(THETA),OMG USED AS TEMP
4463 0001           FSQR
4464 0006           FNEG
4465 1076           FADD      ONE
```

```
4466 0000            FEXT
4467 1045            TAD        45
4470 7700            SMA  CLA
4471 5275            JMP        .+4
4472 4467            JMS  I     LSTRNG
4473 3661            SINLIM
4474 7410            SKP
4475 2200            ISZ        BIANG
4476 4407            JMS  I     7
4477 0002            FSQRT
4500 6735            FPUT I     LBP         /POINTER TO COS (SEE ATAN)
4501 5515            FGET I     POMG
4502 0005            FATN
4503 6515            FPUT I     POMG
4504 3101            FMPY       TWO
4505 6514            FPUT I     PTTH
4506 0000            FEXT
4507 1123            TAD        LHF
4510 1104            TAD        THR
4511 3124            DCA        LHF2
4512 4407            JMS  I     7
4513 5523            FGET I     LHF1        /POINTER TO HF1
4514 6735            FPUT I     LBP         /POINTER TO B
4515 5524            FGET I     LHF2        /POINTER TO HF2
4516 0005            FATN
4517 6517            FPUT I     PPHI        /PHI=ATAN(HF2,HF1)
4520 5523            FGET I     LHF1        /POINTER TO HF1
4521 0001            FSQR
4522 6516            FPUT I     PCHI
4523 5524            FGET I     LHF2        /POINTER TO HF2
4524 0001            FSQR
4525 1516            FADD I     PCHI
4526 0002            FSQRT
4527 6735            FPUT I     LBP         /POINTER TO B
4530 5527            FGET I     LHF3        /POINTER TO HF3
4531 0005            FATN
4532 6516            FPUT I     PCHI
          /CHI=ATAN(HF3,(HF1*HF1+HF2*HF2)**-.5)
4533 0000            FEXT
4534 5600            JMP  I     BIANG       /RETURN
4535 5511 LBP,       B
4536 0000 ROWCOL,    0
4537 4407            JMS  I     7
4540 5551            FGET I     HT2
4541 3540            FMPY I     AX
4542 1550            FADD I     HT1
4543 6550            FPUT I     HT1
4544 7151            FINC       HT2
4545 7140            FINC       AX
4546 0000            FEXT
4547 5736            JMP  I     ROWCOL
4550 7767 M11,       -11                    /-9
          /COLLECT DATA FOR INDIVIDUAL REFLECTIONS
          /CALLED BY /IR
4551 2146 IRR,       ISZ        AUTO
4552 4472            JMS  I     LSTD
4553 4773            JMS  I     LHKL
4554 4200            JMS        BIANG
4555 7410            SKP
4556 4436            JMS  I     POBSCR
```

```
4557 4463            JMS  I    LGO
4560 5351            JMP       IRR
4561 4773 CALCUL,    JMS  I    LHKL
4562 4200            JMS       BIANG
4563 5434            JMP  I    LOMGA
4564 4425            JMS  I    LFOP
4565 7774            -4
4566 0114            PTTH
4567 4772            JMS  I    HKLOPT
4570 4435            JMS  I    LCRLF
4571 5434            JMP  I    LOMGA
4572 4722 HKLOPT,    HKLOUT
4573 4166 LHKL,      HKL
4574 0135 PHB,       HH
          HT1=       NH1
          HT2=       NH2
          HT3=       G
          AX=        GSTP
          LHF=       LFG
          LHF1=      LFG
          LHF2=      PHMH
          LHF3=      PCNEW
          LHV=       COUNT
          UNB=       LACCNT

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      19
          *          4600
          /INDIVIDUAL PARALLEL POSITIONS
          /CALLED BY /IP
4600 2146 IPP,       ISZ       AUTO
4601 4472            JMS  I    LSTD
4602 4624            JMS  I    AHKL
4603 4625            JMS  I    ABIANG
4604 5221            JMP       IPP1
4605 4407            JMS  I    7
4606 5517            FGET I    PPHI
4607 1623            FADD I    LFP90
4610 6517            FPUT I    PPHI
4611 5623            FGET I    LFP90
4612 2516            FSUB I    PCHI
4613 1515            FADD I    PDMG
4614 6515            FPUT I    PDMG
4615 5623            FGET I    LFP90
4616 6516            FPUT I    PCHI
4617 0000            FEXT
4620 4436            JMS  I    LOBSCU
4621 4463 IPP1,      JMS  I    LGO
4622 5201            JMP       IPP+1
4623 5517 LFP90,     K90
4624 4166 AHKL,      HKL
4625 4400 ABIANG,    BIANG
          /ROUTINE TO START DATA COUNT
          /CALLING   SEQUENCE
          /          JMS       STCT
4626 0000 STCT,      0
4627 1134            TAD       CT          /GET NO. OF MONITOR
4630 3270            DCA       CTC         /TRANSFER TO COUNTER
4631 1133            TAD       TIME
4632 6346            CSOSSO                /START DATA COUNT
```

```
4633 3271            DCA       CFLAG
          /WAIT FOR DATA COUNT
4634 1271            TAD       CFLAG
4635 7650            SNA  CLA
4636 5241            JMP       .+3
4637 4432            JMS  I    LSUPER
4640 5234            JMP       .-4
4641 6331            RDH                    /GET DETECTOR HIGH
4642 3045            DCA       45           /HIGH ORDER C(FAC)
4643 6336            RDLCDS                 /READ DETECTOR LOW,AND RESET
4644 3046            DCA       46           /LOW ORDER C(FAC)
4645 1272            TAD       KEXP         /27
4646 3044            DCA       44           /EXPONENT C(FAC)
4647 4407            JMS  I    7
4650 0010            FNOR
4651 6525            FPUT I    COUNT
4652 0000            FEXT
4653 5626            JMP  I    STCT         /RETURN
          /MONITOR OVERFLOW ROUTINE
4654 6302 MONDET,    CMS                    /CLEAR MONITOR SCALER
4655 3271            DCA       CFLAG        /CLEAR COUNT FLAG
4656 2270            ISZ       CTC
4657 1270            TAD       CTC          /TEST LOOP COUNTER
4660 7710            SPA  CLA
4661 5264            JMP       .+3
4662 3270            DCA       CTC
4663 5403            JMP  I    DISMIS
4664 1133            TAD       TIME
4665 6346            CSOSSO
4666 3271            DCA       CFLAG
4667 5403            JMP  I    DISMIS
4670 0000 CTC,       0
4671 0000 CFLAG,     0
4672 0027 KEXP,      27
4673 4424 UBR,       JMS  I    LFNP
4674 7767            -11
4675 0111            LUB
4676 5434            JMP  I    LOMGA
4677 4436 DCCOLL,    JMS  I    LOBSCU
4700 5434            JMP  I    LOMGA
4701 2305 SERIUS,    2305
4702 2211            2211
4703 1725            1725
4704 2340            2340
4705 0000            0
4706 0000 SAVFAC,    0
4707 0000            0
4710 0000            0
4711 0000 TV1,       0
4712 0000            0
4713 0000            0
4714 0000 TV2,       0
4715 0000            0
4716 0000            0
4717 0000 TV3,       0
4720 0000            0
4721 0000            0
4722 0000 HKLOUT,    0
4723 1135            TAD       HH
4724 4462            JMS  I    LIOUT
```

```
4725 1136            TAD       K
4726 4462            JMS   I   LIOUT
4727 1137            TAD       L
4730 4462            JMS   I   LIOUT
4731 5722            JMP   I   HKLOUT
4732 4430 SETTIM,    JMS   I   LINTIN
4733 0076            AND       ONE
4734 1101            TAD       TWO
4735 3133            DCA       TIME
4736 4430            JMS   I   LINTIN
4737 7041            CMA   IAC
4740 3134            DCA       CT
4741 5434            JMP   I   LOMGA

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      20
          *          5000
          /ROUTINE TO GENERATE H,K,L'S CALLED BY /CD
5000 4431 CDR,       JMS   I   LFIXIN
5001 7770            -10                   /-8
5002 0112            LHMIN
5003 4354            JMS       GETARG
5004 2146            ISZ       AUTO
5005 3153 SEL1,      DCA       NEX
5006 7040            CMA
5007 1044            TAD       LMIN
5010 3137            DCA       L
5011 2137 SEL2,      ISZ       L
5012 1137            TAD       L
5013 7041            CMA   IAC
5014 1045            TAD       LMAX        /L > LMAX?
5015 7710            SPA   CLA
5016 5274            JMP       SEL15       /YES
5017 7040            CMA
5020 1042            TAD       KMIN        /NO
5021 3136            DCA       K
5022 2136 SEL3,      ISZ       K
5023 1136            TAD       K
5024 7041            CMA   IAC
5025 1043            TAD       KMAX        /K > KMAX?
5026 7710            SPA   CLA
5027 5211            JMP       SEL2        /YES
5030 7040            CMA
5031 1040            TAD       HMIN        /NO
5032 3135            DCA       HH          /H=HMIN
5033 2135 SEL4,      ISZ       HH
5034 1135            TAD       HH
5035 7041            CMA   IAC
5036 1041            TAD       HMAX        /H > HMAX?
5037 7710            SPA   CLA
5040 5222            JMP       SEL3        /YES
5041 7146 SL1,       CLL   CMA RTL         /-3
5042 3354            DCA       CS1
5043 3305            DCA       SUM
5044 1353            TAD       PH
5045 3050            DCA       PHT
5046 1450 SL2,       TAD   I   PHT
5047 7500            SMA
5050 7041            CMA   IAC
5051 3051            DCA       ABSOL       /-(H)
```

```
5052 1051           TAD       ABSOL
5053 3361           DCA       CS2          /-(H)
5054 1305 SL3,      TAD       SUM
5055 1051           TAD       ABSOL
5056 3305           DCA       SUM
5057 2361           ISZ       CS2
5060 5254           JMP       SL3
5061 2050           ISZ       PHT
5062 2354           ISZ       CS1
5063 5246           JMP       SL2
5064 1305           TAD       SUM
5065 1047 SEL6,     TAD       LIMIT
5066 7450           SNA
5067 5301           JMP       SEL10
5070 7710           SPA  CLA
5071 2153 SEL9,     ISZ       NEX          /SIGMA H**2 >= LIMIT
5072 4463           JMS  I    LGO
5073 5233           JMP       SEL4
5074 1153 SEL15,    TAD       NEX
5075 7650           SNA  CLA
5076 5434           JMP  I    LOMGA        /NOTHING IN RANGES > LIMIT
5077 2047 SEL7,     ISZ       LIMIT
5100 5205           JMP       SEL1
5101 1046 SEL10,    TAD       SYMM         /NOW PERFORM SYMMETRY CHECKS
5102 0307           AND       MASK7
5103 1306           TAD       JUMP1
5104 3305           DCA       .+1
5105 5307 SUM,      JMP       MASK7
5106 5307 JUMP1,    JMP       MASK7
5107 0007 MASK7,    7
5110 5340           JMP       MEASUR       /1,PRIMITIVE
5111 5333           JMP       KPL          /2,K+L EVEN
5112 5321           JMP       HPL          /3,H+L EVEN
5113 5317           JMP       HPK          /4,H+K EVEN
5114 5326           JMP       FCC          /5,F.C.C. ALL EVEN OR ALL ODD
5115 5324           JMP       BCC          /6,B.C.C. H+K+L EVEN
5116 5334           JMP       LONLY        /7,HEX. L EVEN
5117 1136 HPK,      TAD       K
5120 7410           SKP
5121 1137 HPL,      TAD       L
5122 1135           TAD       HH
5123 5335           JMP       SEL13
5124 1135 BCC,      TAD       HH
5125 5333           JMP       KPL
5126 1135 FCC,      TAD       HH
5127 1136           TAD       K
5130 7110           CLL  RAR
5131 7630           SZL  CLA
5132 5271           JMP       SEL9
5133 1136 KPL,      TAD       K
5134 1137 LONLY,    TAD       L
5135 7110 SEL13,    CLL  RAR
5136 7630           SZL  CLA
5137 5271           JMP       SEL9
5140 1112 MEASUR,   TAD       LHMIN
5141 1307           TAD       MASK7
5142 3305           DCA       SUM
5143 1047           TAD       LIMIT
5144 3705           DCA  I    SUM
5145 4472           JMS  I    LSTD
```

```
5146 4764              JMS  I    LBIANG
5147 5434              JMP  I    LOMGA
5150 4436              JMS  I    LOBSCU
5151 4354              JMS       GETARG
5152 5271              JMP       SEL9        /RETURN FROM MEASUREMENT
5153 0135 PH,          HH
5154 0000 GETARG,      0
5155 1112              TAD       LHMIN
5156 3361              DCA       .+3
5157 4423              JMS  I    LCOP
5160 7770              -10
5161 0000 CS2,         0
5162 0040              HMIN
5163 5754              JMP  I    GETARG
          CS1=         GETARG
          HMIN=        40
          HMAX=        41
          KMIN=        42
          KMAX=        43
          LMIN=        44
          LMAX=        45
          SYMM=        46
          LIMIT=       47
          PHT=         50
          ABSOL=       51
          POBSCU=      LOBSCU
5164 4400 LBIANG,      BIANG


          /ND6C2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE        21
          *            5200
          /SUBROUTINE SINE RETURNS SIN(ANGLE IN DEGREES)
5200 0000 SINE,        0                        /ENTRY
5201 7200              CLA
5202 3317              DCA          PNTR
5203 1045              TAD          45          /IS X NEGATIVE
5204 7740              SMA  SZA     CLA
5205 5213              JMP          MOD
5206 1045              TAD          45
5207 7700              SMA  CLA
5210 5600              JMP  I       SINE        /RETURN X=0
5211 4303              JMS          NEG
5212 2317              ISZ          PNTR
5213 4407 MOD,         JMS  I       7
5214 3366              FMPY         RAD
5215 4326              FDIV         TWOPI
5216 6323              FPUT         XSQR
5217 0000              FEXT
5220 4426              JMS  I       LFIX
5221 4354              JMS          FLOAT
5222 4407              JMS  I       7
5223 6320              FPUT         X
5224 5323              FGET         XSQR
5225 2320              FSUB         X
5226 3326              FMPY         TWOPI
5227 6320              FPUT         X
5230 2331              FSUB         PI
5231 0000              FEXT
5232 1045              TAD          45          /IS X LESS THAN PI
```

```
5233 7710            SPA  CLA
5234 5241            JMP       PICH       /YES
5235 4407            JMS  I    7          /NO
5236 6320            FPUT      X          /X=X-PI
5237 0000            FEXT
5240 2317            ISZ       PNTR
5241 4407 PICH,      JMS  I    7          /IS X LESS THAN PI/2
5242 5320            FGET      X
5243 2334            FSUB      PIOT
5244 0000            FEXT
5245 1045            TAD       45
5246 7710            SPA  CLA
5247 5255            JMP       ALG        /YES
5250 4407            JMS  I    7          /NO
5251 5331            FGET      PI         /X=PI-X
5252 2320            FSUB      X
5253 6320            FPUT      X
5254 0000            FEXT
5255 4407 ALG,       JMS  I    7          /MAIN ALGORITHM FOR SINE
5256 5320            FGET      X
5257 4334            FDIV      PIOT
5260 6320            FPUT      X
5261 3320            FMPY      X
5262 6323            FPUT      XSQR
5263 5337            FGET      C9
5264 3323            FMPY      XSQR
5265 1342            FADD      C7
5266 3323            FMPY      XSQR
5267 1345            FADD      C5
5270 3323            FMPY      XSQR
5271 1350            FADD      C3
5272 3323            FMPY      XSQR
5273 1334            FADD      PIOT
5274 3320            FMPY      X
5275 0000            FEXT
5276 1317            TAD       PNTR
5277 7010            RAR
5300 7630            SZL  CLA
5301 4303            JMS       NEG
5302 5600 END,       JMP  I    SINE
5303 0000 NEG,       0                    /ENTRY
5304 3047            DCA       OVER2
5305 4707            JMS  I    LACMIN
5306 5703            JMP  I    NEG        /RETURN
5307 6000 LACMIN,    ACMINS
          /SUBROUTINE COS ADDS 90 DEGREES AND CALLS SINE
5310 0000 COS,       0                    /ENTRY
5311 4407            JMS  I    7
5312 1753            FADD I    LK90
5313 0000            FEXT
5314 1310            TAD       COS        /SET RETURN
5315 3200            DCA       SINE
5316 5201            JMP       SINE+1     /EXIT TO SINE
5317 0000 PNTR,      0                    /SYMBOLS AND CONSTANTS
5320 0000 X,         0
5321 0000            0
5322 0000            0
5323 0000 XSQR,      0
5324 0000            0
5325 0000            0
```

```
5326 0003 TWOPI,    0003
5327 3110           3110
5330 3755           3755
5331 0002 PI,       0002
5332 3110           3110
5333 3755           3755
5334 0001 PIOT,     0001
5335 3110           3110
5336 3755           3755
5337 7764 C9,       7764
5340 2366           2366
5341 5735           5735
5342 7771 C7,       7771
5343 5466           5466
5344 6317           6317
5345 7775 C5,       7775
5346 2431           2431
5347 5053           5053
5350 0000 C3,       0000
5351 5325           5325
5352 0420           0420
5353 5517 LK90,     K90
5354 0000 FLOAT,    0                       /ENTRY
5355 3364           DCA       FLT+1         /FLOATS CONTENTS
5356 4407           JMS   I   7             /OF ACCUMULATOR
5357 5363           FGET      FLT
5360 0000           FEXT
5361 4771           JMS   I   PNORM
5362 5754           JMP   I   FLOAT         /RETURN
5363 0013 FLT,      0013
5364 0000           0000
5365 0000           0000
5366 7773 RAD,      7773
5367 2167           2167
5370 6433           6433
5371 6600 PNORM,    DNORM
5372 1702 DBCURE,   1702
5373 2303           2303
5374 2522           2522
5375 0540           0540
5376 0000           0

        /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
        /PAGE       22
        *           5400
        /FLOATING ARCTANGENT
        /CALLING SEQUENCE
        /           SINE IN C(FAC)
        /           COS  IN C(B,B+1,B+2)
        /           ATAN
5400 0000 ARTN,     0
5401 3303           DCA       FLAG2         /CLEAR INDICATOR
5402 1312           TAD       B+1           /SEE IF COS=0
5403 7640           SZA CLA
5404 5215           JMP       ART1
5405 4407           JMS   I   7
5406 6306           FPUT      A             /SAVE SINE
5407 5317           FGET      K90           /F.A = 90
5410 0000           FEXT
5411 1307 ART2,     TAD       A+1
```

```
5412 7710           SPA  CLA
5413 4420           JMS  I    LNEG      /IF SINE NEGATIVE,NEGATE C(FAC)
5414 5600           JMP  I    ARTN      /RETURN
5415 4407 ART1,     JMS  I    7
5416 6306           FPUT      A         /SAVE SINE
5417 4311           FDIV      B         /CALCULATE TANGENT
5420 0000           FEXT
5421 4421           JMS  I    LABS      /TAKE ABSOLUTE VALUE OF TANGENT
5422 4407           JMS  I    7
5423 6704           FPUT I    LX
5424 2076           FSUB      ONE
5425 0000           FEXT
5426 1045           TAD       45        /TEST IF TAN GREATER THAN 1
5427 7710           SPA  CLA
5430 5240           JMP       ART3      /NO
5431 4407           JMS  I    7         /CALCULATE RECIPROCAL
5432 5076           FGET      ONE
5433 4704           FDIV I    LX
5434 6704           FPUT I    LX
5435 0000           FEXT
5436 7240           CLA  CMA            /SET INDICATOR =-1
5437 3303           DCA       FLAG2
5440 4407 ART3,     JMS  I    7         /COMPUTE ARCTANGENT
5441 5704           FGET I    LX
5442 3704           FMPY I    LX
5443 6705           FPUT I    LXSQR
5444 3341           FMPY      BET2
5445 1336           FADD      BET1
5446 3705           FMPY I    LXSQR
5447 1333           FADD      BETZ
5450 6314           FPUT      DEN
5451 5330           FGET      ALF2
5452 3705           FMPY I    LXSQR
5453 1325           FADD      ALF1
5454 3705           FMPY I    LXSQR
5455 1322           FADD      ALFZ
5456 3704           FMPY I    LX
5457 4314           FDIV      DEN       /RESULT IN RADIANS
5460 3344           FMPY      DEG       /CONVERT TO DEGREES
5461 6314           FPUT      DEN
5462 0000           FEXT
5463 2303           ISZ       FLAG2     /TEST INDICATOR
5464 5272           JMP       ART4
5465 4407           JMS  I    7         /ADJUST WITHIN QUADRANT
5466 5317           FGET      K90
5467 2314           FSUB      DEN
5470 6314           FPUT      DEN
5471 0000           FEXT
5472 1312 ART4,     TAD       B+1       /ADJUST TO CORRECT QUADRANT
5473 7700           SMA  CLA
5474 5211           JMP       ART2
5475 4407           JMS  I    7
5476 5317           FGET      K90
5477 1317           FADD      K90
5500 2314           FSUB      DEN
5501 0000           FEXT
5502 5211           JMP       ART2
5503 0000 FLAG2,    0                   /VARIABLES
5504 5320 LX,       X
5505 5323 LXSQR,    XSQR
```

```
5506 0000 A,        0
5507 0000           0
5510 0000           0
5511 0000 B,        0
5512 0000           0
5513 0000           0
5514 0000 DEN,      0
5515 0000           0
5516 0000           0
5517 0007 K90,      0007                /CONSTANTS
5520 2640           2640
5521 0000           0000
5522 0000 ALFZ,     0000
5523 2437           2437
5524 1643           1643
5525 7777 ALF1,     7777
5526 3304           3304
5527 4434           4434
5530 7773 ALF2,     7773
5531 3306           3306
5532 5454           5454
5533 0000 BETZ,     0000
5534 2437           2437
5535 1646           1646
5536 0000 BET1,     0000
5537 2427           2427
5540 2323           2323
5541 7775 BET2,     7775
5542 3427           3427
5543 7052           7052
5544 0006 DEG,      0006
5545 3451           3451
5546 3560           3560
          /ROUTINE TO TRUNCATE C(FAC) TO AN INTEGER
5547 0000 FIX,      0
5550 1045           TAD       45
5551 7710           SPA  CLA
5552 7040           CMA
5553 3303           DCA       FIX1      /SET INDICATOR =-1 IF NEGATIVE
5554 4370           JMS       ABS
5555 4407           JMS  I    7
5556 1365           FADD      FIXC      /ADD 2**22
5557 0000           FEXT
5560 1046           TAD       46        /INTEGER BELOW 4096 IN C(AC)
5561 2303           ISZ       FIX1
5562 5747           JMP  I    FIX
5563 7041           CMA  IAC
5564 5747           JMP  I    FIX
          FIX1=     FLAG2
5565 0027 FIXC,     0027
5566 2000           2000
5567 0000           0000
          /ROUTINE ABS CALLS LNEG IF NEGATIVE
5570 0000 ABS,      0
5571 1045           TAD       45
5572 7710           SPA  CLA
5573 4420           JMS  I    LNEG
5574 5770           JMP  I    ABS
5575 0310 CHIN,     0310
5576 1140           1140
```

```
5577 0000                0

         /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
         /PAGE      23
         *          5600
         /FLOATING POINT INTERPRETER
5600 0000 FPNT,     0
5601 7300           CLA  CLL
5602 3043           DCA       OVER1
5603 3047           DCA       OVER2
5604 1600           TAD  I    FPNT       /GET NEXT INSTRUCTION
5605 3253           DCA       JUMP
5606 1253           TAD       JUMP
5607 0263           AND       PAGENO     /GET PAGE BIT
5610 7650           SNA  CLA               /PAGE ZERO?
5611 5214           JMP       .+3        /YES
5612 1261           TAD       MASK5      /NO
5613 0200           AND       FPNT       /C(FPNT)0-4 CONTAINS PAGE BITS
5614 3256           DCA       ADDR
5615 1262           TAD       MK0177     /GET 7 BIT ADDRESS
5616 0253           AND       JUMP
5617 1256           TAD       ADDR
5620 3256           DCA       ADDR
5621 1264           TAD       INDRCT     /INDIRECT BIT=1?
5622 0253           AND       JUMP
5623 7650           SNA  CLA
5624 5227           JMP       LOOPC1     /NO-GO ON
5625 1656           TAD  I    ADDR       /YES DEFER
5626 3256           DCA       ADDR
5627 2200 LOOPC1,   ISZ       FPNT
5630 1656           TAD  I    ADDR
5631 3040           DCA       EX1        /EXPONENT
5632 1256           TAD       ADDR
5633 3257           DCA       SAV
5634 2257           ISZ       SAV
5635 1657           TAD  I    SAV
5636 3041           DCA       AC1H       /HIGH ORDER MANTISSA
5637 2257           ISZ       SAV
5640 1657           TAD  I    SAV
5641 3042           DCA       AC1L       /LOW ORDER MANTISSA
5642 1253           TAD       JUMP
5643 7106           CLL  RTL
5644 7006           RTL
5645 0260           AND       MASK3      /GET BITS 0-2, IE OPCODE
5646 1265           TAD       TABLE      /LOOKUP IN TABLE
5647 3254           DCA       JUMP2
5650 1654           TAD  I    JUMP2
5651 3254           DCA       JUMP2
5652 5654           JMP  I    JUMP2      /GO THERE
5653 0000 JUMP,     0
5654 0000 JUMP2,    0
5655 0000 GO2,      0
5656 0000 ADDR,     0
5657 0000 SAV,      0
5660 0017 MASK3,    0017
5661 7600 MASK5,    7600
5662 0177 MK0177,   0177
5663 0200 PAGENO,   0200
5664 0400 INDRCT,   0400
5665 5666 TABLE,    .+1
```

```
5666 5742              EXIT                    /TABLE USED IN INTERPRETING
5667 5716              FLAD                    /BITS 0-2 OF PSEUDO
5670 5737              FLSU                    /INSTRUCTION
5671 5761              FLMY
5672 6305              FLDV                    /IF OPCODE=0, GO TO EXIT
5673 5676              FLGT                    /AND INTERPRET BITS 8-11
5674 5705              FLPT
5675 5772              FINK
5676 1040 FLGT,        TAD       EX1           /FGET=5
5677 3044              DCA       EXP
5700 1041              TAD       AC1H
5701 3045              DCA       HORD
5702 1042              TAD       AC1L
5703 3046              DCA       LORD
5704 5201              JMP       FPNT+1
5705 1044 FLPT,        TAD       EXP           /FPUT=6
5706 3656              DCA   I   ADDR
5707 2256              ISZ       ADDR
5710 1045              TAD       HORD
5711 3656              DCA   I   ADDR
5712 2256              ISZ       ADDR
5713 1046              TAD       LORD
5714 3656              DCA   I   ADDR
5715 5201              JMP       FPNT+1
5716 4770 FLAD,        JMS   I   ALGN          /FLAD=1-FIRST ALIGN EXPONENTS
5717 5201              JMP       FPNT+1        /NO ALIGNMENT IS POSSIBLE
5720 4771              JMS   I   UNORM         /LARGER OF THE TWO IS IN F.A.
5721 7100              CLL
5722 1043              TAD       OVER1         /TRIPLE PRECISION ADDITION
5723 1047              TAD       OVER2         /SINCE BITS ARE SHIFTED
5724 3047              DCA       OVER2         /RIGHT
5725 7004              RAL
5726 1042              TAD       AC1L
5727 1046              TAD       LORD
5730 3046              DCA       LORD
5731 7004              RAL
5732 1041              TAD       AC1H
5733 1045              TAD       HORD
5734 3045              DCA       HORD
5735 4776              JMS   I   NORM          /NORMALIZE THE RESULT
5736 5201              JMP       FPNT+1
5737 4741 FLSU,        JMS   I   OPMINS        /FSUB=2 - NEGATE THE OPERAND
5740 5316              JMP       FLAD          /ADD
5741 6400 OPMINS,      MINUS2
5742 1253 EXIT,        TAD       JUMP          /OPCODE=0
5743 0260              AND       MASK3         /ARE BITS 8-11=0
5744 7450              SNA
5745 5600              JMP   I   FPNT          /YES=FEXT
5746 1360              TAD       ACON6         /LOOKUP ON TABLE
5747 3254              DCA       JUMP2
5750 1654              TAD   I   JUMP2
5751 3254              DCA       JUMP2
5752 1200              TAD       FPNT
5753 3255              DCA       GO2
5754 4654              JMS   I   JUMP2         /CALL AS A SUBROUTINE
5755 1255              TAD       GO2           /RESTORE F.P. POINTER
5756 3200              DCA       FPNT
5757 5201              JMP       FPNT+1        /GET NEXT PSEUDO INSTRUCTION
5760 6544 ACON6,       TABLE6-1                /CALLING TO A DEPTH OF ONE
5761 7201 FLMY,        CLA  IAC                /FMPY=3
```

```
5762 1040             TAD       EX1
5763 1044             TAD       EXP          /ADD EXPONENTS TOGETHER
5764 3044             DCA       EXP
5765 4767             JMS   I   MULT         /MULTIPLY
5766 5201             JMP       FPNT+1
5767 6221 MULT,       DMULT
5770 6020 ALGN,       ALIGN
5771 6564 UNORM,      DUNORM
5772 2656 FINK,       ISZ   I   ADDR
5773 2656             ISZ   I   ADDR
5774 2656             ISZ   I   ADDR
5775 5201             JMP       FPNT+1
5776 6600 NORM,       DNORM

          /ND6C2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE       24
          *           6000
          /FLOATING POINT INTERPRETER
6000 0000 ACMINS,     0                      /ROUTINE TO PERFORM
6001 7300             CLL   CLA
6002 1047             TAD       OVER2        /TRIPLE PRECISION NEGATION
6003 7041             CMA   IAC              /OF FLOATING AC
6004 3047             DCA       OVER2
6005 1046             TAD       LORD
6006 7040             CMA
6007 7430             SZL
6010 7101             CLL   IAC
6011 3046             DCA       LORD
6012 1045             TAD       HORD
6013 7040             CMA
6014 7430             SZL
6015 7101             CLL   IAC
6016 3045             DCA       HORD
6017 5600             JMP   I   ACMINS
6020 0000 ALIGN,      0                      /SUBROUTINE TO ALIGN
6021 1045             TAD       HORD         /BINARY POINTS FOR
6022 7640             SZA   CLA              /ADD-SUBTRCT
6023 5227             JMP       .+4
6024 1046             TAD       LORD         /IS MANTISSA ZERO
6025 7650             SNA   CLA
6026 5337             JMP       NOHERE       /YES
6027 1041             TAD       AC1H         /IS OPERAND ZERO
6030 7640             SZA   CLA
6031 5235             JMP       .+4
6032 1042             TAD       AC1L
6033 7650             SNA   CLA
6034 5620             JMP   I   ALIGN        /BOTH ARE ZERO-EXIT
6035 1040             TAD       EX1
6036 7041             CMA   IAC
6037 1044             TAD       EXP
6040 7450             SNA                    /ARE EXPONENTS EQUAL?
6041 5314             JMP       DONE         /YES
6042 7500             SMA
6043 7041             CMA   IAC
6044 3342             DCA       AMOUNT
6045 1342             TAD       AMOUNT
6046 1343             TAD       TEST2
6047 7700             SMA   CLA              /CAN EXPONENTS BE ALIGNED?
6050 5256             JMP       .+6          /YES
6051 4316             JMS       OUTGO        /NO
```

```
6052 7430            SZL
6053 1350            TAD       TAG2
6054 1347            TAD       TAG1
6055 5325            JMP       NOGO
6056 4316            JMS       OUTGO
6057 7420            SNL                    /SET UP ADDRESSES
6060 1350            TAD       TAG2
6061 1347            TAD       TAG1
6062 3344            DCA       TEST3
6063 1342            TAD       AMOUNT
6064 7041            CMA  IAC
6065 1744            TAD  I    TEST3
6066 3744            DCA  I    TEST3
6067 2344            ISZ       TEST3
6070 1344            TAD       TEST3
6071 3345            DCA       TEST4
6072 2345            ISZ       TEST4
6073 1345            TAD       TEST4
6074 3346            DCA       TEST5
6075 2346            ISZ       TEST5
6076 7100 SHIFT,     CLL                    /THIS ROUTINE DOES
6077 1744            TAD  I    TEST3       /THE ACTUAL SHIFTING
6100 7510            SPA
6101 7020            CML
6102 7010            RAR
6103 3744            DCA  I    TEST3
6104 1745            TAD  I    TEST4
6105 7010            RAR
6106 3745            DCA  I    TEST4
6107 1746            TAD  I    TEST5
6110 7010            RAR
6111 3746            DCA  I    TEST5
6112 2342            ISZ       AMOUNT
6113 5276            JMP       SHIFT
6114 2220 DONE,      ISZ       ALIGN
6115 5620            JMP  I    ALIGN
6116 0000 OUTGO,     0                      /DETERMINE WHICH TO SHIFT
6117 1040            TAD       EX1
6120 7041            CMA  IAC
6121 1044            TAD       EXP
6122 7004            RAL
6123 7200            CLA
6124 5716            JMP  I    OUTGO
6125 3344 NOGO,      DCA       TEST3       /CAN'T BE ALIGNED
6126 1744            TAD  I    TEST3       /LARGEST GOES INTO FAC
6127 3044            DCA       EXP
6130 2344            ISZ       TEST3
6131 1744            TAD  I    TEST3
6132 3045            DCA       HORD
6133 2344            ISZ       TEST3
6134 1744            TAD  I    TEST3
6135 3046            DCA       LORD
6136 5620            JMP  I    ALIGN
6137 1040 NOHERE,    TAD       EX1         /MANTISSA=0
6140 3044            DCA       EXP
6141 5314            JMP       DONE
6142 0000 AMOUNT,    0
6143 0030 TEST2,     0030
6144 0000 TEST3,     0
6145 0000 TEST4,     0
```

```
6146 0000 TEST5,    0
6147 0044 TAG1,     EXP
6150 7774 TAG2,     -4                      /EX1-EXP
6151 5601 RETN2,    FPNT+1
6152 1362 ERRCR1,   TAD       GOOF          /DIVISION BY ZERO
6153 3044           DCA       EXP           /SET TO LARGEST + VALUE
6154 1362           TAD       GOOF
6155 3045           DCA       HORD
6156 7040           CMA
6157 3046           DCA       LORD
6160 2061           ISZ       FLAG          /SET FLAG
6161 5751           JMP   I   RETN2
6162 3777 GOOF,     3777
6163 0000 SQUARE,   0
6164 4407           JMS   I   0007
6165 6052           FPUT      FPAC1
6166 3052           FMPY      FPAC1
6167 0000           FEXT
6170 5763           JMP   I   SQUARE
6171 0000 EXIT6,    0                       /DUMMY SUBROUTINE
6172 5771           JMP   I   EXIT6
6173 4471 ANGPRT,   JMS   I   LANGPT
6174 5434           JMP   I   LOMGA
6175 4427 DRIVE,    JMS   I   LMOTST
6176 7000           NOP
6177 5434           JMP   I   LOMGA

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     25
          *         6200
          /FLOATING POINT INTERPRETER
6200 0000 DIV1,     0                       /SHIFT FAC RIGHT
6201 7300           CLA   CLL
6202 1045           TAD       HORD
6203 7510           SPA
6204 7120           CLL   CML
6205 7010           RAR
6206 3045           DCA       HORD
6207 1046           TAD       LORD
6210 7010           RAR
6211 3046           DCA       LORD
6212 1047           TAD       OVER2
6213 7010           RAR
6214 3047           DCA       OVER2
6215 7100           CLL
6216 2044           ISZ       EXP
6217 7000           NOP
6220 5600           JMP   I   DIV1
6221 0000 DMULT,    0                       /DOUBLE PRECISION MULTIPLY
6222 7300           CLA   CLL               /SAVE PRODUCT TRIPLE PRECISION
6223 1342           TAD       SMACLA
6224 3347           DCA       SNSWIT        /CALLS A SINGLE PRECISION
6225 4336           JMS       SIGH          /MULTIPLY 3 TIMES
6226 1042           TAD       AC1L
6227 3756           DCA   I   MP2PT
6230 1046           TAD       LORD
6231 4755           JMS   I   MP4PT
6232 7200           CLA
6233 1757           TAD   I   MP5PT
6234 3047           DCA       OVER2
```

```
6235 1045            TAD       HORD
6236 3756            DCA   I   MP2PT
6237 1042            TAD       AC1L
6240 4755            JMS   I   MP4PT
6241 1047            TAD       OVER2
6242 3047            DCA       OVER2
6243 7004            RAL
6244 1757            TAD   I   MP5PT
6245 3367            DCA       DD
6246 7004            RAL
6247 3370            DCA       KEEP
6250 1041            TAD       AC1H
6251 3756            DCA   I   MP2PT
6252 1046            TAD       LORD
6253 4755            JMS   I   MP4PT
6254 1047            TAD       OVER2
6255 3047            DCA       OVER2
6256 7004            RAL
6257 1757            TAD   I   MP5PT
6260 1367            TAD       DD
6261 3367            DCA       DD
6262 7004            RAL
6263 1370            TAD       KEEP
6264 3370            DCA       KEEP
6265 1045            TAD       HORD
6266 3756            DCA   I   MP2PT
6267 1041            TAD       AC1H
6270 4755            JMS   I   MP4PT
6271 1367            TAD       DD
6272 3046            DCA       LORD
6273 7004            RAL
6274 1757            TAD   I   MP5PT
6275 1370            TAD       KEEP
6276 3045            DCA       HORD
6277 4760            JMS   I   NORMF
6300 3047            DCA       OVER2
6301 2365            ISZ       SGN
6302 5621            JMP   I   DMULT
6303 4773            JMS   I   MINS
6304 5621            JMP   I   DMULT
6305 1041  FLDV,      TAD       AC1H
6306 7640            SZA   CLA
6307 5313            JMP       .+4
6310 1042            TAD       AC1L
6311 7650            SNA   CLA
6312 5774            JMP   I   ERROR       /DIVISION BY ZERO
6313 1040            TAD       EX1
6314 7041            CMA   IAC
6315 1044            TAD       EXP
6316 7001            IAC
6317 3044            DCA       EXP         /SUBTRACT EXPONENTS
6320 1362            TAD       SPACLA
6321 3347            DCA       SNSWIT
6322 4336            JMS       SIGN        /SET UP SIGNS
6323 4761            JMS   I   DIVIDE      /DIVIDE
6324 1757            TAD   I   MP5PT
6325 1041            TAD       AC1H        /ROUND OFF IN 23RD BIT
6326 7630            SZL   CLA
6327 7001            IAC
6330 3042            DCA       AC1L
```

```
6331 3041            DCA       AC1H
6332 2365            ISZ       SGN          /TEST SIGN
6333 5375            JMP       COMP
6334 5735            JMP  I    .+1          /ADD ROUNDING
6335 5720 LFLAC2,    FLAD+2
6336 0000 SIGH,      0                      /TEST SIGN OF RESULT
6337 1366            TAD       REST         /SET UP BY MULTIPLY AND
6340 3365            DCA       SGN          /DIVIDE
6341 1045            TAD       HORD
6342 7700 SMACLA,    SMA  CLA
6343 5346            JMP       .+3
6344 4773            JMS  I    MINS
6345 2365            ISZ       SGN
6346 1041            TAD       AC1H
6347 7700 SNSWIT,    SMA  CLA                 /OR SPA CLA
6350 5736            JMP  I    SIGH
6351 4771            JMS  I    MINS2
6352 2365            ISZ       SGN
6353 7000            NOP
6354 5736            JMP  I    SIGH
6355 6437 MP4PT,     MP4
6356 6471 MP2PT,     MP2
6357 6465 MP5PT,     MP5
6360 6600 NORMF,     DNORM
6361 6472 DIVIDE,    DUBDIV
6362 7710 SPACLA,    SPA  CLA
6363 0000            SPARE
6364 5601 RETURN,    FPNT+1
6365 0000 SGN,       0
6366 7776 REST,      7776
6367 0000 DD,        0
6370 0000 KEEP,      0
6371 6400 MINS2,     MINUS2                   /-AC1H,AC1L
6372 6420 RAR2,      DIV2                     /AC1H,AC1L/2
6373 6000 MINS,      ACMINS
6374 6152 ERROR,     ERROR1
6375 4771 COMP,      JMS  I    MINS2
6376 4773            JMS  I    MINS         /NEGATE
6377 5735            JMP  I    LFLAD2

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      26
          *          6400
          /FLOATING POINT INTERPRETER
6400 0000 MINUS2,    0                      /NEGATE OPERAND
6401 7300            CLA  CLL               /TRIPLE PRECISION
6402 1043            TAD       OVER1
6403 7041            CMA  IAC
6404 3043            DCA       OVER1
6405 1042            TAD       AC1L
6406 7040            CMA
6407 7430            SZL
6410 7101            CLL  IAC
6411 3042            DCA       AC1L
6412 1041            TAD       AC1H
6413 7040            CMA
6414 7430            SZL
6415 7101            CLL  IAC
6416 3041            DCA       AC1H
6417 5600            JMP  I    MINUS2
```

```
6420 0000 DIV2,     0                          /SHIFT OPERAND RIGHT
6421 7300           CLA  CLL                   /TRIPLE PRECISION
6422 1041           TAD      AC1H
6423 7510           SPA
6424 7120           CLL  CML
6425 7010           RAR
6426 3041           DCA      AC1H
6427 1042           TAD      AC1L
6430 7010           RAR
6431 3042           DCA      AC1L
6432 1043           TAD      OVER1
6433 7010           RAR
6434 3043           DCA      OVER1
6435 7100           CLL
6436 5620           JMP  I   DIV2
6437 0000 MP4,      0                          /SINGLE PRECISION MULTIPLY
6440 3266           DCA      MP1               /12 BITS BY 12 BITS
6441 3265           DCA      MP5
6442 1270           TAD      THIR
6443 3267           DCA      MP3
6444 7100           CLL
6445 1266           TAD      MP1
6446 7010           RAR
6447 3266           DCA      MP1
6450 1265           TAD      MP5
6451 7420           SNL
6452 5255           JMP      .+3
6453 7100           CLL
6454 1271           TAD      MP2
6455 7010           RAR
6456 3265           DCA      MP5
6457 2267           ISZ      MP3
6460 5245           JMP      MP4+6
6461 1266           TAD      MP1
6462 7010           RAR
6463 7100           CLL
6464 5637           JMP  I   MP4
6465 0000 MP5,      0
6466 0000 MP1,      0
6467 0000 MP3,      0
6470 7764 THIR,     7764
6471 0000 MP2,      0
6472 0000 DUBDIV,   0                          /DOUBLE PRECISION DIVIDE
6473 7300           CLA  CLL
6474 3051           DCA      QUOL
6475 1344           TAD      MIF
6476 3267           DCA      MP3
6477 5306           JMP      DVX
6500 1046 DV3,      TAD      LORD
6501 7004           RAL
6502 3046           DCA      LORD
6503 1045           TAD      HORD
6504 7004           RAL
6505 3045           DCA      HORD
6506 1042 DVX,      TAD      AC1L
6507 1046           TAD      LORD
6510 3271           DCA      MP2
6511 7004           RAL
6512 1045           TAD      HORD
6513 1041           TAD      AC1H
```

```
6514 7420             SNL
6515 5321             JMP       DV2-1
6516 3045             DCA       HORD
6517 1271             TAD       MP2
6520 3046             DCA       LORD
6521 7200             CLA
6522 1051 DV2,        TAD       QUOL
6523 7004             RAL
6524 3051             DCA       QUOL
6525 1047             TAD       OVER2
6526 7004             RAL
6527 3047             DCA       OVER2
6530 2267             ISZ       MP3
6531 5300             JMP       DV3
6532 1051             TAD       QUOL
6533 3046             DCA       LORD
6534 1045             TAD       HORD
6535 7106             CLL  RTL
6536 3265             DCA       MP5
6537 1047             TAD       OVER2
6540 3045             DCA       HORD
6541 3047             DCA       OVER2
6542 1265             TAD       MP5
6543 5672             JMP  I    DUBDIV
6544 7751 MIF,        7751
6545 6163 TABLE6,     SQUARE                  /TABLE FOR INTERPRETATION
6546 6656             SQROOT                  /OF BITS 8-11
6547 5200             SINE                    /CONTAINS ABSOLUTE ADDRESSES
6550 5310             COS                     /OF PROGRAMS CALLED AS
6551 5400             ARTN                    /SUBROUTINES
6552 5303             NEG                     /EXIT6=A DUMMY OR NOP
6553 5570             ABS
6554 6600             DNORM
6555 6171             EXIT6
6556 6171             EXIT6
6557 7400             7400
6560 7470             FOUT
6561 7553             DPOUT
6562 6171             EXIT6
6563 6171             EXIT6
6564 0000 DUNORM,     0
6565 4220             JMS       DIV2        /SHIFT OPERAND RIGHT
6566 4772             JMS  I    RAR1
6567 2040             ISZ       EX1
6570 7000             NOP
6571 5764             JMP  I    DUNORM
6572 6200 RAR1,       DIV1
6573 4404 WAVEL,      JMS  I    LFNK
6574 4407             JMS  I    7
6575 6526             FPUT I    LWVL
6576 0000             FEXT
6577 5434             JMP  I    LOMGA

          /ND662-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE       27
          *           6600
          /FLOATING POINT INTERPRETER
6600 0000 DNORM,      0                       /SUBROUTINE TO NORMALISE
6601 7300             CLA  CLL                /FLOATING ACCUMULATOR
6602 3255             DCA       AMT1
```

```
6603 3254            DCA       SIGN1
6604 1045            TAD       HORD
6605 7510            SPA                    /IS MANTISSA NEGATIVE
6606 2254            ISZ       SIGN1        /YES
6607 7640            SZA  CLA               /IS MANTISSA=0
6610 5217            JMP       GO6          /NO
6611 1046            TAD       LORD
6612 7640            SZA  CLA
6613 5217            JMP       GO6
6614 1047            TAD       OVER2
6615 7650            SNA  CLA
6616 5251            JMP       EXIT2        /YES
6617 1254 GO6,       TAD       SIGN1        /NO
6620 7640            SZA  CLA               /NEGATIVE?
6621 4653            JMS  I    NEGIT        /YES
6622 1045 LOP,       TAD       HORD         /WILL SHIFT BE TOO FAR?
6623 7104            RAL  CLL
6624 7710            SPA  CLA
6625 5241            JMP       EXIT1        /YES
6626 1047            TAD       OVER2
6627 7104            CLL  RAL
6630 3047            DCA       OVER2
6631 1046            TAD       LORD         /NO SHIFT MANTISSA LEFT
6632 7004            RAL
6633 3046            DCA       LORD
6634 1045            TAD       HORD
6635 7004            RAL
6636 3045            DCA       HORD
6637 2255            ISZ       AMT1         /COUNT NO. OF TIMES SHIFTED
6640 5222            JMP       LOP
6641 1255 EXIT1,     TAD       AMT1         /CORRECT EXPONENT
6642 7041            CMA  IAC
6643 1044            TAD       EXP
6644 3044            DCA       EXP
6645 1254            TAD       SIGN1        /NEGATIVE?
6646 7640            SZA  CLA
6647 4653            JMS  I    NEGIT        /YES
6650 5600            JMP  I    DNORM
6651 3044 EXIT2,     DCA       EXP          /SET TO ZERO
6652 5600            JMP  I    DNORM
6653 6000 NEGIT,     ACMINS
6654 0000 SIGN1,     0
6655 0000 AMT1,      0
6656 0000 SQROOT,    0                      /FLOATING SQUARE ROOT
6657 3362            DCA       FLAG1        /TAKES ROOT OF ABSOLUTE
6660 4407            JMS  I    OC7
6661 6052            FPUT      FPAC1        /VALUE
6662 0000            FEXT                   /NEWTON'S METHOD IS USED
6663 1045            TAD       HORD
6664 7710            SPA  CLA
6665 5351            JMP       SQEND1       /NUMBER IS NEGATIVE
6666 1044            TAD       EXP
6667 7100            CLL
6670 7510            SPA
6671 7020            CML
6672 7010            RAR
6673 3356            DCA       ITER1        /MAKE FIRST APPROXIMATION
6674 7430            SZL
6675 2356            ISZ       ITER1
6676 7000            NOP
```

```
6677 1361            TAD       SQCON1
6700 3357            DCA       ITER1+1
6701 3360            DCA       ITER1+2
6702 1053            TAD       FPAC1+1
6703 7640            SZA  CLA
6704 5310            JMP       CLCU
6705 1054            TAD       FPAC1+2
6706 7650            SNA  CLA
6707 5354            JMP       SQEND       /NUMBER=0
6710 4407 CLCU,      JMS  I    0007
6711 5052            FGET      FPAC1
6712 4356            FDIV      ITER1
6713 1356            FADD      ITER1
6714 0000            FEXT
6715 7240            CLA  CMA
6716 1044            TAD       EXP
6717 3044            DCA       EXP
6720 1044            TAD       EXP
6721 7041            CMA  IAC
6722 1356            TAD       ITER1
6723 7640            SZA  CLA                /ARE EXPONENTS EQUAL?
6724 5345            JMP       ROOTGO      /NO
6725 1045            TAD       HORD        /HIGH-ORDER MANTISSAS EQUAL?
6726 7041            CMA  IAC
6727 1357            TAD       ITER1+1
6730 7640            SZA  CLA
6731 5345            JMP       ROOTGO      /NO
6732 1046            TAD       LORD
6733 7041            CMA  IAC
6734 1360            TAD       ITER1+2     /DO LOW-ORDER MANTISSAS AGREE
6735 7500            SMA
6736 7041            CMA  IAC                /WITHIN ONE BIT?
6737 7001            IAC
6740 7710            SPA  CLA
6741 5345            JMP       ROOTGO      /NO
6742 1362            TAD       FLAG1
6743 3061            DCA       FLAG
6744 5656            JMP  I    SQROOT
6745 4407 ROOTGO,    JMS  I    0007
6746 6356            FPUT      ITER1
6747 0000            FEXT
6750 5310            JMP       CLCU
6751 4653 SQEND1,    JMS  I    NEGIT       /NEGATE FAC
6752 2362            ISZ       FLAG1
6753 5260            JMP       SQROOT+2
6754 3044 SQEND,     DCA       EXP
6755 5656            JMP  I    SQROOT
6756 0000 ITER1,     0
6757 0000            0
6760 0000            0
6761 3015 SQCON1,    3015
6762 0000 FLAG1,     0
6763 0000 PRTANG,    0                       /ENTRY
6764 4773            JMS  I    LWHERE
6765 4435            JMS  I    LCRLF
6766 4425            JMS  I    LFOP
6767 7774            -4
6770 0114            PTTH
6771 4435            JMS  I    LCRLF
6772 5763            JMP  I    PRTANG      /RETURN
```

```
6773 0615 LWHERE,   WHERE
6774 0004 TEN,      0004
6775 2400           2400
6776 0000           0000

          /FLOATING POINT PACKAGE
          /INPUT-OUTPUT ROUTINES
          /ND6C2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     28
          *         7000
          EX10=     40
          EXPONT=   44
          HORDER=   45
          LORDER=   46
          /SINGLE PRECISION INTEGER INPUT
7000 0000 INTIN,    0                        /ENTRY
7001 4207           JMS       DECONV         /READ AND CONVERT
7002 1060           TAD       SIGN           /TEST SIGN
7003 7640           SZA  CLA
7004 4347           JMS       MSIGN
7005 1056           TAD       LO             /RESULT IN AC
7006 5600           JMP  I    INTIN          /RETURN
          /DOUBLE PRECISION DECIMAL-BINARY
          /INPUT AND CONVERSION
7007 0000 DECONV,   0
7010 3055           DCA       HI
7011 3056           DCA       LO
7012 3060           DCA       SIGN
7013 3303           DCA       DNUMBR
7014 3674           DCA  I    LPRSW
7015 5220           JMP       STRT
7016 1057 STRTOP,   TAD       CHAR
7017 4506           JMS  I    LOPCH
7020 4507 STRT,     JMS  I    LIPCH          / INPUT NEXT CHARACTER
7021 3057           DCA       CHAR
7022 1057           TAD       CHAR           /TEST FOR RUBOUT
7023 1273           TAD       MRBOUT
7024 7650           SNA  CLA
7025 5210           JMP       DECONV+1       /RUBOUT  START AGAIN
7026 1057 DGTST,    TAD       CHAR           /IS IT A DIGIT
7027 1276           TAD       MIN9           /-272=-COLON
7030 7450           SNA                      /TERMINATOR?
7031 5270           JMP       DECON          /YES
7032 7500           SMA
7033 5247           JMP       TEST           /NO
7034 1277           TAD       MZERO
7035 7510           SPA
7036 5247           JMP       TEST           /NO
7037 3302           DCA       DIGIT          /YES
7040 1055           TAD       HI
7041 1301           TAD       MASK
7042 7710           SPA  CLA
7043 5220           JMP       STRT           /YES  IGNORE
7044 2303           ISZ       DNUMBR         /INDEX NUMBER OF DIGITS
7045 4304           JMS       MULTIO
7046 5216           JMP       STRTOP         /CONTINUE
7047 7200 TEST,     CLA
7050 1674           TAD  I    LPRSW
7051 7640           SZA  CLA
7052 5220           JMP       STRT
```

```
7053 1303            TAD       DNUMBR
7054 7640            SZA  CLA
7055 5264            JMP       DECTST
7056 1057            TAD       CHAR        /NO TEST FOR MINUS SIGN
7057 1275            TAD       MINUS
7060 7640            SZA  CLA
7061 5264            JMP       .+3
7062 2060            ISZ       SIGN        /MINUS SET INDICATOR
7063 5216            JMP       STRTOP
7064 1057 DECTST,    TAD       CHAR
7065 1300            TAD       DEC         /IS IT A DECIMAL POINT
7066 7640            SZA  CLA
7067 5220            JMP       STRT
7070 1057 DECON,     TAD       CHAR
7071 4506            JMS  I    LOPCH
7072 5607            JMP  I    DECONV      /RETURN
7073 7401 MRBOUT,    -377
7074 7452 LPRSW,     PRSW
7075 7523 MINUS,     -255                  /TEST FOR SIGN
7076 7506 MIN9,      -272                  /TEST FOR DIGIT
7077 0012 MZERO,     272-260
7100 7522 DEC,       -256
7101 3465 MASK,      3465
7102 0000 DIGIT,     0                     /STORAGE FOR DIGIT
7103 0000 DNUMBR,    0                     /NUMBER OF DIGITS
7104 0000 MULT10,    0                     /ROUTINE TO MULTIPLY
7105 1056            TAD       LO          /DOUBLE PRECISION WORD
7106 3323            DCA       LTEMP       /BY TEN (DECIMAL)
7107 1055            TAD       HI
7110 3322            DCA       HTEMP
7111 4324            JMS       MULT2       /CALL SUBROUTINE TO
7112 4324            JMS       MULT2       /MULTIPLY BY TWO
7113 4335            JMS       DUBLAD      /CALL DOUBLE PRECISION ADD
7114 4324            JMS       MULT2
7115 1302            TAD       DIGIT       /ADD LAST DIGIT RECEIVED
7116 3323            DCA       LTEMP
7117 3322            DCA       HTEMP
7120 4335            JMS       DUBLAD
7121 5704            JMP  I    MULT10
7122 0000 HTEMP,     0                     /DOUBLE RECISION WORD
7123 0000 LTEMP,     0
7124 0000 MULT2,     0                     /MULTIPLY LO, HI BY 2
7125 7300            CLA  CLL
7126 1056            TAD       LO
7127 7004            RAL
7130 3056            DCA       LO
7131 1055            TAD       HI
7132 7004            RAL
7133 3055            DCA       HI
7134 5724            JMP  I    MULT2
7135 0000 DUBLAD,    0                     /DOUBLE PRECISION ADDITION
7136 7300            CLA  CLL
7137 1056            TAD       LO
7140 1323            TAD       LTEMP
7141 3056            DCA       LO
7142 7004            RAL
7143 1055            TAD       HI
7144 1322            TAD       HTEMP
7145 3055            DCA       HI
7146 5735            JMP  I    DUBLAD
```

```
7147 0000 MSIGN,    0                          /ROUTINE TO FORM
7150 7300           CLA  CLL                   /2S COMPLEMENT
7151 1056           TAD       LO
7152 7041           CMA  IAC
7153 3056           DCA       LO
7154 1055           TAD       HI
7155 7040           CMA
7156 7430           SZL
7157 7001           IAC
7160 3055           DCA       HI
7161 5747           JMP  I    MSIGN
          /SET SIGN CHARACTER FOR OUTPUT
7162 0000 STSG,     0
7163 7710           SPA  CLA
7164 1370           TAD       MCH          /NEGATIVE
7165 1371           TAD       SPCH         /POSITIVE
7166 3060           DCA       SIGN
7167 5762           JMP  I    STSG         /RETURN
7170 0015 MCH,      255-240                /MINUS-SPACE
7171 0240 SPCH,     240                    /SPACE
          /READ NO OF STEPS FOR DATA COLLECTION
7172 4430 STEPNO,   JMS  I    LINTIN
7173 3141           DCA       NSTP
7174 5434           JMP  I    LOMGA
7175 0000 ZERO,     0
7176 0000           0
7177 0000           0

          /ND602-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     29
          *         7200
          /VARIABLE-FORMAT FLOATING OUTPUT
7200 0000 VFOUT,    0
7201 1600           TAD  I    VFOUT        /MOVE ARGUMENTS
7202 3251           DCA       AAA
7203 2200           ISZ       VFOUT
7204 1600           TAD  I    VFOUT
7205 3252           DCA       BBB
7206 2200           ISZ       VFOUT        /SET EXIT
7207 1045           TAD       HORDER       /SET SIGN CHARACTER
7210 4660           JMS  I    LSTSG
7211 1252           TAD       BBB          /IS SCALING NEEDED
7212 7450           SNA
7213 5223           JMP       ROND         /NO
7214 7041           CMA  IAC                /YES SET COUNTER
7215 3057           DCA       CHAR
7216 4407 SCAL,     JMS  I    7            /MULTIPLY BY TEN
7217 3657           FMPY I    LTEN
7220 0000           FEXT
7221 2057           ISZ       CHAR         /TEST COUNTER
7222 5216           JMP       SCAL
7223 4421 ROND,     JMS  I    LABS
7224 4407           JMS  I    7
7225 1654           FADD I    HALVE
7226 0000           FEXT
7227 1044           TAD       EXPONT       /IS NUMBER TOO LARGE
7230 1255           TAD       M30
7231 7710           SPA  CLA
7232 5240           JMP       FIXIT        /NO
7233 1256           TAD       AST          /YES PUT OUT *
```

```
7234 4506            JMS  I   LOPCH
7235 1364            TAD      COLON
7236 4506            JMS  I   LOPCH
7237 5600            JMP  I   VFOUT      /RETURN
7240 1243 FIXIT,     TAD      E27        /OBTAIN DOUBLE PRECSN INTEGER
7241 3040            DCA      EXIC       /(PREVIOUS FEXT LEAVES
7242 4661            JMS  I   LALIG      /  AC1H NON-ZERO)
7243 0027 E27,       27                  /CONSTANT, UNUSED ERROR RETURN
7244 1045            TAD      HORDER     /MOVE ARGUMENT
7245 3055            DCA      HI
7246 1046            TAD      LORDER
7247 3056            DCA      LO
7250 4262            JMS      DROP       /OUTPUT
7251 0000 AAA,       0                   /NUMBER OF DIGITS BEFORE
7252 0000 BBB,       0                   /  AND AFTER POINT
7253 5600            JMP  I   VFOUT      /RETURN
7254 3555 HALVE,     HALF
7255 7750 M30,       -30
7256 0252 AST,       252                 /ASTERISK
7257 6774 LTEN,      TEN
7260 7162 LSTSG,     STSG
7261 6020 LALIG,     ALIGN
          /BASIC DOUBLE-PRECISION OUTPUT
7262 0000 DROP,      0
7263 1074            TAD      M8         /SET COUNTERS
7264 3374            DCA      CHRK       /CHRK=-8
7265 1074            TAD      M8         /FLDK=-8+A+B
7266 1662            TAD  I   DROP
7267 2262            ISZ      DROP
7270 1662            TAD  I   DROP
7271 3375            DCA      FLDK
7272 1360            TAD      MIN7       /DPTK=-7+B
7273 1662            TAD  I   DROP
7274 3376            DCA      DPTK
7275 3057            DCA      CHAR       /CLEAR CHARACTER
7276 3377            DCA      PREV       /CLEAR PREVIOUS NON-SPACE IND
7277 1376 TESTU,     TAD      DPTK       /TEST FOR UNITS POSITION
7300 7650            SNA  CLA
7301 5307            JMP      PRSGN
7302 1375            TAD      FLDK       /TEST FIELD COUNTER
7303 7710            SPA  CLA
7304 5315            JMP      TESTK      /NO OUTPUT
7305 1361            TAD      SPACE      /OUTPUT SPACE
7306 5314            JMP      PRCH
7307 1060 PRSGN,     TAD      SIGN       /PRINT SIGN
7310 4506            JMS  I   LOPCH
7311 2377            ISZ      PREV       /SET PREVIOUS NON-SPACE IND
7312 1057 STCH,      TAD      CHAR       /SET UP CHARACTER
7313 1362            TAD      ZCH
7314 4506 PRCH,      JMS  I   LOPCH      /PRINT CHARACTER OR SPACE
7315 2374 TESTK,     ISZ      CHRK       /TEST CHARACTER COUNTER
7316 5322            JMP      .+4
7317 1364            TAD      COLON
7320 4506            JMS  I   LOPCH
7321 5662            JMP  I   DROP       /RETURN
7322 1376            TAD      DPTK       /TEST FOR DECIMAL POINT
7323 7640            SZA  CLA
7324 5327            JMP      ADV
7325 1363            TAD      DCH        /PRINT DECIMAL POINT
7326 4506            JMS  I   LOPCH
```

```
7327 2376 ADV,       ISZ        DPTK       /ADVANCE COUNTERS
7330 7000            NOP
7331 2375            ISZ        FLDK
7332 7000            NOP
7333 3057            DCA        CHAR       /CLEAR CHARACTER
7334 1365            TAD        HIMM       /SET TEMP AT MINUS ONE MILLION
7335 3772            DCA   I    LHTEMP
7336 1366            TAD        LOMM
7337 3773            DCA   I    LLTEMP
7340 4770 GEN,       JMS   I    LDBLAD     /GENERATE CHARACTER
7341 7420            SNL                   /HAS TOO MUCH BEEN SUBTRACTED
7342 5345            JMP        ADJ
7343 2057            ISZ        CHAR       /NO
7344 5340            JMP        GEN
7345 4767 ADJ,       JMS   I    LMSIGN     /YES ADD IT BACK
7346 4770            JMS   I    LDBLAD
7347 4767            JMS   I    LMSIGN
7350 4771            JMS   I    LMLT10     /MULTIPLY BY TEN
7351 1377            TAD        PREV       /TEST FOR PREV NON-SPACE OUT
7352 7640            SZA   CLA
7353 5312            JMP        STCH
7354 1057            TAD        CHAR       /TEST FOR ZERO
7355 7640            SZA   CLA
7356 5307            JMP        PRSGN      /NON-ZERO
7357 5277            JMP        TESTU      /ZERO
          M8=        M10
7360 7771 MIN7,      -7
7361 0240 SPACE,     240
7362 0260 ZCH,       260
7363 0256 DCH,       256
7364 0272 COLON,     272
7365 7413 HIMM,      7413                  /MINUS ONE MILLION
7366 6700 LOMM,      6700                  /DOUBLE PRECISION
7367 7147 LMSIGN,    MSIGN
7370 7135 LDBLAD,    DUBLAD
7371 7104 LMLT10,    MULT10
7372 7122 LHTEMP,    HTEMP
7373 7123 LLTEMP,    LTEMP
7374 0000 CHRK,      0                     /CHARACTER COUNTER
7375 0000 FLDK,      0                     /FIELD COUNTER
7376 0000 DPTK,      0                     /DECIMAL POINT COUNTER
7377 0000 PREV,      0                     /PREVIOUS NON-SPACE INDICATOR

          /ND6602-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE      30
          *          7400
          /FLOATING POINT INPUT
7400 0000 FLINTP,    0
7401 4646            JMS   I    DPCVPT
7402 1057            TAD        CHAR
7403 1244            TAD        PER
7404 7640            SZA   CLA
7405 5211            JMP        FIG01
7406 3651            DCA   I    DPN        /SET NO OF DIGITS TO 0
7407 2252            ISZ        PRSW       /SET PERIOD SWITCH
7410 5647            JMP   I    DPCSPT     /CONVERT REST OF STRING
7411 1252 FIG01,     TAD        PRSW       /PERIOD READ IN PREVIOUSLY
7412 7640            SZA   CLA
7413 1651 FIG02,     TAD   I    DPN        /YES, GET - NO OF DIGITS
7414 7041            CMA   IAC              /NO
```

```
7415 3057            DCA       CHAR
7416 1060            TAD       SIGN        /TEST SIGN
7417 7640            SZA  CLA
7420 4650            JMS  I    MSGNPT
7421 1243 FIG03,     TAD       C27
7422 3044            DCA       EXPONT
7423 1055            TAD       HI
7424 3045            DCA       HORDER
7425 1056            TAD       LO
7426 3046            DCA       LORDER
7427 4407            JMS  I    7           /NORMALIZE F.P. NUMBER
7430 0010            FNOR
7431 0000            FEXT
7432 1057            TAD       CHAR
7433 7650            SNA  CLA
7434 5600            JMP  I    FLINTP
7435 4407 SCALE,     JMS  I    7           /DECIMAL POINT IS TO THE LEFT
7436 4645            FDIV I    TENPT       / DIVIDE BY TEN
7437 0000            FEXT
7440 2057            ISZ       CHAR
7441 5235            JMP       SCALE
7442 5600            JMP  I    FLINTP
7443 0027 C27,       0027
7444 7522 PER,       -256
7445 6774 TENPT,     TEN
7446 7007 DPCVPT,    DECONV
7447 7020 DPCSPT,    STRT
7450 7147 MSGNPT,    MSIGN
7451 7103 DPN,       DNUMBR
7452 0000 PRSW,      0
          /INTEGER OUTPUT
7453 0000 IOUT,      0
7454 3056            DCA       LO          /STORE ARGUMENT
7455 3055            DCA       HI
7456 1056            TAD       LO          /SET SIGN CHARACTER
7457 4676            JMS  I    LSTS
7460 1056            TAD       LO          /MAKE POSITIVE
7461 7510            SPA
7462 7041            CMA  IAC
7463 3056            DCA       LO
7464 4677            JMS  I    LDPOP       /OUTPUT
7465 0004            4
7466 0000            0
7467 5653            JMP  I    IOUT        /RETURN
          /FIXED-FORMAT FLOATING OUTPUT
7470 0000 FOUT,      0
7471 4675            JMS  I    LVFOUT
7472 0004            4                     /NUMBER OF DIGITS BEFORE
7473 0003            3                     /AND AFTER POINT
7474 5670            JMP  I    FOUT        /RETURN
7475 7200 LVFOUT,    VFOUT
7476 7162 LSTS,      STSG
7477 7262 LDPOP,     DROP
          /FLOATING POINT INPUT AND OUTPUT OF A STRING OF NUMBERS
          /CALLING SEQUENCE
          /          JMS       FINP
          /          NUMBER OF NUMBERS IN STRING
          /POINTER TO ADDRESS FOR FIRST NUMBER
7500 0000 FINP,      0
7501 4422            JMS  I    LADR
```

```
7502 7776            -2
7503 0000 FK,        0                          /COUNTER
7504 0000 FADR,      0                          /ADDRESS FOR STORING INPUT
7505 1704            TAD  I     FADR
7506 3304            DCA        FADR
7507 4404 FINS,      JMS  I     4               /GET NO INTO F.A
7510 4407            JMS  I     7               /STORE IN DESIRED LOCATION
7511 6704            FPUT I     FADR
7512 7304            FINC       FADR
7513 0000            FEXT
7514 2303            ISZ        FK              /TEST COUNTER
7515 5307            JMP        FINS
7516 5700            JMP  I     FINP            /RETURN
          /CALLING SEQUENCE
          /          JMS        FOPT
          /          NUMBER OF NUMBERS
          /POINTER TO ADDRESS OF STARTING LOCATION
7517 0000 FOPT,      0
7520 4422            JMS  I     LADR
7521 7776            -2
7522 0000 FOPK,      0                          /COUNTER
7523 0000 FLOC,      0                          /ADDRESS OF NUMBER
7524 1723            TAD  I     FLOC
7525 3323            DCA        FLOC
7526 4407 FOPS,      JMS  I     7               /PUT NO IN F.A
7527 5723            FGET I     FLOC
7530 7323            FINC       FLOC
7531 0000            FEXT
7532 4406            JMS  I     6               /PRINT NUMBER
7533 2322            ISZ        FOPK            /TEST COUNTER
7534 5326            JMP        FOPS
7535 5717            JMP  I     FOPT            /RETURN
          /SUBROUTINE TO READ IN A STRING OF INTEGERS
          /CALLING SEQUENCE
          /          JMS        FIXIN
          /          -NO OF INTEGERS
          /POINTER TO ADDRESS TO STORE FIRST
7536 0000 FIXIN,     0
7537 4422            JMS  I     LADR            /MOVE ARGUMENTS
7540 7776            -2
7541 0000 FIA,       0
7542 0000 FIB,       0
7543 1742            TAD  I     FIB
7544 3342            DCA        FIB
7545 4430 FIC,       JMS  I     LINTIN
7546 3742            DCA  I     FIB
7547 2342            ISZ        FIB
7550 2341            ISZ        FIA             /TEST COUNTER
7551 5345            JMP        FIC
7552 5736            JMP  I     FIXIN           /RETURN
7553 0000 DPOUT,     0
7554 4675            JMS  I     LVFOUT
7555 0007            7
7556 0000            0
7557 5753            JMP  I     DPOUT
          /COPY FROM ONE AREA OF MEMORY TO ANOTHER UNDER INTERRUPT
          /CALLING SEQUENCE
          /          JMS        TRANS
          /          NO OF ARGUMENTS
          /FROM      ADDRESS OF TABLE-1
```

```
          /TO       ADDRESS OF TABLE-1
7560 0000 TRANS,    0                        /ENTRY
7561 1760           TAD  I    TRANS
7562 3377           DCA       COUNTA
7563 2360           ISZ       TRANS
7564 1760           TAD  I    TRANS
7565 3010           DCA       AUTC
7566 2360           ISZ       TRANS
7567 1760           TAD  I    TRANS
7570 3011           DCA       AUT1
7571 1410           TAD  I    AUTC
7572 3411           DCA  I    AUT1
7573 2377           ISZ       COUNTA
7574 5371           JMP       .-3
7575 2360           ISZ       TRANS
7576 5760           JMP  I    TRANS
7577 0000 COUNTA,   0

          /ND6G2-AUTOMATIC DIFFRACTOMETER-PDP8
          /PAGE     31
          *         7600
          /BINARY LOADER, RIM LOADER
2000      $         START
```

APPENDIX 2

THE REFINEMENT PROGRAM CCD/2

(a) The Input

Card 1 – KW, WAV (Format I1, F9.4)

WAV = wavelength in A, KW = 0 if it not to be refined (that is, fixed) and = 1 if it is to be refined (that is, free).

Card 2 – KTTHZ, TTHZ, KOMZ, OMZ, KCIZ, CIZ (Format 3 (I1, F9.4)).

TTHZ, OMZ, CIZ are zeros of 2θ, ω, χ

KTTHZ, KOMZ, KCIZ are 0 or 1 depending on whether these parameters are fixed or free.

Card 3 – KA1, A1, KA2, A2, KA3, A3, KANG1, ANG1, KANG2, ANG2, KANG3, ANG3 (Format 6 (I1, F9.4)).

A1,2,3 are the sides, and ANG1,2,3 the angles, of the unit cell. The K's are 0 for fixed parameters, 1 for free parameters, and 2 if the parameter is free but constrained to equal the previous parameter.

Card 4 – NOL (Format I2)

= the number of reflections, of which the first is taken as the primary orienting reflection and the second is the secondary orienting reflection.

Card 5 – NTYP, H, K, L, 2θ, ω, χ, ϕ (Format I1, 3I3, 4F10.4)

NTYP = 1 if χ was used to centre vertically and = 2 if ϕ was used.

The angles are in degrees and decimals.

This first reflection is the First Orienting Reflection.

Card 6 – KORA(I) (I = 1 to 3) Format 3I1).

These three numbers are 1 or 0 to specify which two of the three angles ω, χ, ϕ of the First Orienting Reflection are to be the free parameters which will define **U**.

Card 7 – Similar to Card 5 and specifying the Second Orienting Reflection.

Card 8 – Similar to Card 6 and specifying which one of the three angles ω, χ, ϕ of the Second Orienting Reflection is to be a free parameter defining **U**.

Cards 9, etc. Similar to Card 5 and describing the remaining reflections.

In preparing the input data the most difficult task is to name the reflections consistently and correctly. A stereographic net is most useful here. One must then choose three angles from two reflections to act as the vehicle for the Orientation Matrix. The following table should be used in making this choice.

(continued)

CHOICE OF PARAMETERS TO DEFINE ORIENTATION

NOTES: (i) 0 means $0° \pm 45°$

(ii) 90 means $90° \pm 45°$

(iii) N.S. means "Not Suitable", the reflections are near parallel and others should be chosen.

(iv) $\nu = \omega - \theta$, that is, in the "Bisecting Position" $\nu = 0$.

(v) When $\nu = 90$ and $\chi = 0$ there is no axis which can centre the reflection vertically so this arrangement should not be used.

If the Observations are:					Choose the following as Free Parameters:	
First Reflection		Second Reflection			First Reflection	Second Reflection
ν_1	χ_1	ν_2	χ_2	$\phi_2 - \phi_1$		
0	0	0	0	0		N.S.
		0	0	90	ω_1 and χ_1, or	χ_2
		0	90	0	χ_1 and ϕ_1	ω_2
		0	90	90		χ_2
		90	90	0		ω_2
		90	90	90		N.S.
0	90	0	0	any	ω_1 and χ_1,	ω_2 or ϕ_2
		0	90	any		N.S.
		90	90	any		ϕ_2
90	90	0	0	0	ω_1 and ϕ_1	χ_2
		0	0	90		N.S.
		0	90	0		χ_2
		0	90	90		ω_2
		90	90	0		N.S.
		90	90	90		ω_2

(b) The Output from CCD/2

The output is self-explanatory. The input data is printed and the free parameters sorted out and listed. At each cycle of refinement the observed and calculated values of all angles are listed. These should be checked, particularly for the first cycle. If there is a large error in one or two angles, due, say, to mis-reading or, quite likely, to a mis-naming of the indices, this will ruin everything. After four cycles of refinement the standard deviations of the final values of the parameters, and the correlation matrix are printed. Finally the UB matrix is printed out, ready to be fed back into the computer-controlled diffractometer, in rows.

(continued)

(c) Program Listing

```
//    EXEC  FORTSCLG,PARM.FORT='DECK,OPT=2'
//FORT.SYSIN    DD  *
C      PROGRAMME CODRUB FOR THE REFINEMENT OF DATA ON < 11 CENTRED REFLNS
C      TO PRODUCE UB-MATRIX
       DIMENSION DIAL(4,10),KANG(3),ANG(3),IH(3),H(3,10),HA(3),HB(3),KORA
      1(3),KORB(3),NTYP(10),A(3),HZ(3),UF(3),B(3,3),UB(3,3),YO(3,10),YC(3
      1,10),YCZ(3,10),VVEC(12),AMAT(12,12),VAR(12),RHO(12,12),KA(3),P(12)
      1,HFI(3),PHIM(3,3),UCI(3),CHIM(3,3),UOM(3),FDP(12),DELP(12),SIG(12)
      1,DYC(12,3,10),HAB(3),HBB(3),SUMS(4),X(12,13),RDIFF(3,10),DIFF(3,10
      1),AVDEV(4)
       LOGICAL SET,AGAIN,FIN
    1  READ(1,901)KW,WAV
  901  FORMAT(I1,F9.4)
       WRITE(3,801)KW,WAV
  801  FORMAT('0WAVELENGTH',I3,F12.4,'ANGSTROMS ')
       READ(1,902)KTTHZ,TTHZ,KOMZ,OMZ,KCIZ,CIZ
  902  FORMAT(3(I1,F9.4))
       WRITE(3,802)KTTHZ,TTHZ,KOMZ,OMZ,KCIZ,CIZ
  802  FORMAT('0ZERO ANGLE SETTINGS KTTHZ=',I2,' TTHZ=',F9.4,' KOMZ=',I2,
     1' OMZ=',F9.4,' KCIZ=',I2,' CIZ=',F9.4,2X,'ALL ANGLES IN DEGREES')
       READ(1,903)(KA(I),A(I),I=1,3),(KANG(I),ANG(I),I=1,3)
  903  FORMAT(6(I3,F9.4))
       WRITE(3,803)(KA(I),A(I),I=1,3),(KANG(I),ANG(I),I=1,3)
  803  FORMAT('0CELL DIMS',2X,3(I3,F8.4),5X,'ANGLES',2X,3(I3,F9.4))
       READ(1,904)NOL
  904  FORMAT(I2)
       WRITE(3,804)NOL
  804  FORMAT('0',I3,2X,'LINES OF WHICH FIRST TWO ARE ORIENTING LINES AND
     1ALL ANGLES IN DEGREES'//)
       WRITE(3,6031)
 6031  FORMAT('    H  K  L                TTH          OM          CHI
     1   PHI'//)
       EPS=0.005
       DO 2 M=1,NOL
       READ(1,905)NTYP(M),(IH(I),I=1,3),(DIAL(N,M),N=1,4)
  905  FORMAT(I1,3I3,4F10.4)
       IF(M.EQ.1)READ(1,906)(KORA(I),I=1,3)
       IF(M.EQ.2)READ(1,906)(KORB(I),I=1,3)
  906  FORMAT(3I1)
       WRITE(3,805)NTYP(M),(IH(I),I=1,3),(DIAL(N,M),N=1,4)
  805  FORMAT(1X,I1,3I3,8X,4F12.4)
       DO 3 I=1,3
    3  H(I,M)=IH(I)
       DO 333 I=1,4
  333  DIAL(I,M)=DIAL(I,M)/57.2957795
       YO(1,M)=-DIAL(1,M)
       YO(2,M)=-DIAL(2,M)
       IF(NTYP(M).EQ.1)YO(3,M)=DIAL(3,M)
       IF(NTYP(M).EQ.2)YO(3,M)=DIAL(4,M)
    2  CONTINUE
       DO 4 I=1,3
       HA(I)=H(I,1)
    4  HB(I)=H(I,2)
       SET=.FALSE.
       FIN=.FALSE.
       DO 2000 I=1,4
       AVDEV(I)=.0
 2000  SUMS(I)=.0
```

```
    ICYC=Ø
  5 ICYC=ICYC+1
    IC=Ø
  6 NP=Ø
    IF(KW.EQ.Ø)GO TO 11
    NP=NP+1
    IF(SET)GO TO 1Ø
    P(NP)=WAV
    WRITE(3,8Ø6)NP,WAV
8Ø6 FORMAT('ØPARAMETER ',I2,'=WAVELENGTH  ',F8.4,2X,'ANGSTROMS')
 1Ø WAV=P(NP)
 11 IF(KTTHZ.EQ.Ø)GO TO 13
    NP=NP+1
    IF(SET)GO TO 12
    P(NP)=TTHZ
    WRITE(3,8Ø7)NP,TTHZ
8Ø7 FORMAT(' PARAMETER ',I2,'=2THETA ZERO ',F8.4,2X,'DEGREES')
 12 TTHZ=P(NP)
 13 IF(KOMZ.EQ.Ø)GO TO 15
    NP=NP+1
    IF(SET)GO TO 14
    P(NP)=OMZ
    WRITE(3,8Ø8)NP,OMZ
8Ø8 FORMAT(' PARAMETER ',I2,'=OMEGA ZERO ',F8.4,2X,'DEGREES')
 14 OMZ=P(NP)
 15 IF(KCIZ.EQ.Ø)GO TO 17
    NP=NP+1
    IF(SET)GO TO 16
    P(NP)=CIZ
    WRITE(3,8Ø9)NP,CIZ
8Ø9 FORMAT(' PARAMETER ',I2,'=CHI  ZERO  ',F8.4,'DEGREES')
 16 CIZ=P(NP)
 17 DO 19 I=1,3
    IF(KA(I).EQ.Ø)GO TO 19
    IF(KA(I).EQ.2)GO TO 191
    NP=NP+1
    IF(SET)GO TO 18
    P(NP)=A(I)
    WRITE(3,81Ø)NP,I,A(I)
81Ø FORMAT(' PARAMETER ',I2,'=CELL EDGE',I1,2X,F8.4,2X,'ANGSTROMS ')
 18 A(I)=P(NP)
    GO TO 19
191 A(I)=A(1)
 19 CONTINUE
    DO 21 I=1,3
    IF(KANG(I).EQ.Ø)GO TO 21
    IF(KANG(I).EQ.2)GO TO 211
    NP=NP+1
    IF(SET)GO TO 2Ø
    P(NP)=ANG(I)
    WRITE(3,811)NP,I,ANG(I)
811 FORMAT(' PARAMETER ',I2,'=CELL ANG',I1,2X,F8.4,2X,'DEGREES ')
 2Ø ANG(I)=P(NP)
    GO TO 21
211 ANG(I)=ANG(1)
 21 CONTINUE
    DO 23 I=1,3
    IF(KORA(I).EQ.Ø)GO TO 23
```

```
      NP=NP+1
      I1=I+1
      IF(SET)GO TO 22
      P(NP)=DIAL(I1,1)
      WRITE(3,812)NP,I1,DIAL(I1,1)
  812 FORMAT(' PARAMETER ',I2,'=PRI.ORIENT.REFL.DIAL',I1,2X,F8.4,'RAD ')
   22 DIAL(I1,1)=P(NP)
   23 CONTINUE
      DO 25 I=1,3
      IF(KORB(I).EQ.Ø)GO TO 25
      NP=NP+1
      I1=I+1
      IF(SET)GO TO 24
      P(NP)=DIAL(I1,2)
      WRITE(3,813)NP,I1,DIAL(I1,2)
  813 FORMAT(' PARAMETER ',I2,'=SEC.ORIENT.REFL.DIAL',I1,2X,F8.4,'RAD ')
   24 DIAL(I1,2)=P(NP)
   25 CONTINUE
      SET=.TRUE.
      IF(IC.GE.1)GO TO 5Ø
      IF(FIN)GO TO 5Ø
      WRITE(3,1ØØ1)ICYC
 1ØØ1 FORMAT('ØCYCLE ',I2)
C     WE NOW ENTER THE CALCULATE SECTION TO COMPUTE YC'S FROM THE P'S
   5Ø CALL GENB(A,ANG,B)
      CALL GMPRD(B,HA,HAB,3,3,1)
      CALL GMPRD(B,HB,HBB,3,3,1)
      AHAB=SQRT(HAB(1)*HAB(1)+HAB(2)*HAB(2)+HAB(3)*HAB(3))
      AHBB=SQRT(HBB(1)*HBB(1)+HBB(2)*HBB(2)+HBB(3)*HBB(3))
      THA=ARSIN(AHAB*WAV/2.)
      THB=ARSIN(AHBB*WAV/2.)
      ROMZ=OMZ/57.2957795
      RCIZ=CIZ/57.2957795
      RTTHZ=TTHZ/57.2957795
      OMB=-DIAL(2,2)+THB-ROMZ
      OMA=-DIAL(2,1)+THA-ROMZ
      ZIAL1=DIAL(3,1)-RCIZ
      ZIAL2=DIAL(3,2)-RCIZ
      IF(IC.GE.1)GO TO 53
      WRITE(3,1ØØ2)OMA,OMB
 1ØØ2 FORMAT(1ØX,' OMA=',1PE12.5,2X,'OMB=',1PE12.5,2X,'RADIANS ')
   53 CALL GENUB(HA,OMA,ZIAL1,DIAL(4,1),HB,OMB,ZIAL2,DIAL(4,2),B,UB)
      IF(IC.EQ.Ø)WRITE(3,52)UB
   52 FORMAT('ØUBMAT ',1P9E13.4,//)
      IF(FIN)GO TO 51
      IF(IC.EQ.Ø)WRITE(3,6Ø28)
 6Ø28 FORMAT(9X,3(3X,'YOBS ',7X,'YCALCS ',5X)//)
      DO 26 M=1,NOL
      DO 27 I=1,3
   27 HZ(I)=H(I,M)
      CALL GMPRD(UB,HZ,HFI,3,3,1)
      AMHFI=HFI(1)*HFI(1)+HFI(2)*HFI(2)+HFI(3)*HFI(3)
      AMHFI=SQRT(AMHFI)
      STHC=AMHFI*WAV/2
      THC=ARSIN(STHC)
      TTHC1=2.*THC+RTTHZ
      TTHC2=-2.*THC+RTTHZ
      CALL SLECT(TTHC1,TTHC2,YO(1,M),YC(1,M))
```

```
      DO 28 I=1,3
   28 UF(I)=HFI(I)/AMHFI
      IF(NTYP(M).EQ.1)GO TO 29
      ACHI=DIAL(3,M)
      TQ1=SIN(ACHI)*UF(1)
      TQ2=SIN(ACHI)*UF(2)
      TQ3=COS(ACHI)*UF(3)
      CALL TREQ(TQ1,TQ2,TQ3,FIC1,FIC2)
      CALL SLECT(FIC1,FIC2,YO(3,M),YC(3,M))
   29 CALL ZROT(PHIM,DIAL(4,M))
      CALL GMPRD(PHIM,UF,UCI,3,3,1)
      IF(NTYP(M).EQ.2)GO TO 30
      CALL RTAN(UCI(3),UCI(1),CIRT)
      CIC1=CIRT+RCIZ
      CIC2=CIRT+3.141592+RCIZ
      CALL SLECT(CIC1,CIC2,YO(3,M),YC(3,M))
   30 ZAL=DIAL(3,M)-RCIZ
      CALL YROT(CHIM,ZAL)
      CALL GMPRD(CHIM,UCI,UOM,3,3,1)
      OSTHC=-STHC
      CALL TREQ(-UOM(2),UOM(1),OSTHC,GNU1,GNU2)
      OMC1=GNU1+ROMZ
      OMC2=GNU2+ROMZ
      CALL SLECT(OMC1,OMC2,YO(2,M),YC(2,M))
      IF(IC.GE.1)GO TO 26
      WRITE(3,892)(YO(N,M),YC(N,M),N=1,3)
  892 FORMAT(5X,3(3X,1PE11.4,2X,1PE11.4))
      DO 6026 N=1,3
      RDIFF(N,M)=YO(N,M)-YC(N,M)
 6026 DIFF(N,M)=RDIFF(N,M)*57.2957795
      WRITE(3,6027)(DIFF(N,M),N=1,3)
 6027 FORMAT(5X,3(6X,'DIFF= ',F6.3,1X,'DEGREES ')//)
   26 CONTINUE
      IF(IC.NE.0)GO TO 31
      DO 32 I=1,3
      DO 32 M=1,NOL
   32 YCZ(I,M)=YC(I,M)
      GO TO 33
   31 DO 34 I=1,3
      DO 34 M=1,NOL
   34 DYC(IC,I,M)=(YC(I,M)-YCZ(I,M))/EPS
      IF(IC.GE.NP)GO TO 35
   33 IC=IC+1
      P(IC)=P(IC)+EPS
      ICM=IC-1
      IF(IC.GT.1)P(ICM)=P(ICM)-EPS
      GO TO 6
   35 P(IC)=P(IC)-EPS
      DO 36 I=1,NP
      VVEC(I)=.0
      DO 36 J=1,NP
   36 AMAT(I,J)=.0
      DO 37 N=1,3
      DO 37 M=1,NOL
      SUMS(ICYC)=SUMS(ICYC)+(YO(N,M)-YCZ(N,M))**2
      DO 37 I=1,NP
      VVEC(I)=VVEC(I)+DYC(I,N,M)*(YO(N,M)-YCZ(N,M))
      DO 37 J=1,NP
```

```
   37 AMAT(I,J)=AMAT(I,J)+DYC(I,N,M)*DYC(J,N,M)
      AVDEV(ICYC)=SQRT(SUMS(ICYC)/(3*NOL))*57.296
      WRITE(3,1000)SUMS
 1000 FORMAT(10X,'  SUMS ',1P4E14.5,//)
      WRITE(3,7000)AVDEV
 7000 FORMAT(10X,'  AVDEV ',1P4E12.3,//)
      NP1=NP+1
      DO 3005  I=1,NP
      X(I,NP1)=VVEC(I)
      DO 3005 J=1,NP
 3005 X(I,J)=AMAT(I,J)
      CALL  SID(X,-1,12,NP,NP1,DET)
      WRITE(3,100)DET
  100 FORMAT(12X,'DET=',1PE15.6,//)
      PDELP=.0
      DO 3007 I=1,NP
      DELP(I)=X(I,NP1)
 3007 PDELP=PDELP+DELP(I)*VVEC(I)
      AGAIN=.FALSE.
      DO 38 I=1,NP
      P(I)=P(I)+DELP(I)
      FDP(I)=ABS(DELP(I))
      IF(FDP(I).GT.1.0E-4)AGAIN=.TRUE.
   38 CONTINUE
      IF(ICYC.GT.3)AGAIN=.FALSE.
      WRITE(3,815)(DELP(I),I=1,NP)
  815 FORMAT(' DELPS ',1P12E12.3)
      WRITE(3,814)(P(I),I=1,NP)
  814 FORMAT('0NEWPARMS ',1P12E12.3,//)
      IF(AGAIN)GO TO 5
      DO 8000 I=1,NP
      DO 8000 J=1,NP
 8000 AMAT(I,J)=X(I,J)
      FT=3*NOL-NP
      WRITE(3,6030)
 6030 FORMAT(8X,'SIGMAS ',12X,'CORRELATIONS ')
      DO 39 I=1,NP
      VAR(I)=AMAT(I,I)*(SUMS(ICYC)-PDELP)/FT
      SIG(I)=SQRT(VAR(I))
      DO 40 J=1,NP
   40 RHO(I,J)=AMAT(I,J)/(SQRT(AMAT(I,I)*AMAT(J,J)))
   39 WRITE(3,816)I,SIG(I),(RHO(I,J),J=1,NP)
  816 FORMAT('0',I3,1PE13.3,0P12F6.2)
      FIN=.TRUE.
      GO TO 6
   51 WRITE(3,9003)
 9003 FORMAT('0UBMAT FINAL ')
      DO 9001 I=1,3
 9001 WRITE(3,9002)(UB(I,J),J=1,3)
 9002 FORMAT(3F12.5,//)
      WAV=WAV+0.00005
      WRITE(2,10850)((UB(I,J),J=1,3),I=1,3),WAV
      WAV=WAV-0.00005
10850 FORMAT('UB',2X,2(3(F8.5,3X)/4X),3(F8.5,3X),'WAV=',F6.4)
      CALL BANG(UB,H,NOL,WAV,TTHZ,OMZ,CIZ)
      CALL CELCAL(UB)
      GO TO 1
      END
```

```
      SUBROUTINE GENB(A,D,B)
C     CALCULATION OF CARTESIAN XFORMATION MATRIX B(REFER B.AND L.EQN.3)
      DIMENSION A(3),D(3),B(3,3),ANG(3),G(3),CANG(3),CANG2(3),SANG(3)
      DO 3 I=1,3
      IF(D(I).EQ.90.0)GO TO 5
      DNG=D(I)/57.2957795
      CANG(I)=COS(DNG)
      CANG2(I)=CANG(I)*CANG(I)
      SANG(I)=SIN(DNG)
      GO TO 3
    5 CANG(I)=0.0
      CANG2(I)=0.0
      SANG(I)=1.0
    3 CONTINUE
      ALPH=SQRT(1-CANG2(1)-CANG2(2)-CANG2(3)+2*(CANG(1)*CANG(2)*CANG(3))
     1)
      DO 6 K=1,3
      G(K)=SANG(K)/(A(K)*ALPH)
    6 CONTINUE
      CSTA2=CANG(3)*CANG(1)-CANG(2)/(SANG(3)*SANG(1))
      CSTA3=CANG(1)*CANG(2)-CANG(3)/(SANG(1)*SANG(2))
      SSTA2=SQRT(1-CSTA2**2)
      SSTA3=SQRT(1-CSTA3**2)
      B(1,1)=G(1)
      B(1,2)=G(2)*CSTA3
      B(1,3)=G(3)*CSTA2
      B(2,2)=G(2)*SSTA3
      B(2,3)=-(G(3)*SSTA2*CANG(1))
      B(3,3)=1./A(3)
      B(2,1)=0.
      B(3,1)=0.
      B(3,2)=0.
      RETURN
      END
      SUBROUTINE GENUB(HA,OM1,CI1,FI1,HB,OM2,CI2,FI2,B,UB)
C     GIVEN THE MATRIX B AND THE INDICES AND ANGLES OF THE TWO ORIENTING
C     REFLECTIONS COMPUTE THE MATRIX UB
      DIMENSION HA(3),HB(3),B(3,3),UB(3,3),H1CRYS(3),H2CRYS(3),U1FI(3),U
     12FI(3),TR1PC(3,3),TR1PF(3,3),TTR1PC(3,3),COANG(3),SIANG(3),SIAND(3
     1),COAND(3),U(3,3)
      CALL GMPRD(B,HA,H1CRYS,3,3,1)
      CALL GMPRD(B,HB,H2CRYS,3,3,1)
      COANG(1)=COS(OM1)
      SIANG(1)=SIN(OM1)
      COANG(2)=COS(CI1)
      SIANG(2)=SIN(CI1)
      COANG(3)=COS(FI1)
      SIANG(3)=SIN(FI1)
      COAND(1)=COS(OM2)
      SIAND(1)=SIN(OM2)
      COAND(2)=COS(CI2)
      SIAND(2)=SIN(CI2)
      COAND(3)=COS(FI2)
      SIAND(3)=SIN(FI2)
      CALL UFI(COANG,SIANG,U1FI)
      CALL UFI(COAND,SIAND,U2FI)
      CALL TR1P(H1CRYS,H2CRYS,TR1PC)
      CALL GMTRA(TR1PC,TTR1PC,3,3)
```

```
      CALL TR1P(U1FI,U2FI,TR1PF)
      CALL GMPRD(TR1PF,TTR1PC,U,3,3,3)
      CALL GMPRD(U,B,UB,3,3,3)
      RETURN
      END
      SUBROUTINE TR1P(X,Y,Z)
C     SETS UP MATRIX Z WHOSE COLUMNS ARE A R.H. SET OF UNIT ORTHOG VECS.
      DIMENSION X(3),Y(3),Z(3,3),T(3),U(3),V(3)
      XS=SQRT(X(1)**2+X(2)**2+X(3)**2)
      DO 7 I=1,3
      T(I)=X(I)/XS
    7 CONTINUE
      U(1)=(X(2)*Y(3))-(X(3)*Y(2))
      U(2)=(X(3)*Y(1))-(X(1)*Y(3))
      U(3)=(X(1)*Y(2))-(X(2)*Y(1))
      US=SQRT(U(1)**2+U(2)**2+U(3)**2)
      DO 8 I=1,3
    8 U(I)=U(I)/US
      V(1)=(U(2)*T(3))-(U(3)*T(2))
      V(2)=(U(3)*T(1))-(U(1)*T(3))
      V(3)=(U(1)*T(2))-(U(2)*T(1))
      Z(1,1)=T(1)
      Z(1,2)=V(1)
      Z(1,3)=U(1)
      Z(2,1)=T(2)
      Z(2,2)=V(2)
      Z(2,3)=U(2)
      Z(3,1)=T(3)
      Z(3,2)=V(3)
      Z(3,3)=U(3)
      RETURN
      END
      SUBROUTINE RTAN(Y,X,RT)
C     RETURNS RT IN THE QUADRANT WHERE THE SIGNS SIN(RT)AND COS(RT) ARE
C     THOSE OF Y AND X
      IF((X.EQ..Ø).AND.(Y.EQ..Ø))GO TO 1
      IF(X.EQ..Ø)GO TO 4
      IF(ABS(Y).LT.ABS(X))GO TO 2
      RT=1.57Ø81-ATAN(ABS(X/Y))
      GO TO 3
    2 RT=ATAN(ABS(Y/X))
    3 IF(X.LT..Ø)RT=3.1415926-RT
      GO TO 5
    4 RT=1.57Ø81
    5 IF(Y.LT..Ø)RT=-RT
      RETURN
    1 RT=.Ø
      RETURN
      END
      SUBROUTINE TREQ(A,B,C,CAL1,CAL2)
C     ANGLES CAL1 AND CAL2 ARE THE ROOTS OF THE EQN.A*COS(X)+B*SIN(X)=G
      CALL RTAN(B,A,ETA)
      BSS=ABS(A*A+B*B-C*C)
      IF(BSS.LT.1.ØE-5)BSS=Ø.
      BS=SQRT(BSS)
      CALL RTAN(BS,C,GAM)
      CAL1=ETA+GAM
      CAL2=ETA-GAM
```

```
      RETURN
      END
      SUBROUTINE SLECT(C1,C2,Y,SEL)
C     ANGLE SEL=WHICHEVER OF C1AND C2 IS CLOSEST TO THE OBSERVED ANG. Y
      DIMENSION C(2)
      C(1)=C1
      C(2)=C2
      DMIN=3.1415926
      NC=-3
      DO 1 M=1,5
      DO 1 I=1,2
      FC=NC+M
      CIR=FC*6.28325
      AB=ABS(C(I)+CIR-Y)
      IF(AB.GT.DMIN)GO TO 1
      DMIN=AB
      SEL=C(I)+CIR
    1 CONTINUE
      RETURN
      END
      SUBROUTINE YROT(R,B)
C     R IS THE MATRIX WHICH ROTATES BY B ABOUT Y-AXIS
      DIMENSION R(3,3)
      R(1,1)=COS(B)
      R(1,2)=.Ø
      R(1,3)=SIN(B)
      R(2,1)=.Ø
      R(2,2)=1.
      R(2,3)=.Ø
      R(3,1)=-SIN(B)
      R(3,2)=.Ø
      R(3,3)=COS(B)
      RETURN
      END
      SUBROUTINE ZROT(R,B)
C     R IS THE MATRIX WHICH ROTATES BY B ABOUT Z-AXIS
      DIMENSION R(3,3)
      R(1,1)=COS(B)
      R(1,2)=SIN(B)
      R(1,3)=.Ø
      R(2,1)=-SIN(B)
      R(2,2)=COS(B)
      R(2,3)=.Ø
      R(3,1)=.Ø
      R(3,2)=.Ø
      R(3,3)=1.
      RETURN
      END
      SUBROUTINE BANG(UB,H,NOL,WAV,TTHZ,OMZ,CIZ)
C     CALCULATES ANGLES FOR BISECTING POSITION
      DIMENSION UB(3,3),H(3,1Ø),HA(3),IH(3),HFI(3)
      WRITE(3,51Ø)
  51Ø FORMAT('Ø  ANGLES FOR BISECTING POSITION ON CENTRED REFLNS.'//)
      WRITE(3,5Ø9)
  5Ø9 FORMAT('  H    K    L       TTH          OMG         CHI       FI'//)
      DO 3 M=1,NOL
      DO 2 I=1,3
      HA(I)=H(I,M)
```

```
      IH(I)=H(I,M)
    2 CONTINUE
      CALL GMPRD(UB,HA,HFI,3,3,1)
      TTH=2*(57.2958*(ARSIN(WAV*SQRT(HFI(1)*HFI(1)+HFI(2)*HFI(2)+HFI(3)*
     1HFI(3))/2.)))+TTHZ
      OMG=TTH/2.+OMZ
      CALL RTAN(HFI(2),HFI(1),RFI)
      FI=RFI*57.2958
      DS=SQRT(HFI(1)*HFI(1)+HFI(2)*HFI(2))
      CALL RTAN(HFI(3),DS,RCHI)
      CHI=RCHI*57.2958+CIZ
      WRITE(3,506)IH(1),IH(2),IH(3),TTH,OMG,CHI,FI
  506 FORMAT(3I4,4F12.3)
    3 CONTINUE
      RETURN
      END
      SUBROUTINE CELCAL(UB)
C     CALCULATE AND PRINT CELL PARMS. FROM UB-MATRIX
      DIMENSION UB(3,3),UBT(3,3),G(3,3),SG(3,4),ACAL(3),ANGCAL(3)
      CALL GMTRA(UB,UBT,3,3)
      CALL GMPRD(UBT,UB,G,3,3,3)
      DO 1 I=1,3
      SG(I,4)=1.
      DO 1 J=1,3
    1 SG(I,J)=G(I,J)
      CALL SID(SG,-1,3,3,4,DET)
      DO 2 I=1,3
      DO 2 J=1,3
    2 G(I,J)=SG(I,J)
      DO 3 I=1,3
    3 ACAL(I)=SQRT(G(I,I))
      DO 4 I=1,3
      JN=MOD(I,3)+1
      KN=MOD(JN,3)+1
      ANGCAL(I)=ARCOS(G(JN,KN)/(ACAL(JN)*ACAL(KN)))*57.2958
    4 CONTINUE
      WRITE(3,9000)ACAL,ANGCAL
 9000 FORMAT('0CELL DIMS CALCULATED',3F8.4,5X,3F9.4,//)
      RETURN
      END
      SUBROUTINE UFI(A,B,C)
C     REFER EQUATION 22 B.AND L.
      DIMENSION A(3),B(3),C(3)
      C(1)=A(1)*A(2)*A(3)+B(1)*B(3)
      C(2)=A(1)*A(2)*B(3)-B(1)*A(3)
      C(3)=A(1)*B(2)
      RETURN
      END
```

APPENDIX 3

THE DATA-HANDLING PROGRAM CODOEP

This program accepts the paper tape output from a data-collection run and edits it. The input consists of the tape plus two cards containing the wavelength and the UB matrix. These two cards are punched by the program CCD/2 (since re-named CODRUB) when it refines the UB matrix for the run in question.

The program edits the tape, checks that the angles calculated and set by the PDP-8 were correct, checks and comments on various points connected with the placement of the scan, and finally punches cards containing the Miller indices, the net intensity, the background and sin 2θ. These cards are then processed in further programs to analyze for physical parameters.

```
C       CODOEP
        INTEGER BUFFER,SM,EM,CR,DATA,G,GSTP,COUNT,BGR,LITS,BEG,END,RH,
       1RING,REZERO,ONE,H
        DIMENSION BUFFER(256),RING(4096),DATA(200),LITS(4096)
        LOGICAL TYPE
        COMMON KOBS,NOL,TYPE,LINES,NOP,IHEAD1,IHEAD2,IHEAD3,NSTD,KSTD,LSTD
       1,NOS,ITAG
  850 FORMAT(4X,A4,1X,2A4,1X,A4/8X,3(1X,I3))
  900 FORMAT('1 COD ',A4,T36,2A4,T73,'PAGE',I3)
  901 FORMAT('1 COD ',A4,T36,2A4,T73,'PAGE',I3//'   H  K  L  INTEN   BGR
     1   2THETA ERROR  G GES GSTP NSTP, NH1  NH2  FWHMH   CORR'/)
  999 FORMAT('0NUMBER OF LINES',I5)
        DATA SM,EM,CR,BEG,END,RH,LH,REZERO,ONE/Z81,Z83,Z8D,0,0,0,0,2049
       1,Z100000/
        TYPE=.TRUE.
        NOL=0
        NOP=0
        NOS=0
        READ(1,850)IHEAD1,IHEAD2,IHEAD3,ITAG,NSTD,KSTD,LSTD
        WRITE(3,900)IHEAD1,IHEAD2,IHEAD3,NOP
        CALL CALCAS(0,0,0,0,0,0)
    1   CALL FILL(BUFFER,256,&2)
        DO 3 INDEX=1,256
        I=BUFFER(INDEX)
        IF(I)4,3,5
    4   I=-I-1
        J=I/16777216
        I=I-J*16777216
        I=-I+16777215
        J=-J+255
        GO TO 6
    5   J=I/16777216
        I=I-J*16777216
    6   IF(J.EQ.0.OR.J.EQ.128)GO TO 7
        END=END+ONE
        RING(END/ONE+REZERO)=J
    7   J=I/65536
        I=I-J*65536
        IF(J.EQ.0.OR.J.EQ.128)GO TO 8
        END=END+ONE
        RING(END/ONE+REZERO)=J
    8   J=I/256
        I=I-J*256
        IF(J.EQ.0.OR.J.EQ.128)GO TO 9
        END=END+ONE
        RING(END/ONE+REZERO)=J
    9   IF(I.EQ.0.OR.I.EQ.128)GO TO 3
        END=END+ONE
        RING(END/ONE+REZERO)=I
    3   CONTINUE
   19   IF(BEG.EQ.END.OR.RH.EQ.END)GO TO 1
        IF(TYPE)GO TO 10
C       DATA
   12   RH=RH+ONE
        IF(RING(RH/ONE+REZERO).EQ.EM)GO TO 11
        IF(RH.EQ.END)GO TO 1
        GO TO 12
   11   LH=RH
```

```
      LINES=LINES+1
      IF(NOL+NOS.NE.Ø.AND.(LINES.LT.5Ø.OR.NOL/5*5-NOL.NE.Ø))GO TO 13
      WRITE(3,999)NOL
      NOP=NOP+1
      LINES=5
      WRITE(3,9Ø1)IHEAD1,IHEAD2,IHEAD3,NOP
   13 LH=LH-ONE
      I=RH/ONE-LH/ONE-1
      IF(I.LT.-1)I=I+4Ø96
      IF(I.LT.2Ø1.AND.RING(LH/ONE+REZERO).NE.SM)GO TO 13
      IF(I.GT.2ØØ.OR.I.LT.26)GO TO 14
      DO 15 J=1,I
      LH=LH+ONE
   15 DATA(J)=RING(LH/ONE+REZERO)
      IL=1
      IR=I
      CALL SIGN(DATA,IL,IR,    H,  R,&14)
      CALL SIGN(DATA,IL,IR,    K,  R,&14)
      CALL SIGN(DATA,IL,IR,    L,  R,&14)
      CALL SIGN(DATA,IL,IR,    I,TTH,&14)
      CALL SIGN(DATA,IL,IR,    I, OM,&14)
      CALL SIGN(DATA,IL,IR,    I,CHI,&14)
      CALL SIGN(DATA,IL,IR,    I,PHI,&14)
      CALL SIGN(DATA,IL,IR,    G,  R,&14)
      CALL SIGN(DATA,IL,IR, GSTP,  R,&14)
      CALL SIGN(DATA,IL,IR,  NH1,  R,&14)
      CALL SIGN(DATA,IL,IR,  NH2,  R,&14)
      CALL SIGN(DATA,IL,IR,COUNT,  R,&14)
      CALL SIGN(DATA,IL,IR,  BGR,  R,&14)
   16 IF(H.EQ.NSTD.AND.K.EQ.KSTD.AND.L.EQ.LSTD)GO TO 17
      CALL LIST(H,K,L,TTH,OM,CHI,PHI,G,GSTP,NH1,NH2,COUNT,BGR)
      GO TO 18
   14 CONTINUE
      CALL LIST(Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,1)
      GO TO 18
   17 CALL STAND(H,K,L,TTH,OM,CHI,PHI,G,GSTP,NH1,NH2,COUNT,BGR)
   18 TYPE=.TRUE.
      LH=RH
      GO TO 19
C     LITS
   1Ø RH=RH+ONE
      IF(RING(RH/ONE+REZERO).EQ.SM)GO TO 2Ø
      IF(RH.EQ.END)GO TO 1
      GO TO 1Ø
   2Ø I=Ø
   22 LH=LH+ONE
      IF(LH.EQ.RH)GO TO 21
      I=I+1
      LITS(I)=RING(LH/ONE+REZERO)
      GO TO 22
   21 L=1
      DO 23 J=1,I
      IF(LITS(J).NE.CR)GO TO 23
      CALL PRINT(LITS,L,J-1)
      L=J+2
   23 CONTINUE
   24 TYPE=.FALSE.
      GO TO 19
```

```
  2 WRITE(3,999)NOL
    CALL STAND(Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø)
    STOP
    END
    SUBROUTINE SIGN(DATA,LH,RH,I,R,*)
    INTEGER DATA,RH,COLON,POINT,POWER
    DIMENSION DATA(2ØØ)
    DATA COLON,MINUS,LT,POINT/ZBA,ZAD,ZBC,ZAE/
    IF(LH.LE.Ø)RETURN 1
    IF(RH-LH.LT.1)RETURN 1
    DO 1 J=LH,RH
  1 IF(DATA(J).EQ.COLON)GO TO 2
    RETURN 1
  2 POWER=Ø
    TEMP=Ø
    M=J+1
    J=J-1
    IF(J.LT.LH)RETURN 1
    DO 3 KQ=LH,J
    L=LH+J-KQ
    K=DATA(L)
    IF(K.EQ.MINUS)GO TO 4
    IF(K.EQ.LT)GO TO 5
    IF(K.EQ.POINT)GO TO 7
    IF(K.LT.176.OR.K.GT.185)GO TO 6
    TEMP=TEMP+(K-176)*1Ø**POWER
    POWER=POWER+1
    GO TO 3
  7 TEMP=TEMP/1Ø**POWER
    POWER=Ø
  3 CONTINUE
    GO TO 6
  5 TEMP=Ø.Ø
  4 TEMP=TEMP*(-1.Ø)
  6 R=TEMP
    I=TEMP
    LH=M
    RETURN
    END
    SUBROUTINE PRINT(LITS,I,J)
    INTEGER PRIB,PRIX,BLANK,LITS
    DIMENSION PRIB(132),PRIX(64),LITS(4Ø96)
    COMMON KOBS,NOL,TYPE,LINES,NOP,IHEAD1,IHEAD2,IHEAD3,NSTD,KSTD,LSTD
9ØØ FORMAT(132A1)
    DATA PRIX(Ø1)/       Z4ØØØØØØØ/
    DATA PRIX(Ø2)/       ZC9ØØØØØØ/
    DATA PRIX(Ø3)/       Z7FØØØØØØ/
    DATA PRIX(Ø4)/       Z7BØØØØØØ/
    DATA PRIX(Ø5)/       Z5BØØØØØØ/
    DATA PRIX(Ø6)/       Z6CØØØØØØ/
    DATA PRIX(Ø7)/       Z5ØØØØØØØ/
    DATA PRIX(Ø9)/       Z4DØØØØØØ/
    DATA PRIX(Ø8)/       Z7DØØØØØØ/
    DATA PRIX(1Ø)/       Z5DØØØØØØ/
    DATA PRIX(11)/       Z5CØØØØØØ/
    DATA PRIX(12)/       Z4EØØØØØØ/
    DATA PRIX(13)/       Z6BØØØØØØ/
    DATA PRIX(14)/       Z6ØØØØØØØ/
```

```
      DATA PRIX(15)/       Z4B000000/
      DATA PRIX(16)/       Z61000000/
      DATA PRIX(17)/       ZF0000000/
      DATA PRIX(18)/       ZF1000000/
      DATA PRIX(19)/       ZF2000000/
      DATA PRIX(20)/       ZF3000000/
      DATA PRIX(21)/       ZF4000000/
      DATA PRIX(22)/       ZF5000000/
      DATA PRIX(23)/       ZF6000000/
      DATA PRIX(24)/       ZF7000000/
      DATA PRIX(25)/       ZF8000000/
      DATA PRIX(26)/       ZF9000000/
      DATA PRIX(27)/       Z7A000000/
      DATA PRIX(28)/       Z5E000000/
      DATA PRIX(29)/       Z4C000000/
      DATA PRIX(30)/       Z7E000000/
      DATA PRIX(31)/       Z6E000000/
      DATA PRIX(32)/       Z6F000000/
      DATA PRIX(33)/       Z7C000000/
      DATA PRIX(34)/       ZC1000000/
      DATA PRIX(35)/       ZC2000000/
      DATA PRIX(36)/       ZC3000000/
      DATA PRIX(37)/       ZC4000000/
      DATA PRIX(38)/       ZC5000000/
      DATA PRIX(39)/       ZC6000000/
      DATA PRIX(40)/       ZC7000000/
      DATA PRIX(41)/       ZC8000000/
      DATA PRIX(42)/       ZC9000000/
      DATA PRIX(43)/       ZD1000000/
      DATA PRIX(44)/       ZD2000000/
      DATA PRIX(45)/       ZD3000000/
      DATA PRIX(46)/       ZD4000000/
      DATA PRIX(47)/       ZD5000000/
      DATA PRIX(48)/       ZD6000000/
      DATA PRIX(49)/       ZD7000000/
      DATA PRIX(50)/       ZD8000000/
      DATA PRIX(51)/       ZD9000000/
      DATA PRIX(52)/       ZE2000000/
      DATA PRIX(53)/       ZE3000000/
      DATA PRIX(54)/       ZE4000000/
      DATA PRIX(55)/       ZE5000000/
      DATA PRIX(56)/       ZE6000000/
      DATA PRIX(57)/       ZE7000000/
      DATA PRIX(58)/       ZE8000000/
      DATA PRIX(59)/       ZE9000000/
      DATA PRIX(60)/       Z4D000000/
      DATA PRIX(61)/       Z6D000000/
      DATA PRIX(62)/       Z5D000000/
      DATA PRIX(63)/       Z4A000000/
      DATA PRIX(64)/       Z5F000000/
      DATA BLANK,PRIB/Z40000000,Z40000000/
      IF(J.LE.I.OR.I.LE.0.OR.J.GT.4096)RETURN
      DO 1 K=2,132
    1 PRIB(K)=BLANK
      DO 2 K=I,J
      N=K-I+2
      IF(N.GT.132)N=132
      L=K-I+1
```

```
      M=LITS(K)-159
      IF(M.EQ.95)RETURN
      IF(M.GT.64.OR.M.LT.1)M=1
    2 PRIB(N)=PRIX(M)
      WRITE(3,900)PRIB
      LINES=LINES+1
      RETURN
      END

      SUBROUTINE LIST(H,K,L,TTH,OM,CHI,PHI,G,GSTP,NH1,NH2,COUNT,BGR)
      INTEGER BGR,COUNT,FLAG,G,GEST,GSTP,H,REV
      COMMON KOBS,NOL,TYPE,LINES,NOP,IHEAD1,IHEAD2,IHEAD3,NSTD,KSTD,LSTD
     1,NQQ,ITAG
      LOGICAL BIT,TYPE,STD
901   FORMAT(' ',3I3,I7,I6,F9.3,6X,I3,4X,4I5,15X,A1)
902   FORMAT('+',T33,F6.3,3X,I4,20X,2F7.3)
903   FORMAT('+',T82,I3)
910   FORMAT('+',T35,'X',9X,'X',23X,'X       X')
911   FORMAT('+',T83,'HKL>20:')
914   FORMAT('+',T90,'S/N<0.7:')
915   FORMAT('+',T90,'INT LOW:')
916   FORMAT('+',T98,'TTH ERROR>0.2:')
917   FORMAT('+',T98,'NO NH1:')
918   FORMAT('+',T105,'NO NH2:')
920   FORMAT('+',T83,'DATA FORMAT ERROR')
931   FORMAT('+',T83,'NARROW SCAN:')
932   FORMAT('+',T96,'G ERROR>100%:')
941   FORMAT('+',T83,'T',F5.2,':')
942   FORMAT('+',T90,'O',F5.2,':')
943   FORMAT('+',T97,'C',F5.2,':')
944   FORMAT('+',T104,'P',F5.2,':')
950   FORMAT(3I5,2I8,2X,F8.5,20X,I4,6X,A4)
951   FORMAT(3I5,2I8,2X,F8.5,15X,I4,11X,A4)
      BIT(N)=FLAG*2/2**N-FLAG/2**N*2.NE.0
      DATA NOS/0/
    3 STD=.FALSE.
      IF(H.EQ.NSTD.AND.K.EQ.KSTD.AND.L.EQ.LSTD)STD=.TRUE.
      IF(STD)NOS=NOS+1
      IF(.NOT.STD)NOL=NOL+1
      CALL ANGLES(H,K,L,TTH,OM,CHI,PHI,REV,ET,EM,EC,EP)
      NSTP=G*GSTP
      INT=COUNT-BGR
      CEN=(GSTP-NH1-NH2)*G/400.0
      WRITE(3,901)H,K,L,INT,BGR,TTH,G,GSTP,NSTP,NH1,NH2,REV
      IF(G+GSTP+NH1+NH2+COUNT.EQ.0)GO TO 4
      IF(STD)WRITE(3,903)NOS
      R=SIN(TTH/57.2958)
      IF(STD)WRITE(2,951)H,K,L,INT,BGR,R,NOS,ITAG
      IF(.NOT.STD)WRITE(2,950)H,K,L,INT,BGR,R,NOL,ITAG
      FLAG=0
      IF(IABS(H).GT.20)  FLAG=FLAG+1
      IF(IABS(K).GT.20)  FLAG=FLAG+2
      IF(IABS(L).GT.20)  FLAG=FLAG+4
      IF(10*INT/BGR.LT.7)FLAG=FLAG+8
      IF(COUNT.LT.1000)  FLAG=FLAG+16
      IF(ABS(CEN).GT.0.2)FLAG=FLAG+32
      IF(NH1.LE.2)       FLAG=FLAG+64
      IF(NH2.LE.GSTP/2-2)FLAG=FLAG+128
```

```
      IF(FLAG.NE.Ø)GO TO 1Ø
      FWHMH=G*(NH2-NH1)/2ØØ.Ø
      IF(FWHMH.NE.Ø)CORR=1.Ø/ERF(NSTP*Ø.ØØ416/FWHMH)
      GEST=42ØØ.Ø*FWHMH/INT
      IF(GEST.EQ.Ø)GEST=1
      WRITE(3,9Ø2)CEN,GEST,FWHMH,CORR
      IF(KOBS.EQ.Ø)GO TO 2
      FLAG=KOBS
      IF(BIT(1))WRITE(3,941)ET
      IF(BIT(2))WRITE(3,942)EM
      IF(BIT(3))WRITE(3,943)EC
      IF(BIT(4))WRITE(3,944)EP
      GO TO 1
    2 IF(CORR.GT.1.2)WRITE(3,931)
      IF(IABS(GEST-G)/MINØ(G,GEST).GT.1)WRITE(3,932)
      GO TO 1
   1Ø WRITE(3,91Ø)
      IF(BIT(1).OR.BIT(2).OR.BIT(3))WRITE(3,911)
      IF(BIT(5))GO TO 11
      IF(BIT(4))WRITE(3,914)
   11 IF(BIT(5))WRITE(3,915)
      IF(BIT(7).OR.BIT(8))GO TO 12
      IF(BIT(6))WRITE(3,916)
   12 IF(BIT(7))WRITE(3,917)
      IF(BIT(8))WRITE(3,918)
      GO TO 1
    4 WRITE(3,92Ø)
    1 TYPE=.TRUE.
      RETURN
      END
      SUBROUTINE STAND(H,K,L,TTH,OM,CHI,PHI,G,GSTP,NH1,NH2,INT,BGR)
      COMMON KOBS,NOL,TYPE,LINES,NOP,IHEAD1,IHEAD2,IHEAD3,NSTD,KSTD,LSTD
     1,NOS
      INTEGER H,G,GSTP,BGR
      DIMENSION INTEST(9,2ØØ),REALST(4,2ØØ)
  9ØØ FORMAT('1 COD ',A4,T36,2A4,T73,'PAGE',I3//'  STANDARDS'//'   H  K
     1 L  INTEN   BGR   2THETA ERROR  G GES GSTP NSTP  NH1  NH2  FWHMH
     2CORR'/)
  9Ø1 FORMAT(' *** STANDARD',I3,' ***')
      IF(H+K+L+TTH+OM+CHI+PHI+G+GSTP+NH1+NH2+INT+BGR.EQ.Ø)GO TO 1
      NOS=NOS+1
      WRITE(3,9Ø1)NOS
      INTEST(1,NOS)=H
      INTEST(2,NOS)=K
      INTEST(3,NOS)=L
      REALST(1,NOS)=TTH
      REALST(2,NOS)=OM
      REALST(3,NOS)=CHI
      REALST(4,NOS)=PHI
      INTEST(4,NOS)=G
      INTEST(5,NOS)=GSTP
      INTEST(6,NOS)=NH1
      INTEST(7,NOS)=NH2
      INTEST(8,NOS)=INT
      INTEST(9,NOS)=BGR
      RETURN
    1 IF(NOS.LE.Ø)RETURN
      DO 2 I=1,NOS
```

```
      I1=INTEST(1,I)
      I2=INTEST(2,I)
      I3=INTEST(3,I)
      R1=REALST(1,I)
      R2=REALST(2,I)
      R3=REALST(3,I)
      R4=REALST(4,I)
      I4=INTEST(4,I)
      I5=INTEST(5,I)
      I6=INTEST(6,I)
      I7=INTEST(7,I)
      I8=INTEST(8,I)
      I9=INTEST(9,I)
      IF(I-I/50*50.NE.1)GO TO 2
      NOP=NOP+1
      WRITE(3,900)IHEAD1,IHEAD2,IHEAD3,NOP
    2 CALL LIST(I1,I2,I3,R1,R2,R3,R4,I4,I5,I6,I7,I8,I9)
      RETURN
      END
      SUBROUTINE ANGLES(H,K,L,TTH,OM,CHI,PHI,REV,ET,EM,EC,EP)
      INTEGER BLANK,H,REV,STAR
      COMMON KOBS
      DATA BLANK,STAR/' ','*'/
      REV=BLANK
      KOBS=0
      CALL CALCAS(H,K,L,CAT,CAC,CAP)
      ET=CAT-TTH
      EM=CAT/2-OM
      EC=CAC-CHI
      EP=CAP-PHI
      ER=ABS(ET)+ABS(EM)+ABS(EC)+ABS(EP)
      IF(ER.LT.0.05)RETURN
      CALL CALCAS(-H,-K,-L,CAT,CAC,CAP)
      RET=CAT-TTH
      REM=CAT/2-OM
      REC=CAC-CHI
      REP=CAP-PHI
      RER=ABS(RET)+ABS(REM)+ABS(REC)+ABS(REP)
      IF(RER.GE.ER)GO TO 1
      REV=STAR
      H=-H
      K=-K
      L=-L
      ET=RET
      EM=REM
      EC=REC
      EP=REP
    1 IF(ABS(ET).GT.0.05)KOBS=KOBS+1
      IF(ABS(EM).GT.0.05)KOBS=KOBS+2
      IF(ABS(EC).GT.0.05)KOBS=KOBS+4
      IF(ABS(EP).GT.AMAX1(0.05,ABS(CHI/200.0-0.2)))KOBS=KOBS+8
      RETURN
      END
      SUBROUTINE CALCAS(N,K,L,TTH,CHI,PHI)
      DIMENSION H(3),HPHI(3),UB(3,3)
      LOGICAL GETUB
      DATA GETUB/.TRUE./
  801 FORMAT(2X,A1,1X,2(3(F8.5,3X)/4X),3(F8.5,3X),4X,F6.4)
```

```
901 FORMAT('ØUB',A1,2(3(5X,F8.5)/4X),3(5X,F8.5)/'ØWAV',4X,F6.4/)
    IF(.NOT.GETUB)GO TO 1
    GETUB=.FALSE.
    READ(1,801,END=2)IUB,((UB(I,J),J=1,3),I=1,3),WAV
    WRITE(3,901)      IUB,((UB(I,J),J=1,3),I=1,3),WAV
    HWAV=WAV/2.Ø
  2 RETURN
  1 H(1)=N
    H(2)=K
    H(3)=L
    CALL GMPRD(UB,H,HPHI,3,3,1)
    X=HPHI(2)
    Y=HPHI(1)
    PHI=ARCT(X,Y)
    X=HPHI(3)
    Y=SQRT(HPHI(1)*HPHI(1)+HPHI(2)*HPHI(2))
    CHI=ARCT(X,Y)
    TEMP=HWAV*SQRT(HPHI(1)*HPHI(1)+HPHI(2)*HPHI(2)+HPHI(3)*HPHI(3))
    TTH=Ø
    IF(TEMP.LE.1.Ø)TTH=114.5916*ARSIN(TEMP)
    RETURN
    END
    SUBROUTINE FILL(BUFFER,N,*)
    INTEGER BUFFER,BASE,STORE
    DIMENSION BUFFER(256),STORE(512)
    LOGICAL END
    DATA BASE,END/Ø,.FALSE./
    IF(END)RETURN 1
    IF(BASE.NE.Ø)GO TO 3
    DO 1 I=1,512
  1 STORE(I)=Ø
    CALL TREAD(STORE,SSI,LENGTH,&2,&2)
    GO TO 3
  2 END=.TRUE.
  3 DO 4 I=1,N
    J=I+BASE
  4 BUFFER(I)=STORE(J)
    BASE=BASE+N
    IF(BASE.GE.512)BASE=Ø
    RETURN
    END
```

APPENDIX 4

PROGRAM LOADING

The computer program is loaded using the paper tape reader but before this an initial loader must be toggled into the computer by using the Load-Address, Deposit, and Address switches at the console. This initial loader is called the Read-in Mode Loader (RIM Loader) and once it is toggled into the computer it should not be necessary to toggle it in again unless it is destroyed by errors occurring in the main program.

The complete RIM loader is as follows:

Read-in Mode Loader High Speed Reader

```
7752    7300    TEST,   CLA     CLL
7753    1376            TAD             TEMP    /Test for exit
7754    7640            SZA     CLA
7755    5776            JMP     I       TEMP    /Exit to loaded program
7756    6014    BEG,    RFC                     /Fetch character
7757    6011            RSF                     /Skip if flag set
7760    5357            JMP             .-1     /Wait for flag
7761    6016            RRB             RFC     /Read Char., get next
7762    7106            CLL     RTL
7763    7006            RTL
7764    7510            SPA                     /Column 8 in sign
7765    5352            JMP             TEST    /Leader?
7766    7006            RTL                     /Column 7 in link
7767    6011            RSF
7770    5367            JMP             .-1
7771    6012            RRB
7772    7420            SNL                     /Address or contents?
7773    3776            DCA     I       TEMP    /Store contents
7774    3376    START,  DCA             TEMP    /Store address
7775    5356            JMP             BEG     /Get next char.
7776    0000    TEMP,   0
```

The RIM loader is toggled into core memory using the console keys as follows:

1. Set the first address (7752) in the switch register (SR).
   (N.B. SR switches are UP for 1, DOWN for 0).

2. Press LOAD ADDRESS key.

3. Set the first instruction (7300) in the SR.

4. Press DEPOSIT key.

(continued)

5. Set the next instruction (1376) in the SR.
6. Press DEPOSIT key.
7. Repeat 5 and 6 until complete loader is toggled in.

To check that the RIM loader is in core memory and has not been destroyed the following steps should be followed:

1. Set the first address (7752) in the SR.
2. Press LOAD ADDRESS key.
3. Press EXAMINE key. The memory buffer (and accumulator) lights will display the instruction (which for the first examine should be 7300).
4. Repeat 3 until all instructions have been checked.

Once the RIM loader is in core, the main program tape can now be loaded as follows:

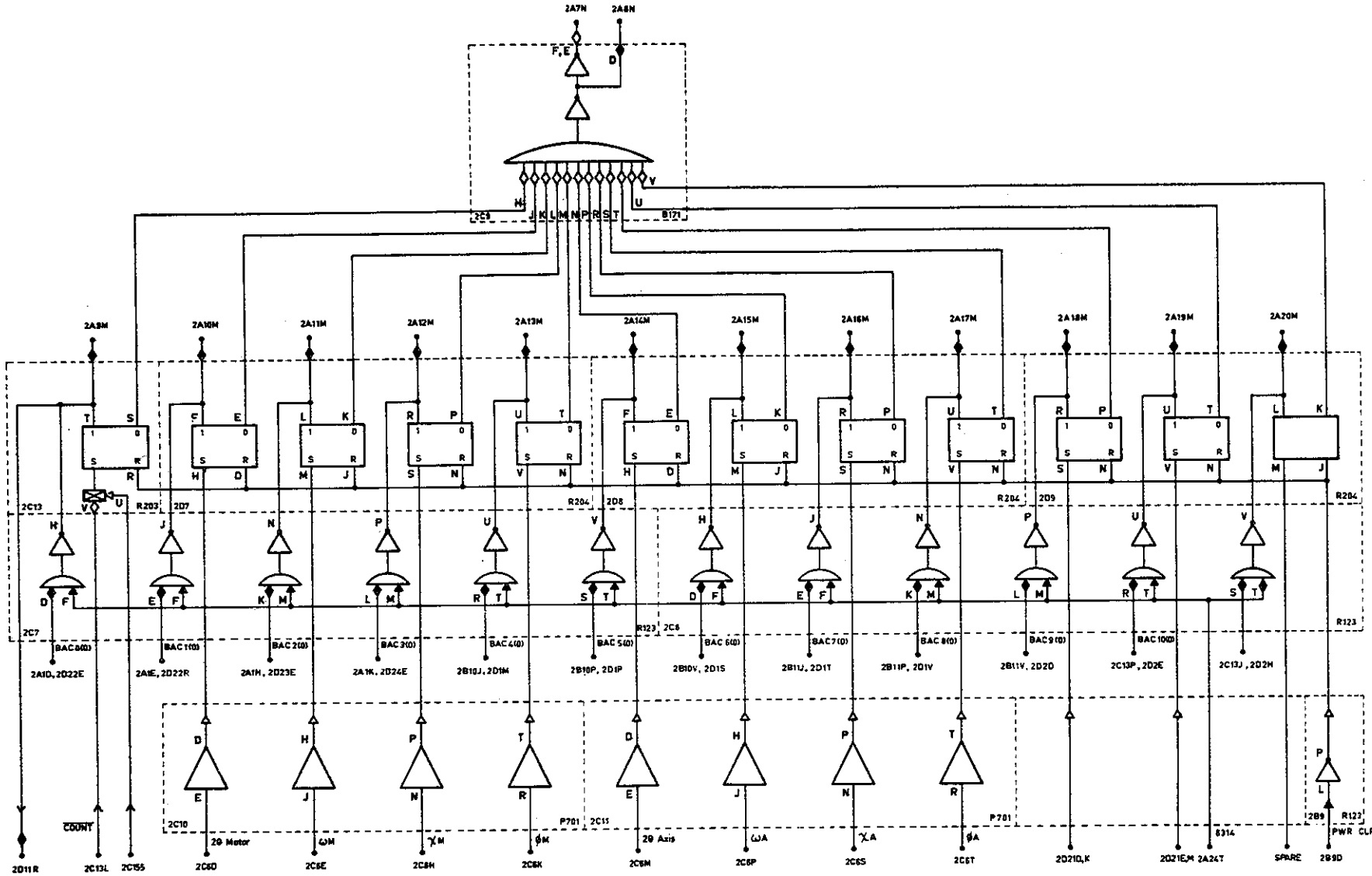
1. Place the main program tape into the reader head ensuring that the leader code at the start of the tape is over the head.
2. Set the RIM loader starting address (7774) into the switch register.
3. Press LOAD ADDRESS key.
4. Press START key.

The main program tape will now load and once loaded will commence operating. If the loading program detects an error the computer will halt with the error difference in the accumulator. If this occurs the loading sequence should be repeated and if the fault persists the RIM loader should be checked. If the fault still persists, the duplicate paper tape should be used and a note to this effect recorded in the log.

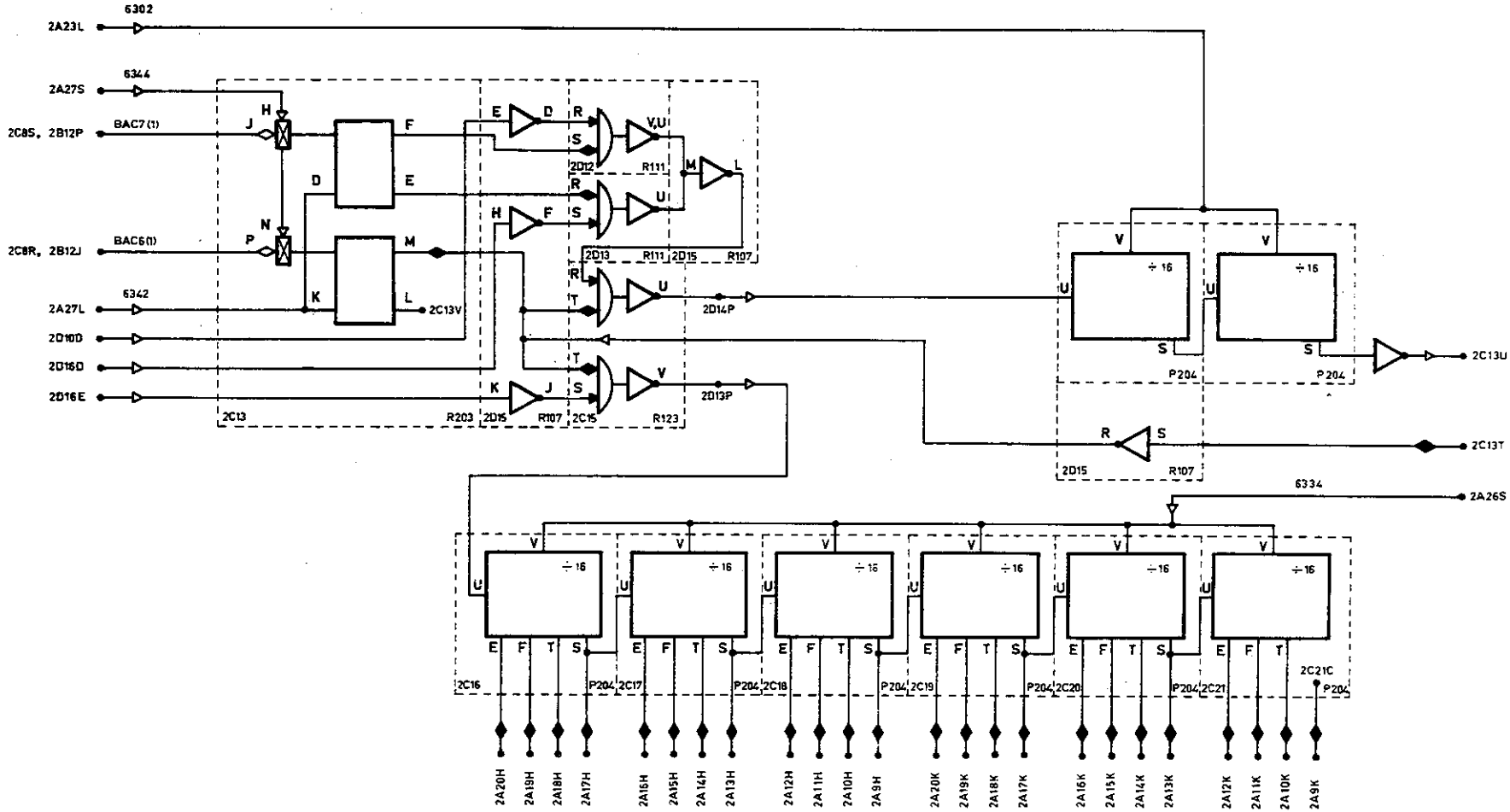
APPENDIX 5

INTERFACE CIRCUITS

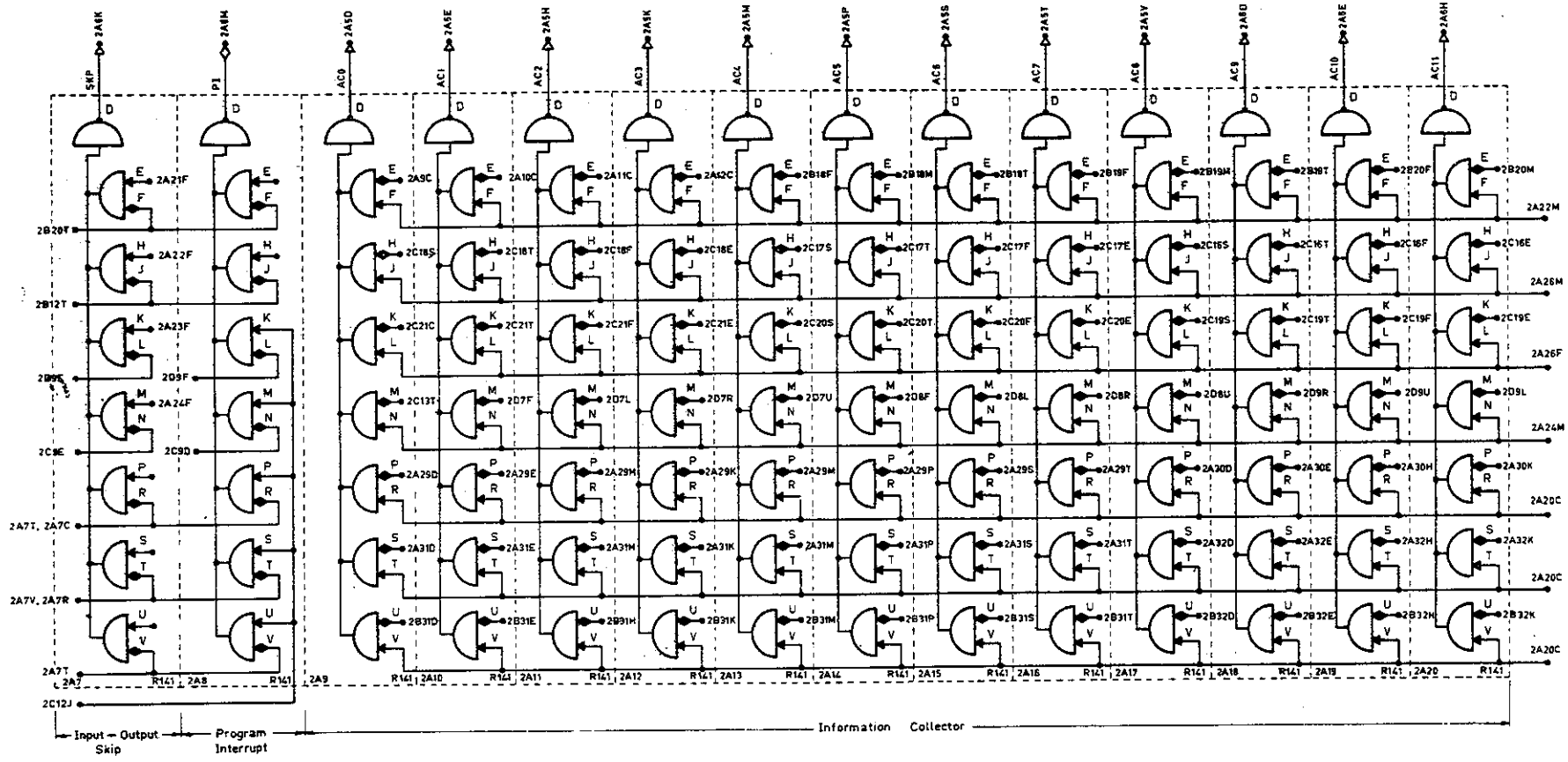
These schematic circuits of the interface hardware are included here as an aid to understanding the system. For detailed maintenance, up-to-date complete circuits should be obtained from the Instrumentation and Control Section.



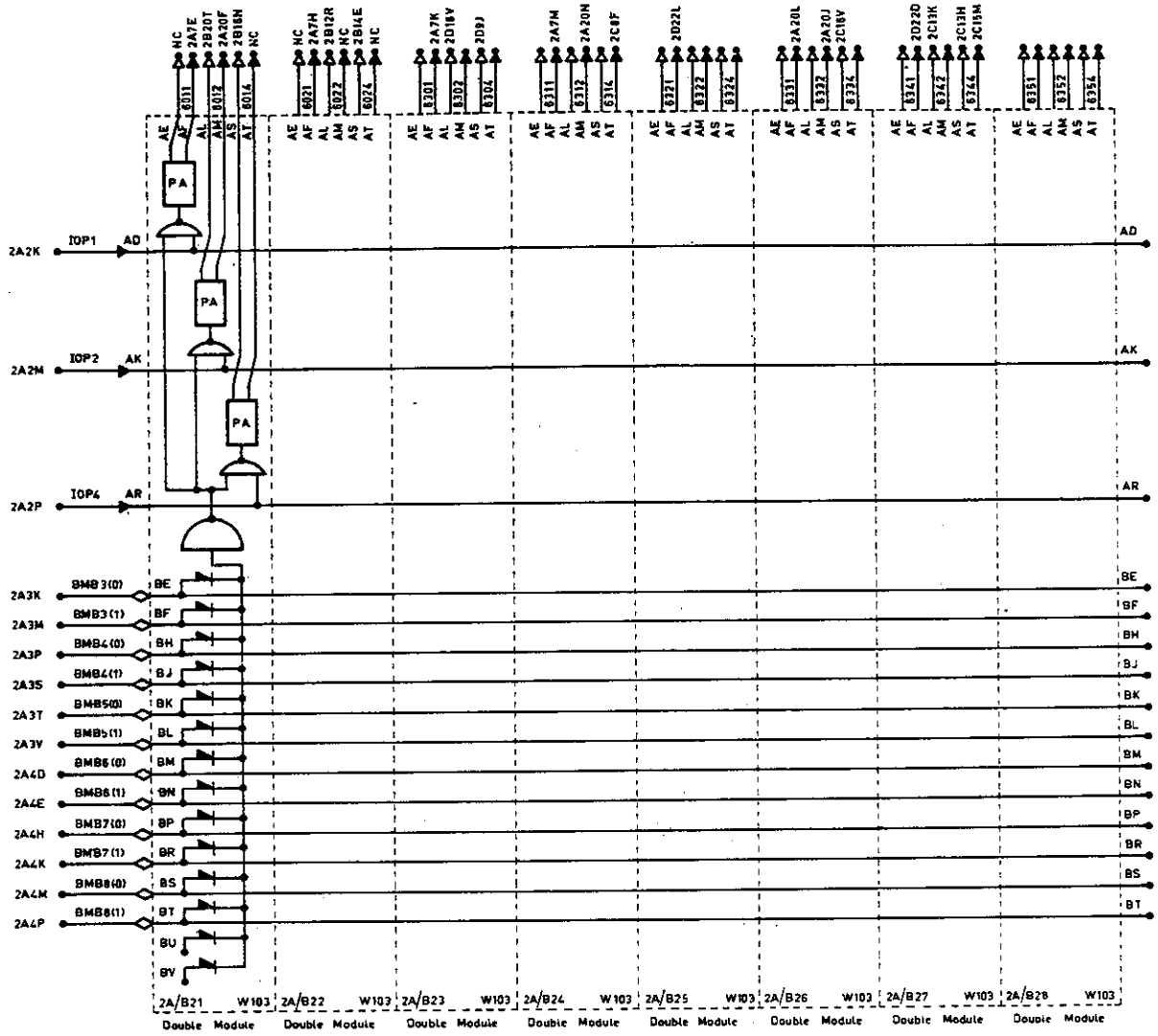
AUTOMATIC DIFFRACTOMETER - STATUS IN (DRAWING CE 24813)



AUTOMATIC DIFFRACTOMETER - STATUS OUT AND SCALERS (DRAWING CE 24812)

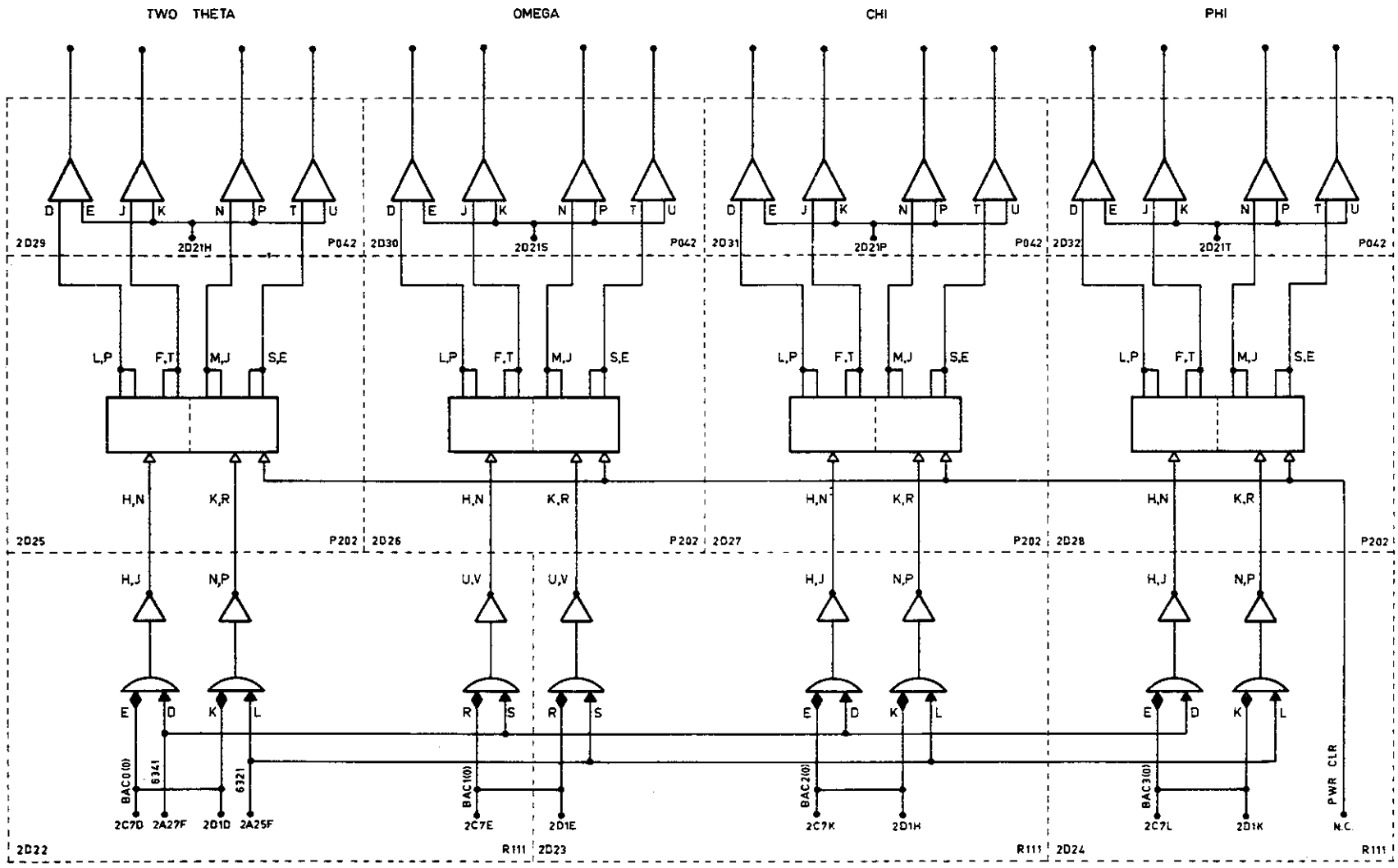


AUTOMATIC DIFFRACTOMETER - INFORMATION COLLECTOR - SKIP AND PROGRAM INTERRUPT (DRAWING CE 24811)

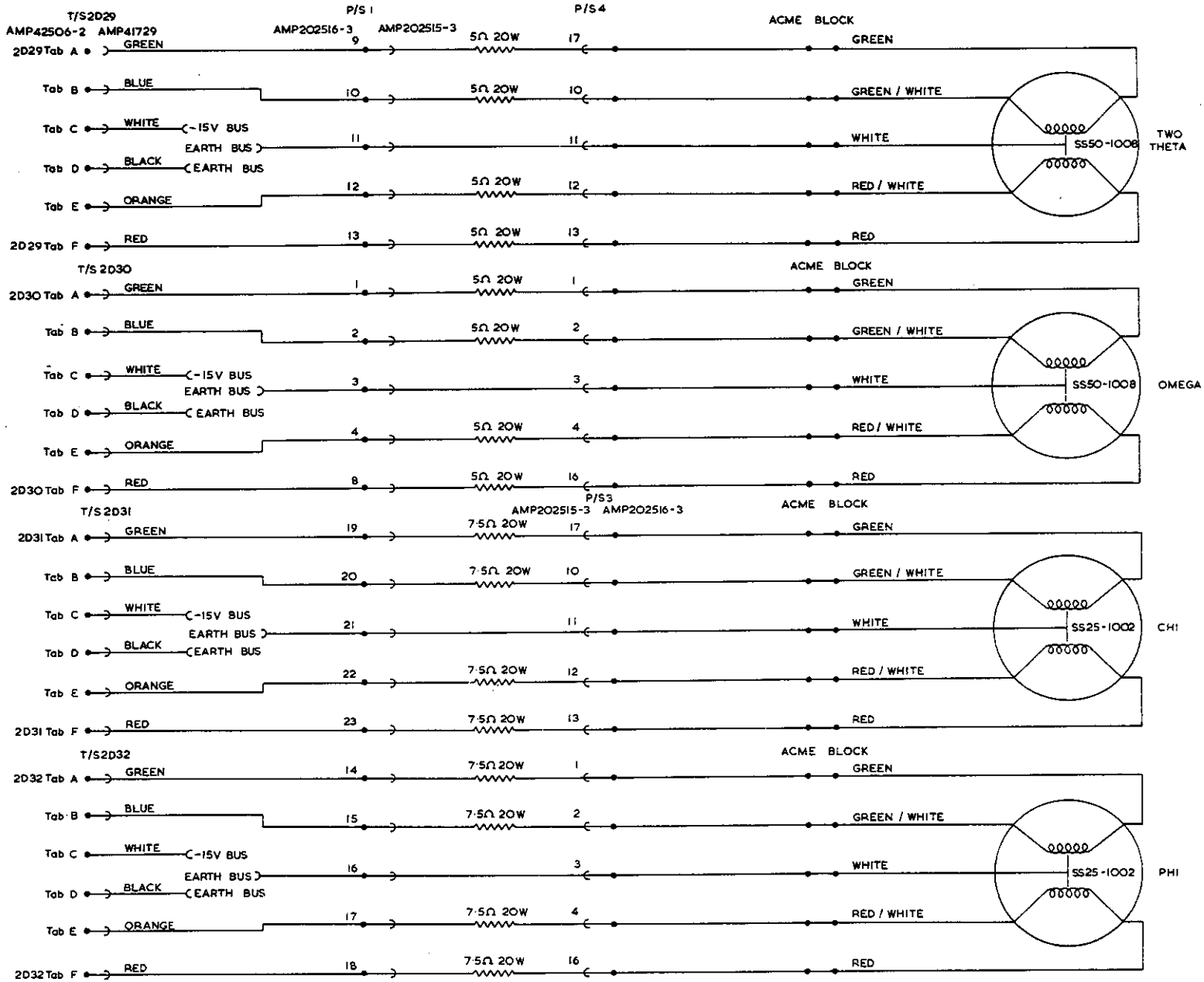


NOTE: ALL CARDS 2A21 TO 2A28 TO HAVE H-J, N-P, AND U-V INDIVIDUALLY CONNECTED TOGETHER.

AUTOMATIC DIFFRACTOMETER - DEVICE SELECTOR (DRAWING CE 24810)



AUTOMATIC DIFFRACTOMETER - MOTOR DRIVES (DRAWING CE 24814)



AUTOMATIC DIFFRACTOMETER - MOTOR WIRING (DRAWING CE 27842)

TABLE 1

CUMULATIVE ERRORS IN ANGLES DURING AZIMUTH ROTATION
WHEN USING THE APPROXIMATE METHODS OF SECTION 3.7 EQUATION 7

PSI	OMEGA	OMEGA ERROR	CHI	CHI ERROR	PHI	PHI ERROR
-89.993	45.000	-0.090	89.995	0.052	-89.995	0.116
-79.994	44.561	-0.080	82.943	0.051	-82.889	0.113
-69.995	43.218	-0.072	76.001	0.049	-75.563	0.109
-59.996	40.892	-0.063	69.292	0.047	-67.789	0.101
-49.997	37.452	-0.053	62.964	0.043	-59.315	0.089
-39.998	32.731	-0.041	57.201	0.037	-49.877	0.074
-29.999	26.564	-0.028	52.238	0.031	-39.230	0.055
-20.000	18.881	-0.015	48.359	0.022	-27.236	0.035
-10.000	9.851	-0.004	45.864	0.012	-14.002	0.015
0.0	0.0	0.000	45.000	0.000	0.0	0.000
10.000	-9.851	0.004	45.864	0.012	14.002	0.008
20.000	-18.881	0.015	48.359	0.022	27.236	0.011
29.999	-26.564	0.028	52.238	0.031	39.230	0.013
39.998	-32.731	0.041	57.201	0.037	49.877	0.017
49.997	-37.452	0.053	62.964	0.043	59.315	0.025
59.996	-40.892	0.063	69.292	0.047	67.789	0.036
69.995	-43.218	0.072	76.001	0.049	75.563	0.051
79.994	-44.561	0.080	82.943	0.051	82.889	0.069
89.993	-45.000	0.090	89.995	0.052	89.995	0.089
-89.993	54.738	-0.092	119.994	0.077	-54.731	0.058
-79.994	57.446	-0.072	111.687	0.074	-48.538	0.068
-69.995	59.102	-0.058	103.175	0.068	-43.032	0.072
-59.996	59.895	-0.048	94.554	0.062	-37.900	0.071
-49.997	59.915	-0.040	85.897	0.048	-32.891	0.067
-39.998	59.163	-0.033	77.272	0.040	-27.771	0.060
-29.999	57.557	-0.027	68.753	0.032	-22.292	0.050
-20.000	54.910	-0.020	60.432	0.022	-16.144	0.037
-10.000	50.897	-0.011	52.444	0.012	-8.910	0.020
0.0	45.000	0.000	45.000	0.000	0.0	0.000
10.000	36.473	-0.013	38.447	0.013	11.389	0.027
20.000	24.514	-0.024	33.335	0.028	26.109	0.051
29.999	9.032	-0.016	30.416	0.045	44.292	0.059
39.998	-8.134	0.020	30.337	0.057	64.138	0.039
49.997	-23.772	0.069	33.117	0.064	82.475	0.007
59.996	-35.928	0.106	38.132	0.061	97.388	-0.009
69.995	-44.619	0.125	44.624	0.059	108.934	-0.007
79.994	-50.635	0.133	52.029	0.055	117.954	0.009
89.993	-54.733	0.136	59.994	0.052	125.260	0.031

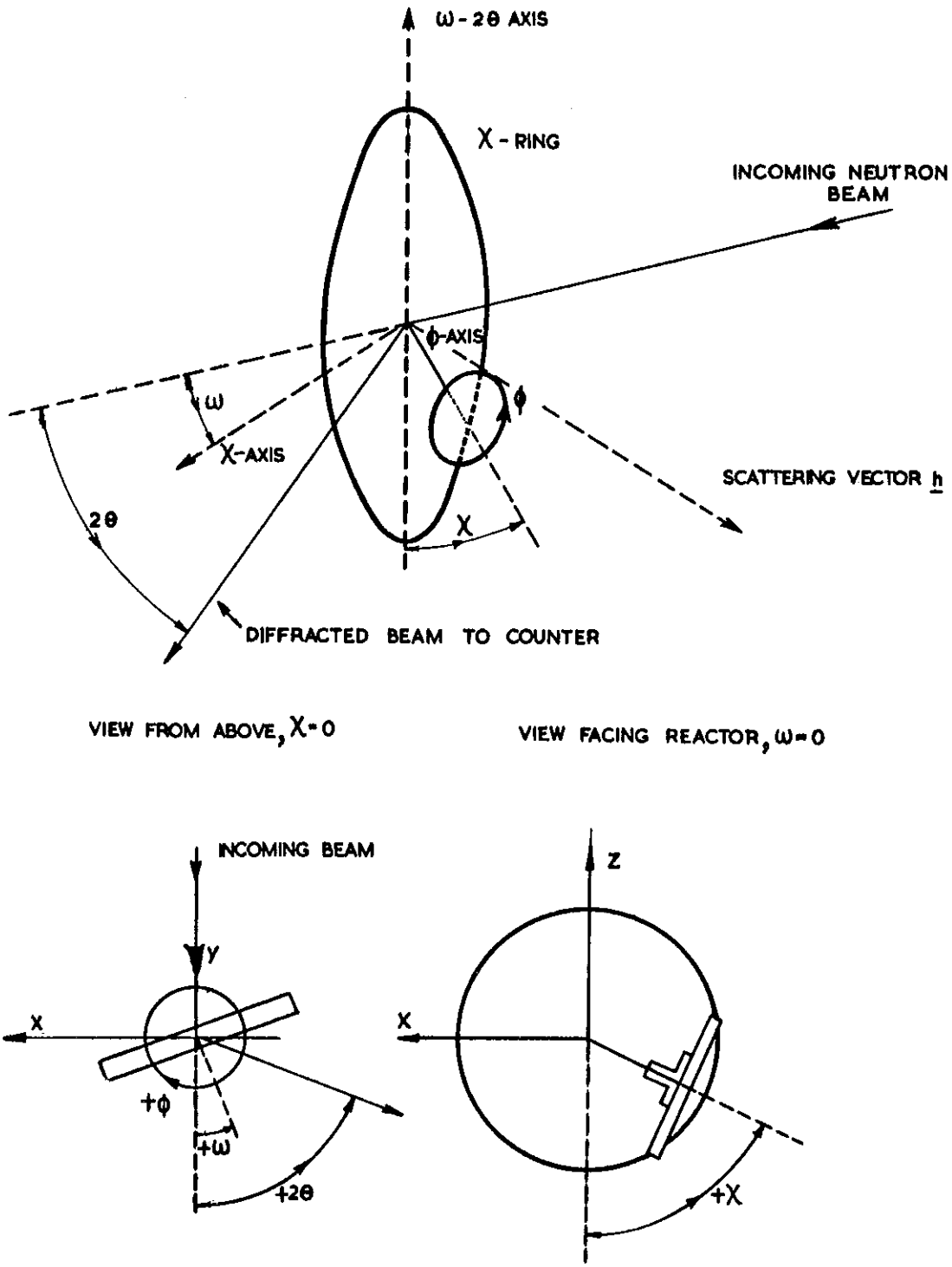


FIGURE 1. DEFINITION OF ANGLES FOR 4-CIRCLE DIFFRACTOMETER