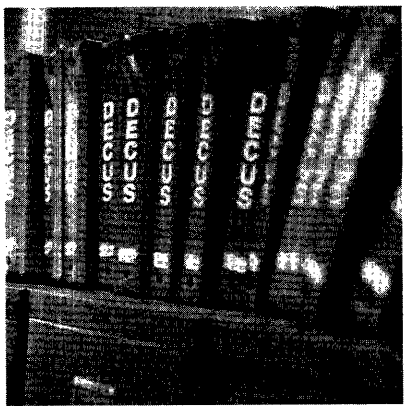
DECUSCOPE

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY

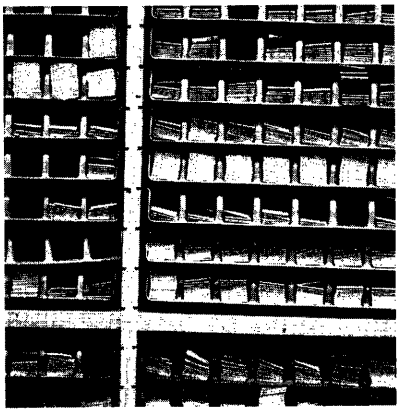
Vol.10, No.1

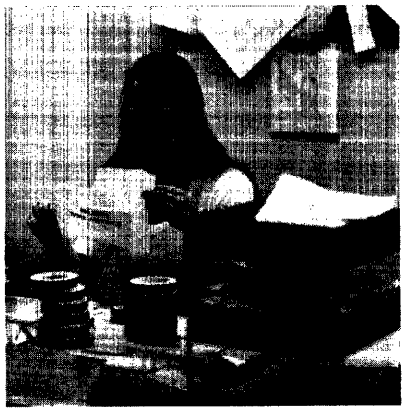
1



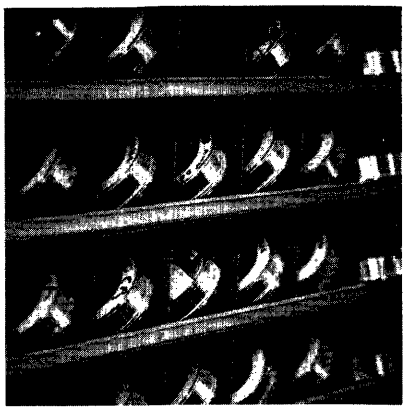
2

4





5



3

1 Angela J. Cossette
Executive Director

2 Maryann Oskirko
Group Secretary

3 Ferne Halley
Program Librarian

4 Helen Tucker
PDP-8 Library Controller

5 Barbara Kowalczyk
PDP-10/12/15 Library Controller

DECUS EXECUTIVE BOARD - 1971

PRESIDENT

Mr. Richard J. McQuillin
Composition Technology, Inc.
Cambridge, Massachusetts

PRESIDENT-ELECT

Mr. George S. Cooper
LOGIC, Inc.
Detroit, Michigan

MEETINGS CHAIRMAN

Mr. John C. Alderman, Jr.
Georgia Institute of Technology
Atlanta, Georgia

PUBLICATIONS CHAIRMAN

Mr. John O'Donnell
TIME, Inc.
Stamford, Connecticut

STANDARDS CHAIRMAN

Mr. John Sauter
Sanders Associates
Nashua, New Hampshire

EXECUTIVE DIRECTOR

Mrs. Angela J. Cossette
Digital Equipment Corporation
Maynard, Massachusetts

DEC DELEGATE

Mr. William Segal
Digital Equipment Corporation
Maynard, Massachusetts

DEC SOFTWARE REPRESENTATIVE

Mr. Ken Stone
Digital Equipment Corporation
Maynard, Massachusetts

MAINFRAME COMMITTEE CHAIRMAN (PDP-10)

Mr. George Zepko
Stevens Institute of Technology
Hoboken, New Jersey

MAINFRAME COMMITTEE CHAIRMAN (PDP-11)

Mr. Gene Stengstock
Canberra Industries
Meriden, Connecticut

EUROPEAN DECUS EXECUTIVE BOARD - 1971

CHAIRMAN/PAPERS SECRETARY

Mr. John Elder
University of Manchester
Manchester, United Kingdom

PAPERS COMMITTEE

Mr. C. P. Malpus
University of Leeds
Leeds, United Kingdom

Mr. B. Perrette
Banque de France
Puteaux, France

Hans-Jurgen Trebst
Tandem Laboratory
Erlangen, Germany

Mr. I. Widegren
FOA 1
Sundbyberg, Sweden

DEC REPRESENTATIVE

Mr. Myron H. Myers
Digital Equipment International Europe
Geneva, Switzerland

DECUS EUROPEAN SECRETARY

Miss Martha Ries
Digital Equipment International Europe
Geneva, Switzerland

DECUS - Digital Equipment Computer Users Society

ESTABLISHED - MARCH 1961

First President - Charleton M. Walter
AFCRL
Cambridge, Massachusetts

Current President - Richard J. McQuillin
Composition Technology
Cambridge, Massachusetts

Membership 1961 - 12

Membership January 1971 - 9,444

First Symposium held in May 1962, Itek Corporation, Lexington, Massachusetts.

First Issue of DECUSCOPE - April 1962.

DECUS Program Library Established - 1962.

CONTENTS

DECUSCOPE HAS BEEN PUBLISHED SINCE APRIL 1962 AND IS THE OFFICIAL NEWSLETTER FOR DIGITAL EQUIPMENT COMPUTER USERS SOCIETY.

IT IS PUBLISHED PERIODICALLY AT THE DECUS OFFICE, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754.

TELEPHONE: Area Code 617, 897-5111, Ext. 2414

EDITOR: ANGELA J. COSSETTE, DECUS

CIRCULATION: 10,500 PER ISSUE

CONVERSATIONAL, TWO-DIMENSIONAL FITTING PROGRAM FOR PDP-9/L

R. D. Barton, J. D. Kittle, and J. S. Wadden
Physics Department
Carleton University
Ottawa, Ontario, Canada

ABSTRACT

We describe a program with two important features. First, it will do least-squares fitting of curves to sets of data points (X_i, Y_i) where both X and Y have random scatter. Second, the fitting is done by a man-machine conversational system that proves to be extremely convenient. The program contains subroutines for fitting the following function: polynomial (including powers -2 to +3), Gaussian with quadratic background, exponential with quadratic background, and a sophisticated Gaussian-exponential convolution with quadratic background and pure Gaussian core.

INTRODUCTION

This program, PKFIT, was written as a general purpose curve fitting system for data from pulse height analysis. It is common practice to use large computer centers for such data analysis, but there are two important advantages gained by using a small computer such as the PDP-9/L. First, man-machine conversation is made possible so that, by responding to the various questions asked by PKFIT, the operation has the choice of a wide range of options included in the program and access to immediate recycling of the fit iterations. Second, the operator can obtain results immediately during an experiment without the inconvenient waiting periods involved with using a large computer center for data analysis.

PKFIT is a multi-function program, i.e., the operator can choose one of many fitting functions. PKFIT has the capability of fitting data with random scatter in either the Y- or the X-direction of both. Curve fitting is done by a three-point quadratic minimization of chi-squared with operator controlled cycling. The simultaneous oscilloscope display is used for visual comparisons of experimental data and the vectors generated by the fit function.

THE FIT TECHNIQUE

Consider a set of two-dimensional data points (X_i, Y_i), $i = 1, 2, \ldots, n$, that is to be fit by a function $Y = C(X; p_1, p_2, \ldots, p_m)$, where the p_j are parameters appearing in C. That is, the expectation values over many identical experiments, $(\langle X_i \rangle, \langle Y_i \rangle)$, are related by the function C for some choice of the p_j. The problem is to determine these values of the parameters. If we assume that both the X's and Y's have random scatter we solve the problem as follows. We choose a set of X's (called ξ_i) and p's and let $C(\xi_i; p_j) = \eta_i$. Then the ξ_i and p_j are varied until

$$\chi^2(\xi_i, p_j) = \sum_{i=1}^{n} \left[W_i^y (\eta_i - Y_i)^2 + W_i^x (\xi_i - X_i)^2 \right]$$

is a minimum.

W_i^y and W_i^x are weight factors. In PKFIT, there are four options for specifying the weight factors.

W_i =
- i) constant for all i
- ii) input for each i
- iii) reciprocal of datum for each i
- iv) zero

Anyone of these can be chosen independently for x and y. Note that case iv) allows the reduction of the χ^2 equation to the case of 1-dimensional scatter in either the Y- or the X-direction.

Now, since χ^2 is a function of the m parameters p_j, we may consider it a function of one of these parameters, say p_j. A Taylor expansion for this function is:

$$\chi^2(p_j) = \chi^2(p_j^M) + \frac{\partial \chi^2}{\partial p_j}(p_j - p_j^M) + \frac{1}{2}\frac{\partial^2 \chi^2}{\partial p_j^2}(p_j - p_j^M)^2 + \cdots$$

where p_j^M is the value of the j^{th} parameter that minimizes χ^2. Then, we have:

$$\left.\frac{\partial \chi^2}{\partial p_j}\right|_{p_j^M} = 0$$

and,

$$\chi^2(p_j) = \chi^2(p_j^M) + \frac{1}{2}\frac{\partial^2 \chi^2}{\partial p_j^2}(p_j - p_j^M)^2 + \cdots$$

Therefore, near the minimum, χ^2 is a quadratic function of p_j. See fig. 1. We can then define a method for quadratic minimization of chi-squared. This is justified if the initial "guesses" at the parameter values are close to the correct values. In order to simplify the algebra, let us define:

$$Z = \chi^2; \quad P = p_j$$

Consider the general parabola $Z = aP^2 + bP + c$.

Then
$$\left.\frac{dZ}{dP}\right|_{P_M} = 0 \qquad P_M = \frac{-b}{2a}$$

Computing a value of Z for each of three different values of P, we have,

$$Z_1 = aP_1^2 + bP_1 + c$$

$$Z_2 = aP_2^2 + bP_2 + c$$

$$Z_3 = aP_3^2 + bP_3 + c.$$

We solve for a and b by Kramer's Rule and finally write

$$P_M = \frac{-1}{2} \begin{vmatrix} P_1^2 & Z_1 & 1 \\ P_2^2 & Z_2 & 1 \\ P_3^2 & Z_3 & 1 \end{vmatrix} \cdot \begin{vmatrix} Z_1 & P_1 & 1 \\ Z_2 & P_2 & 1 \\ Z_3 & P_3 & 1 \end{vmatrix}^{-1}$$

$$= \frac{1}{2} \frac{P_1^2 (Z_2 - Z_3) + P_2^2 (Z_3 - Z_1) + P_3^2 (Z_1 - Z_2)}{P_1 (Z_2 - Z_3) + P_2 (Z_3 - Z_1) + P_3 (Z_1 - Z_2)} .$$

Using the original notation again, this is the value p_i^M that minimizes χ^2 when the ξ's and other p's are held fixed.

In PKFIT, the operator inputs an estimate of the true value of i^{th} parameter, p_i, along with an increment, δp_i. The program then defines three points of the parabola by calculating χ^2 at $p_i - \delta p_i$. It then minimizes χ^2 using the above equation and outputs the corresponding parameter value p_i^M, along with the value of χ^2 at that point. It also outputs a statistical error on the parameter and the value of χ^2 per degree of freedom. If any parameter is to be fixed, the operator simply inputs zero for its increment, δp_i. Thus, any parameter with a non-zero increment is an "active" parameter in the fit. One fit cycle of the program occurs when all active parameters in the fitting function have serially minimized χ^2 once in the above way.

Since χ^2 is also a function of each X-value, ξ_i, by a similar argument we can define a procedure for an X-direction fit. The operator inputs an increment $\delta\xi$, and the program does a 3-point parabolic minimization of χ^2 for each i by calculating χ^2 at ξ_i, at $\xi_i + \delta\xi$, and at $x_i^{\xi} - \delta\xi$. The x-values corresponding to minimum χ^2 for each i, ξ_i^M, may then be used to refine the parameter values in the fitting function.

Figure 2 is a flow diagram of the fit procedure. By looping through the fitting cycle over the ξ's and p's a few times with suitably chosen increments $\delta\xi$ and δp_i one obtains an acceptable set of p's, their standard deviations and the final values of χ^2 and χ^2 per degree of freedom.

MOMENT CALCULATIONS

PKFIT includes a subroutine which computes and outputs statistical moment information about the raw experimental data to facilitate making the initial estimates of the parameter values. The computations are as follows. The program computes the area, N, the mean, M, and the first three central moments of the data by the formulas:

$$N = \sum_{i=1}^{n} Y_i ; \quad M = \frac{1}{N} \sum_{i=1}^{n} X_i Y_i$$

$$M^{(s)} = \frac{1}{N} \sum_{i=1}^{n} (X_i - M)^s Y_i , \quad s = 2,3,4$$

It outputs the following:

1) AREA
2) MEAN
3) STANDARD DEVIATION: $SD = [\, M^{(2)} \,]^{1/2}$
4) SKEWNESS: $= M^{(3)} / (SD)^3$
5) KURTOSIS: $= M^{(4)} / (SD)^4 - 3$
6) FWHM = (SD) · (2.3548) (= full width at half maximum for a Gaussian).

SUPPLIED FITTING FUNCTIONS

The PKFIT program provides the choice of four parameterized fitting functions for shape analysis of experimental data.

A) POLYNOMIAL: (6 Parameters)

$$P(x) = a_1 x^{-2} + a_2 x^{-1} + a_3 + a_4 x + a_5 x^2 + a_6 x^3$$

B) GAUSSIAN: (5 Parameters)

$$G(x) = \frac{N'}{\sigma\sqrt{2\pi}} e^{-(x-\bar{x})^2/2\sigma^2} + \left[B + a_1(x-\bar{x}) + a_2(x-\bar{x})^2 \right]$$

where:

$$N' = \sum_i Y_i - \sum_i \left[B + a_1(x_i - \bar{x}) + a_2 (x_i - \bar{x})^2 \right]$$

where: N' is the area of the peak with background removed.

B = background constant

a_1 = coefficient of background slope

a_2 = coefficient of background curvature

σ = (FWHM of Gaussian) / 2.3548

$\bar{x}$ = mean of Gaussian

C) EXPONENTIAL: (6 Parameters)

$$E(x) = N_{\bar{x}} e^{-\lambda(x-\bar{x})} + \left[B + a_1(x-\bar{x}) + a_2(x-\bar{x})^2 \right]$$

where: $N_{\bar{x}}$ = height of exponential at origin

λ = logarithmic decrement

$\bar{x}$ = origin x-value

B = background constant term

a_1 = coefficient of background curvature

a_2 = coefficient of background curvature

D) GAUSSIAN-EXPONENTIAL CONVOLUTION: (12 Parameters)

$$C(x) = N' \left[f_1 D_1(x) + f_2 D_2(x) + f_3 D_3(x) + f_4 F(x) \right] + \left[B + a_1(x-\bar{x}) + a_2(x-\bar{x})^2 \right]$$

where

$$N' = \sum_i Y_i - \sum_i \left[B + a_1(x_i - \bar{x}) + a_2(x_i - \bar{x})^2 \right]$$

N' is the area of the peak with background removed

B = background constant term

a_1 = coefficient of background slope

a_2 = coefficient of background curvative

and the $D_K(x)$ are the three convolution terms for the three exponentials folded into the Gaussian given by:

$$D_k(x) = \int_{\bar{x}}^{\infty} \lambda_k e^{-\lambda_k(u-\bar{x})} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-u+\bar{x})^2} du$$

$$= \frac{\lambda_k}{2} e^{-\lambda_k(x-\bar{x})} e^{(\lambda_k\sigma/\sqrt{2})^2} \left[1+\mathrm{erf}\left(\frac{x-\bar{x}}{\sigma\sqrt{2}} - \frac{\lambda_k\sigma}{\sqrt{2}}\right)\right]$$

Further: λ_K = decrement of K^{th} exponential (K=1,2,3,)

σ = standard deviation of Gaussian

$\bar{x}$ = centroid of Gaussian

f_1, f_2, f_3 = fractional intensities of the three exponential convolutions.

$f_4 = 1 - (f_1 + f_2 + f_3)$ = fractional intensity of additional Gaussian term given by:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\bar{x})^2/2\sigma^2}$$

Note that any of the $D_K(x)$ may be omitted from the convolution expression by setting its fractional intensity, f_K, equal to zero. Also, the slope of each exponential, λ_K, may be either positive or negative corresponding to convoluting the exponential into the Gaussian from either the left side or the right side respectively.

HARDWARE REQUIREMENTS

PDP-9/L (or PDP-9) with 16K memory, EAE, high speed reader and punch, teletype, DECtape and Oscilloscope Display Type 34H.

SOFTWARE REQUIREMENTS

PDP-9 Advanced Software System, PDP-9 Keyboard Monitor System

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial assistance of the National Research Council of Canada. The ideas of Dr. R. J. McKee were invaluable to this work and we thank him for many helpful conversations.

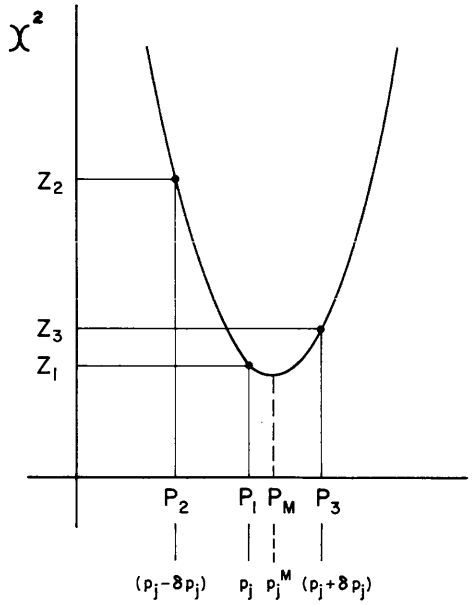


Fig. 1 Schematic plot of χ^2 vs p_j in the Neighborhood of p_j^M.

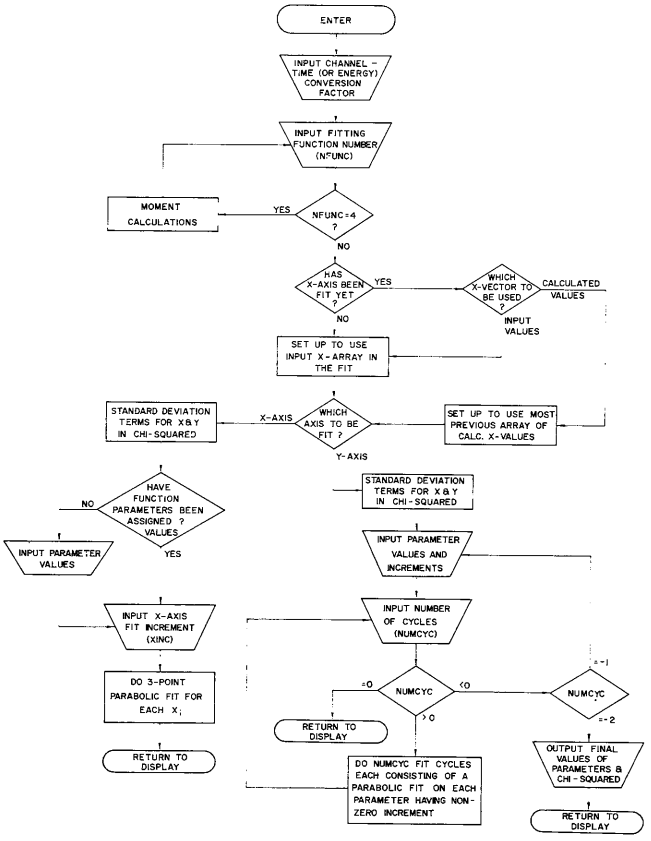


Fig. 2 Flow Chart of the Man-Machine Conversational Fitting Procedure.

A PDP-9/L PROGRAM TO HISTOGRAM DATA FROM FOUR-WORD EVENTS

R. D. Barton, J. D. Kittle and J. S. Wadden
Carleton University
Physics Department
Ottawa, Ontario, Canada

ABSTRACT

A program has been developed for a PDP-9/L computer that histograms data from four-word events in a nuclear physics experiment. The data from any of the four registers can be histogrammed subject to restrictions placed on the values of the four words of each event. Many such spectra can be simultaneously accumulated. A number of keyboard-controlled facilities are available to the user. The program can be used for any number of registers less than five.

INTRODUCTION

We have built a delayed coincidence system for the measurement of life-times of excited states of nuclei and for the investigation of timing problems in nuclear physics. Whenever the fast logic declares an event valid, four registers are loaded with data. These four words are a time interval (register 0, 12 bits), the energies deposited in the start and stop counters (register 1 and register 2, 8 bits each) and a tag word (register 3, 4 bits). The system then interrupts a PDP-9/L computer which deposits the four words into memory for processing and possible histogramming.

This paper describes the data processing system and various other keyboard-controlled facilities. The program can be used in any application involving histogramming data from less than five registers. It contains the usual assortment of display and output facilities found in pulse height analysis programs as well as a few original features. We have found that some of these features are extremely convenient.

DATA PROCESSING

The purpose of the processing subroutine is to accumulate histograms of time-interval or energy spectra. Each spectrum is defined by setting a condition on each of the four words of an event and specifying which register output is to be histogrammed. This is accomplished by typing AC (for accumulate) command. Each line of this command defines one spectrum. Such a line has the following format:

n OR0,EX0. OR1, EX1. OR2,EX2. $b_3 b_2 b_1 b_0$

where n (= 0,1,2,3) is the register whose output is being histogrammed; OR0 is, for this spectrum, the origin of the acceptable range of register 0; EX0 is the extent of this range; OR1, EX1, OR2, EX2 have analogous definitions for registers 1 and 2; b_3, b_2, b_1, and b_0 are the acceptable values of the bits of register 3.

There may be up to 65 such lines in one AC command.

After an AC line is typed in, a region in bank 1 of our 16K memory is assigned to this spectrum. The absolute address of the origin of this region and its extent are typed out for later use in typing display and output commands.

After an AC line is typed in, a region in bank 1 of our 16K memory is assigned to this spectrum. The absolute address of the origin of this region and its extent are typed out for later use in typing display and output commands.

After the desired number of AC lines is typed in, the system is ready to accumulate data. On receiving an interrupt the program first tests for a flag from the apparatus. If this is set the four data registers are loaded into memory. Each word is tested against the appropriate condition in the first AC line. If all four conditions are satisfied the number from register "n", as specified by this AC line, is used to calculate an address in the region of memory assigned to this spectrum. This location is then incremented. In the same manner each subsequent line of the AC command is tested and the corresponding histogram may receive an increment. After the data has been tested against all the lines the program clears and enables the apparatus and jumps to the display routine.

The process described above is sufficiently fast so that with the fourteen particular AC lines we used in one experiment, the limiting data rate was 250 per second. With one AC line defined, the limiting data rate is 1400 per second. These rates are faster than necessary in our application.

A further feature of the processing subroutine is that the original four data words of all events can be stored on DECtape for later analysis. This enables the user, after gaining knowledge from the first try at an experiment, to change the definitions of the spectra and not have to redo the experiment. He only has to read the original data off the DECtape and reanalyze each event.

DISPLAY AND OUTPUT

The program is normally executing a display loop. The user can type in a command to display any region of memory in bank 1 whose extent is less than 1025. Then the contents of the first location in this region is loaded into the Y display buffer and the X display buffer is set to zero. After the oscilloscope beam is momentarily intensified, the contents of the next memory location is loaded into the Y buffer and the X buffer is incremented. In this way every location in the region is plotted. Since the X buffer is a 10 bit register, it is incremented by the greatest integer less than 1024/EX, where EX is the extent of the region to be displayed. Thus, the display nearly always uses up most of the oscilloscope screen. The Y buffer is also a 10 bit register. In order to keep the memory contents on scale or to expand plots of histograms with only a few counts per channel, the user types in a full scale factor, F, between 0 and 99. The data is multiplied by $1024/2^F$ before being loaded into the Y buffer. Thus, 2^F-1 is plotted at full scale.

In order to locate features of a displayed spectrum, the typing of simply a number on the keyboard (followed by a period) causes that channel number of the current display to be plotted a number of times in succession. By then typing S the user obtains the ADC channel number and the absolute address in memory of this intensified location. This information has proved quite convenient for the simple peak analysis that we do.

There is also a facility by which the logarithms of the contents of a region of memory are displayed. This is useful for enhancing the low intensity portions of spectra or for straightening the tails of nuclear lifetime data.

The output facilities of the system are straightforward and quite standard. There are four such facilities. First the picture on the oscilloscope display can be plotted on an incrementing X-Y plotter. Second, the contents of any region of bank 1 can be typed out. At the start of any spectrum the AC line that defines it is included in the type-out. Third, data in this same format can be put on punched paper tape for storage and off line listing. Lastly, the data, along with the AC command, can be stored on DECtape. This gives a convenient method of entering data into our least squares fitting program.

R COMMAND

The R command is a keyboard controlled facility that computes and types out some statistics of any selected region of memory bank 1. It is meant to be applied to a single peak. The statistics are as follows:

$$\text{AREA} \quad N = \sum_i N(i)$$

$$\text{MEAN} \quad M = \sum_i iN(i)$$

$$\text{FWHM} = \frac{2.951}{N} \sum_i |i - M| N(i)$$

$$\text{SDOM} = \frac{1}{\sqrt{N}} \frac{\text{FWHM}}{2.3548}$$

where N(i) is the contents of channel i. The latter two are the Full Width at Half Maximum and the Standard Deviation of the Mean, respectively, if the peak is a Gaussian curve and N is reasonably large. They are surprisingly good approximations to these quantities for curves that are quite far from being Gaussian. The FWHM is calculated from the mean deviation rather than the more usual mean squared deviation in order to minimize the effects of statistically poor points far from the mean. This facility has made convenient many investigations of system properties that would otherwise have been inhumanly laborious.

DRIFT TEST

One of the important factors in our application is the stability of the system. In order to monitor this throughout a measurement one can specify a region of memory bank 1 to be drift-tested (usually containing a single peak). After a preset elapsed time the R command is automatically applied to that region, the results typed out and the contents of that region zeroed. This cycle repeats until the measurement is stopped. The individual centroid results are stored and upon command their mean and standard deviation are typed out. This facility has made possible studies of our system that would have otherwise been completely impractical. It is also a reassuring system monitor during any long run.

CONCLUDING REMARKS

We have described the main facilities in this program available to the user. There is a number of other keyboard commands for zeroing part or all of the data, starting and stopping the system, monitoring memory overflows, etc. The simple facilities we have described have proved to be as sophisticated as is reasonably useful in our application. Moreover, this program has the virtue of being easy for a new user to operate correctly with confidence.

HARDWARE REQUIREMENTS

PDP-9/L (or PDP-9) with 16K Memory, EAE, High Speed Reader and Punch, Teletype, DECtape, and Type 34H Oscilloscope Display.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial assistance of the National Research Council of Canada. Mr. M. Labonte and Mr. V. K. Carriere wrote PDP-8 programs for one of us upon which this program was based.

THE LAB-8 AS AN INSTRUCTIONAL TOOL

Dr. James W. Cooper
Digital Equipment Corporation
Maynard, Massachusetts

ABSTRACT

A simple program for the teaching of IUPAC organic chemical nomenclature has been developed for the Lab-8 computer system. Since this program runs easily in 4K, the computer-assisted instructional techniques generally considered to be available only to large computers are now readily available on the minicomputer.

INTRODUCTION

In general, the concept of computer-assisted instruction calls forth thoughts of timesharing computers with extensive hardware, CRT displays, complex systems programming and extreme cost. Alternatively, one thinks of small computers used mainly to solve simple mathematical equations using BASIC or FOCAL. In fact, all previously reported designs for chemical CAI have used extremely large computer systems. [1-3]

EDUCATIONAL MOTIVATION

In an effort to develop the potential of the minimum configuration Lab-8 as an educational tool, the program ISOMER[4] (Interactive Study of Organic Molecules by Educational Reinforcement) was developed. ISOMER is designed to assist organic chemistry students in mastering organic nomenclature, by asking them to identify all possible isomers of a given empirical formula.

Since IUPAC nomenclature is highly logical in structure, the small computer is an ideal choice to help point out to students the fallacies which arise from an illogical approach to this study. In this author's experience, some students have a great deal of difficulty grasping the essential logical simplicity of

the IUPAC system. The subset of these rules neccessary for identification of dibromopentanes is as follows:

1. Find the longest continous carbon chain
2. Number the substituents from that end of the chain such that the substituent numbers are lowest at the occurrence of the first difference.

The principle difficulty that students seem to have is in finding the longest carbon chain in a compound, if it is not drawn in a straight line. Conversely, students tend to draw some very convoluted representations of structures which can be most clearly drawn in a straight chain fashion, believing that they have found additional isomers. By introducing the Lab-8 computer as an instructional tool, one can prevent these bad habits from being reinforced, by having the computer "slap the student's hand" each time he re-enters the same compound, and by listing out any isomers which he missed immediately, allowing him to try again if he desires.

PROGRAM DESCRIPTION

ISOMER was developed to step students through the formation and identification of the twenty-one isomers of $C_5H_{10}Br_2$. It will accept virtually any twisting of the carbon chains of the pentanes, 2-methylbutanes, and dimethylpropanes as legitimate input, and correctly recognize the IUPAC name for the compound. It lists this name and whether the student has selected this structure previously. When the student indicates that he has identified all the isomers, not knowing how many there really are, he types D, and the program lists any structure which he omitted.

The logic of this program is extremely simple; it could be expanded to identify many more compounds with negligible coding. The program occupies locations 0-4200, with messages filling another 1000_8 words.

The program proceeds by displaying C's on the Lab-8 scope each time C is typed on the teletype. Bonds are placed between connected carbons. The relative positions of these C's are controlled by moving a pointer controlled by space, up-arrow, back-arrow, line feed and return. When a legal 5-carbon skeleton has been constructed, two Br's appear on the scope, whose coordinates are controlled by the four knobs on the Lab-8. The program then, on command, names the compound, tells the student if he has entered it before, and allows the Br's to be moved to generate a new isomer.

The naming process takes place exactly as a chemist would proceed. The program examines the x-y coordinates of each carbon, determines to how many it is connected and by proceeding through the chain with backtracking prevented, finds the longest chain and stores this order for naming purposes. There is need for only one trap for an illogical name, that of 1-bromo-2-bromomethlybutane. In all cases, the indices are determined by finding which carbon in the ordered list the Br is attached to. If the sum of the two coefficients is larger than five for the 2-methylbutanes or larger than six for the pentanes, the chain is renumbered from the other end.

The expandibility of this system for educational purposes is obvious: it would be possible to include an in-core accounting system, naming of all the C-2, C-3, and C-4 compounds and even timesharing with other teletypes and scopes, all in 4K to 8K. This makes it possible to utilize the inexpensive computer as a powerful skinnerian educational tool at an extremely low per student cost.

BIBLIOGRAPHY

1. D. Alpert and D.L. Bitzer, Science 167 1582 (1970)
2. E.J. Corey and W.T. Wippke, Science 166 178 (1969)
3. F.D. Tabbutt, Chem. Eng. News, Jan. 19, 1970, p.44
4. This program has been submitted to the DECUS Library. Until it is accepted, copies may be obtained directly from the author.

MODERNIZING A PDP-7

A. R. Atherton
Cavendish Laboratory
Cambridge, England

Abstract

New core store has recently been fitted to a PDP-7 and the opportunity taken to reduce the cycle time of the computer from 1750 nsec to 875 nsec.

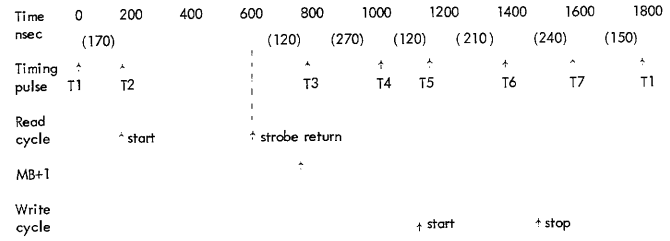
Introduction

The PDP-7 (8K, EAE) owned by the Bubble Chamber Physics Group of the Cavendish Laboratory has recently been replaced with a PDP-15/30. It transpired, however, that the group would not be able to use the PDP-7 for an alternative project unless the core store could be increased from 8 to 16K. Since the available funds were extremely low, we determined to expand the computer with commercial core store, interfaced by ourselves. While searching for a supplier of suitable core store we learned of another PDP-7 user engaged on a similar hunt. It was then sensible for us to provide them with our old core store, and to replace it with a single unit of 16K. This was not only advantageous financially but opened up the interesting speculation of determining the maximum speed at which the PDP-7 could operate. Even superficially it was obvious that the PDP-7 must be highly limited by its obsolete core store, since the CP was constructed from 10 MHz B-series modules and should be capable of as high speeds as the nominal 6 MHz M-series modules of the PDP-15.

Timing Consideration

Upon inspection of the former timing structure of the PDP-7, shown in Figure 1, it was clear that there must be a great deal of wasted time. The computer cycle was constructed around 7 timing

Figure 1



pulses (T1 to T7) whose time relationships are shown in the figure. It was noticeable that several of the delays were much greater than the B-series minimum of 100 nsec., and that the long delay between the memory read and write cycles was necessitated by the incrementation of the memory buffer (MB) that occurred on very few cycles (actually E.ISZ, CLOCK.B and auto index). Furthermore, there were more timing pulses between T4 and T1 of the next cycle than were really necessary.

Thus the minimum time of operation of the computer could be ascertained as follows. If T2 would start the memory read cycle, the T1 to T2 delay should be set by the time to set the memory address, say 70 nsecs. T2 to T3 should be the memory access time, plus 100 nsecs for the memory buffer to load. However this would be for computer cycles not involving incrementation of the MB. Cycles requiring MB + 1 would insert an extra delay between the memory data available pulse (strobe return) and T3 sufficient to allow for ripple propagation through the MB. Thus, T3 could also start the write cycle. The remaining delays (T3 to T4, T4 to T5, T5 to T6, T6 to T1) would be approximately the minimum possible for B-series modules, 100 nsec each for a total of 400 nsec. We thus arrived at a figure of 570 nsec plus the memory access time. Assuming a reasonable figure for modern memory of about 300 nsec, it appeared that the PDP-7 could operate at a cycle time of about 900 nsec.

The cheapest core store available in England at the time was probably Store/33, manufactured by Core Memories, Inc. and sold here by Data Products Ltd. This was available at two cycle times, 650 and 850 nsec., with access times of 275 and 350 nsec respectively. However, these times were for full cycle operation (read/restore or clear/write) and would be increased to 800 and 900 nsec for split cycle. We decided to purchase 16K x 18 bits of the faster model, so that the computer would be CP and not memory limited.

Installation

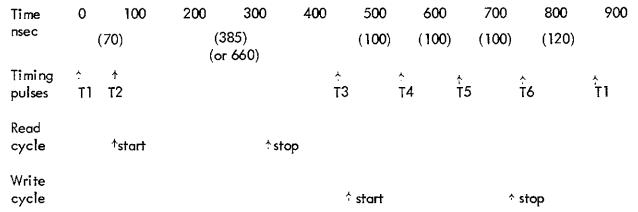
During the changeover the computer remained in day-to-day control of a film measuring machine. Though this delayed the rate of changeover, it was extremely useful since every alteration that rendered the computer inoperable could be discovered immediately. And of course it ensured that the modifications of the computer were installed on a step-by-step basis with rapid reversibility in case of malfunction.

The program of alteration was to install the new core store and make it operational on the basis of simple plug interchangeability with the old core store. Only when satisfied that the new core store was fully operational was the old core store finally removed and the process of decreasing the cycle time undertaken. The various modifications necessary (chiefly to remove the delay of T7 and to circumvent the DCD gate of a few R-series modules) were added gradually with few problems. For simplicity the necessary memory extension control was to operate in extend mode alone.

Final Configuration

The final cycle time of 875 nsec is constructed as shown in Figure 2. It is seen that for operations including MB + 1 this is increased to 1150 nsec. T7 is in fact still present in the computer but has few functions and is coincident with T1. The higher speed

Figure 2



required restructuring the IOT command (essentially by forcing it into permanent slow cycle), and the computer is now much the same as the PDP-15 in that an IOT takes 2μsec if IOP 1 only is present, and 4μsec if IOP 2 or 4 is present. Of course this allows proper gating against the MB. The opportunity has also been taken to add an increment accumulator instruction, 740030, which operates as on the PDP-15 to complement the link on overflow.

Assessment

In order to assess the full effect of these changes a FORTRAN numerical integration program was written, since this would use the EAE most fully and it was thought that the PDP-7 EAE was marginally faster than that on the PDP-15. The times taken on the various DEC 18-bit computers are shown in the table below, but it should be noted that the program was compiled on the PDP-7 and the resulting binary code run in each computer with the PDP-7 OTS. Thus, the times do not allow for the improved OTS (to the extent of approximately 20%) issued with the PDP-15. These times include the printing of some 200 characters of the computer teletype.

Computer	Time in Secs	Time Ratio to PDP-15
PDP-7 unmodified	1023.1	1.93
PDP-9	640.8	1.21
PDP-7 modified	548.6	1.03
PDP-15	531.4	1.00

Any modification of this sort would, of course, be useless if it reduced the extremely high reliability of the computer. However, with so reliable a machine it is difficult to construct sufficiently severe tests that would show up any minor decrease of reliability. In practice the day-to-day programs in use in the computer proved a more stringent test than any combination of the Maindecs. No program failures attributable to machine failure have shown up in several hundred hours' running time of these programs. Furthermore the margin tests for many Maindecs remain at least as good as before modification. We are satisfied that in fact there has been no change in reliability.

In conclusion we should add that the increased speed and size of the computer seem a very good return on a total outlay somewhat less than $6,000, and the part-time efforts of one person for three months. It is still not obvious that the computer is running near its maximum speed since most instructions involve very few operations in the CP. It should be possible to construct different length cycles, similar to that existing for MB + 1 operations, that would allow most cycles to use the full speed of the core store. However, these changes would necessitate restructuring the memory control to use read/restore mode as far as possible, and to use read/modify/write mode only for operations where the memory is to be altered. It is clear that such changes cannot increase the speed of the machine by a factor anywhere near as large as that already obtained, and in any case might well pose greater problems in the interrupt control.

DESIGN OF A PATCHABLE INTERFACE FOR ON-LINE CONTROL DEVELOPMENT

Mahesh K. Seth and John G. Bollinger
University of Wisconsin
Department of Mechanical Engineering
Madison, Wisconsin

INTRODUCTION

Users of minicomputers for control purposes are faced with the important problem of interface selection or design. Every user's problem tends to be unique and hence may require variations of the interface hardware and software. This is particularly true in a research and development laboratory where many projects are being conducted and needs vary. However, all interface systems may be thought of as being composed of basic building blocks which, when properly assembled, serve a specific input/output function.

A careful design of the logic that goes along with the hardware is very rewarding in the long run. Where the same computer has to be used for several projects, it becomes very important that the interface be easily adaptable to change. The computer for which the applications of problems are discussed and an interface described is the PDP-8/L.

No matter what the control application is, there are several aspects which are common to all. To mention some of them: an interface should have means of outputting and accepting as inputs, analog and digital signals; have access to a timing device (a real time clock); be able to assign priorities to control actions; have level converters to make external signals compatible to the computer logic; drive relays; detect contact closures, etc.

Hence, one approach could be to build a complete general purpose interface which can perform all input/output functions. However, this is a costly solution since it involves dedicating each piece of hardware to one particular problem. It is more economical to assemble the interface in small subsystems and then make these subsystems patchable as desired to meet a variety of demands with a minimum of problem setup time. The inputs to and outputs from these subsystems are brought to jacks on the patchboards, thus making the interface readily changeable.

This paper describes a general purpose, patchable interface wherein input/output aspects were undertaken individually. At the same time, the subsystems were made compatible with each other so that they would be patched together and used as one complete system in any control application. Some of the basic subsystems made from DEC modules are first described and then a few examples of using these subsystems together for control purposes are given.

PATCHBOARD LAYOUT

Layouts of the two patchboards are shown in Figures 1 and 2. The numbers of the patchboard refer to the module coding in the DEC Logic Handbook. The list of patched components is given in Table 1. Some of these components are dedicated to one of the prewired subsystems including the A/D converter, D/A converter and the relay drivers. The inputs to and outputs from prewired systems, several logic and memory elements, and I/O lines needed for "communication" with the process are available in the patchboard.

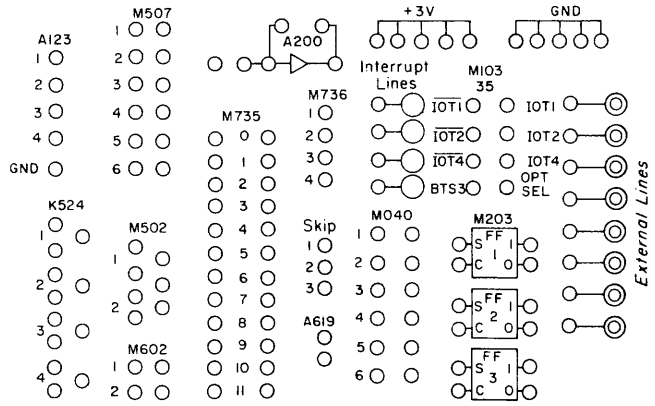


Figure 1 Patchboard 1

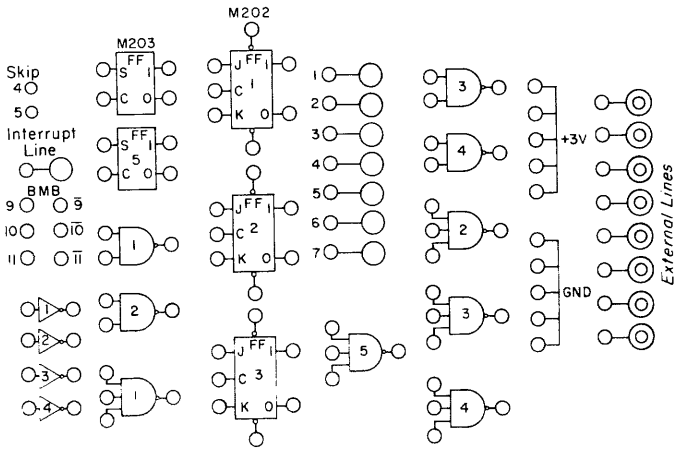


Figure 2 Patchboard 2

PREWIRED INTERFACE SUBSYSTEMS

Device Selector (M103)

In many cases, a user is required to use the I/O pulses from the computer to serve specific functions for solutions of specific interface problems. Bits 3 through 8 of an I/O instruction are used to code the pulses and the IOT pulses for a prewired code are available on the patchboard. Memory buffer bits 9, 10 and 11 are also available at the patchboard so that these may be combined with the IOT's to increase the capability of the device selector and obtain seven different IOT's as shown in Figure 3.

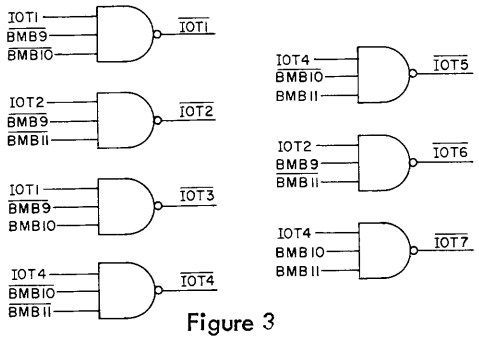


Figure 3

Increasing Capability of the Device Selector (M103)

Analog-Digital Conversion

The A/D conversion system consists of a multiplexer, a sample and hold and a converter. The schematic for the system is shown in Figure 4.

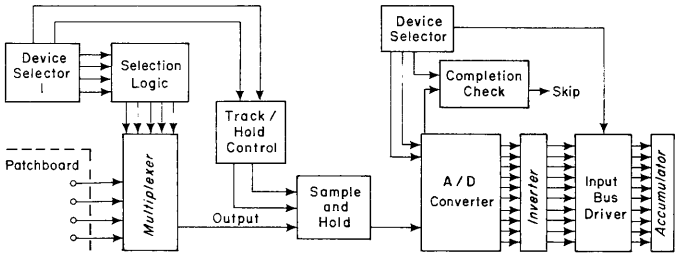


Figure 4 A/D Conversion Schematic

One device selector services the multiplexer and sample and hold. The logic which goes between the device selector and the multiplexer is a decoding circuit which connects the correct channel to the output of the multiplexer and hence to the input of the sample and hold.

The sample and hold is serviced by the same device selector which controls the state of an R-S flip-flop which puts the sample and hold in the track or hold mode. The time for which the input should be tracked depends on the acquisition time of the sample and hold.

A second device selector services the analog to digital converter. On a proper I/O instruction, a START pulse is sent to the converter which then produces a DONE pulse when the conversion is complete. This sets up a flag which is detected by using the SKIP facility. The digital output can then be transferred to the accumulator using the bus driver. If the two device selectors used to service the A/D conversion system have codes 30 and 31, the program would be as given in Program 1.

```
CLA       /Clear accumulator
630X      /X = 1,2,3,4 depending which channel is
            desired.
6306      /Track
6307      /Hold
6312      /Convert
6311      /Skip if DONE flag is set
JMP.-1    /Test flag again
6314      /Transfer to accumulator and clear DONE
            flag
```

Program 1 - A/D Software

Digital to Analog Conversion (A619)

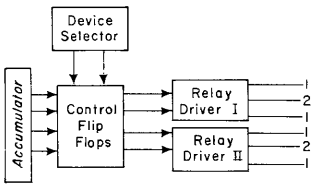
The digital to analog converter has its own buffer register and, hence, the inputs are connected directly to the ten most significant bits of the accumulator.

There is one pulse needed to load the buffer register and to start the conversion. This pulse is IOT1 obtained from the device selector which services the D/A converter. The selector has a prewired code, in this case, 14. The output of the D/A converter is bipolar and the output is available at the patchboard.

Relay Driver

There are two M040 Solenoid driver modules, each capable of controlling two relays. More modules may be added to the interface and all can be controlled by the same I/O instruction.

Each relay driver has four inputs, any of which can turn a relay on or off. In this case, all inputs are tied together and connected to the controlling flip flops which are set or reset to turn a relay on or off. The schematic for the circuit is shown in Figure 5.



1. Relay Coils Terminal
2. Supply Voltage to protect against back voltage.

Figure 5 Relay Drivers

Bit zero controls the first relay, bit one the second and so on. Hence, when the accumulator contains the correct number corresponding to the relays to be turned on, this information is available at the data input of the clock data flip flop. IOT4 from a device selector provides the clock input for these flip flops and the relay is then turned on or off according to the contents of the accumulator.

To prevent any relay from being turned on when the computer is switched on, the INITIALIZE pulse is used to reset all the flip flops and hence turn all relays off.

Sensor Converter (K524)

The sensor converter is a very useful device to have available on the patchboard so that every user does not have to build a separate interface to make his signals compatible with the system logic. It is basically a threshold sensing differential amplifier and can be used both with AC or DC coupling or adapted for resistence thresholds. Applications of this are presented in the next section.

Digital Data I/O Register (M735)

The M735 has two 12-bit registers, one for the input, and the other for the output. The left column of the M735 on the patchboard forms the input register which is connected to the accumulator through input bus drivers. The output register is buffered from the accumulator. The two registers are serviced by one device selector with a prewired code of 36.

For the various I/O instructions and their functions, the reader is referred to the DEC Logic Handbook.

Interrupt Facility

The patchboard provides access to the interrupt bus both directly and through a priority interrupt system formed with the M736. The input on the patchboard are the input to a hardwired AND gate, the output of which pulls the interrupt bus to ground if any one of the inputs is low.

The SKIP lines are also connected through a wired AND gate and any one low input pulls the SKIP BUS to ground.

Logic Gates and Flip Flops

Logic gates and flip-flops which form the building blocks of any computer interface are available on the patchboard, to be connected together as desired by the user.

Miscellaneous

Other useful devices available on the patchboard are an operational amplifier, a pulse amplifier and a negative input converter. The pulse amplifier is useful since it converts level changes to standard negative pulses. The negative input converter can be used to convert negative logic signals to the positive logic used in the computer.

APPLICATION EXAMPLES

In this section, some typical I/O functions are illustrated. It is common to encounter several I/O functions while attempting computer control of a process.

Detecting Contact Closure

In some control problems, it is necessary to detect contact closures such as limit switches. The K524 module can be used with DC coupling to produce a logic level and then this logic level can be checked using the SKIP facility.

The K524 produces a high output if the (+) input is more positive than the (-) input by at least 0.3 volts. The circuit can be so made that contact closure connects the test line to ground.

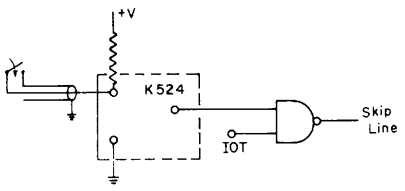


Figure 6 Detecting Contact Closure

The connections to the K524 inputs are shown in Figure 6. When the contact is open, the (+) input is ONE. When the contact is closed, the (+) input is at ground potential and hence the output is ZERO.

Now this logic level output can be detected using the SKIP facility. The output of the K524 and an IOT are connected to the inputs of a nand gate and the output from the gate is connected to the skip line. When the contact is open, the IOT pulse pulls the skip line to ground and hence the skip condition is satisfied.

The necessary patching on the patchboard is obvious from the Figure 6. The device selector which is available on the patchboard has a device code of 35 and if IOT1 is used for checking the skip condition, the necessary instruction will be 6351. The shield of the co-axial cable which carries the contact signal is connected to the computer ground.

If the program is to wait until the contact is open, the necessary waiting loop will be

```
6351    /Skip if contact is open

JMP.-1
```

The interrupt facility on the computer can also be used to detect contact closure.

The contact closure is detecting as before using the K524 sensor converter but now logic level output from it is connected to the interrupt line. When the contact closes, the interrupt is pulled to ground. The program then jumps to location 0000, stores the return address. The routine starting in location 0001 then detects the signal which is causing the interrupt.

Counting Pulses

In cases where the number of times an event happens has to be counted, the user has several alternatives which may be used. The source of external information can be a pulse generator, an oscillator or a magnetic pick-up. In each case the external signal has to be conditioned to produce a logic level signal which can then be used to set a flip-flop. The program is written to detect this setting of the flip-flop, add one to the count and then clear the flip-flop so that the next external pulse can set it again.

As an example, consider the case of a magnetic pick-up as the external signal source. The first step is to convert the output of the pickup into a rectangular wave so that the number of level changes can be detected. The K524 signal converter can produce the necessary wave shape.

The external signal is patched to the (+) input of the K524 and the (-) input is connected to ground. As the external voltage swings above and below ground potential, the K524 output will change levels from ONE to ZERO, thus producing a rectangular wave.

The rectangular wave is used as the signal as the clock input of a J-K flip-flop, the J and K inputs of which are tied to ONE. The flip-flop will change its state on a high to low transition of the clock, i.e., act as a trigger flip-flop.

The state of the flip-flop can now be detected by using either the skip or the interrupt facility. The use of the interrupt facility is now considered. The flip-flop is normally set, a clock pulse causes it to get reset and hence cause the interrupt. During the interrupt service routine, the flip-flop state is again set so that the next ONE to ZERO transition of the clock causes another interrupt to occur. The final complete patching on the patchboard is shown in Figure 7. It is assumed that the interrupt service routine does not take longer than the time period between interrupts

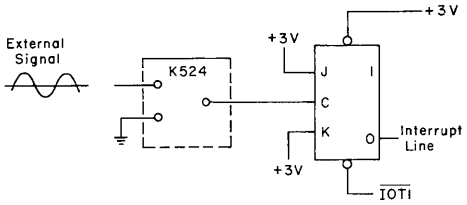


Figure 7 Counting Pulses

Relay Control

A commonly used form of control is pulse width modulation. This necessitates basically turning on relays for a certain length of time and then turning them off.

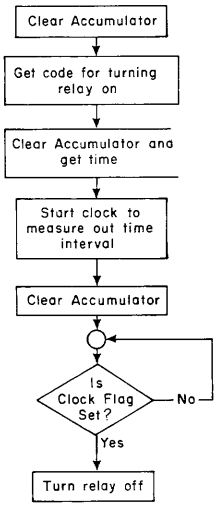


Figure 8

The time is calculated based on the set clock frequency and is then loaded into the clock. The flow diagram will be as shown in Figure 8. Program 2 shows the program necessary.

```
CLA
TAD CODE    /Code for turning relay on
6144        /Turn relay on
CLA
TAD TIME    /TIME = pulse duration
6136        /Start Clock
CLA
6133        /Skip if flag is one
JMP.-1      /Check again
6144        /Turn relay off
```

Program 2 - Relay Control Software

CONCLUSIONS

A patchable interface designed for the PDP-8/L has been described. The application of this equipment has been in reseach and development of open-loop and closed-loop feedback control systems. The system has proven to provide valuable flexibility and expandability in a laboratory where needs vary greatly. Where limited funds are available and must provide as wide a variety of opportunity for differenct projects as possible, a patchable interface as described could be invaluable.

There are many hidden advantages in building an interface as advanced here including the education and in depth understanding that one acquires by building up control systems using the patchboard modules.

A123	Positive Logic Multiplexer
A200	Operational Amplifier
A619	10-Bit D/A Converter Single Buffered
K524	Sensor Converter
M040	Solenoid Driver
M050	Indicator Driver
M103	Device Selector
M111	Inverters
M115	3-Input Nand Gates
M202	J-K Flip-Flops
M203	R-S Flip-Flops
M502	Negative Input Converter
M507	Bus Converter
M602	Pulse Amplifier
M735	I/O Bus Transfer Register
M736	Priority Interrupt Module

Table 1 - Patchable Interface Components

A SLOW-RUN OPTION FOR THE PDP-8*

D. W. Chamberlin
Lawrence Radiation Laboratory
University of California
Mercury, Nevada

INTRODUCTION

When debugging a PDP-8 program, it is often helpful to slow down the CPU and observe the console lights. You can watch ISZ loops zero out, IOT's get sent and so on. On a standard PDP-8, the only way to do this is to set SINGLE INST and wear out the CONTINUE key. This article describes a simple modification to slow down the CPU without using the KEY logic. Only two logic cards, a R602 and a R303, are added (spare slots are available) and a handful of wires changed.

A switch must be mounted to select NORMAL or SLOW and a speed control as shown is desirable. With the values shown, operation of 1 to 15 machine cycles per second is provided. A range switch could be added to the R303 timing capacitor to add faster speeds up to NORMAL.

For interface and CPU maintenance, it was desired to keep SLOW mode as similar to NORMAL as possible. Thus, the simple method of a CONTINUE pulser was not considered. (Key functions introduce extra CPU circuitry and timing; the RUN shuts down after each cycle, etc.) Also extra gating would be needed to recognize the HLT instruction.

THEORY

In the PDP-8, each machine cycle is split into two 750NS periods called T1 and T2. This is done by toggling the TG flip-flop with a 1.33 Mhz pulser called CLOCK and calling the set state T2 and the clear state T1 (See Figure 1). CPU

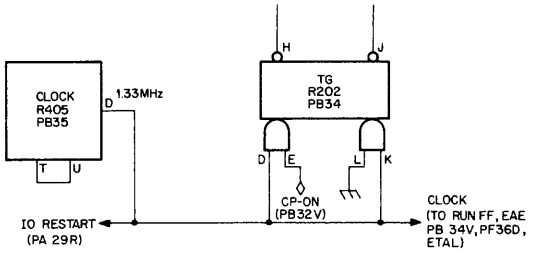


Figure 1
ORIGINAL CIRCUIT

and memory timing demand that a transition to T2 follow T1 by exactly 750NS so merely reducing the rate of CLOCK causes great grief. But T1 can come any time greater than 750 NS after T2, or T2 can be of any length. In fact, setting the CPU halts, TG stays on T2.

The approach then is to slow the cycle time by stretching T2 times. Figure 2 shows the circuit. The 750NS CLOCK rate is always present at the set side of TG, but only at the reset side if a) RUN-STOP is true of b) the R303 is not timing. Taking case b) first, if the switch is in NORMAL, then R303 will not trigger and operation is as originally designed. If the switch is in SLOW, then R303 times out each transition to T2 and during this period the pulse amplifier PA is disabled.

Case a) was the result of KEY CONT timing problems while in SLOW and the annoyance of having to hold the STOP key down for up to one second. RUN-STOP is true on any key function so all keys operate at normal speed regardless.

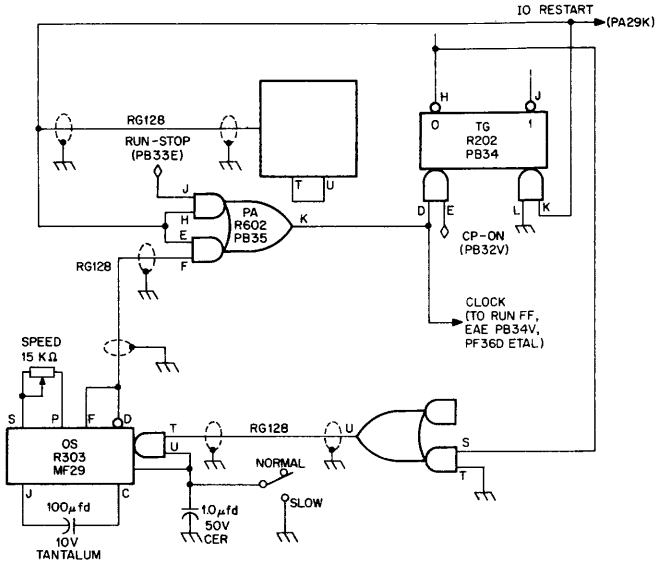


Figure 2
MODIFICATION

MODIFICATION

The speed control pot and NORMAL/SLOW were mounted between the two wood-grained panels on a small bracket placed on top of the center channel. When seated at the console, this requires a long reach. If available, a Data or Instruction field switch could be used.

Even with Extended Memory, EAE and all the rest there are two spare slots at ME 29 and MF 29. The CLOCK, a R405, was removed from PB 35 and relocated at ME 29. A R303 one-shot was then added at MF 29 and a R602 wired in at PB 35. Since ME 29 and MF 29 were blank except for power and PB 35 had very little circuitry, only a few wires need to be removed. The R602 buffers the TG, provides the needed gating and regenerates the remotely located CLOCK. Twisted pair with its higher Z would probably be a better choice than the 50 ohm RG128 since terminations were not used.

I would appreciate hearing of timing problems not provided for should any reader come across any. The modification has been in for six months at the time of this writing without problems. Note that this PDP-8 does not use a separate clock for the EAE.

*Work performed under the auspices of the U. S. Atomic Energy Commission.

EQUIPMENT CABINET FRONT COVER HANGUP SOLUTION

Submitted by: Richard R. Plum
Bell Telephone Laboratories, Inc.
Holmdel, New Jersey

If you are having a problem with the DEC equipment cabinet cover being loose or falling off, try the following "patch".

1. Order X* yards of one inch hook and loop No. 65 velcro ®black tape with No. 0140 precoat and one quart of V45D adhesive from Mr. Warren Auty of the F. Schumacher & Co. of New York City. The tape costs $2.82 per yard and the adhesive costs $3.55 per quart, which is the minimum amount obtainable.

2. Remove the nylon snaps on the cabinet brackets by drilling out the rivits with a No. 30 drill. The remaining snaps on the plastic covers may be removed with a putty knife or a similar instrument.

3. Allowing for two pieces of each type of tape per cover, cut the hook and loop tapes into two inch pieces. Following the directions on the adhesive can label, mount the hook tape on the brackets and the loop tape on the covers.

4. Replace the covers after the adhesive sets (about 15 minutes) and note that the "HANGUP" problem has disappeared. It is interesting to note that repeated attachment and removal of the covers will strengthen the hook-loop bond instead of the opposite.

*To determine the total yards required: Multiply the number of front covers on your cabinet by four inches. Divide the total inch figure by 36 and round to the next whole yard.

Editors Note: The editor has discussed this problem with individuals concerned at DEC. They are fully aware of the problem and are presently working on a solution. It is expected that the next issue of DECUSCOPE will carry an article with suggested solution by DEC. A DEC representative also commented regarding the above solution that if a machine is to be moved by truck or other means, further precautions should be taken to insure the covers will not fall off. Equipment currently being built by DEC include the new "pop-on" type covers.

AN INTRODUCTION TO VECTOR-8 A NEW PROGRAMMING LANGUAGE

Richard Rothman
Groton School
Groton, Massachusetts

Vector-8, a new interpretive language, is designed to provide a PDP-8 family computer sporting 8K and a DF32 disk with power previously unavailable to such a machine. The reasons for this are manifold. First, Vector-8 is not a language written for a basic 4K machine and slightly modified to accommodate extra hardware, but has rather been developed with the knowledge that 8K and a 32K disk, if properly utilized, can add considerable versatility to a language.

Secondly, Vector-8 applies a powerful concept to the manipulation of vectors. A vector is considered to be a unit rather than a collection of individual elements that have in common only the name of the vector. For example, in such languages as BASIC or FORTRAN, it is necessary, when applying an operation to an entire vector, or part of a vector, to do so element by element. In Vector-8, though, an operation is applied to an entire vector. There is no need to loop, and this removes much of the drudgery from programming. In addition, the ability to do in a single expression what would take several commands in other languages, allows for a flexible and powerful syntax. Vector-8, above all, provides one with the means of manipulating, with ease, in an on line environment, large amounts of data.

Vector-8 offers 15 operators (5 arithmetic, 6 relational, 3 logical, and cantenation), 32 functions, powerful commands, and a complete character string facility. User commands and functions can be easily interfaced to the Vector-8 interpreter, which is also equipped to work with high speed paper tape, if present. Now for a more detailed explanation of the language.

A vector can be specified in four ways:

1. A number string.
2. A character string.
3. By creating a variable.
4. By cantenating together separate vectors specified as per 1, 2, 3.

1. A number string consists of a series of numbers separated by spaces. The dimension of the vector specified is the number of numbers. Some examples:

1 2 3 4 specifies a vector of dimension whose first element is 1, whose second element is 2, whose third element is 3, and whose fourth element is 4.

5 4 1 specifies a vector of dimension 3 whose first element is 5, whose second element is 4, and whose third element is 1.

2. A character string consists of a series of characters contained within quotes. Carriage returns may be imbedded by typing line feeds. Line feeds will only be recognized within quotes. The dimension of the character string is equal to the number of characters in the string. Some examples:

"ABCD" specifies a vector of dimension 4 whose first element is "A", whose second element is "B", whose third element is "C", and whose fourth element is "D".

"CDX" specifies a vector of dimension 3 whose first element is "C", whose second element is "D", and whose third element is "X".

3. A variable is a symbol representing a vector. It is specified via a SET command. Some examples:

   SET A=1 2 3 4 specifies A as a vector of 4 whose first element is 1, whose second element is 2, whose third element is 3, and whose fourth element is 4.

A variable consists of at most two recognized characters. The first must be a letter, but not an "F". The second can be any letter or digit. Any subsequent letters or digits are ignored.

4. By cantenating together specified vectors, new vectors can be specified. Some examples:

   1 2 3,3 2 1 $\Rightarrow^{1}$ 1 2 3 3 2 1

   "ABC","DEF" $\Rightarrow$ "ABCDEF"

   SET A=1 2

   A,1 2 3 4 5 $\Rightarrow$ 1 2 1 2 3 4 5

All the vectors specified above have been of dimension 1 or greater. It is also possible to specify a vector of dimension 0. For example:

   "" specifies a null character string.

The ways of specifying null number strings will be shown later.

Specified vectors may be indexed by following the vector with an expression enclosed in parenthesis. Some examples:

   1 2 5 7(1 3 4) > 1 5 7

   1 4 8 9 [1] > 1

   "ABCDEF" (6 3 2) > "ECB"

   SET A=1 2 3

   A<2 3> > 2 3

Indexed vectors may themselves by indexed. Some examples:

   1 4 6 8(1 3 2) [1 2] > 1 6

   "ABCDEFG"(1 3 2) <1 2> $\Rightarrow$ "AC"

Indexing may go to any depth, and indices do not have to be integers, for they are truncated anyway.

1. reads "produces"

Vector-8, as mentioned before, has 15 operators, with which vectors are combined in various ways. All operators, with the exception of cantenation, can have their arguments in three forms. In the ensuing discussion, let :o: represent any operation except for cantenation. Let also d_n stand for a vector of dimension n. The three forms are:

1) d_1 :o: $d'_{n>1}$

2) $d_{n>1}$:o: d'_1

3) $d_{n>0}$:o: $d_{n>0}$

In case 1, the one element on the left is applied to the n elements on the right, to produce, in most cases, a result of dimension n. This process can be described by:

   for i=1,n apply d_1 to d'_i and set the result r_i

In case 2, the n elements on the left are applied to the one element on the right, to produce, in most cases, a result dimension n. This process can be described by:

   for i=1,n apply d_i to d'_1 and set the result r_i

In case 3, each element on the left is applied to its corresponding element on the right. This process can be described by:

   for i=1,n apply d_i to d'_i and set the result r_i

It should be noted that the :o: operations can not be used with null arguments.

Cantenation takes the following form:

$$d_{n\geq 0} \ , \ d_{n\geq 0}$$

The following are the Vector-8 operations:

		symbol	priority	description
1)	Arithmetic:	+	3	addition
		-	3	subtraction (or unary minus)
		*	4	multiplication
		/	4	division
		↑	5	exponentiation. a^b computed $e^{b\ln a}$.
2)	Relational:	:EQ:	2	relational =
		:GT:	2	relational >
		:LT:	2	relational <
		:GE:	2	relational $\geq$
		:LE:	2	relational $\leq$
		:NE:	2	relational $\neq$
3)	Logical:	&	2	logical and ($\wedge$)
		!	2	inclusive or ($\vee$)
		#	2	compression
4)	Cantenation:	,	1	cantenation.

Note: The higher the priority number, the higher is the priority of the operation. So in an expression, exponentiation is performed first, followed by multiplication and division, addition and subtraction, the relational and logical operators, and finally cantenation.

Group 1, the arithmetic group, does not operate on character strings. A few examples:

   1+1 2 3 $\Rightarrow$ 2 3 4

   1 2 3*2 $\Rightarrow$ 2 4 6

   4/2 $\Rightarrow$ 2

   1 2 3+4 5 6 $\Rightarrow$ 5 7 9

The second group, the relational group, compares the two arguments in the same manner as :o:. If the condition is true, a 1 is placed in the result. If the condition is false, a 0 is placed in the result. Thus, the two logical quantities, true and false, are defined to be 1 and 0 respectively.

Character strings may be compared with each other, but not with number strings. The two modes cannot be mixed. When character strings are used with the relational operation, their character codes are compared, thus, "A" is less than "B". Some examples:

1:EQ:1 2 3 ⇒ 1 0 0

1 2 3:GT:4 1 2 ⇒ 0 1 1

"ABC":NE:"DEC" ⇒ 1 1 0

The logical operators must have for arguments only 1's and 0's. Some examples:

1 1 0&0 1 0 ⇒ 0 1 0

1 0 1 0 ⇒ 1 1 1

The operator # has logical quantities for its left argument, and a number or character string for its right. It functions in the following way: Where there is a 1 in the left arg, the corresponding element in the right arg is retained in the result. Where there is a 0, the corresponding element is not retained in the result. Some examples:

1#1 0 5 8 ⇒ 1 0 5 8

1 0 1 1 1#8 ⇒ 8 8 8 8

1 0 1#1 2 3 ⇒ 1 3

1 2 3:EQ:1 2 4#"ABC" ⇒ "AB"

0#1 ⇒ null vector of numerical mode

SET A=2 2 3 2 4 5 6 2 2

2:EQ:A#A ⇒ 2 2 2 2 2

2:EQ:A#1 2 3 4 5 6 7 8 9 ⇒ 1 2 4 8 9

These are the Vector-8 operators. Vector-8 also offers 32 functions. A function is represented by an "F" followed by the name. A list follows:

Numerical Functions:

FSIN ($d_{n>0}$)	sine. argument in radians.
FCOS($d_{n>0}$)	cosine. argument in radians.
FTAN($d_{n>0}$)	tangent. argument in radians.
FATN($d_{n>0}$)	arc-tangent.
FLOG($d_{n>0}$)	natural logrithim.
FEXP($d_{n>0}$)	exponential. e^x
FABS($d_{n>0}$)	absolute value.
FSGN($d_{n>0}$)	sign function.
FNOT($d_{n>0}$)	logical not function. Argument must contain only 1's and 0's.
FINT($d_{n>0}$)	integer function.
FFRC($d_{n>0}$)	returns fractional part of argument.

Reductive Functions:

FMLT($d_{n>1}$)	reductive multiplication.
FADD($d_{n>1}$)	reductive addition.
FSUB($d_{n>1}$)	reductive subtraction.
FDIV($d_{n>1}$)	reductive division.
FMAX($d_{n>1}$)	returns largest number.
FMIN($d_{n>1}$)	returns smallest number.
FAND($d_{n>1}$)	reductive logical and ($\wedge$).
FIOR($d_{n>1}$)	reductive inclusive or ($\vee$).
FEQ($d_{n>1}$)	reductive :EQ: .
FGT($d_{n>1}$)	reductive :GT: .
FLT($d_{n>1}$)	reductive :LT: .
FGE($d_{n>1}$)	reductive :GE: .
FLE($d_{n>1}$)	reductive :LE: .
FNE($d_{n>1}$)	reductive :NE: .

Other Functions:

FCHN($d_{n>0}$)	changes mode of argument.
FROT($d_{n>2}$)	rotates the argument.
FSER($d_{1\leq n\leq 3}$)	generates a series.
FREP($d_{n>0}$)	repeats the argument.
FDIM($d_{n\geq 0}$)	returns dimension of agrument.
FRAN($d_{n>0}$)	generates n random numbers.
FNEW()	user defined function

Numerical functions have for agruments vectors whose dimensions are greater than 0. The reason being that it doesn't make sense to, for instance, take the sine of a null vector. Each function applies its particular operation to each element of its vector argument, and places in the element operated on the result of the operation. So for example:

FNOT(1 0 0 1 1) ⇒ 0 1 1 0 0

FSIN(1 2 3) ⇒ sin 1 sin 2 sin 3

Each reduction function reduces its vector argument, whose dimension must be 2 or greater, to a single number by applying its operation to the elements of the vector. For example, FMLT applies its operation, multiplication, to produce a result which is equal to all the elements of the argument multiplied together. This process can be visualized by the following: Let a_1, a_2, a_3, a_4, . . . a_n be the arguments of the reductive function whose operation is :q:. Then,

a_1 :q: a_2 :q: a_3 :q: a_4 :q: . . . a_n ⟹ result.

The other functions must be gone into further:

1. FCHN changes the mode of the argument. If the argument is a character string, then the result is a number string whose elements are the codes of the characters. If the argument is a number string, then the result is a character string.

Some examples:

FCHN("ABC") ⟹ 193 194 195

FCHN(193 194 195) ⟹ "ABC"

This function is designed so that functions such as FROT and FREP can be used with character strings.

2. FROT rotates a vector. The last element of the vector argument is used to indicate the direction and number of times that the vector is to be rotated. Call the last element N.

If N=0 the vector is not rotated.

If N>0 the vector is rotated N times right.

If N<0 the vector is rotated -N times left.

Because it is not reasonable to rotate a vector of dimension 1, the dimension of FROT's argument must be 3 or greater. (N plus the vector to rotate.) Some examples:

FROT(1 2 3 1) > 3 1 2 -rotate 1 2 3 once right.
FCHN(FROT(FCHN("ABC"),1)) > "CAB" -
change "ABC". rotate once right. change back.

3. FREP repeats a vector. The last element of the argument, as in FROT, is used to indicate how many times.

If N=0 a null vector is returned.

If N>0 the vector is repeated N times.

If N<0 an error results.

If the dimension of the argument is 1, then a null vector is returned.

Some examples:

FREP(1) ⟹ null vector of numerical mode.

FREP(1 4) ⟹ 1 1 1 1 -repeat 1 four times.

FREP(1 2 2) ⟹ 1 2 1 2 -repeat 1 2 two times.

4. FDIM returns the dimension of the argument. Some examples:

FDIM(1 2) ⟹ 2

FDIM(FREP(1 100)) ⟹ 100

FDIM ("") ⟹ 0

FDIM(0#4) ⟹ 0

FDIM("ABCDEFR") ⟹ 7

5. FSER takes three forms. One with the argument of dimension 1, another with the argument of dimension 2, and the third with the argument of dimension 3.

a. FSER(d_1). Let N be the one element. If N≤0 an error results. Otherwise the series 1,2,3,4,. . . N is generated. Some examples:

FSER(1) ⟹ null vector of numerical mode.

FSER(5) ⟹ 1 2 3 4 5

FSER(10) ⟹ 1 2 3 4 5 6 7 8 9 10

b. FSER(d_2). Let M be the first, and N be the second element. A series is generated, in increments of 1, from M to N. If N<M then a null vector is returned. Some examples:

FSER(1 5) ⟹ 1 2 3 4 5

FSER(3 6) ⟹ 3 4 5 6

FSER(2 1) ⟹ null vector

FSER(1.2 4.2) ⟹ 1.2 2.2 3.2 4.2

FSER(2 2) ⟹ 2

c. FSER(d_3). Let N,M,P be the first, second, and third elements respectively. A series is produced from M to P in increments of N. If M>P then a null vector is returned. If N=0 then an error results. If P>M and N<0 then an error results, for it would be impossible for M to reach P with a negative increment. Some examples:

FSER(1 3 12) ⟹ 1 4 7 10

FSER(1,(-2),10) ⟹ error

FSER(1 0 5) ⟹ error

FSER(.2 .01 .3) ⟹ .20 .21 .22 .23 .24 .25
.26 .27 .28 .29 .30

d. FRAN generates as many random numbers as the dimension of the argument. A null argument is not legal. Some examples:

FRAN(FSER(2)) ⟹ 2 random numbers.

FRAN(FSER(1000)) ⟹ 1000 random numbers.

We have now seen the functions and operators that make up a Vector-8 expression. Now for the commands that make up a Vector-8 program.

1. TYPE EXP;!;#;↑;% (EXP stands for a Vector-8 expression)

!	CR-Lf
#	CR
↑	LF
%	Changes output to the high speed punch (HSP) till the TYPE command is terminated.

If EXP is a character string, then the characters are outputted. If EXP is a number string, then it is outputted according to the parameters specified in the FORMAT command.

2. SET VAR=EXP -create a new variable and set to EXP

   SET VAR(INDEX)=EXP -replace certain elements of VAR with EXP.

3. ASK VAR -create a new variable, VAR. Get values from the TTY.

   ASK VAR(INDEX) -replace certain elements of VAR. Get values from the TTY.

   ASK %VAR -create a new var, VAR. Get values from the HSR.

   ASK %VAR(INDEX) -replace certain elements of VAR. Get values from the HSR.

Note: HSR stands for High Speed Reader.

4. GOTO EXP -EXP must result in a number string. Control is transferred to line EXP(1).

5. GOSUB EXP -EXP must result in a number string. Control is transferred to a subroutine at line EXP(1).

6. RETURN -This command is used to exit a subroutine. Control is transferred to the line following the GOSUB that called the subroutine.

7. CALL PNAME -The program whose name is PNAME is called down from the disk and executed. When the program is done, control is returned to the line following the CALL command in the original program.

8. WHEN EXP;COMMAND -If the result of EXP equals 1, the COMMAND is executed.

   -If the result of EXP equals 0, control is transferred to the next line.

   -If the result of EXP does not equal 1 or 0, then an error results.

9. REM -This command transfers control immediately to the next line and ignores any argument. It is the comment command.

10. RUN -This command runs the program in core.

11. DELETE EXP -EXP must result in a number string. The lines specified by the elements of the number string are deleted.

12. KILL VARNAM -The variable, VARNAM, is deleted.

13. EDIT EXP -EXP must result in a number string. The line EXP(1) is edited in a manner similar to FOCAL's MODIFY command.

14. WRITE EXP -EXP must result in a number string. The lines specified by the result are listed on the teletype. If EXP results in a null vector, the entire program is listed.

    WRITE % EXP -Same as above, except that the listings are produced on the HSP.

15. DUMP -This command dumps the symbol table in the following format:

    NAME DIMENSION MODE(char or number string)

16. SELECT I;O;%O;%I -This command selects I/O devices.

    I - low speed input (teletype)

    O - low speed output (teletype)

    %I - high speed input (HSR)

    %O - high speed output (HSP)

17. FORMAT EXP -EXP must result in a number string of dimension 3. This command supplies format parameters concerning the output of numerical vectors to the TYPE command. Let e_1, e_2, e_3, be the three elements:

    e_1 specifies the number of spaces between elements.

    e_2 specifies the number of elements per line.

    e_3 specifies the number of CR-LF's between lines.

    This command comes in useful, when, for instance, a vector of dimension 200 is being outputted. That many elements can not possibly fit on a line, so the option is left open to the user as to formatting the output.

18. LIB S PNAME -Saves program in core in the library under the name PNAME.

    LIB D PNAME -Deletes the program in the library whose name is PNAME.

These are the Vecotr-8 commands as they are currently defined.

LIB C PNAME -Calls the program whose name is PNAME from the library into core.

LIB L -Produces a listing of all current entires in the library.

19. LOAD ONAM; NNAM -A command or function of the name ONAM is replaced by NNAM, and the binary for the new command or function is read in from the low speed reader and written up onto the appropriate block on the disk.

LOAD ONAM; NNAM; -Same as above, except that the binary is read in from the HSR.

20. QUIT -Program execution is halted, and a return to direct mode is made.

These are the Vector-8 commands as they are currently defined. Programs are created in much the same way that they are in FOCAL. If a line of inputted text begins with a line number, the text is stored away for later reference. If the line doesn't begin with a line number, the text is assumed to be a command, and is executed in direct mode. Thus, Vector-8 can operate as a calculator, as well as execute stored programs.

A line number is a whole number from 0 to 2047 inclusive. Line number 0 may not be deleted, written, or operated on in any way except by a GOTO command. A GOTO line number 0 causes the program to halt, just as if a QUIT command had been executed. This is in effect a computed halt.

Vector-8 operates with 7 significant digits. Floating point output is formatted, but the format cannot be set by the user, for there is not enough core for the lengthy routine that would make this possible.

A more complete and detailed description of Vector-8 will be written during the next few months. This article should serve, though, as an introduction to Vector-8.

PROGRAMMING NOTES

Program to Convert Decibels to Volts and vice-versa (Ødbm= .7746 volts across 600 ohms).

Submitted by: George Kowalski
Ampex Corporation
Colorado Springs, Colorado

FOCAL FRACTION-DECIMAL CONVERSION

Randall S. Battat
55 San Rafael Way
San Francisco, California

I have developed a short program written in FOCAL that will convert proper or improper fractions to decimal. The program is:

```
1.10  ERASE
1.20  ASK "NUMERATOR" N,!
1.30  ASK "DENOMINATOR" D,!
1.40  TYPE D/N,!
1.50  GOTO 1.10
```

To stop this program at any time, type CTRL/C.

MODIFICATION TO DECUS NO. 5/8-29

Steuart G. C. Dewar
University of Chicago
Billings Hospital
Clinical Chemistry Department
Chicago, Illinois

Users of computers with limited core storage may be interested in the following improvement upon a frequently needed utility subroutine. At present the minimal storage subroutine offered for BCD to binary conversion by DECUS (DECUS No. 5/8-29) occupies 29 locations. The following program improves upon this subroutine by reducing the number of necessary locations to 27. A BCD number in the form $D_1D_2D_3$ is converted in a fixed time by evaluating the expression: $(10D_1 + D_2)10 + D_3$:

```
BINIZE,  Ø
         DCA NUMBER
         DCA STORE
         TAD NUMBER
         RTR
         RTR
         JMS MULT
         JMS MULT
         AND MASK17
         TAD STORE
         JMP I BINIZE
MULT,    Ø
         RTR
         RTR
         AND MASK17
         TAD STORE
         CLL RAL
         DCA STORE
         TAD STORE
         RTL
         TAD STORE
         DCA STORE
         TAD NUMBER
         JMP I MULT
NUMBER,  Ø
STORE,   Ø
MASK17,  17
```

ERASE NEGATIVE NUMBERS

W. J. Blanchard
Department of Physics and Astronomy
Louisiana State University
Baton Rouge, Louisiana

In our experimental work in nuclear spectroscopy, we have made extensive use of a Nuclear Data 50/50 system (with PDP-8/L interfacing). The dual parameter system has its principle function in directional correlation studies where background subtraction of coincidence data frequently leave negative numbers in the storage and display memory unit. This is undesirable if data is outputted in six digital BCD form. The following program has been developed to test each storage channel (4K in all). If the data in a particular channel is positive it is unaltered, if the data is negative a zero is put in the storage channel. The program is used as an overlay to the ND basic software package program number ND40-1042.

```
                    /    PG:001  WJB   12/12/69
                    / ERASE NEGATIVE NUMBERS ROUTINE TO BE
                    / USE AS OVERLAY WITH PHYSICS
                    / ANALYZER PROGRAM. CMD = EN THEN ALTMODE ECT....

                    / DEFINITIONS FOR ASSEMBLY

                    PRINT = 4467
                    M10   = 0120
                    MGC   = 0161
                    ICON  = 0164
                    ICOF  = 0163
                    RMLA  = 6457
                    CRMM  = 6443
                    CLMM  = 6341
                    CLML  = 6351
                    EACS  = 6335
                    DELAY = 6770
                    END   = 5471


              *0276
0276  7262            7262            /EN


              *0606
0606  4467  BEGN,    PRINT           /PRINT ERASE
0607  4040            4040
0610  0522            0522
0611  0123            0123
0612  0500            0500
0613  5620            JMP I ISTRT
0614  7000            NOP
0615  7000            NOP
0616  7000            NOP
0617  7000            NOP
0620  5000  ISTRT,    5000


              *5000
5000  4467            PRINT           /PRINT NEG. #'S
5001  4016            4016
5002  0507            0507
5003  0124            0124
5004  1126            1126
5005  0540            0540
5006  1625            1625
5007  1502            1502
5010  0522            0522
5011  2300            2300
5012  7300            CLA CLL
5013  1120            TAD M10
5014  3240            DCA RPT
5015  3161            DCA MGC
5016  4564  EN1,      JMS I ICON
5017  7410            SKP

5020  6457  EN2,      RMLA
5021  7300            CLA CLL
5022  6443            CRMM
5023  7700            SMA CLA         /IS # NEGATIVE
5024  5231            JMP EN3         /NO: PROCEED
5025  6341            CLMM            /YES: ERASE A NUMBER
5026  6351            CLML
5027  6335            EACS
5030  6770            DELAY

5031  7300  EN3,      CLA CLL
5032  2161            ISZ MGC
5033  5220            JMP EN2
5034  4563            JMS I ICOF
5035  2240            ISZ RPT
5036  5216            JMP EN1
5037  5471            END
5040  0000  RPT,      0
```

FOCAL, AMITY

Robert W. Tuttle
Amity Regional Senior High School
Woodbridge, Connecticut

An advanced version of DEC FOCAL '69 has been in use at Amity Regional for approximately six months. The following features have been implemented. (1) A Logical-If-Test; (2) Double Subscripts; (3) Function Subprograms (with multiple arguments); (4) Computed Line-Numbers; (5) Improved FRAN (); (6) Decrementing (negative increment) For-Loop Control; (7) Character String Functions (previously published as FIN() and FOUT()); (8) True (one figure left of decimal) Scientific Notation; (9) Line-feed without carriage for faster plotting; (10) Automatic "Load-and-Go" starting.

All extended functions are still available and no reduction has been made in the user's text buffer. However, this program is strictly for a single-user, 4-K system without Disc or Tape Monitor and a special version of the BIN loader, occupying less than half a page, must be used. Also display control and ADC control have been deleted. Nevertheless, the author has found this version of FOCAL extremely powerful especially for matrix calculations. In addition, students learn to write better organized programs more quickly with the Logical-If-Test. (The standard arithmetic test is still available.)

This version has been submitted to DECUS (DECUS Number FOCAL8-136) and the author welcomes comments from users. A subset of the above options could be easily programmed into FOCAL for a larger hardware configuration.

FORMAT-FREE INPUT FOR FORTRAN

Daniel Graboi
University of California/San Diego
Department of Psychology
La Jolla, California

The information load carried by a computer user may be considerably reduced by using format-free input. With INFREE, rubouts work as usual; a carriage return terminates the entry. INFREE may be called as an integer function or a subprogram. When called as a function, the floating point value is returned in the argument and the rounded fixed point value is returned as the value of the function. The routine assembles alphanumeric data into numeric data. The last statement before RETURN may not be deleted since it sets up the accumulator properly for an integer function exit. INFREE is written in FORTRAN for the PDP-9/15.

Calling INFREE from FORTRAN:

As a function- J=INFREE(X)

As a subprogram- CALL INFREE(X)

Examples of format-free input:

```
-127,  1.38475,  +82,  6,  28.9 COMMENTS
                           ARE IGNORED,
```

SOURCE LISTING

```
C       FILE TITLE  -  INFREE
C
        SUBROUTINE INFREE (A)
        DIMENSION B(10),C(14)
        DATA C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),
        1C(9),C(10),C(11),C(12),C(13),C(14)
        2/5H0    ,5H1    ,5H2    ,5H3    ,5H4    ,5H5    ,
        35H6     ,5H7    ,5H8    ,5H9    ,
        45H-     ,5H+    ,5H.    ,5H     /
        K=0
        READ (5,1) B
C  READ FROM KEYBOARD.   .DAT SLOTS VARY
C   FROM SYSTEM TO SYSTEM.
1       FORMAT (10A1)
        DO 2 I=1,10
        DO 2 J=1,14
        IF (B(I) .EQ. C(J)) B(I)=J
        IF (K .NE. 0) GO TO 2
        IF (B(I) .EQ. 14.) K=I
2       CONTINUE
        A=0.
        X=.1
        DO 5 I=1,10
        X=10.*X
        K=K-1
        IF (K .EQ. 0) GO TO 6
        IF (B(K) .LE. 10.) GO TO 4
        IF (B(K) .NE. 13.) GO TO 3
        A=A/X
        X=.1
        GO TO 5
3       J=B(K)
        IF (J .EQ. 12) GO TO 6
        IF (J .EQ. 11) A=-1. * A
        GO TO 6
4       A=A+(B(K)-1.)*X
5       CONTINUE
6       K=A
        IF (A .GE. 0.) K=A+.5
        K=K
        RETURN
        END
```

DECUS NOTES

AUSTRALIA REGIONAL MEETING

A DECUS Meeting will be held on May 28 and 29 in Australia. Details on the meeting are available by contacting Digital Equipment Australia Pty. Ltd., 75 Alexander Street, Crows Nest, N.S.W. Australia 2065.

PDP-9/15 LOCAL MEETING

A meeting highlighting PDP-9 and PDP-15 applications will be held at Brookhaven National Laboratories, Stony Brook, New York on April 6th. Details are available by contacting Digital Equipment Corporation, 1919 Middle Country Road, Centereach, L. I., New York 11720.

POLICY CHANGE

By vote of Executive Board, the following policy is in effect: "DECUS will no longer publish in DECUSCOPE articles which are deemed to be of a commercial nature.

The Customer Exchange Board will continue on a space-available basis. Items published under the Customer Exchange Board will be limited in text."

MEETING DISPLAYS

DECUS will continue to allow the display of literature at its meetings, provided the literature has been approved (contact DECUS office for procedure). Hardware displays are definitely not allowed.

Correction

On page 18, Column 2, DECUSCOPE Vol. 9, No. 5, the signature to the letter addressed to Miss Ries was erroneously omitted. The letter was sent by Mr. R. P. Lorétan, General Direction of PTT, Speechergasse 6, 3000 Bern, Germany. Our apologies to Mr. Lorétan.

NEW PUBLICATIONS

Fall 1970 Proceedings and the 1970 Canadian Proceedings now available upon request. Fee for non-delegates $5.00 per copy. Copies have been mailed to meeting attendees.

LIBRARY NOTE

As an interim to catalog updates, new programs available from the DECUS Library are being announced as often as possible through flyers to delegates. These flyers include the program number, title, submitter, and material available. The flyers are prepared by computer line, therefore, delegates will only receive the flyer for computers they represent.

Individual members should see their installation delegate for this information or may write to DECUS specifying computer interest.

CUSTOMER EXCHANGE BOARD

For Sale, Rental or Lease

PDP-9, PDP-8's - Contact M. Zimmerman, Telco Leasing, Inc., 301 East Erie Street, Chicago, Illinois 60611.

For Sale

PDP-9, PDP-8's, PDP-1's, Teletype - Contact Delos International Group, Inc., 100 State Street, Boston, Mass. 02109.

Disk File, Controller and Interface for PDP-9, Fixed-frequency clock interrupt for PDP-9, contact - Steven Deller, Northwestern University, The Medical School, Chicago, Illinois 60611.

Time Available

On 4K PDP-8 Installation - Contact S. Fidell, Bolt, Beranek and Newman, Inc., 21120 Vanowen Street, Canoga Park, California 91303.

During the year DECUS has grown at a 50% rate, indicating that the Society is becoming a very significant force in the computer user community. This growth rate is very encouraging, having been achieved in a year of national economic recession. Last year the growth rate was 73% with 2,998 new members. This year's 50% represented 3,147 new members.

The DECUS office in Maynard now has eleven full time employees. There is one more in the Geneva office.

The year 1970 has been a good year and a significant year for DECUS, and all indications are that this will continue.

New DECUS Bylaws

During the year the DECUS delegates approved a new set of bylaws. These bylaws called for radical reorganization of the administration of the Society. The DECUS Executive Board is now structured into two broad areas: Administration and Operating. The Administrative Officers consist of a President, Past President and President-Elect. A successful candidate is elected to be President-Elect for one year, and then moves the following year to be President, and the following year Past President. Being elected President-Elect now implies a three year commitment.

The Operating Officers on the Executive Board are now Standards Chairman, Meetings Chairman, Publications Chairman, and Mainframe Committee Chairmen. In this latter case, the membership votes along Mainframe lines to elect their own chairman of the Committee that is concerned with a particular PDP family. At year end, the PDP-10 users had chosen a Mainframe Chairman with the other product lines close behind.

The offices of Recording Secretary and Equipment Chairman were abolished.

Our capable Executive Secretary, Angela Cossette was promoted to be Executive Director, a title that better approximates her current duties.

The new Executive Board structure should make the DECUS administration more responsive to the membership as DEC Product lines continue to diverge.

Executive Board Changes

Elected members of the new Executive Board are: Richard J. McQuillin, Composition Technology, (President); George S. Cooper, LOGIC, Inc. (President-Elect); John Sauter, Sanders Associates (Standards); Sean O'Donnell, TIME, Inc. (Publications); John Alderman, Georgia Institute of Technology (Meetings); and George Zepko, Stephens Institute of Technology (PDP-10 Mainframe).

Non-Voting members are Angela Cossette, Executive Director, and William Segal and Ken Stone, DEC Representatives.

The European Chairman, John Elder (England) is also a voting member.

The Executive Board feels deep appreciation for faithful services of retiring members. Michael Wolfberg contributed many hours as Programming Chairman. Also, Sypko Andreae worked very hard on organizing the LUGs (Local Users Groups), as Equipment Chairman.

Special appreciation is due to retiring John Jones, the long-time DEC Representative. John served very well for six years despite his busy schedule as PDP-9/15 Product Line Manager. Certainly DECUS would be in a far different stage of development had not we had John working so effectively for DECUS. We wish him well as DEC's Manager in Australia. We expect to see a DECUS/Australia very soon.

DECUS Finances

DECUS has income from the following (net income shown):

1.	Meetings (registrations, display charges, sale of Proceedings)	$ 1,702.96
2.	Program Library (Distribution and handling charges to non-delegates)	8,419.82
3.	Mailing Services (Distribution of advertising for products directly related to member interests)	5,935.47
4.	Program Library Catalog Binders	1,013.55
	Total	$17,071.80

Under terms of our Support Agreement with Digital, we reduce our treasury balance to $1,500 twice a year, 60 days after our meetings. In 1970 we turned back $8,706.00 to Digital to held defray the costs of running DECUS. Incidently, the costs of running DECUS are now about $150,000 per year! Last year $3,500.00 was turned back to Digital.

The Society policy is to run the meetings on a break even basis as closely as possible. This year the Spring meeting in Atlantic City showed a net loss of $204.11. The Fall meeting in Houston, however, had about a $2,100 profit. The Canadian DECUS meeting showed a loss, which DECUS covered to an extent of $224.00. Last year the Canadian DECUS meeting had a loss of $130.00.

In 1970 there was considerable activity in both program library distribution and mailing service. In the program library, the net income is computed at gross income less cost of supplies and mailing. These charges are made only to individual members.

DECUS offers a mailing service to companies that want to promote their products through the DECUS mailing list. In practice a sample must be sent to DECUS beforehand to get approval by the Finance Committee. It is our policy to mail out information that we feel is of interest to the DECUS membership. We also exercise discretion if the products are in direct competition with DEC mainframes. If the advertising is approved, then the company sends the literature to DECUS in bulk, and it is mailed out by the DECUS mailing area. Thus, our mailing list is never released to any of the companies.

The DECUS Program Library Binders continue to be popular. The demand seems to be quite continuous except for increases in sales after they have been on display at the meetings.

During 1970 the DECUS/Europe treasury was established in Switzerland. Charging policies were also established in accordance with established policies in DECUS/U.S. In 1970 the expenses for the European Chairman to attend the DECUS Spring Meeting were paid from the DECUS/U.S. treasury. In 1971 the DECUS/Europe treasury should cover these costs.

Membership Growth

During 1970 the membership growth was 50%. There were 3,147 new applications with a total membership of 9,110. Of these 4,043 are delegates and 5,067 are individual members. The mailing list is 9,110 with some 600 non-members (individuals and institutions) being dropped. Of the 600 non-members, it was decided that maintaining this list was an unnecessary administrative burden. Letters were sent to these people informing them of our intention to drop them from our mailing list, and if they had a desire to maintain ties, they should become individual members.

Program Library

The program library distribution staff has had its busy year. This year there were 333 programs submitted with 13 obsolete. The total number of programs in the library now stands at 874.

As of December 31 the DECUS staff had processed 5,384 request forms with 96 more in process. This amounted to 23,601 programs; meaning 34,504 paper tapes, 1,590 DECtapes, 49 magtapes, and 78 card decks.

In 1970 the FOCAL section of the Program Library Catalog was expanded. During the year DEC implemented the interpretive language on more of their machines so now users can write FOCAL programs on the PDP-8, PDP-9, and PDP-12.

The number of programs submitted to the library increased by 16% from last year. The number of request forms processed showed an increase of 125% from the previous year. The number of programs showed an increase of 103% over 1969.

If one compares the 125% increase in requests with the 50% growth in membership, it would be clear that the program library is becoming more important to the membership.

Meetings

DECUS continues to have well attended meetings although severe restrictions on travel funds had a definite effect. The Spring meeting in Atlantic City had 424 attendees with 54 papers presented. This was understandably down from the 610 in Boston the year before, because the proximity of Maynard made Boston so attractive to both users and DEC employees. The Fall meeting was held in Houston with 313 attendees and 57 papers presented.

The topics of the Spring meeting were PDP-9/15 Operating Systems, PDP-9/15's in a Laboratory Environment, PDP-10 General, User Expansions of FOCAL, Hardware Interfacing Techniques, Biomedicine, PDP-8,12 Operating Systems, Education, Graphics, Cardiology and Patient Monitoring Neurophysiology, Business Applications, Clinical Laboratory Applications, PDP-11 General, and product panels.

The Society was pleased to hear an address by Winston R. Hindle, Jr., Vice President, Digital Equipment Corporation at the banquet.

The Fall Meeting featured a keynote address by Samual M. Keathley, Chief Simulation Branch, NASA Manned Spacecraft Center. Mr. Keathley discussed the role of the minicomputer in the guidance system of the lunar landing module. Topics covered were Industrial and Real-Time Applications, Computers in Physiology, Biomedicine, Education, General Applications, Hardware Interfacing Techniques, Astrophysics, FOCAL Applications, and PDP-8,-9,-10 Applications. The meeting also featured intensive discussions in the product panels.

The most significant point of the 1970 meetings was a definite trend away from the applications papers, and a very fast growing interest of the membership for workshops and intense interaction with the DEC technical people. This is very strongly evidenced in the PDP-10 users. It was found that the allotted time for their workshop was nowhere near enough, so almost continuous meetings were scheduled during the meeting. As DEC supplied system software becomes more and more complex, the users look to the DECUS meetings as the place to discuss their technical problems with the implementers. Indeed, although keen interest is voiced on applications, the users' real need is felt to be in understanding the technical aspects of the system software, and being aware of new developments at DEC in both software and hardware.

DECUS Publications

At year-end the circulation of DECUSCOPE was 10,000 copies. During the year only five issues were published. DECUSCOPE has been under study by the Executive Board during the year. It has been observed that the publication was not fulfilling its original aim of being a newsletter because of infrequency of publication. The users are submitting more and more technical articles for publication in DECUSCOPE. The format of DECUSCOPE will change during 1971. The Technical Journal (see below) will take the technical articles, and hopefully DECUSCOPE will be published every month with much less typographic quality, but with timely news and notices.

The DECUS Proceedings have been published with tight schedules this year. This has been due to good cooperation of the authors in getting their papers in on time, and typed on the DECUS forms properly. Also, the drawings and pictures were submitted in good shape. (Xerox copies of drawing aren't good enough!)

DECUS has reached an agreement with University Microfilms in Ann Arbor, Michigan, to distribute on microfilm (or Xerox full size) all current and back copies of the Proceedings, DECUSCOPE and European Proceedings. Some of these publications are out-of-print from DECUS, so this is the only way they can be obtained. The membership is encouraged to ac-

quire these sets of film from University Microfilm so that company will continue to distribute DECUS publications. Continuing availability depends on requests.

Late in the year the DECUS Executive Board announced the publication of the DECUS Technical Journal. This publication is to be entirely run by users, including setting of publication policy, standards, etc. The Editor-in-Chief was appointed, Professor John Elder of Manchester University, England, with the Associate Editor for North America, Mr. John O'Donnell of TIME, Inc. Mr. O'Donnell is the DECUS Publications Chairman. At year end, the Finance Committee appropriated $2,000 as seed money for the publication and instructed the editorial staff to ask for support from the membership.

It is the intent to make the DECUS Technical Journal a high quality journal containing articles about DEC computers, including applications, techniques, standards, evaluation, education and tutorials. In order to achieve quality it was decided to set up review committees. For this reason the Executive Board felt that the Journal must function as a completely user-run publication. It must be distributed on a subscription fee basis. The first issue is expected to be published mid-1971, with three or four issues per year initially. At year-end the editorial staff was busy setting up the mechanics for publication, including format, style, etc.

International DECUS

As DECUS/Europe becomes more structured, it has become clear that DECUS needs some sort of overall government. While the operation of DECUS/USA and DECUS/Europe are similar, they are also different. Certain functions are performed in Maynard for both organizations, while Geneva handles others. For economic reasons it may be desirable to transfer some functions to Europe that are currently being done in the United States.

We also have the question of what to do when users become active and numerous in other places, such as Canada, Australia, and Asia.

In order to better cope with future problems relating to growth of DECUS in geographical remote areas, a set of DECUS/International bylaws are being prepared. It is expected that this document will go through many revisions, but that it will be approved in 1971.

DECUS/Europe

DECUS/Europe has had an active year. The DECUS office in Geneva has been firmly established, and Martha Ries is busy building up her staff. There has been considerable contact with DECUS in Maynard to help set up DECUS in Europe. We hope that the experience gained in running DECUS/USA can be applied to the problems of setting up DECUS/Europe.

During the year the DECUS/Europe Executive Board developed a set of bylaws. By year-end these bylaws were approved by the European delegates, and a support agreement was signed with DEC/Europe. The European membership stands at about 1,700 up by 700 from last year.

In May the European Chairman, Birger Kvaavik, came to the Spring meeting in Atlantic City. He reported a very active year for the Executive Board so far. They had a meeting in Frankfurt, Germany, in January, and that much had been accomplished at this meeting. He reported that most Europeans, especially in universities, face severe restrictions or international travel, even for relatively short distances. He acknowledged DEC's assistance in funding this travel so the Executive Board could meet.

In September the Sixth Annual European DECUS Seminar was held in Munich, Germany. There were 262 attendees, with 63 papers presented. The DECUS/USA President, Executive Director, and DEC Representative were present. The meeting featured very strong participation from DEC/Maynard technical personnel. Such participation is essential to the success of DECUS/Europe meetings, and the success was evident. The European DECUS members have the same strong desire to hear the latest news from key DEC personnel as their American counterparts. The meeting featured a magnificant cocktail party at the DEC/Munich office, as well as the OKTOBERFEST festival. Next year the meeting is scheduled for Amsterdam. DECUS/USA participation is welcomed.

It was apparent at this meeting that one of the main problems facing DECUS/Europe was the language barrier. The meeting was attended by many who had difficulty with English, and the Board agreed to look into ways of giving speakers better instructions in talking slowly without slang, and effective use of visual aids, etc. The meeting had no parallel paper sessions, and it was agreed that in the next meeting there would be parallel sessions. Like the American meetings, the users wanted to talk about system software with DEC/Maynard technical personnel. This was especially true with the PDP-10 users. DEC/Maynard was very ably represented by Dave Stone in this area.

The new officers of the DECUS/Europe Executive Board are, John Elder, Chairman (England), Chris Malpus (England), Bernard Perrette (France), Hans-Jurgen Trebst (Germany), I. Widegren (Sweden), Myron Myers, DEC Representative (Geneva) and Martha Ries, Executive Secretary (Geneva).

DECUS/Canada

The annual Canadian DECUS meeting was held in September. It was attended by 96 people, with 22 papers presented. John Alderman was the Executive Board representative to the meeting, which was held concurrently with the European DECUS meeting.

The Executive Board has discussed the possibility of setting up a DECUS/Canada along the same lines as DECUS/Europe. Mr. Alderman was instructed to approach the Canadians with this question. It was voted to appoint a steering committee to study the question and report back the next year with recommendations. The Committee was to poll the members and meet with the Executive Board. No further action was taken, except to vote on having a Canadian DECUS meeting the following year.

The Executive Board voted to have the Spring 1972 meeting ing Montreal. By this time the question of whether or not to form an independent DECUS/Canada should be resolved.

Newspaper User Group

At the Executive Board meeting in Houston, the PDP-8 Newspaper Users Group was voted in as a special interest group of DECUS. This group consists of newspaper publishers and their production personnel that use the DEC Typeset-8 system. The group has been functioning independently for several years. They have two meetings a year to discuss solutions to common problems and to foster basic programming knowledge. They plan to establish a program library, and their primary aim of joining DECUS was for our distribution services. The group has 75 newspaper members.

The group will function as a Special User Group within DECUS. Their Chairman and Vice Chairman will meet with the Executive Board to determine how DECUS can serve their membership. Currently their members are mostly operating personnel, unsophisticated in computers. DECUS may be able to sponsor tutorial sessions on computing so that these people can participate more actively in DECUS affairs.

Local User Groups (LUGS)

The Local User Groups continued to flourish within DECUS. By the end of the year there were seventeen LUG's active with four more in various stages of formation. The seventeen are: Berkeley, Stanford, Oklahoma City, Livermore, Florida, Maritime Provinces, East Tennessee, Benelux, Deleware Valley, New England Secondary Schools, PDP-12 Users of New England, New Mexico, LAB-8 Users (Los Angeles Area), Mid-atlantic 6/10 Users, Pittsburgh, Duke University, Midwest.

Some LUG's operate under their own bylaws, and some have their own newsletter. As a matter of policy, the DECUS Executive Board does not actively participate in LUG activities. How a particular LUG functions is a matter of local prerogative.

Joint User Group Activities

DECUS has continued active participation in JUG, the Joint User Group. DECUS President McQuillin was elected to be Chairman of JUG. He is also Chairman of the JUG Program Library Committee.

JUG conducted a Workshop in Houston for Executive Boards of the unit members. About fifteen groups were represented, with DECUS Executive Board members McQuillin, Cooper, Cossette and Alderman present. The Workshop discussed problems facing the user groups, and developed areas of possible communication among the groups.

The JUG Program Library Committee expects to see the JUG Program Directory in print in 1971 after a long delay. The Directory will contain about 1,000 programs in its first issue, with new editions planned for every three to six months. DECUS members will be able to get program documentation about programs in other user group libraries by sending in the appropriate form to the DECUS office. The purpose of the Directory, beyond the documentation of information, will be to raise the standards of program documentation. We expect participation by most of the computer user groups in the country.

Prospects for Next Year

The year 1971 will be the tenth anniversary of the founding of DECUS. The Executive Board is currently looking into plans to celebrate this event.

Next year should be a very eventful one for DECUS. The meetings are being restructured to reflect the members' growing intent to have seminars and workshops along hardware lines. The DECUS publications will undergo significant changes with the emergence of the Technical Journal and the publication of DECUSCOPE as a regular newsletter. We expect the Standards Committee to be very active with the growing communication with other users through JUG.

Next year the membership will pass the 10,000 mark. Such a fast growing membership will present challenges to the Executive Board. DECUS is becoming a large Society.

Respectfully Submitted,

Richard J. McQuillin
President

LETTERS

"Dear Mrs. Cossette:

"I am enclosing a MACRO-9 subroutine (symbolic and expanded) and an example FORTRAN IV program for tabbing control in FORTRAN IV programs. You may wish to publish these as Programming Notes for the PDP-9.

"The subroutine may be of interest to the user of PDP-9 with a KSR-35 teletype for the output from FORTRAN IV programs. It allows the programmer to use the horizontal and vertical tabbing facilities of the teletype for faster arrangement of page. The program is initialized with the instruction

CALL TAB 35 (HT,VT)

where:

TAB 35 is the subroutine called,

HT is the variable for the horizontal tab,

VT is the variable for the vertical tab.

"After that the tabbing is called with an A1 format and may appear anywhere in the format statement after the vertical spacing (first) character.

"The FORTRAN IV program shows how to find the tabbing set-up of the teletype.

"Hoping these will be of some value to other PDP-9 users,

I remain,

N. M. Sandler
Atomic Energy of Canada, Ltd.
Chalk River Nuclear Laboratories
Chalk River, Ontario, Canada"

```
TAB35     PAGE  1

                                        .TITLE  TAB35
                                        .GLOBL  TAB35,..DA
                                   /
                                        .DEFINE  TAB,TABS,ADDR
                                        LAC TABS          /MOVE THE TAB
                                        DAC* ADDR         /TO THE F4 WORD
                                        ISZ ADDR          /2 WORDS
                                        DZM* ADDR         /BOTH SET
                                        .ENDM
                                   /
     00000  R  000002  A        TAB35   0         /TAB SET-UP SUBROUTINE
     00001  R  120020  E                JMS* ..DA         /CALL ADDRESS
     00002  R  600005  R                JMP .+2+1
     00003  R  000000  A        ADDRH   .DSA 0  /HORIZONTAL TAB ADDRESS
     00004  R  000000  A        ADDRV   .DSA 0  /VERTICAL TAB ADDRESS
                                        TAB TABH,ADDRH  /HORIZONTAL
     00005  R  200016  R GEN*           LAC TABH
     00006  R  060003  R GEN*           DAC* ADDRH
     00007  R  440003  R GEN*           ISZ ADDRH
     00010  R  160003  R GEN*           DZM* ADDRH
                                        TAB TABV,ADDRV  /VERTICAL
     00011  R  200017  R GEN*           LAC TABV
     00012  R  060004  R GEN*           DAC* ADDRV
     00013  R  440004  R GEN*           ISZ ADDRV
     00014  R  160004  R GEN*           DZM* ADDRV
     00015  R  620000  R                JMP* TAB35        /RETURN
                                   /
     00016  R  044000  A        TABH    044000  /HORIZ TAB "A" FORMAT
     00017  R  054000  A        TABV    054000  /VERT TAB "A" FORMAT
                000000  A                .END
     00020  R  000020  E *ETV
                                   NO ERROR LINES

MACRO V5D

>
```

```
  ↑P
C       TO FIND THE TABBING & PAGING SET-UP ON A KSR-35 TELETYPE
C       CODE NAME -- TESTAB
C       N.M. SANDLER
C       SPECIAL PROJECTS DIVISION
C       ATOMIC ENERGY OF CANADA LIMITED
C       CHALK RIVER, ONTARIO, CANADA
        DIMENSION N(12)
100     FORMAT (1H1,72I1)
101     FORMAT (1H ,9X,63I1)
102     FORMAT (I3)
103     FORMAT (1H1,A1,9(A1,I1))
104     FORMAT (1H ,2A1,I2)
        CALL TAB35(TABH,TABV)
        DO 200 K=1,10
200     N(K)=K-1
        WRITE (1,100) (I,I=1,9),((N(K),J=1,12),K=2,7),7,7,7
        WRITE (1,101) ((N(K),K=1,10),I=1,6),(N(K),K=1,3)
        DO 201 J=3,66
201     WRITE (1,102) J
        READ (1,102)
C   THIS DELAY IS TO ALLOW YOU TO MOVE THE PAPER BACK TO THE PAGE BEFORE
C   THE OUTPUT DISPLAY PAGE
C   MAKE SURE THE LINE-FEED GEARS ARE MESHED PROPERLY
C   PRESS RETURN TO CONTINUE
        WRITE (1,103) TABV,(TABH,J,J=1,9)
        DO 202 J=2,17
202     WRITE (1,104) TABV,TABH,J
        STOP
        END

F4Q V10A
>
```

```
                .TITLE  TAB35
                .GLOBL  TAB35,..DA
        /
                .DEFINE  TAB,TABS,ADDR
                LAC TABS        /MOVE THE TAB
                DAC* ADDR       /TO THE F4 WORD
                ISZ ADDR        /2 WORDS
                DZM* ADDR       /BOTH SET
                .ENDM
        /
TAB35           0               /TAB SET-UP SUBROUTINE
                JMS* ..DA       /CALL ADDRESS
                JMP .+2+1
ADDRH           .DSA 0          /HORIZONTAL TAB ADDRESS
ADDRV           .DSA 0          /VERTICAL TAB ADDRESS
                TAB TABH,ADDRH          /HORIZONTAL
                TAB TABV,ADDRV          /VERTICAL
                JMP* TAB35      /RETURN
        /
TABH            044000          /HORIZ TAB "A" FORMAT
TABV            054000          /VERT TAB "A" FORMAT
                .END
```

WANTED TO BUY

All or part of the following:

PDP-8 or PDP-8/I CPU
Extended memory (one field)
EAE
CRT controller 34D, VC8/I, or 30N
DF32 disc
DS32 disc

Contact: Mr. Garth Peterson
Institute of Atmospheric Sciences
South Dakota School of Mines and Technology
Rapid City, South Dakota 57701

PROGRAMS AVAILABLE FROM AUTHOR

COMPUTER - LAB-8

Title: DISK EDITOR WITH VIEW FOR LAB-8

Author: K. W. Ranatunga
University of Bristol
Department of Physiology
The Medical School
Bristol, 8, England

Disk Editor (DEC-D8-ESAB-PB, 1968) has been modified slightly so that a 'V' (view) command made via the teletype is recognized. This command is like a 'L' (list) command except that the requested line of the text buffer is displayed on a CRO screen along with the 17 succeeding lines. Further, the reference numbers of these lines as given by the Editor are also displayed.

Figure 1 is a photograph of a display obtained with a 77V command. ASCII characters are displayed in a new 9 x 7 format. Each line is terminated when a CARRTN/LNFEED is encountered. However, a line may be displayed in more than one row if it is longer than 27 characters (see line 77 in figure 1).

For each view command the corresponding display is issued only once, and thus the display should be stored on a storage CRO screen.

Program Description

"Disk Editor with View" consists of the Disk Editor, and two other sub-programs. One of the sub-programs (= MOD) is resident in core with the Editor, and the other (= VIEW) is stored on the disk.

Following a 'V' command, MOD writes a portion of the Editor (400-1777) on the disk and reads View into the same area in core. View is executed once, and the Editor is restored subsequently.

When started, View issues a pulse via the AXØ8 (Relay 1 ÷ option XR) and waits until the RC clock flag is set by an External pulse (1). This arrangement has been used to erase the screen of the storage oscilloscope prior to a display. An oscillator (Digitimer, DEVICES) is triggered by the relay pulse. One pulse from the oscillator is used to erase the screen, and another, issued after an appropriate delay, is used to set the RC clock flag via the EXTERNAL. View proceeds to display the selected lines of the Editor Buffer when the RC clock flag is set. This loop may be avoided by replacing the contents of the location 14Ø5 of View with a NOP.

Loading & Saving

MOD should be loaded with the Disk Editor. When loading and saving Disk Editor + MOD the teletype printout will take approximately the following format:

```
÷ LOAD↵
* IN - T:, T:↵   (Two tapes)
*
*
*
* OPT - 2        (MOD is in page 74ØØ)
*ST= ↑↑↑↑↑
. SAVE EDIT ! Ø-3177, 74ØØ; 26ØØ↵
```

View could be loaded in one pass mode, and may be saved as a system program (2) by the following command.

```
. SAVE VIEW ! 4ØØ-1777; 4ØØ.
```

Prior to using the Disk Editor, View should be called in (÷ VIEW), or loaded and started at 400. This operation will write View on the track 15 of the disk.

Symbolic listings, binary tapes, & ASCII tapes of MOD & VIEW are available.

Hardware Requirements

PDP-8/I, Laboratory Peripheral type AXØ8 with option XR & a storage oscilloscope, Disk Fike (DF 32).

REFERENCES

1. AXØ8 LABORATORY PERIPHERAL (1968) Instruction Manual (DEC, Maynard, Mass.)

2. PDP-8/I DISK MONITOR SYSTEM (1969) Programmers Reference Manual (DEC Maynard, Mass.)

```
77 /A LAB-8 PROGRAM TO DISPLAY
            ASCII CHARACS. IN A
            9 X 7 FORMAT
78 /FOLLOWING ARE RECOGNISED
79 /1234567890 :-QWERTYUIOP
80 /ASDFGHJKL ;ZXCVBNM, .!" $%
81 /&'( )*=< >+, SPACE, TAB
82 /LNFEED, CARRIAGE RETURN
83
84 *400
85 FILTER, HLT
86      TAD VERTIC   /Y AXS, -9
87      DCA COUNT1
88      TAD HORIZN   /X AXS, -7
89      DCA COUNT2
90      TAD ADDRES   /GET ADDR
91      DCA LOCN1
92      TAD XCONST   /SET SCALE
```

FIGURE 1

A display of a section of the Disk Editor buffer, obtained by using the Disk Editor with View.

COMPUTER - PDP-15 (9)

Title: FFI - Free Format Input for FORTRAN Programs

Author: Clayton Hull, Aeronomy Laboratory, Electrical Engineering Department, University of Illinois, Urbana, Illinois 61801

This program allows data input from the teletype without format restrictions. The data types handled are signed or unsigned integers, real numbers entered with or without decimal points or exponents, and ASCII strings of five characters or less. All data items are separated by a comma, tab, carriage return, or one or more spaces. Leading and trailing spaces are allowed.

Storage Required:	660 Words (Octal) (567 Words if string facility is removed)
External Calls:	FIOPS, Integer Arithmetic, Real Arithmetic

Title: SUBRG - Object Time Subscript Range Check

SUBRG is intended for inclusion with user programs on the .DAT -4 device for linking loader input. It replaces the FORTRAN OTS routine .SS (Calculate array element address). When zero or negative subscripts are used, or the element referenced is outside the bounds of the array, a diagnostic message is output to the teletype. The program is useful in debugging FORTRAN programs.

Storage Required:	123 Words (Octal)
External Calls:	SPMSG, Integer Arithmetic

Write-ups and paper tapes (source and binary) of the above programs are available on request from the author.

COMPUTER - PDP-7,-8,-9,-15, TSS/8

Title: "MOO" or "BULLS and COWS"

Available from: L. Johnston
Phoenix Services
56a High Street
Barkingside, Ilford, Essex, England

MOO or BULLS and COWS, a computer game, which involves the user in evaluating a randomly chosen string of digits from minimal information supplied by the machine, is useful for testing random number generators. At present, versions for the PDP-7,-8,-9,-15, and TSS/8 are available.

The Game

At the end of a game (and when first started) the computer chooses a series of four random digits between Ø and 9. These are remembered between guesses. At each turn the user may guess at the string of four digits. The following rules apply:-

(a) the four digits are different, and must be quoted as such;

(b) a correct digit in the correct place is a BULL (B);

(c) a correct digit in the wrong place is a COW (C);

(d) the game continues until the user gives up (?) or guesses the number.

Example

```
READY

3162
BC

READY

1065
BBB

READY

1069
BBC

READY

1965
BBBB - END OF GAME IN 4 ATTEMPTS
```

The user may request for the score to be typed out by typing one of W, G, M or I. W gives a weighted average on which comparative scores are made. This is defined as:

$$\text{WT. av.} = \frac{(1 + 2 * \text{moves to date})}{(20 + 2 * \text{games to date})}$$

R prints the rules and E sets scores to zero.

Details are available by writing to the address above. There is a nominal charge of $25.00 (£ 10 sterling) to cover expenses.

COMPUTER - LINC-8

Author: A. Cleary
The University of Newcastle Upon Tyne
Department of Psychology
7 & 8 Sydenham Terrace
Newcastle Upon Tyne
NE17TU
England

"I have available two collections of programs which may be of interest to DECUS members. I expect to submit the programs to the DECUS library in due course.

1. FOCAL Statistics Programs

"The following 16 programs are available together with a comprehensive write-up giving sample printout for typical behavioral science applications. The programs and write-up

were intended for use by psychology students. LINC-8 users may request the whole collection of programs to be supplied on tape by sending a tape marked with 128 word blocks (LEAP-1 XMARK). Otherwise the listings are given in the write-up.

Programs on the tape are:

1-SAM for mean, variance, standard deviation, standard error and degrees of freedom

2-SAM for means, variances, standard deviations, standard errors, difference between means, t for independent samples, F ratio and degrees of freedom

CORELT for t-test for related samples

SIGMAX for sums of squares, sums of scores and correction factor preliminary to normal analysis of variance, also means and variances

AV1 for one way analysis of variance (completely randomized design) using data processed by SX, also gives eta.

SHEFFE for examining differences between means after a one way analysis of variance (Scheffe test)

AV2FAC for two way analysis of variance (factorial design; two factors) using data processed by SX

AV2RM for two way analysis (treatments-by-subjects or repeated measures design) from raw scores

RANK for ranking raw scores

PRODMO for Pearson product-moment correlation, z or t-test, regression of Y on X and X on Y

SPEAR for Spearman rank-order correlation

CHISQ for chi square test on frequency data. Tests for differences and gives degree of relationship

MANWIT for Mann-Whitney U-test for differences between independent samples

WILCOX for Wilcoxon test for differences between related samples

SX for cell totals and uncorrected total sum of squares preliminary to analysis of variance programs

LINFIT for a least squares line of best fit to a set of data points, also gives standard error of the estimate and the calculated values of Y for the X values input

2. LINFOC or LINC-8 FOCAL

LINFOC is a version of DEC FOCAL which is used in this laboratory to teach students the use of computers in controlling on line experiments. Although LINFOC is slow, it is extremely useful for many experiments and for testing interfaces to the LINC-8.

LINFOC incorporates functions to operate the LINCscope, read the ADC's and external level lines and load the relay register.

An improved pseudorandom number generator has been incorporated and functions for single character input/output. Versions incorporating common data storage functions are available, in which data may be stored on LINCtape and swapped between programs. A new command, OPERATE, is included which simplifies the formulation of statements to operate the LINCscope, relay register and common data storage functions.

Versions of standard and common storage LINFOC will be supplied to those sending a marked LINCtape. A write-up describing in detail the use of LINFOC is also available. This was written primarily for students in this department. Listings of the patches are also available.

BITS AND PIECES

PUBLICATION AVAILABLE

FOCAL Programming Self-Instruction Manual (150 pages) covers all languages elements, subscripting, alphanumeric variable values, and graphing. Linear programmed instruction format interlaced with programming assignments (answers included). Suitable for independent learning, grade 7 to adult. Single copy $5.00. Quantity discounts available. Contact: Robert N. Haven, Project LOCAL, Inc., 44 School Street, Westwood, Massachusetts 02090.

CODING SHEETS NOW AVAILABLE

I have developed four coding sheets for PDP computers. Two of them are for ALGOL, BASIC, FOCAL, and FORTRAN. The other two are for the languages PAL III, PAL-D, MACRO-8/9 and other assembly languages.

The four coding sheets include two rough drafts and two final copies.

Only 10 can be ordered per one month period free of charge. For additional copies a charge of 25¢ per copy is added.

If interested contact:

Ronald S. Battat
55 San Rafael Way
San Francisco, CA 94127

WANTED

Volunteers to comment LISP (DECUS No. 8-102a) listing from Dutch to English.

Also, a volunteer to adapt existing LISP program for PS/8 Operating Systems.

Contact:

Ferne Halley
DECUS Program Library
DECUS
146 Main Street
Maynard, Massachusetts 01754

WANTED - INFORMATION

Any PDP-12 or LINC-8 user who has done any work in coherence or transfer functions or cross correlation analysis either frequency or time domain, please contact:

Dick Wilson
Barrow Neurological Institute
Department of Neurobiology
350 W. Thomas Road
Phoenix, Arizona 85013

DID YOU KNOW?

There are two libraries at Digital Equipment Corporation and each has a separate operating function.

DEC Program Library is a distribution center for Digital authorized and maintained software. Programs distributed with DEC machines make their home here.

DECUS Program Library is an organization which distributes programs written by users which are not maintained by DEC.

MAINDEC CHANGE NOTICES

Users interested in receiving a list of Maindec Change Notices should complete the reply card below and return it to DECUS. Amount of interest expressed will determine the generation of the list.

Fold and Staple

I would like to receive a list of Maindec Change Notices

Name _____

Organization _____

Address _____

_____

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

DECUS

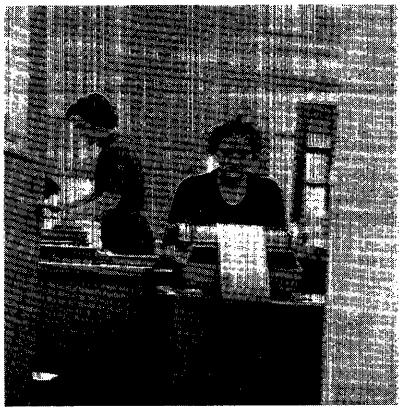
DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
MAYNARD, MASSACHUSETTS 01754

6 Pat Davies
PDP-11/FOCAL8 Library Controller

7 Norma Newsham
Techical Typist

8 Diana Lindorfer
Membership/Mailings

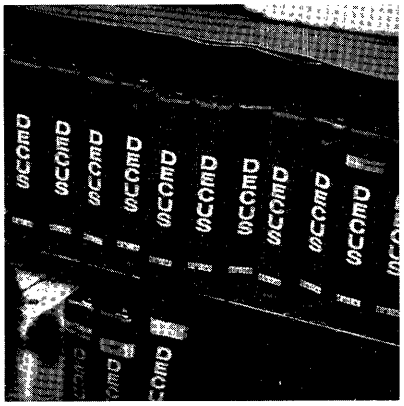
9 Anna del Giudice
Membership/Mailings



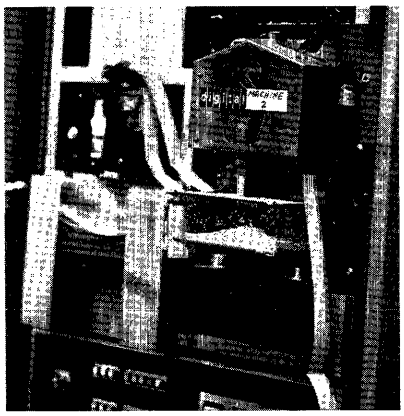
7 6

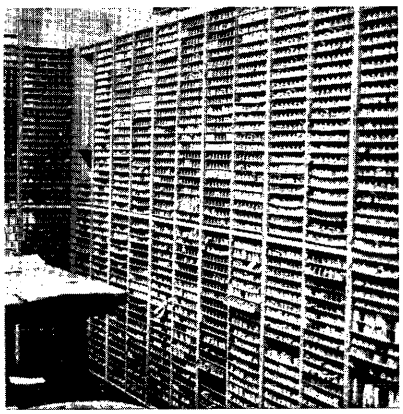
8

9



11





10

12

10 Steven Oskirko
Mail/Stock Clerk

11 Jackie Page
Tape Duplication/Mailings

12 Karen King
Accounting