DECUSCOPE

INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

Best Wishes for the New Year

A MESSAGE FROM YOUR DECUS PRESIDENT

It is a great personal pleasure to begin my term as DECUS President by wishing all DECUS members and DECUSCOPE readers a very happy and successful New Year.

DECUS was little more than a good idea a few years ago and now through the action of the PDP community and active DECUS officials, DECUS is a successful, growing, users group. A number of difficult problems have been overcome during the past two years and the organization can look forward to 1964 as a year of continued achievement.

In this message, I would like to summarize some of our plans for the year and ask you to come forward with your own suggestions and comments.

Program Exchange and Review

Of course, the major function of a users group is to promote communication among its members and every action that we take must be evaluated from this point of view. One form of this communication is through the mutual exchange of operating programs. DECUS has done this very well in the past and now that procedures and techniques have been standardized this activity will continue and proceed more rapidly in the future. But your help is needed if this activity is to be efficient.

First, programs must be donated to the

PDP-4 SYMPOSIUM ON PROCESS CONTROL

Place: The Foxboro Company
21 Strathmore Road
Natick, Massachusetts

Time: Tuesday, January 21, 1964, 9:45 a.m.

9:45 Greetings by the Foxboro Company

10:00 THE KEYDATA ON-LINE PROCESSOR (KOP-2)
Charles W. Adams, President, Charles W. Adams Associates

An executive routine for time sharing a PDP-4 among dozens of teletypewriters, each under the control of any one of several "logic tables" - KOP language programs describing the applications for which the various teletypes are to be used.

10:30 APPLICATION OF PDP-4 TO MEDICAL RESEARCH
Stephen Lorch, Massachusetts General Hospital

A discussion of the application of the PDP-4 in analyzing neurophysiological data, x-ray diffraction of bone crystal and simulation of radio-isotope brain scans.

11:00 APPLICATION OF PDP-4 TO A DIRECT DIGITALLY CONTROLLED BATCHING PROCESS

Part 1: Saul Dinman, The Foxboro Company

The organization and operation of the digital equipment.

Part 2: Allen Rousseau, Charles W. Adams Associates

The design of the real time program.

12:00 PDP-4 AS A CONTROL ELEMENT FOR THE AUTOMATIC MODULE TEST SYSTEM
Russell Winslow, Digital Equipment Corporation

The talk will describe the automatic module test system showing what the program does and the advantages of it in financial, personnel and time savings; also its flexibility in making engineering additions.

12:30 LUNCH

1:30 THE PDP-4 FORTRAN ALGEBRAIC STATEMENT COMPILER
Stephen Piner, Digital Equipment Corporation

A brief description of the techniques used in FORTRAN for the compilation of algebraic statements.

2:00 DECSYS, AN INTEGRATED PROGRAMMING SYSTEM FOR THE PDP-4
David Fellows, Digital Equipment Corporation

A programming system is described which provides job-processing capability for the PDP-4 in its standard configuration; 8K memory, 57A magnetic tape control and 2 Type 50 magnetic tape units.

library. Note, however, that a program which is not adequately documented is of little use to other members.

Secondly, the program must be checked and reviewed by the party that requests it. This means that if you request a program that has not yet been certified, it becomes your responsibility to review it as quickly as possible and send your comments to DECUS for certification by the Programming Committee. Programs are certified by the Programming Committee after review by two or more users.

Finally, the person who donates a program should take on the responsibility of maintaining it.

It has been my experience that the author-reviewer link weakens with time, so it is desirable to get programs reviewed and certified as soon as possible. Everybody benefits when these three phases are handled quickly by the people involved.

DECALRITHMS

The recent addition of DECAL-BBN to the list of useful PDP-1 compilers opens the way to a new form of communication. Since DECAL-BBN is a dialect of the Algol 60 language, we can communicate small routines, algorithms, instruction generators and action operators by means of the printed page rather than via the exchange of paper tapes. Recently, William Fletcher of BBN-LA was able to take a standard ACM algorithm and compile it directly on the PDP-1 within a few minutes.

This suggests that the pages of DECUSCOPE can be used to publish "Decalrithms" in the future. To start the ball rolling, we have published a small example of this type of information exchange elsewhere in this issue of DECUSCOPE. I hope that you will find this method of communication useful and will contribute to it whenever possible.

Technical Meetings and Symposia

Another form of technical communication is through meetings. DECUS sponsors a Spring and Fall Technical Meeting each year which are invariably overwhelming successes. In addition there are occasional local symposia, such as the forthcoming January 21 symposium on the PDP-4 at the Foxboro Company in Natick, Mass. I

PDP-4 SYMPOSIUM

2:30 THE PDP-4 FORTRAN LIST PROCESSOR (FLIP)
Stephen Piner, Digital Equipment Corporation

A description of a new program to facilitate list processing on the PDP-4. The language will be discussed in general and a sample program given in detail.

3:00 THE PDP-4 IN AN INDUSTRIAL PROCESS CONTROL SYSTEM
David McAvinn, The Foxboro Company

This talk will describe the elements of a general purpose executive routine for process monitoring and control.

3:30 DEMONSTRATION OF THE FOXBORO SYSTEM

Be sure you notify Mr. McAvinn, OL 3-5660, if you plan to attend.

LIBRARY CATALOG ADDITIONS (PDP-1)

The following PDP-1 programs have been added to the DECUS Program Library in December 1963. Address requests for programs to the DECUS Secretary.

BBN 121a Floating Point Interpreter Modified for DECAL-BBN — DECUS No. 63

This floating point interpreter program is modified to handle the new multiply and divide instruction generators in DECAL-BBN, and also to handle conditional statements involving floating point variables.

BBN-23 Random, Randmodn, Coin (DECAL-FRAP) — DECUS No. 64

Random, when called, will return with a pseudo random number in the accumulator, the routine is initialized by depositing a known number in two preassigned registers. The routine occupies 12_8 registers.

Ranmodn, if called with a mod number in the accumulator, will return with a random number in the accumulator in in-out register which is between zero and the mod number minus one. It uses random and idvd.

Coin, when called, has a number in the accumulator which will be considered by the routine as a binary fraction with the radix at zero. If, at call time, Coin is at L then the routine will return to L+ 1 with a probability of N and L+ 2 with a probability of 1-N. It occupies 12_8 registers and uses random.

This program was announced in August 1962 but never distributed.

UAC-14 Digital Function Generate and Display — DECUS No. 65*

*Write-up of DECUS Program No. 65 is added as an insert to this DECUSCOPE to serve as an example for users wishing to submit programs to the DECUS Library.

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY

PROGRAM LIBRARY | DECUS No. 65

Title: UAC-14 Digital Function Generate and Display

Authors: S. Jackson, G. Paquette, United Aircraft Corporation

Date: September, 1963

Purpose: Scope display for univariant or bivariant functions stored in UAC-10 (DECUS No. 34) format.

Options include:

1. Typewriter command for first data location, number of points per curve, and number of curves per function.
2. Sense switch control for point, curve, calibrated axes, multi or single sequenced curve displays.
3. Light pen modification or generation of functions.

Locations:

0-15 Loader (UAC-9)
16-502 Character Display (BBN-37)
503-1510 Program

Start: 506

Stops: 506 Program ready to receive typewriter command for new function after continue.

Sense Switches:

1 off - plot function values only.
on - fill space between function values with straight line segments.

2 off - axes not plotted.
on - axes plotted with data.

3 off - plot all curves of bivariate.
on - plot single curve of family. First curve to appear when switch turned on will be first curve in data sequence.

4 off - no affect
on - step to next sequential curve if sense switch 3 is on. This switch must be turned off and on again before subsequent curve can be displayed.

5 off - no affect
on - return to 506 stop.

6 off - permit light pen pick up of existing function value
on - alter function ordinate in memory to value provided by light pen. Function abscissa unaffected. Inhibits light pen pick up.

Description: UAC-14 permits generation and/or display of any function stored above 1510_8 in UAC-10 format.

When loaded, UAC-14 will stop at 506. Pressing continue will allow the program to interrogate the user for function first octal location, decimal number of points per function, and decimal number of curves per function. Each request is followed by a tab after which the user types the requested information and a carriage return. Scope display will follow the last carriage return.

Sense switches 1-4 are used to control the display format. With all switches off, all function values will be displayed with values of each curve equispaced across the face of the scope. Sense switch 1 on will produce straight line segements between points.

Sense switch 2 on will add axes to the display. The vertical axis contains 5 volt least divisions in analog scaling (s7/10)* with each 25v division emphasized and each 50v interval numbered. The horizontal axis shows a point at each function abscissa location with every fourth point emphasized.

Sense switches 3 and 4 are used to view single curves of a family. The first sequential curve will appear when sense switch 3 is turned on. Stepping through the curves is permitted on at a time by turning sense switch 4 on and off. If sense switch 4 is turned on when the last curve is being displayed, the step will be made back to the first curve in the sequence.

Sense switch 5 is used to terminate the program at its loading stop, 506.

Light pen generation or modification is provided. In generating new functions, use of single curve option is required. The light pen will pick up function values only, regardless of options displayed. When the light pen "sees" a point, the function data and axes will be blanked as they normally would appear. In place, a cross at the light pen position and the vertical axis described above will appear. The vertical axis remains stationary passing through the abscissa of the selected function point. The cross will move horizontally or vertically with the light pen until the desired new function value is reached. The horizontal motion is permitted to aide in viewing the cross horizontal bar relative to the vertical axis.

The selected function ordinate can be changed to the position of the light pen by turning sense switch 6 on. Light pen pick up is inhibited when switch 6 is on to prevent "slipping" of the desired value.

Write DECUS Program Library, Digital Equipment Corporation, Maynard, Massachusetts for binary operational tape and macro symbolic tape.

*Sign plus 7 bits left of binary point and 10 bits right of binary point.

ATTENTION PROGRAMMERS
Send DECUSCOPE your favorite quick trick or other routine.

This write-up is published as an insert to the December 1963 DECUSCOPE.

DECUS PROGRAM LIBRARY

COMMENTS from Programming Chairman, Dick McQuillin, BBN

The new year, 1964, finds a much larger DECUS organization than ever and a rapidly expanding Program Library. We look forward to a significant increase in user requests.

The distribution of large systems like DECAL-BBN and the AFCRL Matrix Package puts considerable strain on the DECUS-DEC reproduction system, and delays have resulted. We hope to be able to eliminate these delays soon.

Users are urged to consider the DECAL-BBN system to fill their compiler needs. It is a fairly sophisticated system to learn, but I think that once the effort has been made, the user will be repaid in having a very powerful compiler at his disposal. Incidentally, if there are questions concerning the DECAL-BBN system, they may be directed to me, and I will try to answer them individually and have them printed in DECUSCOPE if they are of interest to the group. I am interested in getting feedback from the users both on the documentation and on the system itself.

Re: Floating Point Routines (BBN 121) - DECUS 10. When one tries to run programs under floating point interpretation that have been compiled with DECAL-BBN, one finds that they don't work. This is because the interpreter has to be changed to handle the new instruction generators for multiply and divide. I am submitting a floating point interpreter that will work in the new system. This is called DECUS 66. It does not need the imp and idv routine anymore (for machines with automatic multiply and divide), and it will handle conditional statements with floating point variables:

if a > b then goto c.

(Note: The = sign has no meaning in floating point,

if a = b then goto c.)

In DECAL-BBN, by the way, the mathematical symbol instruction generators (sin, cos, exp, etc.) are defined, but fip and fop are not. The user may define them as follows:

```
fip dig beg lv7 rs1 nlc
        jsp  ths  lst  end

fop dig beg lv7 op1
      jsp  ths  lst  end
```

Suggestions

I would suggest that the users adopt DECUS 47, the OAL 18-18 arithmetic package to replace the corresponding package in DECUS 10. All in all, DECUS 10 is obsolete and needs to be updated. The mathematical routines (sin, cos, etc.) are much slower than need be. New ones have been written, but have not been checked.

It would be advisable to have an article in DECUSCOPE every month from a user, describing work on new programs going on in his installation. I am trying to contact everyone, a few at a time, but do not wait till you hear from me. Write today.

Richard McQuillin
Programming Committee
Chairman, DECUS
Maynard, Massachusetts

A MESSAGE FROM YOUR DECUS PRESIDENT Continued

hope that 1964 will see an increasing number of the local meetings which give people in a community a chance to meet and exchange ideas. If you would like to set up such a local meeting in your area with DECUS help, you can contact Joseph Lundy, our Meetings Chairman at Inforonics, Inc., Maynard, Massachusetts.

Join JUG?

Finally there is a third form of technical communication which we are beginning to explore. This is communication with other users groups. An organization has been set up within the ACM called the Joint Users Group (JUG). JUG's objective is ". . .the establishment of communications among digital computer user groups to promote study, exchange of information and cooperative effort in areas of common interest." At last count thirteen different users groups, including SHARE, POOL (LGP-30), H-800, UUA (Univac) and GET (GE-225), were included in the JUG membership. There is a motion before our executive board to investigate the possibility and desirability of joining JUG and some action may be taken at the next executive board meeting. In the meantime, I hope you will let me know of your reaction to the proposal.

The DECUS Delegate and Executive Board

The DECUS Delegate's responsibility is to act as a central information coordinator between his installation and DECUS. He requests and distributes programs, and helps his organization play an active role in DECUS. See June 1963 DECUSCOPE for a general description of a DECUS Delegate. In the coming months we hope to introduce Delegates from various member institutions through the pages of DECUSCOPE.

Well, there you have some of our plans for 1964. We have an active group of participants on the DECUS Executive Board: (Elsa Newman, Bill Fletcher, Dick McQuillin, Joe Lundy and Ed Fredkin) and an eager membership. It looks like a good year ahead.

Lewis C. Clapp
DECUS President

ATTENTION ALL DELEGATES

You will be invited to a Delegates Get - together on February 7, 1964.

Please keep the "high-tea" hours (5-7 p.m.) open for DECUS.

New DECUS Delegates

PDP-1

Donald Sordillo
Information International Inc.
Maynard, Massachusetts

PDP-5

Dave Stackpole
Power Control Division
Westinghouse
Pittsburg, Pennsylvania

Lt. Robert M. O'Hagan
U.S. Coast Guard

EDITOR'S NOTE

Congratulations to United Aircraft Corporation's S. Jackson and G. Paquette for submitting programs and prompt reviews. Although the UAC Program Write-up (DECUS 65) was edited in November, it was not included in the December DECUSCOPE and now prevents this issue from being a Bolt Beranek and Newman exclusive for PDP-1 information.

February DECUSCOPE will have an interesting article by Inforonics Inc.

Contributions to DECUSCOPE should reach the Editor before the 7th day of the current month for publication in that issue. Material received after the 7th will be considered for publication in the months following.

DECUSCOPE
Published Monthly
for
Digital Equipment Computer
Users Society
(DECUS)
Maynard, Massachusetts
Elsa Newman, Editor

THE DECUS PROGRAM LIBRARY

PDP-1 Program Library Facts and Figures - 1963
from Angela Cossette, DEC

290 Programs requested by Users
280 Program requests filled
50 Programs contributed by Users
47 Programs reviewed by two or more Users
17 Programs certified by the DECUS Programming Committee
30 Programs to be certified by February, 1964
10 Program requests outstanding at the present time

DECALRITHMS

This is a new column to communicate information among users of DECAL-BBN. Among the items suitable for publication are algorithms, procedures, useful instruction generators and action operators. All entries should be written in DECAL-BBN notation. Certification and comments about previously published items will also be accepted.

DECALRITHM 1

True-False Igs

By: C. M. Fletcher and L. C. Clapp; Bolt Beranek and Newman Inc.

```
... The following igs permit the writing of statements
... as true=> abc and makes Decal-BBN logical statements
... compatible with Algol 60 format.

 .. true dig beg lv7 rs1 nlc
        fde lst; law' 0 end
 .. false dig beg lv7 rs1 nlc
        fde lst; law 0 end
```

DECALRITHM 2

Bessel J_o (Z)

By: L. C. Clapp, Bolt Beranek and Newman Inc.

. . . procedure to calculate the zero order Bessel functions in floating mode by the method of repeated ratios. Argument is z and series calculated to an accuracy of epsilon. Result is left in the floating accumulator.

```
Procedure J0 (z, epsilon)

 beg efm 2
 one => sum => k
 for term <= one while abs term< epsilon do beg
     jda sfm; efm 2
    (~term)' × z × z / (four × k × k) => term
     one + k => k; term + sum => sum
     jda rfm; end; sum; lfm end
 one:    lac ; ..1
 two:    lac ; ..2
 four:   lac ; ..3
```

DECUSCOPE

INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

February 1964 Vol. 3 No. 2

"INFORMATION PROCESSING" SUBJECT OF SPRING MEETING IN WASHINGTON, D.C. APRIL 24, 1964

The Office of Naval Research will cosponsor the DECUS Spring Meeting on April 24, 1964

Meetings Chairman, Joseph Lundy, issued a call for papers at the Delegates Get-Together held at the Fairway on Feb. 7, 1964.

The subject for a symposium is "Information Processing: Its Recording, Retrieval and Presentation". Those persons wishing to participate should notify their DECUS Delegate and mail 75-100 word summaries on or before 1 March to Joseph Lundy, Inforonics, Inc., Maynard, Mass. It would be helpful, since time is so short, if a copy of the topic and summary were mailed to the DECUS Secretary.

Speakers should specify the approximate amount of time required for delivery and plan to have a complete paper by April 10th.

The tentative program will be published in the March DECUSCOPE when all suggestions for sub-topics and proposed papers have been received.

For further information, please contact Mr. Lundy at Inforonics, Inc., Maynard, Mass., Telephone 897-2073. Although "Information Processing" has been chosen as a theme for the meeting, other timely topics will be welcome.

Typesetting Using the PDP-1

L. F. Buckland and J. T. Lundy

Inforonics, Inc., Maynard, Massachusetts

A programming system for publishing scientific information is being developed by Inforonics, Inc., under a contract with the National Science Foundation. One project task is the development of computer aided typesetting methods for scientific article composition. The system for recording and processing journal text data begins at the final stages of manuscript editing, when the manuscript is typed on a perforated-tape typewriter. After correction and proofreading, the tape is converted by a computer process to form both a typesetting tape for the journal article and published indexes, and a digital storage for subsequent uses. The typesetting tape is entered into a phototypesetter to produce typeset copy for making printing plates.

Introduction

There are a number of sizeable projects underway in computer aided typesetting, and the following system characteristics distinguish our PDP system:

1. The input manuscript is typed on a perforated-tape typewriter so as to *identify* text items rather than to specify their *typographic form* as is done by other systems.

2. The typographic form is specified by relating it to the type of text item. In this way any output form can be devised from a manuscript tape.

3. The typesetter control portion of the program is kept independent, so that any output typesetter, such as Photon, Linofilm, Linotype, Intertype, or Justowriter, can be used.

System Description

The typist receives the manuscript and rules for dividing the text into items. A computer-readable sequence of delimiters, such as double carriage returns, isolates an item from the surrounding text. These isolating sequences are ones that a Flexowriter typist performs naturally, so that she needs no familiarity with the selected output typesetter. Typographic shifts (e.g. roman to *italics*) which occur within an item are also treated as input functions. The typist types a transliteration symbol followed by a font identifier and continues typing in the normal way.

The PDP-1 typesetting program contains typographic control words arranged in a list

FIRST PDP-4 SYMPOSIUM HELD

The first PDP-4 Symposium was hosted by The Foxboro Company, Natick Office, on January 21, 1964.

Nine papers and a demonstration of the Foxboro Process Control System (see January DECUSCOPE for program) were presented during the all-day meeting. Attendance was so good (forty-five users registered) that a conference hall had to be rented at Armand's Beacon Terrace in Framingham.

The speakers described many interesting applications of the PDP-4. President Adams of Charles W. Adams Associates spoke on "The Keydata On-Line Processor (KOP-2)". Stephen Lorch of Massachusetts General Hospital detailed some PDP-4-aided medical research. Early PDP-4 successes in the field of industrial process control were discussed by Allen Rousseau of Adams Associates and Saul Dinman and David McAvinn of The Foxboro Company. Russ Winslow of Digital Equipment Corporation spoke about the use of the PDP-4 as a module tester. Steve Piner and Dave Fellows, also of Digital, discussed advances in software for the PDP-4.

After the formal presentation of papers and discussions, the guests adjourned to Foxboro's Digital Systems Division plant in Natick for a demonstration of the Foxboro M97000 Process Control System. Among the functions demonstrated were: 1) gathering of measurements from instruments, 2) computer communication with the process operator, 3) logging and alarming of process data and 4) set-point control of a simulated valve.

At the request of many of the persons attending and those who could not attend, the Symposium will be reported in a special Proceedings. Those interested in receiving the Proceedings should write to the DECUS Secretary, Digital Equipment Computer Users Society, Maynard, Massachusetts.

whose entries are associated with the various text items. Each control word governs the composition of a text item until the item boundary is reached. It contains the following information: font (the type style), point size (vertical character dimension), set-size (horizontal character dimension), leading (vertical spacing), and quadding (horizontal layout instructions). The program records this information prior to processing and selects a path based on the quadding command which specifies flush left, flush right, center, or justify.

The problem of justification, brought to light in recent months by the reported success of computer aided newspaper publication, is probably the best known aspect of the program. For each font available on the output typesetter, a table is stored in memory whose entries contain the output code configuration of the selected character and its width in relative units. The input punched tape code (currently FIODEC) is used to address its output counterpart, which is serially stored in a buffer. Its width, in relative units, is added to a running count for the word and to a running count for the line. When a word causes the running line count to overshoot the preselected column width (in relative units), it is temporarily set aside. The program now uniformly expands the inter-word spaces until the line fills the prescribed column width. If the inter-word space reaches its maximum allowable expansion and the line is still short, then the overshooting word is recalled and presented to the computer operator via the on-line typewriter for manual hyphenation.

The line is now justified in relative units and is available for code conversion to any tape operated typesetting machine. For the project described, however, a Photon typesetter was used. This required a two level code containing (1) the character code, and (2) its width in Photon Basic Machine Units. The BMU's, which control the horizontal escapement of the prism carriage, are derived by multiplying relative width times set-size (from the control word). One BMU is equal to 1/512 centimeter. In addition to the character and width codes, the Photon typesetter requires certain function codes (lens shifts, font shifts, etc.). These are supplied, when necessary, by the typographic control words described earlier.

Use of the System

This system is economically applicable to those situations where multiple uses are made of the data. That is the situation where technical reports, indexes, entries for abstract journals, and data for handbooks can be derived from the record produced at a single source. A potential user of the system must analyze his publishing and secondary use requirements to determine the following system specifications:

1. A list of the types of text items which must be identified in the input record for use in primary or secondary publications.

2. The output format of the printed publications.

3. The range of type fonts which are required.

4. A description of the text processing operations which must be performed prior to automatic typesetting, such as item extraction, conversion, sorting, and merging.

Once these specifications have been developed, appropriate subroutines of the text processing program are selected, and their format control and font tables modified to suit the requirements. A short sample publication containing examples of the full range of requirements is selected as a text sample and is processed to uncover any errors before lengthy production runs are made.

*Note

The format illustrated by this article is the result of using the PDP-1 computer program described by the authors.

DECUS PROGRAM LIBRARY

PDP-1 Library Catalog

The following programs have been added to the PDP-1 DECUS Program Library.

MRR - Short, Read Magnetic Tape Routines — DECUS No. 66

Author: William A. Fahle, Systems Research Laboratories, Inc.

Purpose: Reads a record of a standard low density IBM format tape into a block of core. The entire record on tape must be transferred into core, thus allowing records of a non-predetermined length to be transferred. An entry is available for skipping records. The routine has an end of file return and a missed character indication.

Storage used: 86_{10}

"SIN-COS": Floating-Point Sine-Cosine Routines — *DECUS No. 67

Author: A. R. Zacher and E. Mallach, Princeton-Pennsylvania Accelerator

Purpose: These routines compute the sine or the cosine of the number in the floating accumulator and leave the result in the floating accumulator. The routines use 155 (octal) registers.

"SQRT": Floating-Point Square Root Routine — *DECUS No. 68

Author: A. R. Zacher and E. Mallach, Princeton-Pennsylvania Accelerator

Purpose: This routine computes the square root of the floating-point number in the floating accumulator and leaves the result in the floating accumulator. The routines use 56 (octal) registers.

*These single-precision floating-point function routines - sine-cosine and a square root routine are compatible with the SINGLEDEC arithmetic routines. They are somewhat faster than the corresponding SINGLEDEC programs, and the storage requirements and the accuracy are comparable. The versions are not for use with the floating-point interpreter, but all that is required for such use is the insertion of the instructions "jda sfm" and "jda rfm" in the proper places.

"These routines are part of a complete set of fast floating-point routines which Efrem Mallach and I recently finished writing. These routines allow floating-point operands to be located in any memory field; they are fairly fast (300 μs add, 173 μs multiply, 176 μs divide); and their storage requirements are moderate (442_8 registers for the arithmetic and data-handling routines). If there is any interest in such routines we will furnish them also."

Write Mr. Zacher or DECUS for further information.

DECUS DELEGATES AND EXECUTIVE BOARD EXCHANGE IDEAS

The Officers of Digital Equipment Computer Users Society met with DECUS Delegates in an informal get-together at the Fairway in Concord on February 7, 1964. Delegates exchanged ideas and discussed future programs and goals of the organization. Of immediate interest was the announcement of the Spring Meeting at Washington, D.C.

Executive Board to Meet

The Executive Board of DECUS, composed of five elected officers, will meet on February 14 to vote on some of the suggestions made and to approve applications for membership. The Society is open to all Digital PDP users (although originally formed by PDP-1 users) for the purpose of interchanging ideas and information of mutual interest. The Society has grown and includes PDP-4 and PDP-5 users as well as PDP-1 users.

The Society publishes DECUSCOPE every month, DECUS Proceedings every year, (the Proceedings report papers presented at Technical Symposia and Meetings) and maintains a Program Library.

1963-64 Executive Board

Lewis Clapp, President
Elsa Newman, Secretary

Committee Chairman:

Richard McQuillin, Programming
William Fletcher, Equipment
Joseph Lundy, Meetings

Edward Fredkin, Retiring President
Ex-officio DEC Member -
Robert Beckman

PDP-4 PROGRAMMING COMMITTEE CHAIRMAN APPOINTED

Mr. Gerald Mahoney was appointed Programming Committee Chairman for the PDP-4. Please help him build a good PDP-4 Program Library.

W A N T E D

PDP-5

PROGRAMMING CHAIRMAN

MAIL
YOUR
SUGGESTIONS
TO DECUS

C A L L F O R P A P E R S

SPRING DECUS MEETING
TITLES AND SUMMARIES
MARCH 1, 1964

DECUSCOPE

Published Monthly
for
Digital Equipment Computer
Users Society
(DECUS)
Maynard, Massachusetts
Elsa Newman, Editor

Contributions to DECUSCOPE should reach the Editor before the 7th day of the current month for publication in that issue. Material received after the 7th will be considered for publication in the months following.

PDP-4 Library Catalog

KUS - Kie Utility System — DECUS No. 4-5

Author: Allen Rousseau, Charles W. Adams Associates

KUS is an octal debugging and utility system. It consists of five basic routines.

Octal Correcting Routine - to examine and/or correct the contents of any core memory location with the option of punching a correction tape in RIM mode.

Word Search Routine - to examine the contents of a specified portion of core memory for a particular bit pattern and print the address of the register (or registers) if found.

Octal Dump - to print and/or punch (in RIM mode) the contents of a continuous section of core memory specified by the user. The print out is 1-8 words per line as specified by the user.

Compare Tape Equal Routine (What's Changed) to compare RIM tape with the corresponding area of core memory and type out the contents of the tape and core words which differ. The contents of core memory are not altered. What's changed can be used to verify a newly punched tape or compare current memory contents with a previously punched tape for debugging.

Jump Options - The user may jump to any location in memory with interrupt on and off.

DDT MODIFICATION FOR FLOATING POINT DEBUGGING (PDP-1)

While debugging floating point programs, it is often desirable to type out the contents of two adjacent registers in floating point. This can be accomplished by a simple modification to DDT and a special tape which calls routines in the floating point package. A jump to 100 should be put in register 7406 in DDT, and the following program loaded at 100.

```
...debug

dss fop

debug'    dap a
     lio b
     tyo' ; rir 6 ; tyo'
     efm 2
a:   lac ..
     jsp fop
     jsp 6000
b:   ..3672
fin
```

Upon typing location] this routine will be called and the contents of location and location + 1 will be typed out as a floating point number. The program above is written for DECAL-BBN but a similar routine may be written for use with MACRO.

```
200/      350000
201/      1

200]      1.8125+00
```

Ralph Zaorski, Raytheon Company

DECUSCOPE

INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

March 1964 Vol. 3 No. 3

3RD

ANNIVERSARY
FOR DECUS TO BE
CELEBRATED AT THIRD
SPRING
SYMPOSIUM

YOU CAN'T RUSH HOME FROM THE SJCC! PLAN TO COME TO THE HOTEL WASHINGTON AND HEAR ALL ABOUT THE THIN-SKINNED COMPUTER OR THE PDP-1 vs. THE 7094 et al.

IN THIS ISSUE

SPRING 1964 DECUS SYMPOSIUM*

Hotel Washington, Washington, D.C.

April 24, 1964 - 9 a.m. to 5 p.m.

Subject: INFORMATION PREPARATION, RETRIEVAL AND PRESENTATION

ABSTRACTS **

A LANGUAGE TEACHING SYSTEM WITH REAL-TIME SPEECH PROCESSING, ERROR DISCRIMINATION AND ERROR PRESENTATION

Roger Buiten and Harlan Lane
The University of Michigan
Ann Arbor, Michigan

Parameters which carry information in word-group (sentence, clause, etc.) presentation of the human voice are intonation variation, intensity variation, and rhythm variation. A system is described which extracts these parameters from word-group waveforms, converts them to units such that equal changes in the modified parameter correspond to equal sound differences as distinguished by the human ear, and compares word-group parameters spoken as a pattern to word-group parameters spoken as an imitation of the pattern. Analog equipment is described which extracts the intonation and intensity variation parameters. A PDP-4 digital computer is used to determine the rhythm variation parameter, to perform the units conversions, to store the three pattern parameters, and to make real-time comparisons of pattern word-groups to imitation word-groups. Output display of the difference between the pattern and imitation parameters are available for each parameter, or in combination, in both continuous (meters, graphic recorders, etc.) and discrete (flashing lights) forms. Any continuous display will indicate to the student in real-time the magnitude and direction he must modify his imitation to decrease the error level. Results of research with the various output displays with the system are described.

*See April 1964 DECUSCOPE for final program.

**Final papers should be submitted by April 10, 1964.

Editorial

SPRING SYMPOSIA

This year, the Spring Symposium will be held in Washington, D.C. The one-day meeting takes place on April 24, (the closing date of the Joint Spring Computer Conference). The subject, "Information Preparation, Retrieval and Presentation" will include new approaches to old problems. There will be a panel discussion of general interest and it will be open to attendees of the JSCC as well as DECUS members and friends. Because of its Washington locale, the 1964 Symposium assumes organization wide coverage but the concept of the "Symposium" still remains regional. Originally suggested by the editor as a means of bringing users together informally, the Spring Symposium is now part of DECUS tradition.

The first symposium took place at Itek Laboratories, May 1962, on: "Image Processing and Displays." Users in the Boston area attended and found it stimulating. In May of 1963, another one-day symposium on "Time-Sharing" was held. Nearly 100 scientists and programmers assembled at Kresge Auditorium (M.I.T.) for day and evening sessions of presentation, including demonstrations of the PDP-1 as a time-sharing computer. Both the M.I.T. and BBN time-sharing systems were demonstrated after long and lively discussions on varied and numerous other applications. This was a "regional" meeting in the sense that nearly all users in the East were present.

(cont.)

COMPUTER AIDS TO VOCAL CORD STUDIES AT AFCRL

Lt. John L. Ramsey
Air Force Cambridge Research Laboratories
Bedford, Massachusetts

The paper describes the use of the PDP-1 as a tool in reducing and analyzing data which is originally obtained from a high speed motion picture of the vocal cords during voicing. The motion picture simultaneously portrays the vibrations of the vocal cords and a visual representation of the electrical signal picked up by a microphone held in front of the subject during the filming.

For each frame of the motion picture, the computer transcribes onto magnetic tape the following data: (a) the detailed out-line of the glottal opening; (b) the computed glottal area; and (c) the microphone output. This frame-by-frame data can be subsequently processed so that the glottal area and/or the microphone output can be displayed as a function of time. This data can in tern be processed further to obtain the Fourier series representation of the glottal area and the microphone output, and to obtain the transfer function of the vocal tract.

SYSTEM FOR THE RECORDING, PROCESSING AND EXCHANGE OF LIBRARY CARD DATA

Lawrence Buckland
Inforonics Incorporated
Maynard, Massachusetts

A system is proposed for the recording, processing and filing of library card data. The purpose of this system is to provide a data base needed to promote exchange of data between libraries on a national basis. The prime advantage of such a system would be the elimination of the duplicate cataloging and card processing effort presently performed by libraries throughout the country. A second benefit of the system is the assembling of a card catalog in machine readable form so that reference tools other than card form catalogs can be made automatically from the data.

A PDP-1 system is programmed to process the card catalog data. Its functions are to:

(1) edit and correct recorded data

(2) process card data to produce multiple cards in different formats

(3) process card data to control automatic typesetting of cards

(4) process data to form magnetic tape catalog file for loan and distribution.

One unusual aspect of the processing is the handling of non-Latin signs and symbols. These symbols are encoded so that in the output processing they can be typeset on the cards, or converted to an equivalent symbol printer of lesser symbol capacity, or left blank for hand entry.

A second problem which had to be overcome in the design is the identification of text item on the card when the text items are selected from a large set of possible items. This was accomplished by a recording procedure which uses sequences of natural delimiters and specified positions on a typing form.

THE COMPUTER AS A CONTROL DEVICE FOR PSYCHOLOGICAL EXPERIMENTATION

Raymond S. Nickerson
Decision Sciences Laboratory, ESD, AFSC
Bedford, Massachusetts

During the short history of computers, psychologists' involvement with them has been primarily in connection with data analysis and simulation. The utility of such a device for both activities is obvious. A third, less tested, but no less promising application, viz., real-time control of experimentation, has excited the interest of a number of experimenters, including some at the Decision Sciences Laboratory. The purpose in this note is to mention briefly what we feel to be a few of the advantages, and contingencies of employing a computer in this fashion.

Typically, to prepare for, conduct and report a psychological experiment, one must:

- Prepare a set of stimulus materials
- Instruct and train a group of subjects
- Record the salient aspects of subjects' behavior in the experimental situation
- Produce a permanent record of the experimental results
- Extract the pertinent information from the data
- Display the observed relationships between dependent and independent variables in graphical, or some other easily communicated form

In so far as each of these tasks is programmable it may be delegated to a computer. The object, of course, is to program them all if possible.

Certainly, interest in automating experimentation is not new. "Apparatus" sections of 19th century journals are crammed with descriptions of ingenious devices for expediting, or automating, the data collection process. In more recent times, but long ago in terms of the computer age, automated experimentation reached a sort of ultimate with Skinner's push-a-pedal-pluck-a-pellet box. Why then the excitement about computers?

For one thing, use of the computer makes it feasible to conduct experiments which would be very difficult, if possible, to do without it. For example, a class of experiments currently in the "talking about" stage in this Laboratory would involve making certain signal parameters, including the precise moment of delivery, contingent upon a subject's instantaneous level of "arousal" or "alertness" as evidenced by various physiological indicants fed momentarily to the computer via an A-D converter. Such an experiment obviously implies a quantity of fairly involved computations in an extremely short period of time. An example of a much simpler experiment which proved to be dissuadingly difficult without the computer - and straight forward with it - involved a key pressing task in which the time from a subject's last key press to the delivery of the next signal was to be a function of his last response time (i.e., the time from signal to response).

It might, of course, be argued that it is theoretically possible to do any experiment which can be done with the computer with an assemblage of equipments constructed for that specific purpose. However, as the desired computations become more complex, and time more critical, the differences in complexity and function between the special purpose device and a computer would quite probably become less and less distinct. And a computer by any other name is a computer.

Even if we could be convinced that nothing can be done with a computer that cannot be done without it, we still would be excited about its prospective role as an experiment controller. As any trustworthy computerphile will candidly admit, computers are versatile, general purpose devices. One computer suitably tied to a few pieces of in-out equipment provides an adequate hardware base for a variety of experiments whose implementation would otherwise necessitate the construction of numerous special purpose devices. To illustrate this point: currently at DSL, the PDP-1 with typewriter, scope and telegraph keys, is being used to conduct experiments in decision making, logical reasoning, perception, memory and reaction time.* Without the computer each of these experiments would require a specially devised assemblage of gear.

Detailed descriptions of individual experiments fall outside the scope of this note. However, a few specific examples of the computer's involvement in them should give an indication of the diversity of functions it is serving even in this small group of experiments, representing, for the most part, first efforts along this line. A partial list of programmed functions includes:

1. Presentation of verbal information on the CRT to instruct subject, to provide stimulus materials, and to give periodic feedback to subjects concerning their performance;

2. Recording of responses made by subjects with light pen, typewriter, or telegraph keys;

3. Measurement of response latencies with millisecond accuracy;

4. Scheduling of temporal order of events or spatial arrangements of display elements, according to programmed rules which may include randomization with or without forcing constraints;

5. Modification of experimental parameters on the basis of performance;

6. Adjustment of the difficulty of a judgmental task to match the capability of the individual subject;

7. Production of a punched tape record of the trial by trial progress of the experiment;

8. Performance of statistical analyses on data as it is collected thus providing the experimenter with the results of the analyses immediately at the termination of the experimental session.

Discussions of this sort generally evoke some remarks about excessively high cost usually associated with the use of computers. The issue cannot be skirted. Even "inexpensive" computers are still way beyond the means of many research groups. However, it may be taken as axiomatic that information acquired through experimentation is seldom an inexpensive commodity. Although assessment of the worth of information may involve some debatable value judgments, its cost can be measured fairly objectively in terms of equipments and individuals' time. In terms of cold cash, figures exist which indicate that it is not unusual for a single research effort culminating in a single published report - of uncertain worth - to total an amazingly large fraction of the cost of a moderately priced computer.

Among the factors which accrue to make research such an expensive endeavor are some which a computerized facility should help to alleviate. Research is presumably to a great extent a "creative" activity. To ask a scientist to spell out in detail what he intends to do in the future is like asking a poet to recite a poem before he has composed it. In a sense, a scientist's task is to think about some aspect or area of nature and to formulate a conceptual model of it. Experimentation is simply a methodological tool for aiding, checking and stimulating his thinking - and for making the activity public. The observe - hypothesize-check cycle is the method of science. The rub is that the time involved in closing the loop is too frequently discouragingly long. Weeks or months of planning, building apparatus, and running pilot studies often precede "the experiment." Then when the experiment is completed, in many cases, the main thing that one has learned is how the experiment should have been done. Probably every research laboratory has its mausoleum of equipment relics built for researchers who, while the gear was being fabricated, either lost interest in the problem, found a better way to approach it, or moved on to a better job. Or, worse still, in some cases, one has invested so much time and effort, or solicited so much support, that by the time he discovers that an idea is not worth pursuing, it is difficult and embarrassing to back off. If the computer, serving as the major experimental control device, can help tighten the loop - reduce the time from the inception of an idea to the testing of it, it will earn its keep. We perceive this as a distinct possibility.

It should be emphasized that we see the computer playing a role in all stages of experimentation. The researcher should not feel compelled to wait until he has a sure thing ready to go. He should be able to use the computer to run quick exploratory, hunch checking, studies, to get early assistance in his thinking, to identify procedural or conceptual blind alleys, to evolve the proper experiment. All this, of course, depends on the availability of computer time, which, in turn, suggests the importance of time sharing. Full exploitation of the computer in this role is contingent upon the development of techniques which will allow several experimenters to share the computer at any given time.

No less important than time-sharing techniques for freeing computer time is the availability of programs and procedures. The need for a well-documented program library drawing from and accessible to all users of similar systems has been a continuing concern of DECUS. It is our conviction that the importance of this project cannot be exaggerated. Time spent preparing, editing and debugging programs which already have been developed by other users is thoroughly wasted.

Finally, without denying our obvious enthusiasm for what appears to be a new vista of research opportunities, we should perhaps terminate this discussion on a less ebullient note. The computer is not a panacea for all the problems involved in doing significant research. Ultimately the quality of research must depend on the quality of the thinking of the people doing it. No amount of sophisticated instrumentation can compensate for a paucity of researchable ideas.

*The experiments referred to are being conducted by J. Baker, C. Feehrer, I. Goldstein, J. Hayes, P. Weene and the writer.

A PDP-5 INTERPRETIVE-SIMULATOR FOR THE PDP-4 (SLOFIN)

Herbert M. Norris
The Foxboro Company
Natick, Massachusetts

A program (SLOFIN) has been developed to simulate, on an 8K PDP-4, the functions of a 4K PDP-5 with Teletype Model 33ASR I/O capabilities. Any program written for a PDP-5 can be run on a PDP-4 using SLOFIN, within system configuration limitations. SLOFIN interprets assembled PDP-5 instructions and executes them in a manner logically analogous to that of the PDP-5.

In its present configuration SLOFIN will handle all PDP-5 internal operations and 33ASR I/O operations. SLOFIN also includes the capability of simulation of a Foxboro M/97400 System.

THE PDP-1 AS AN ON-LINE CORRELATOR

William Fahle
Systems Research Laboratories
Dayton, Ohio

This paper describes the use of the PDP-1 as an on-line correlation device. Beginning with a brief explanation of the use of the auto-correlation and crosscorrelation functions as on-line analysis techniques, the paper proceeds to discuss three methods for using the PDP-1 as an on-line correlator and the relative advantages and disadvantages of the methods. Comparisons are made between the PDP-1 as an on-line correlator and some other digital on-line correlation devices. The methods described were developed for and in conjunction with the Biodynamics and Bionics Division, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base, Ohio.

POTENTIAL APPLICATIONS OF A TIME-SHARED COMPUTER IN SECONDARY SCHOOL EDUCATION

Lewis C. Clapp
Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

Jesse O. Richardson
Massachusetts State Department of Education

There are many educational advantages to be gained by providing access to a digital computer in the classroom. The computer can be used as a demonstration device in mathematics, physics and general science classes. Equally important, it can provide the student with a means for experimenting with important mathematical and physical concepts. For example, the student can gain first-hand experience with the concepts of convergence and limits, concepts which at present can only be taught in the abstract. Heretofore, however, most schools have found that the cost and maintenance problems were too severe to permit them to acquire a reasonable computer for their needs. A time-shared computer serving many schools in a community has the potential to make the introduction of a computer in a classroom economically feasible. This paper discusses some of the educational advantages that may be gained by the introduction of remote time-shared stations in the classroom and will describe an experiment just started in Massachusetts.

Editorial-cont.

DECUSCOPE would like to announce more "regional" symposia and looks forward to a West Coast Symposium in the future. There are rumblings at present that our Canadian members may host a symposium soon.

WELCOME TO DECUS

New DECUS Delegates

PDP-1

Dr. Richard J. Plano
Physics Department
Rutgers University
New Brunswick, New Jersey

Mr. Clay Fox
System Development Corporation
Santa Monica, California

Miss Jane Levy takes over as Delegate for Professor John Dennis at M.I.T. Electrical Engineering Department.

PDP-5

Mr. William H. Ninke
Bell Telephone Laboratories, Inc.
Murray Hill, New Jersey

New Individual Member

Mr. Ralph Miller
Physics Department
Rutgers University
New Brunswick, New Jersey

NEWS

INFORMATION INTERNATIONAL INC. HAS NEW OFFICES

Edward Fredkin has been busier than ever these days. One of his problems has been expansion. I.I.I.'s address after April 6th will be:

600 Sixth Street
Cambridge, Mass.

The new offices will extend over three times the floor space occupied in Maynard.

DIGITAL HAS BUSY EDUCATIONAL CENTER

The Educational Center of the Customer Relations Department (DEC) is located in the recently renovated Main Building, (Bldg. 12) of Digital's huge complex. Classrooms sparkle with new paint, furniture and teaching aids.

Customer Relations has been extending services to new and old users which include just about everything that could possibly insure better utilization of PDPs.

Mr. Raymond Bernier conducts Familiarization and Maintenance classes for PDP-1, PDP-4 and PDP-5. PDP-6 courses will be starting in late June, 1964. Mr. Ronald Leonard has a red workbook he uses for PDP-1 programming courses. Mr. Jack Richardson teaches PDP-4 and PDP-5 programming.

The Department has issued certificates to approximately 250 attendees. Classes will be continued as follows:

<u>March 30</u>

PDP-1 Programming - 1 week

<u>April 6</u>

PDP-1 Maintenance - 2 weeks

<u>April 20</u>

PDP-5 Maintenance and Familiarization - 1 week

<u>April 27</u>

PDP-4 Maintenance and Familiarization - 2 weeks

<u>May 11</u>

PDP-4 Programming - 1 week

DECUSCOPE
Published Monthly
for
Digital Equipment Computer
Users Society
(DECUS)
Maynard, Massachusetts
Elsa Newman, Editor

AN INFORMATION RETRIEVAL PROGRAM TO AID IN PROVING THEOREMS

William H. Henneman
Information International, Inc.
Maynard, Massachusetts

A program to aid in proving theorems by retrieving the most relevant theorems is described. Theorems are matched for relevance by the descriptor word match and count method. Results are given for the fields of number theory and point set topology. Various extensions and modifications of the program in the light of the experience gained are discussed. Possible applications in areas that are more directly related to the "practical world" are given.

COMPUTER AIDS TO LITERATURE SEARCHES

Mario Grignetti
Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

In the present paper we describe two operating computer programs for the PDP-1 computer, designed to facilitate searches of bibliographies to technical literature.

The first, Permuted Title Indexer, is a program which converts a direct file into an inverse one.

The second, Automated Card Catalogue, is a program that using some of the features of the first, allows a user at the computer console, to search a file for entries related to his interest. The results of the search are visually displayed.

A NEW APPROACH TO TIME-SHARED DISPLAY SYSTEMS

Dan M. Forsyth
Information International, Inc.
Maynard, Massachusetts

A display system will be described which utilizes scan-conversion techniques to permit the simultaneous operation of ten or more different displays (all flicker free) from a single PDP-1. Light pen interrogation and writing facilities are provided at each console. After a given display has been generated, no further computer time is required to maintain it.

PANEL DISCUSSION

There will be a panel discussion on the "Thin-skinned computer vs. the other computers like the 7094." Details and panel members will be announced.

DECUSCOPE
INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

July-August 1964 Vol. 3 Nos. 7, 8

DR. HANS L. OESTREICHER TO MODERATE PANEL

The 1964 DECUS Annual Meeting will take place 24-25 September at The Stratford House, 330 W 1st Street, Dayton, Ohio. The program will include sessions on the applications of the Digital Programmed Data Processors -1, -4, -5 and -6.

Dr. Hans L. Oestreicher, Mathematics and Analysis Branch, Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base will moderate a panel discussion on "PDP Applications in Medicine and Biology."

Charlton M. Walter, Air Force Cambridge Research Laboratories, will give a paper on "Intrinsic Analysis Versus Fourier Analysis in the Representation of Signal Data Structures." There will be several papers on scientific applications and a guest speaker to be announced.

Papers Due August 20th

All PDP Users are invited to participate. Two copies of proposed papers with a 100-word abstract should reach Meetings Chairman, Joseph Lundy, Inforonics Inc., Box 207, Maynard, Mass. by August 20, 1964. DECUS Delegates should contact the DECUS Secretary for reservations, etc.

PHYSICS SYSTEMS GROUP SEMINAR FOR WEST COAST PDP-5 & 4 USERS

Sypko W. Andreae of the Physics Systems Group, LRL, will meet with interested PDP-5 and 4 users on July 28 at LRL, Berkeley, to plan a September Seminar. Users are invited to exchange ideas and discuss program objectives.

Please contact Mr. Andreae, Lawrence Radiation Laboratory, Berkeley, Calif., Bldg. 50A Room 6115, Ext. 5662 and plan to attend.

IN THIS ISSUE

Recording of Medical Data Simplified by Unique Computer System Shown *

A prototype of a new "interpretative communications" system designed to simplify handling of communications and medical records in hospitals was demonstrated in the Clinical Center auditorium June 15.

The National Institute of General Medical Sciences sponsored the demonstration conducted by Bolt Beranek and Newman, Inc., and the Massachusetts General Hospital, which began development of the system two years ago with support from NIGMS and the American Hospital Association.

As part of the demonstration of a new "interpretative communications" system for hospital use, presented in the NIH Clinical Center June 15, Dr. David Poskanzer, Associate Neurologist of the Massachusetts General Hospital uses a teletypewriter in a two-way "conversation" with central computer facilities in Cambridge, Mass.—Photo by Bob Pumphrey.

Computer in Cambridge

A unique system in many respects, it consists of a central time-shared digital computer located in Cambridge, Mass., connected through private-wire teletype lines to a number of teletypewriters located in various nursing stations and in the pharmacy in the hospital.

There are no direct connections between the teletype stations. All communications pass through, and are monitored by, the computer. Thus, hospital personnel may "converse" with and through the computer from the different stations located throughout the hospital.

When a member of the hospital staff wishes to make an entry in the patient's record, or order medication, he calls up the necessary program on the computer, which then produces a series of relevant questions. The hospital personnel type in the answers.

Computer Interprets

The computer "interprets" each message and checks for discrepancies against previously "memorized" criteria. After checking, it stores data in its memory as a permanent part of patients' records.

The system detects technical misspellings and facilitates their correction. It can check discrepancies between patients' stated ages and birth dates, and may provide reminders if a patient's record indicates that a prescribed drug may produce side effects.

By providing a comprehensive record of patient-care that is readily accessible this system is designed to aid studies of patterns of diseases and patient-care.

The system also includes the possibility of providing service to a number of hospitals utilizing only the one central computer unit.

Cost Depends on Use

As a result, the cost to each hospital can be made proportionate to the hospital's use of the system. In addition, this arrangement may enable hospitals in a given region to share information.

In addition to regulating the prescription of drugs and recording patient-care information in a permanent medical record, other applications for this system may include the ordering of laboratory tests, X-ray examinations, diets, entry of technicians' information, and assignment of bed space.

Developed as a small scale pilot study for test purposes, the system which was demonstrated is a prototype. During the next year it is planned to introduce an expanded version of the system into Massachusetts General Hospital.

*Published with permission of "THE NIH RECORD," Public Health Service, U. S. Department of Health, Education, and Welfare.

FOR DECUS PRESIDENT 1964-65

Mr. William A. Fahle received his B.A. in Mathematics from Marian College and his Master's degree also in Mathematics from Indiana University. He taught undergraduate courses in mathematics at Indiana University for several years. In addition to his experience as a member of the faculty at Indiana University, he was engaged in applied mathematics and data processing at the Allison Division of General Motors.

At Systems Research Laboratories, Mr. Fahle is chief PDP-1 programmer for the Mathematics and Analysis Branch, Biodynamics and Bionics Division of the Aerospace Medical Research Laboratories at Wright-Patterson Air Force Base. He has been devising techniques for the analysis of neurophysiological data. He has been implementing these techniques on the PDP-1 computer.

Mr. Fahle's interests in the computer field also include computer languages, real-time scientific data analysis, and biological data processing.

FOR SECRETARY - Elsa Newman

Mrs. Newman is the originator and Editor of DECUSCOPE. She came to Digital nearly three years ago with a varied background in education, administration, technical editing and project engineering. She received her B.A. in Romance Languages and M.S. in Educational Psychology from the University of New York in 1943. Subsequent graduate study included physics, technical writing, production control and scientific editing. Mrs. Newman became interested in computers when she participated in an international seminar on "Automation: Its Effect on Women" in 1959.

With more than 20 years experience in government, private industry, teaching and scientific research Mrs. Newman is uniquely qualified to serve DECUS.

TAPELIBRARY PROGRAM FOR THE PDP-1

Daniel Forsyth*

Tapelibrary Program, DECUS No. 74, stores the image of punched paper tapes on magnetic tape and reproduces them on command. Read-checking is performed for paper-mag and mag-paper conversion. Records may be added to mag tape at any time, but not rewritten. Initial typed instructions are shortened when an operation is repeated. The program occupies 0-3117 registers.

At read-in time a rewind command is given to the tape transport, and typed inquiry is made as to whether reading, writing or listing of mag tape is desired.

1. If reading is desired, the record number will be asked for.
2. If writing is desired, records will be scanned until the last existing record is found. The number of the next record to be written is then typed out, followed by further writing instructions.
3. If listing is desired, mag tape is read and the titles of the successive records typed out, numbered consecutively (decimal).

When reading is the selected operation, the record number will be requested. This decimal number will be available from a listing of the records on mag tape. When it is typed in, the program goes to this record and reads it in, typing out the title for verification. It then punches leader, the program, and trailer, in order. If the record being read occupies more than one block, each block will be read and punched successively, the final block followed by trailer. If SS 2 is up, all punch operations are omitted.

After the record selected has been punched, instructions will be typed out to verify the punched tape. During this operation, the mag tape record is re-read and verified against the punched paper tape.

Because the Tapelibrary Program was written for a PDP-1 without a check-status instruction, when paper tape is being read it will be read completely out of the reader. It is necessary to restart the program at address 0. The last ten lines of paper tape are then removed from the buffer (to permit error-free verification of tape with jagger trailer) and the final block written on mag tape. This is followed by a request to verify, as described under "To Read" and the mag-tape record is checked against a re-read of the paper tape.

During either "read" or "write" operations, if an error is detected a diagnostic will be typed out, specifying the mag and paper tape words that do not agree. The mag tape will then be returned to the position it held before the operation which caused the error, and the query associated with that previous operation will be typed out again. It should be noted that the first record written on a new (blank) mag tape must be written differently than successive records. Read the program in with SS 1 up, to inhibit the jump, and start at address 43. Everything will then proceed as above.

Records may be added to those already on mag tape. The program will search the tape until it encounters an end of file mark, then stop, backspace over it, and type out the number of the next record to be written. Typed instructions will then direct the operator to type in the title of the record, and place the paper tape in the reader. The title which is provided at this time becomes a part of the mag tape record, but <u>does not</u> get punched out when the record is being duplicated.

*Dr. Forsyth is Director of Life Sciences Division, Information International, Inc., Cambridge, Massachusetts.

PDP-1

DECUS NO. 73 - (MADCAP and MADCAP CHECKUP)

MADCAP: MAmmoth DeCimal Arithmetic Program for the PDP-1 Computer

Author: E. Myrvaagnes, Parke Mathematical Laboratories, Inc.

Purpose: To perform arithmetic calculations on decimal floating-point numbers of arbitrary length, using programs written in MADCAP's own simple source language.

Programming Language: Macro Assembly Program

Storage Used: Octal 3362 + 10n + t registers consecutively, anywhere in memory, where n is the number of resistors used to store each floating-point number, and t is the number of decimal places in input and output.

Special Hardware Required: Automatic Multiply and Divide

Restrictions: $3 \leq n$; $2 \leq t \leq 4(n - 1)$; uses sense switch 1 and all program flags.

Other Programs Required: Expensive Typewriter, Macro Assembly Program, Macro Symbol Package, Dec Debugging Tape (optional).

MADCAP CHECKUP

Purpose: To check the operation of individual instructions in MADCAP

Language: Macro Assembly Program, with MADCAP

Storage Used: From mmm + 1 to mmm + 361 + 12n (octal), where mmm is the highest location used by MADCAP, and n is the number of registers used to store numbers in MADCAP.

Usage: MADCAP CHECKUP checks the instructions in the MADCAP source language one by one. Binary tapes of MADCAP and MADCAP CHECKUP should be assembled according to the MADCAP write-up, with MADCAP CHECKUP treated as a program using MADCAP. Read in the binary MADCAP and MADCAP CHECKUP. The program will ask for a floating point number to be typed in in MADCAP format. This will then be typed out and the computer will halt. This checks the "enter" and "type" instructions. Pressing "continue" will check "zero." The computer halts after each operation is completed. The entire sequence of operations is summarized in the following table. A, B, C, etc., will represent any floating point numbers in MADCAP format; i and j will represent any integers in MADCAP format.

DECUS NO. 74

Tapelibrary Program

Author: Daniel Forsyth, Information International, Inc.

Tapelibrary Program, DECUS No. 74, stores the image of punched paper tapes on magnetic tape and reproduces them on command. Read-checking is performed for paper-mag and mag-paper conversion. Records may be added to mag tape at any time, but not rewritten. Initial typed instructions are shortened when an operation is repeated. The program occupies 0-3117 registers.

From: R. J. McQuillin, Programming Chairman

Re: Single Dec and Floating Point Interpreter (DECUS No. 10 & 63)
(Please edit your copy of these programs.)

An algebraic statement like:

efm 2
if a>b ∨ c>∨ e<f then . . .

may not work properly because the and, ior, and xor instructions are not interpreted by the interpreter in floating point.

John Goodenough (DSL) suggests the following to correct this situation.

Change the entries in xctable:

from:	jmp normal . . .02 and	to:	jmp Boolean . . .02 and
	jmp normal . . .04 ior		jmp Boolean . . .04 ior
	jmp normal . . .06 xor		jmp Boolean . . .06 xor

Add these instructions:

```
Boolean:    lac fac
            xct instr
            dac fac
            dac ac
            jmp loop
```

DECUS PROGRAM LIBRARY†

Programs presently available from DECUS are listed by assigned DECUS numbers. One asterisk* indicates that the program listed is being reviewed by one or more users at the present time. Two asterisks** indicate the program has been "certified" by the DECUS Library Programming Committee.

Please see your DECUS Delegate when contributing or requesting programs from DECUS. Individual members not associated with an Installation Delegate may write to the Secretary of DECUS, Digital Equipment Computer Users Society, Maynard, Massachusetts.

Programs presently available from DECUS are listed by assigned DECUS numbers. One asterisk* indicates that the program listed is being reviewed by one or more users at the present time. Two asterisks** indicate the program has been "certified" by the DECUS Library Programming Committee.

Please see your DECUS Delegate when contributing or requesting programs from DECUS. Individual members not associated with an Installation Delegate may write to the Secretary of DECUS, Digital Equipment Computer Users Society, Maynard, Massachusetts.

DECUS NO. 5-1 and 5-2*

Service and Debugging Subroutines for the PDP-5

Author: A. D. Hause, Bell Telephone Laboratories

Two basic subroutine packages for use in communicating with the PDP-5 for service and debugging functions have been written at Bell Telephone Laboratories. The first package, called the Binary Package, is a completely independent one-page subroutine for handling paper tape input-output. The second, called the Octal Package, is a two-page set of subroutines which facilitates exchange of information between the operator and the computer via the Teletype 33ASR unit. As an addition to the Octal Package, there is a symbolic instruction dump subroutine package which occupies three pages of storage.

1. The Binary Package contains a BIN format loader, and provisions for dumping of bracketed locations in BIN or RIM format, punching leader trailer, punching a check sum on BIN tapes, and punching a starting address for self-starting RIM tapes. If entered as a subroutine, the appropriate return can be made. Control of the package is made by the switch register and CONTINUE switch on the PDP-5 control panel.

2. The Octal Package contains provisions for an eight to the line octal dump of bracketed locations, moving of blocks of information in the memory, and loading of memory from the Teletype. Links to the Binary Package and the symbolic dump are also included. Control for this package is through the Teletype and the switch register.

   The symbolic dump portion of the Octal Package provides for a printing on the Teletype of up to four pieces of information for each location of a bracketed group of locations. These include the address, the octal contents of that address, the interpretation as two trimmed Teletype code characters of the octal contents and the symbolic instruction decoding of the contents. Any combination of these pieces of information can be selected through use of the switch register.

DECUS NO. 5-3

BRL - A Binary Relocatable Loader with Transfer Vector Options for the PDP-5 Computer

Author: P. T. Brady, Bell Telephone Laboratories, Inc.

Description: BRL is a binary loader program occupying 4640_8 to 6177_8 registers: also 160 to 177. It has two main functions:

1. It allows a PDP-5 operator to read a suitably prepared binary program into any page location in memory except the registers occupied by BRL. Thus, a program need not be reassembled from symbolic to binary tape whenever it is desired to relocate it.

2. It greatly simplifies the calling of programmed subroutines by allowing the programmer to use an arbitrary subroutine callingsequence when writing his program, instead of having to remember the location of the subroutines. For example, if a programmer wishes to call "Square Root," he does not have to know where Square Root is stored in memory; BRL will instead keep track of this for him. This feature is available to the IBM 7094 programmer and is known as a "transfer vector" option.

Programs presently available from DECUS are listed by assigned DECUS numbers. One asterisk* indicates that the program listed is being reviewed by one or more users at the present time. Two asterisks** indicate the program has been "certified" by the DECUS Library Programming Committee.

Please see your DECUS Delegate when contributing or requesting programs from DECUS. Individual members not associated with an Installation Delegate may write to the Secretary of DECUS, Digital Equipment Computer Users Society, Maynard, Massachusetts.

DECUS NO. 4-1*

PDP-4 Automonitor Program

Author: M. Kawahara, Mesa Scientific Corporation

Purpose: This program is used to trace the execution of a program and provide a printout of the result of each command executed. The printout is of the form:

XXXX	XXXXXX	XXXXXX
Address of Instr.	Instruction	Contents of the Accumulator after executing this instruction.

DECUS NO. 4-2*

Teletype Input and Output Package

Author: M. Stein, Harvard University

Purpose: This is a revision of the DEC teletype output package. Several input routines are added and the typewriter can be initialized for either figures or letters.

DECUS NO. 4-3*

Random Number Subroutine

Author: Donald Norman, Harvard University

Purpose: This program generates a random number and leaves it in AC.

DECUS NO. 4-4*

Read a Decimal Number From Keyboard

Author: Donald Norman, Harvard University

Purpose: This program reads the teletype keyboard and converts typed decimal number to a binary number, leaving the results in AC. Written as a subroutine.

KUS - Kie Utility System

Author: Allen Rousseau, Charles W. Adams Associates

KUS is an octal debugging and utility system. It consists of five basic routines.

> Octal Correcting Routine - to examine and/or correct the contents of any core memory location with the option of punching a correction tape in RIM mode.
>
> Word Search Routine - to examine the contents of a specified portion of core memory for a particular bit pattern and print the address of the register (or registers) if found.
>
> Octal Dump - to print and/or punch (in RIM mode) the contents of a continuous section of core memory specified by the user. The print out is 1-8 words per line as specified by the user.
>
> Compare Tape Equal Routine (What's Changed) to compare RIM tape with the corresponding area of core memory and type out the contents of the tape and core words which differ. The contents of core memory are not altered. What's changed can be used to verify a newly punched tape or compare current memory contents with a previously punched tape for debugging.
>
> Jump Options - The user may jump to any location in memory with interrupt on and off.

DECUS NO. 4-6

DDT-4 With Floating Point Input/Output and Drum Read-Write

Authors: James E. Curry and Jerome L. Abel, C. W. Adams Associates

Purpose: To enable on-line DDT debugging of programs using floating-point data and to enable convenient reading and writing of drum tracks while using DDT.

Storage: $10{,}000_8$-12677_8 15101_8-1777_8

Programming Language: Machine (PDP-4 Symbol Language)

DECUS NO. 4-7

Bivariate Curve Fit (FOX PR-3)

Author: Judith Green, Foxboro Company, DSD

Purpose: This program generates an approximating polynomial by the least squares technique to a set of points (x,y,z) where z = f(x,y). The data, which need not be equally spaced, may be weighted or not at the option of the user. The program allows for any number of points. The number of powers of x and y may be increased by altering the dimension statement.

Storage: 4511_8

Programming Language: DEC Fortran II

Listing is automatic. The title of each record is typed out and numbered decimally. These are the numbers to be used in asking for records when reading. When the last record on tape has been read, the program will encounter an end-of-file, type out this information, and halt. If further reading or writing is desired, the addresses given in the following section must be used. It might be noted here that an end-of-file mark is written after each writing of a record, and that this occurs even before a record is verified. After verification of a written record, the transport stops in a position to write over the end-of-file mark.

EDITOR'S NOTE:

Now that we have a mag-paper tape conversion program, we will reproduce the DECUS Library or DECAL BBN for users who supply DECUS with magnetic tape. Yes, users may have copies of the tapes. Please ask for DECUS No. 74.

DECUS PROGRAM LIBRARY ON MAGNETIC TAPE?

DECUS is growing--and with this growth comes occasional growing pains. Take, for example, DECUS's work in program distribution. Recently, many installations requested a complete set of symbolic tapes for DECAL-BBN (51 tapes in the set). A large backlog developed and it was several weeks before some installations received their tapes.

It has been suggested that the entire program library be maintained on magnetic tape, to facilitate program distribution. A reel of tape could easily hold the entire DECUS library, and a mag tape can be copied in less time than it now takes to duplicate one program on paper tape. According to the suggested plan, every installation would periodically receive an updated version of the library on magnetic tape (and would send back the old tape). For those installations who do not have magnetic tapes, DECUS would still handle paper tapes.

The DECUS Executive Board is working on more rapid reproduction and distribution of DECUS programs. We welcome suggestions and contributions to the DECUS Program Library.

Lewis Clapp, President

PDP-1 TIME-SHARING TERMINAL ON MIT MAC SYSTEM*

The PDP-1 computer at M.I.T.'s Project MAC has been connected as a terminal having access to the time-sharing system on MAC's IBM 7094. The connection was made through Bell System data sets operating at 1200 bits per second.

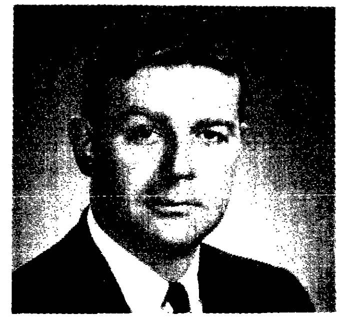
At the 7094 end, the terminal equipment is the IBM Programmed Transmission Control (7750), which also handles the teletype and typewriter terminals connected to the system.

At the PDP-1 end, a special adapter was built to provide an interface between the DATA-PHONE data set (202B) and the I/O register of the PDP-1. The PDP-1 operates in the sequence break mode to maintain communication with the 7750.

In addition to the 1200-bit-per-second connection, the PDP-6 will be connected to the 7094 Direct Data channel to permit communication at rates up to approximately 157,000 36-bit words per second.

*See "Project MAC: A Major Time-Sharing Development Effort," September 1963 DECUSCOPE and 1963 DECUS PROCEEDINGS, Pg. 33.

FOR MEETINGS CHAIRMAN



John T. Gilmore joined the staff of M.I.T. Digital Computer Laboratory after receiving a B. S. degree in Physics from Boston College (1950). He was in charge of the operation of Whirlwind I for all non-military uses and developed the first symbolic assembly program for the Whirlwind I.

Following a 4-year tour of duty as U.S. Navy carrier pilot, Mr. Gilmore joined the M.I.T. Lincoln Laboratory's Advanced Computer Development Group in 1956. As research programmer, he experimented with various new techniques on the TX-0 computer and developed a "moving-window" oscilloscope display of electro-encephalograph data which enabled examination of brain wave data. Other development work included a scopewriter program and contributions to the logical development of the TX-2 computer.

In 1959 Mr. Gilmore became co-founder and vice president of Charles W. Adams Associates, Inc. Since then he has been actively engaged in the logical design of computer systems, the development of techniques for employing automatic digital computers in the processing of graphical data, and the further refinement of dynamic data processing programs and techniques.

FOR PROGRAMMING CHAIRMAN

Richard McQuillin

Mr. McQuillin received a B. Sc. (1955) at the University of Puget Sound, and an M.Sc. (1959) at Brown University. His major fields were Physics and Mathematics.

Mr. McQuillin joined the staff of Bolt Beranek and Newman Inc. specializing in physical acoustics in 1958. The earliest computer experience was with an LGP-30, in which he was involved in writing a compiler to process complex algebraic statements. When BBN acquired the first PDP-1 computer, Mr. McQuillin's interests turned to this machine. Major computer activities included work on the Floating Point Package (DECUS No. 10), and on DECAL-BBN. Mr. McQuillin headed the work on the DECAL-BBN Project.

Since June 1964, Mr. McQuillin has been with Inforonics Inc., where he is presently director of computer applications. His major interests include computer-aided printing as well as programming languages.

FOR EQUIPMENT CHAIRMAN

John B. Brown

J. B. Brown received his B.S. in electrical engineering at M.I.T. in 1955 and M.B.A. at Harvard Graduate School of Business Administration in 1959.

Mr. Brown was Systems Engineer at the Instrumentation Laboratory (M.I.T.) from 1955-1957; Computer Design Engineer at Digital Equipment Corporation from 1959-1962. He was an independent consultant in 1962 before coming to Bolt Beranek and Newman Inc.

At the Instrumentation Laboratory (M.I.T.), Mr. Brown participated in research programs in analog fire control systems for the Air Force. At the Digital Equipment Corporation he had responsibility for the design and construction of digital computer input-output equipment. As an independent consultant, Mr. Brown helped several companies in specifying and designing computer products. At Bolt Beranek and Newman Inc. he has continued his work with computer oriented products and man-machine systems.

DECUS CANDIDATES 1964-65

President - William Fahle
Secretary - Elsa Newman

Committee Chairmen:

Meetings - John T. Gilmore
Programming - Richard McQuillin
Equipment - John B. Brown

DECUS Delegates will receive ballots for voting in August. The new officers will be installed at the Annual Meeting in Dayton, Ohio on September 25, 1964.

DECUSCOPE
INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

Digital Equipment Computer Users Society
Maynard, Massachusetts

LETTERS TO THE EDITOR

From: John Mott-Smith, Decision Sciences Laboratory, AFCRL

Re: Comparison Instruction Generator in DECAL-BBN

"In reviewing the uncertified tapes of DECAL-BBN which you recently sent me the following bug was noted. The comparison ig's > < ≥ ≤ fail to check overflow and hence can cause errors if two numbers of opposite sign and large magnitudes are compared. A correction tape for these ig's and the symbolic listing are enclosed. The user should be careful to clear overflow before running a program using these ig's. Care must also be taken in using the comparison ig's together with arithmetic instructions which may also overflow. I haven't yet gotten around to checking what the various floating point interpreters do with these new ig's or how they handle overflow, but this should be checked also."

<u>Symbolic Listing</u>

```
jm-s, 8 june 64

. . .correction tape for comparison ig's which use add or sub
. . .takes overflow into account
xsy < > ≥ ≤
> dig beg lv4 op2 rs1 nlc
                    lac 2
                    sub 1
                    fde; szo
                    fde lst; cma                end

< dig beg lv4 op2 rs1
                    sub 2
                    fde; szo
                    fde lst; cma                end

≥ dig beg lv4 op2 rs1
                    sub 2
                    fde; szo'
                    fde lst; cma                end

≤ dig beg lv4 op2 rs1 nlc
                    lac 2
                    sub 1
                    fde; szo'
                    fde lst; cma                end
fix.
fin.
```

WELCOME DECUS DELEGATES

<u>PDP-4</u>

Reginald Gwin
L. W. Weston (Member)
Oak Ridge National Laboratory
Union Carbide Nuclear Division
Oak Ridge, Tennessee

<u>PDP-5 Delegates</u>

Stephen Carr
R. S. Langer
University of California, Berkeley

Phillip R. Carter, Brown Eng. Co.
Bradley Harder, Bendix Corp.
Billy J. Doran, MSFC
Huntsville, Alabama

(PDP-5 Continued)

W. H. Highleyman, Data Trends
Parsippany, New Jersey

Otto H. Schmitt
Univ. of Minnesota, Minneapolis

Joseph Naughton
University of Pittsburgh

Donald A. Molony
Edward Della Torre (Member)
Rutgers, New Brunswick, N. J.

George J. Safford
F. J. LoSacco (Member)
Union Carbide Nuclear Division
Tuxedo, New York

DECUSCOPE
INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

October 1964 Vol. 3 No. 10

THE ANNUAL MEETING IN BRIEF

The DECUS Annual Meeting (Sept. 24, 25) closed a busy 1963-64 season for DECUS. It was the fourth technical meeting* this year. Meeting headquarters was at the Stratford House, Dayton, Ohio. Dr. Hans Oestreicher, WPAFB, and Mr. William Fahle, SRL, were obliging hosts. Two special luncheons were served in the Crown Room and a dinner was attended by the delegates on September 24.

Mr. Joseph Lundy, DECUS Meetings Chairman, opened the meeting. Col. Burt Rowan of Wright-Patterson Air Force Base welcomed the group to Dayton. Computer applications in medicine and biology were discussed on the first day which included an afternoon at the Aerospace Medical Research Laboratories, Wright-Patterson Air Force Base. Demonstrations of the research in biological and neurological phenomena were observed. The demonstrations followed a lively session with panelists -- Prof. William Uttal, University of Michigan, Mr. David H. Brand, Systems Research Laboratories, Dr. Joseph Mundie and Dr. Hans L. Oestreicher of Wright-Patterson Air Force Base. Dr. Mundie reminded the computer enthusiasts that even the PDP computer had a few shortcomings when making medical diagnoses. Though the best man-machine systems were not perfected yet, there was general agreement that the PDPs were doing certain functions better than man. Details of the discussion will be reported in the 1964 DECUS PROCEEDINGS now being prepared for distribution. In addition to the panel presentation, fifteen other papers were given. Abstracts of these papers were published in the September DECUSCOPE.

Papers on the second day included the PDP-1's role in decision making, oceanographic data acquisition, natural language processing, automatic justification and hyphenation and a FORTRAN compiler. The afternoon was devoted to papers on the PDP-5.

---

*PDP-4 Symposium on Process Control
Annual Spring Symposium
PDP-5 West Coast Regional Seminar

PDP-4 AND PDP-5 IN DIRECT DIGITAL CONTROL SYSTEMS

Gerald E. Mahoney
The Foxboro Company

The Foxboro Company is currently developing two Direct Digital Control Systems. One system is being built at the Digital System Division at Natick, Massachusetts for Foxboro's Research Department, which is engaged in a joint venture with a leading chemical producer. The system, one of Foxboro's M/97600 Digital Computing Systems, includes a PDP-4. The other Direct Digital Control System is under development at Natick, Massachusetts under a joint project with the Esso Research and Engineering Company. The second system makes use of a Foxboro M/97400 Digital Computing System, containing a PDP-5.

Direct Digital Control - also called "time-shared" control - is the use for process plant control of a digital computer instead of multiple analog controllers which have been used conventionally over the years.

The conventional automatic controller variable operates in a single control loop, receiving a measurement of a controlled variable and determining how much it is in error. It then applies the correction to a control valve, measures the effect of the change by feedback and applies further corrections as needed.

In a direct digital control system, the computer samples each controlled variable at intervals through a "multiplexing" or switching mechanism. It converts the sampled measurement to a digital signal, comparing it with the desired control point. The difference, or error, enters into the controller equation in the computer which calculates the corrective valve position and issues a command to the valve. In this way, the computer is "shared" among all the control loops, eliminating the need for separate controllers.

DIGITAL'S PDP-8 COMPUTER

Digital Equipment Corporation's new small, high-speed computer, PDP-8, was announced October 12 at the meeting of the Instrumentation Society of America. The low price of $18,000 includes a 4K random access magnetic core memory with a complete cycle time of 1.75 μsec. (285,000 add./sec.), Teletype 33 ASR and a software package, tested and debugged on PDP-5 systems. The PDP-8 and PDP-5 are program compatible.

It is with regret that we must announce the resignation of Mrs. Elsa Newman as DECUSCOPE Editor.

DECUS is a cooperative effort on the part of many people, but there is no doubt that Elsa's work has been the greatest single factor in the growth and development of the organization. She has served as DECUS Secretary since shortly after the Society was organized. Almost single-handed, she developed DECUSCOPE and has produced it on a regular basis since April 1962. Her untiring efforts have been a major factor in the success of the various DECUS meetings during the last few years. The editing and publication of the proceedings of the meetings has been another one of her contributions.

All of this has taken a great deal of work on Elsa's part and she has given unstintingly of her time and energy. She will be leaving behind her a growing, active, professional society for which she can justly take a great deal of credit.

Elsa will not be easy to replace, but DECUS, represented by President William Fahle, and DEC are working on the problem. In the meantime, the day to day operations of the Society will be maintained.

William Fahle
DECUS President

Robert Beckman
Manager, Customer Relations (DEC)

DECUS SPRING MEETING 1965

The Center for Cognitive Studies of Harvard University has offered its facilities for the DECUS Spring Symposium. DECUS is pleased to accept the invitation and we look forward to an interesting meeting in the Spring.

WEST COAST REGIONAL DECUS SEMINAR

A West Coast Regional DECUS Seminar was held on September 9, 1964 at Lawrence Radiation Laboratory, Berkeley, California. Approximately 26 users were present.

Mr. Sypko Andreae is compiling a proceedings which will be distributed to all PDP-5 users as soon as it is ready. The participants and subjects discussed were:

Tony Schaeffer - UCLRL, Berkeley,
"PDP-5 Software Activities at LRL"

Philip Bevington - Stanford University,
"Stanford Computer for Analysis of the Nuclear Structure"

Steve Carr - University of California, Berkeley,
"The Berkeley Time-sharing System"

Jon Stedman - UCLRL, Berkeley,
"Spiral Reader Control and Data Acquisition with PDP-4"

Bob Swenson - UCLRL, Livermore,
"A PDP-5 Pulse-height Analyzer with Special Equipment"

Sypko Andreae - UCLRL, Berkeley,
"PDP-5 Systems with Nuclear Physics Equipments"

Lloyd Robinson - UCLRL, Berkeley
"Data Processing Installation at the 88" Cyclotron"

Ken Larsen - Digital Equipment Corporation, Los Angeles
"New Applications for the PDP-5 in Nuclear Physics"

A general tour of Lawrence Radiation Laboratory was held for those interested.

LETTERS TO THE EDITOR

From: Tony Schaeffer, Lawrence Radiation Laboratory, Berkeley

Re: PDP-5 Programming

At the Radiation Laboratory in Berkeley, we are working on the following programs for the PDP-5.

1. An assembler that runs on the IBM 7000 series.

   This program is operative, but is still evolving.

2. A debug package. This is an amalgamation of small useful programs put under keyboard control.

   These are:

   a. Insert octal constants
   b. Type octal (one word per line)
   c. Dump octal (eight/line plus address)
   d. Rim punch
   e. Binary punch with parity
   f. Binary loader with parity

   We have gone to a binary loader with a parity bit. This program halts on a parity error.

3. Memory dump on the CRT

4. Disassembler from rim paper tape to the typewriter

5. An elaborate paper tape reader test program

These programs are in varying degrees of completion, but I would be glad to correspond with people about any of them. When they are finished, I will submit them formally to DECUS.

CATALOG ADDITIONS FOR PDP-1

DECUS NO. 74

Tapelibrary Program

Author: Daniel Forsyth, Information International, Inc.

Tapelibrary Program, DECUS No. 74, stores the image of punched paper tapes on magnetic tape and reproduces them on command. Read checking is performed for paper-mag and mag-paper conversion. Records may be added to mag tape at any time, but not rewritten. Initial typed instructions are shortened when an operation is repeated. The program occupies 0-3117 registers.

DECUS NO. 75

SEETAPE - A Magnetic Tape Dump Program

Author: John B. Goodenough, DSL

SEETAPE is a PDP program for dumping magnetic tape information on the scope. The user may specify the structure of the tape block to be dumped, space forward and backward any number of blocks, and rewind the tape. Binary versions are available for Type 51 control and Type 52 control (SEETAPE-51 and SEETAPE-52).

Equipment Required:

1. At least two cores
2. Type 30 scope with Type 33 character generator
3. Standard Typewriter
4. Type 51 or Type 52 Tape Control

DECUS NO. 76

A 28 Bit Floating Point Package for the PDP-1

Author: Richard D. Smallwood, AFCRL

Purpose: To provide a 28 bit mantissa and 8 bit exponent floating point numbers for more accuracy than the standard 18-18 package. Performs floating point arithmetic, conversion, input, and output.

Typical Call:

```
(floating add)
lac a
lio a + l
jda fadd
lac b
```

Storage used: Various depending on which routines used, but a total of 1673 octal.

DECUS NO. 76 (Continued)

Special hardware required: Automatic multiply and divide

Typical execution time: (for fadd) 1100 μsec.

Programming Language: AMP, DECAL

Transparency: AC: Uses them IO: Flags - Yes, Overflow - No

Subroutines Required: tpo

System symbols defined: fadd, fsub, fmpy, fdvd, r??, ø??, f??, fx?, float, ifix, flip, flop, flex, fpov, fpun, fdve, ifxe

Comments:

1. The following operations are available:

a) add	e) float
b) subtract	f) fix
c) multiply	g) input
d) divide	h) output

2. The following error diagnostic are used:

   a) overflow
   b) underflow
   c) divide by zero
   d) fix (number too big)

3. The system uses roundoff in the arithmetic operations.

4. The arguments to the arithmetic routines can reference across cores.

DECUS NO. 77

DSL SORT ROUTINES - SORT 2, SORT 3

Author: John B. Goodenough, DSL

SORT 2

Purpose: Order (ascending) a two dimensional array

Call: sort2 (A, N, W, b, s, t)
A is name of an N x W array
N is number of items to be sorted
W is number of computer words per item
b is register where sort key begins (1 if the first register)
s number of consecutive registers in sort key
t sort type
= 0 for alphabetic (Concise coded) key
= 1 for signed 17 bit key (algebraic)
= 2 for positive 18 bit key (magnitude)

Transparency: AC: no IO: no Flag 6: used but status is preserved

Storage Used: 217_8 Subroutines Required: None

Special Hardware: These routines will work in either single core or multi-core machines. The program must be called in extend mode with the operands referencing the proper core in multi-core machines.

SORT 3

Purpose: Order (ascending) a two dimensional array

Call: sort3 (A, B, C, D, N, R, n, x_1, y_1, z_1, . . ., x_n, y_n, z_n)

A is name of array to be sorted - an N x R array
B is name of output array of dimension N x R not the same as A
C is name of array with N registers available for storage
D is name of an array with 3n + 2 registers available for storage
N is number of items to be sorted
R is number of computer words per item
n is number of keys

There are n triples describing the keys in the following form:

x_i beginning of i^{th} sort key in item (x_i = 1 means the key begins with the first register of the item)

y_i number of (consecutive) registers in the i^{th} key

z_i sort type for i^{th} key
= 0 for alphabetic key
= 1 for positive 18 bit key (magnitude sort)
= 2 for signed 17 bit key (algebraic sort)

Transparency: AC: no IO: no Flag 6: used but status is preserved

Storage Used: 465_8 Subroutines Required: none

Special Hardware: These routines will work in either single core or multi-core machines. In multi-core machines the program must be called in extend mode with the operands referencing the proper core.

DECUS NO. 78

TAPE 52 Magnetic Tape Control Subroutines

Author: John B. Goodenough, DSL

Abstract: This report describes a set of subroutines which exist on a library tape for using the tape transports connected to the Type 52 Control unit. The routines allow the user to read or write, in even or odd parity, a block of consecutive words in core as one tape block, write and detect on IBM end-of-file mark, space forward and backward, rewind, maintain a block count and automatically take corrective action when errors are detected during read or write operations. All routines may be used in extend mode and called from any core.

Equipment Required: Type 52 control designated as MTCU O, modified so that bit 15 of the status register is a one if a longitudinal parity error is detected and bit 9 is a one if end point on tape is detected.

Additional Programs Required: tpo tdn

These routines are on the standard DSL library tape and will be called automatically.

Title	Purpose
block	Define registers for block counts
read	Read a DECAL-BBN array from tape into core
write	Write a DECAL-BBN array from core onto tape
rewind	rewind tape unit
space	Space tape unit forward and backward N blocks
weof	Write an IBM end-of-file mark on tape
feof	To search for an end-of-file mark
mgrd	Read a tape block into a specified area of core memory
mgwrt	Write a specified core block onto tape as a tape block
mgbks mgfws	Space tape forward or backward a specified number of blocks

From: Jon D. Stedman, Lawrence Radiation Laboratory, Berkeley

Re: A Short Loader for the PDP-4

Enclosed is a write-up of a simple loader for those lazy souls who don't like to key-in the Rim loader. It is appropriately named the "Flynn Loader" because two words and you're in.

As the PDP-4 has no "LOAD" button, it is necessary to have a small resident loading program in core storage. This loader, called the Read-In Loader, will load binary tapes of the following format:

```
dac A
c (A)
dac B
c (B)
  .
  .
jmp Y
 Ø
```

Most tapes in the "funny format" are preceded by the Funny Format Loader punched in the read-in format.

In the past, whenever anything happened to the Read-In Loader, it was necessary to key in 13 words--unpleasant if the loader is lost very often. A few sessions with the accumulator switches directed thought toward a shorter loader.

The final result is the Flynn Loader--only two words long.

Operation

1. Fill core storage with zeros.
2. Deposit 700156 in location 21.
3. Deposit 060010 in location 22.
4. Position the loader in the reader so the first sprocket hole before the loader is above the photo-cells.
5. Place the computer in medium speed single step, repeat mode.
6. Press start. (The address keys may be set to anything except 22.)
7. Put the continue switch up.

Method

The two-word program is: 21/RRBVRSB
22/DAC I 1Ø

1. The contents of the reader buffer are read into the accumulator and the paper tape reader is started.

2. The contents of the accumulator (garbage) are deposited in location 1.

3. The operation code of the CAL instruction is ØØ, so control is returned to location 21.

This loop reads the following instructions into core storage:

```
Ø2/ 7ØØ157    RSBVRSFVRRB
Ø3/ 6ØØØØ2    JMP 2
04/ Ø6ØØ11    DAC I 11
Ø5/ 6ØØØØ2    JMP 2
Ø6/      Ø
Ø7/      Ø
1Ø       1Ø
11/ 017761
.
.
21/ 600002    JMP2
```

(Continued on Page 4)

Heard at the DECUS Dayton Meeting

- Programmers underestimate how long it takes to write a program.

- Procedures are working well.
- Has DECUS reached maturity?

- One learns a lot more at meetings like these than at the big ones.

- What about the PDP-8?

- Let's have more technically oriented information about new DEC products.

- Could you put me on the mailing list for DECUSCOPE?

- You must be an analog man.
- No, I'm a digital man; I'd rather switch than fight.

NEW DECUS MEMBERS

Delegates

PDP-1

Darrell W. Felty
AFMDA - MDSCP
Holloman AFB, New Mexico

John W. McLaughlin
RCA
Moorestown, New Jersey

Cyril Broude
Chalk River Nuclear Laboratory
Ontario, Canada

PDP-4

Jon Stedman
Lawrence Radiation Laboratory
Berkeley, California

PDP-5

Leonard W. Shinn
EE2, NASA - MSC
Houston 1, Texas

Stephen Stasak
Carnegie Institute of Technology
Pittsburgh, Pennsylvania

(Continued on Page 4)

(Continued from Page 3)

PDP-6

D. W. G. Moore
Computing Center
Univ. of Western Australia
Nedlands, Western Australia

PDP-7

Prof. Philip R. Bevington
Dept. of Physics
Stanford University, Calif.

Individual Members

John P. Butler
Atomic Energy of Canada, Ltd.
Ontario, Canada

Linwood M. Culpepper
U.S. Naval Weapons Laboratory
Dahlgren, Virginia

Richard DeJohn
University of Michigan
Mental Health Institute

Duncan E. Morrill
Joseph C. Michael
Woods Hole Oceanographic Institute
Woods Hole, Massachusetts

Donald V. Weaver
Box 567
New York, N. Y.

Ronald E. Weisdock
RCA
Moorestown, New Jersey

Dr. Norman Weissman
NASA - Ames Research Center
Moffett Field, California

Larry McGraw
Donna Smith
Carnegie Institute of Technology
Pittsburgh, Pennsylvania

Published Monthly
for
Digital Equipment Computer
Users Society
(DECUS)
Maynard, Massachusetts

Contributions to DECUSCOPE should reach the Editor before the 7th day of the current month for publication in that issue. Material received after the 7th will be considered for publication in the months following.

Elsa Newman, Editor

(Continued from Page 3)

When "JMP 2" is read into location 21, control is transferred to the short program just read in, which reads in the Read-In Loader. The program counter "wraps around to zero" after the Read-In Loader is read in, and several words of "JUMP Ø" stop the loading loop.

MULTI-CLOCK SIMULATION SUBROUTINE

Jon D. Stedman
Lawrence Radiation Laboratory
Berkeley, California

A real time program may have several external devices which have different reaction times that require the program to maintain a timing surveilance on each device. A clock feature is the hardware solution, but it is unlikely that a computer system will have a separate clock for each external device. Hence the programmer must fill the gap between only one clock and numerous simultaneous users of the clock facility. A subroutine that would simulate any required number of clocks is sketched here.

First, initialize the Clock Simulation Subroutine. A time increment "TICK" is defined to be some multiple of the real clock time interval. In the case of the PDP-4, TICK = N * (1/60 sec). All clock requests must be cancelled and all banks and flags put into initial condition. Since there are no requests for clocks in this condition, the real clock will be disabled initially.

Clock Request Calling Sequence

```
LAM -T+1         / T = number of TICKS
JMS CLOCK        / ROUTINE is a program to process when
JMP ROUTINE      / time has run out.
```

When a clock request is made, the real clock will be enabled and set to interrupt in a time defined by "TICK." Each clock request establishes a new entry in a clock bank and an associated entry in a JMP ROUTINE bank. When a real clock trap occurs, the CLOCK routine will increment all of the entries in the clock bank. Assuming first that none of the entries have been reduced to zero, a check will be made to determine if any previous JMP ROUTINE has not returned to CLOCK as it is required to do. If the ROUTINE does not return to CLOCK before the next "TICK" than a fault will be recorded and a return is made to the previous "trapped" program. If any of the clock entries became zero, then the associated JMP ROUTINE will be placed into a JMP list. Clock requests that have been ticked to zero are naturally removed from the request bank. Briefly, the real clock traps every TICK into the CLOCK routine which then performs the service functions for all of the requests in the clock bank.

DECUS PROGRAM LIBRARY FIGURES - 1964

247	Requests for Programs
247	Requests for Programs Filled
25	New Programs Contributed by Users
28	Programs in Review by Users and in Process of Certification by Programming Committee

The 1964 Library Catalog will contain a complete listing of all DECUS Programs.

DECUSCOPE

INFORMATION FOR DIGITAL EQUIPMENT COMPUTER USERS

December 1964 Vol. III No. 12

Season's Greetings

DECUS SPRING TECHNICAL MEETING

HARVARD UNIVERSITY

May 20 and 21, 1965

The two-day DECUS Spring Technical Meeting will be held at William James Hall, Harvard University, Cambridge, Massachusetts on May 20 & 21, 1965. Dr. Donald Norman, Center for Cognitive Studies will be host. This year's meeting will have an open theme in order to encourage papers on the various phases of the computer industry. Presentations on all the Programmed Data Processors are welcome.

Call For Papers

Authors are asked to submit a 75-100 word abstract of their paper before February 15. Accepted papers will be notified by March 15. Papers in final form must be submitted ...he DECUS office on or before the 1st of ...il. Authors may plan on approximately twenty minutes for their presentation and ten minutes for a question and answer period.

When submitting papers, please send an

(Continued on Page 2)

A BRIEF DESCRIPTION OF THE PDP-6 EXECUTIVE SYSTEM

Harrison R. Morse III
Digital Equipment Corporation

INTRODUCTION

One of the more important needs of the programming system being built for the PDP-6 was the necessity to run on an extremely wide range of configurations. This need affected the design of the system in a number of ways, and had a particularly strong effect on the processor-system interface and the actual implementation techniques used in the I/O section of the system.

As a result, each processor (MACRO, FORTRAN, etc.) is independent of the I/O configuration on which it must run, and all accept commands controlling their respective processors in nearly identical forms.

This was made possible by choosing a level of I/O interface at which all I/O devices could be treated similarly when used for linear I/O (the normal use), but still retained enough control to allow replacing data within files on microtape, for example.

This also means the I/O system routines may be changed in a central place affecting only the efficiency of operation and not changing the interface to the programs.

SYSTEM I/O

The System I/O was designed to make the handling of input (and output) from physically dissimilar devices to as similar as possible. In particular, for normal I/O (standard ASCII or binary) there are only two classes of devices; devices on which the data is retrievable, (i.e. magtape, DECtape, drum and disc) and devices on which the data is not immediately retrievable (the hard copy devices).

In use, the difference is apparent by the fact that on retrievable devices data is contained in named files, while on the hard copy devices, the only file is the "next one."

Primary Commands

The five basic I/O commands available to the user are:

INIT	Select and initiate a device.
CLOSE	Terminate output or input on a device.
RELEASE	Release a device for use by other users of the system.
INPUT	Perform and/or synchronize with input for the device.
OUTPUT	Make available data for output to the device.

A user selects and initiates device by use of the INIT command, which includes as an argument the name of the device in which he is interested. If the device is unassigned or assigned to the requesting user, then it is placed in operation and assigned to one of his 16 I/O channels. Future commands to the device refer to it by its channel number.

The CLOSE command terminates output on the device by performing whatever action is necessary to output the remainder of the data. If CLOSE is applied to an input device, it insures that the next input command will input new data by removing any residual data from the buffers at the time the CLOSE command is given.

(Continued on Page 2)

original and two carbons. Original should be either on reproducible masters or typewritten on bond paper suitable for reproduction. It is planned that copies of papers will be available at the meeting.

For further information, please contact the Meetings Chairman, Mr. John T. Gilmore, Jr., Charles W. Adams Associates, Bedford, Mass. or Angela Cossette, DECUS, Maynard, Mass.

NIGMS OFFERS FILM ON NEW COMPUTER DEVELOPED FOR HOSPITALS

A 38-minute film showing the operation of a pilot version of a new computer-based system designed to simplify handling of communications and medical records in hospitals has been produced by the National Institute of General Medical Sciences.

Copies of the black and white, 16 mm. film are available on a loan basis from the Information Office, NIGMS, National Institutes of Health, Bethesda, Md. 20014.

Demonstrated at NIH

The film was processed from an edited videotape made during a demonstration at NIH of the experimental system being developed by Bolt Beranek and Newman Inc., and Massachusetts General Hospital. Work started on the system two years ago with contract support from NIGMS and the American Hospital Association.

Unique in many respects, the prototype system consists of a central time-shared digital computer in Cambridge connected through private telephone lines to a number of teletypes located on nursing stations and in the pharmacy at the hospital in Boston.

The system is presently designed to handle 60 teletype stations simultaneously within Massachusetts General Hospital. Since the capacity of the computer is greater than that required by this hospital, the system is potentially capable of providing similar service to a number of other hospitals in the area on a shared-time basis.

The system accepts messages from all stations, checking for discrepancies against previously "memorized" criteria. After checking, it stores the data in its memory as a permanent part of the patient's records, while automatically distributing the new information to authorized teletype stations.

By maintaining a comprehensive record of patient-care that is readily accessible in a variety of combinations, the system will be a useful resource in clinical research and in studies of patient-care procedures.

Reprinted from the NIH Record.

RELEASE causes a device to be deassigned.

INPUT will initiate input if the device is not currently performing input, and in any case insures that a buffer full of data is available for processing by the user before returning. Once a device is started, it remains in operation until the entire buffer area becomes full, so further INPUT commands normally serve only to syncron operation of the user's program with the I/O device.

OUTPUT does the final preparation to make data available for output by the system. In addition, OUTPUT insures that there is space in the user's buffer area for storing more data before returning control to the program.

Data Modes

There are a number of modes in which data from a device may be accessed. The three primary modes are:

ASCII data	-	data which is packed 5 characters per word, left justified;
BINARY data	-	one 36 bit word of data per word;
DUMP	-	applicable to retrievable devices in which the data consists of continguous records on the device of the size specified by the user or the data.

Auxiliary Commands

There is a pair of commands (ENTER, LOOKUP) which serve to enter and retrieve file names from file directories. These commands, if given for the hard copy media (papertape, cards, line printer, teletype), will act as if the name were always found in the case of LOOKUP or always entered into the directory in the case of ENTER to allow all I/O devices to be treated identically.

STATUS Commands

There are other commands for testing the status of an I/O operation which corresp exactly to the hardware condition testing commands - (D is the device-channel).

STATZ D, E	skip if all the indicated bits are zero
STAT0 D, E	skip if any of the bits are one
STATUS D, E	read the status into E

Buffer Areas

A buffer area consists of a series of buffers, linked in a list which is set up in the users's area by either explicit or implicit request. The I/O device fills successive buffers in the buffer ring until the entire buffer ring is full. Since only the state of the current buffer kept is stored in the device routine, the number of buffers in the ring is irrelevant to the system. The motivation for designing a buffering scheme of this sort is to allow the buffering efficiency to be determined by the user's choice of the number of buffers in the ring.

The state of each buffer area is defined by a triplet of words which contain the address of the buffer being dealt with, a byte pointer which when incremented will point to the next item within the buffer (either a character or a full word), and an item count of the number of items in a buffer (number of item spaces in a buffer in the case of output).

Since all I/O devices use the same buffering scheme and the input/output commands set up the byte and item count from either the data on input or size of the buffer area on output, all devices may be treated in a similar fashion. This allows a processor such as an assembler or compiler to be written independent of I/O. Therefore, the processing portion of these programs need not change for different configurations, an important consideration since the PDP-6 may exist in such a large variety of configurations. In addition, since the interface of all I/O routines to the user is the same, the I/O routines may share in common a very large portion of code which c trols the I/O and only the data handling portions of the I/O service routines need written for each device.

PROCESSOR-USER INTERFACE

Processors within the system (such as MACRO, FORTRAN, ATLATL, EDITOR, and

LOADER) are designed to interface with the user in a common fashion. They all take a specification of their source and destination files from a command string, which, in the case of a time-sharing system, may be typed on the user's console. A command string contains information such as the device, and file name of any file, the data modes in which data is to be input or output if applicable and certain control information which is processor dependent. The general form of the command statement is identical for all processors, i.e. a command statement consists of the destination file list with each destination file separated by a comma, a left arrow indicating the files in the list to the left are destination files, followed by a source file list where the source files are separated by commas. For example: The command string to MACRO

DTA4: ZORCH, LPT: ←DTA2: A,B,C

says, "assemble files A, B and C from DECtape 2, putting the object program on DECtape 4, file ZORCH, and the assembly listing on the line printer." If devices are to be specified, then the device name is included followed by a colon. Files are on the last specified device if no other device is specified. Control information may be enclosed in parentheses wherever it applies.

In addition to giving the actual command, the user may specify a command file by use of the at sign @, which means the device/file specified after the at sign is to be processed as commands at this point. Users may then prepare and edit a complete specification of how to compile, assemble, load, dump, run a job, and cause it to be executed whenever desired.

This feature is also being heavily used in the construction of all the system components and the system.

SYSTEM MODULARITY

It is worth noting that this design insures that the processors are independent of the configuration on which they are to run, with the possible exception of storage size (there is a smaller version of the Fortran compiler which runs under the system in a 16K machine). In addition, since there is a common interface between system programs and the system, the non-existence of an I/O device in a configuration merely means the user of a processor may not specify input and output to that device. (If he does, a message "DEVICE XXX NOT AVAILABLE" is typed. However, the system programs do not change.)

SYSTEM USER INTERFACE

The system is designed to run all programs in user mode, in which all halts, input/output instructions, UUO's and memory protection violation trap to the executive system. Halts terminate a program with an appropriate message to an on-line user. Hardware I/O commands are illegal. UUO's either initiate system functions or are returned to the user to be processed by him. The UUO's (UUO's are unused operation codes, i.e., instruction codes 000-077) trap to the monitor and may be processed there. Half of these instructions (codes 040-077) are reserved for monitor use, where the other half (000-037) are available for the user to use as he sees fit.

One UUO labelled CALL has been reserved as a general system subroutine call, the address of which points to a word containing, in 6 bit ASCII, the name of the system subroutine being called. The executive system does a symbol table search on this name (in a hash-coded symbol table) and transfers control to the called subroutine. In this way, the user has access to system subroutines without the necessity of knowing where they are located within the monitor, or as is planned for more advanced systems, even if it is resident (the system will load called, non-resident subroutines).

The following is an example of a program to copy paper tape to punched cards.

```
TITLE PTR TO CDP

        GO:    INIT 1,
               SIXBIT /PTR/
               EXP PTRBUF          ;buffer state words
               JRST 4,.            ;device not available return
               INIT 2,
               SIXBIT /CDP/
               XWD CDPBUF, Ø       ;output buffer state words
               JRST 4,.
```

(Continued on Page 4)

NEWS ITEMS

For those interested in time sharing, an article by E. L. Glaser and F. J. Corbato of M.I.T. entitled "Introduction to Time Sharing" appeared in the November issue of DATAMATION.

The Physics Department of the University of Arizona has ordered a PDP-5 for use in studying cosmic ray phenomena. The PDP-5 will perform on-line scanning of wire spark chambers. The chambers will function with particle detectors to record the passage of each particle. The computer will make a permanent record of all the events and will be used to perform kinematic analyses seeking to identify the particles.

Atomic Energy Commission's Oak Ridge Y-12 Plant in Tennessee, currently under construction, has ordered a PDP-5 which will be the control element in a new, automatic, high-precision measuring system for use in their Physical and Electrical Standards Laboratory.

Lawrence Radiation Laboratory has installed a PDP-5 system for use in on-line instrumentation of nuclear reaction studies on the department's 88-inch cyclotron. The laboratory is operated by the University of California for the Atomic Energy Commission. The computer system will be used by the laboratory's Nuclear Chemistry Department at Berkeley, California.

Others joining the ranks of PDP owners are:
DESY, Germany - PDP-5
University of Tokyo - PDP-5
University of Delft, Germany - PDP-7
Max Planck Institute for Physiki, Germany - PDP-8

A DECUS Executive Board Meeting was held on December 14 at Charles W. Adams Associates, Bedford, Mass. Several topics were under discussion including various questions regarding the DECUS Technical Meetings. The next issue of DECUSCOPE will contain a report of the meeting.

PDP-5 Seminar at Rutgers University

Professor Donald A. Molony of Rutgers University has expressed the desire to hold a PDP-5 Seminar at the University. A definite date has not been set, but Prof. Molony is planning for sometime soon after the first of February. As soon as details have been finalized, notifications will be sent to all PDP-5 users in the area. Information will also be published in DECUSCOPE.

All DECUS members are welcome to attend. Those interested in participating or attending may contact Professor Molony at the College of Engineering, Rutgers University, New Brunswick, New Jersey.

(News Items Continued on Page 4)

Literature Available

Brochures

"Bioelectrical Signal Anaylsis with PDP-4"

"Multichannel, Multiparameter Analysis with a General-Purpose Computer"

"Time of Flight PDP Multianalyzer"

PDP-5 - "Time - Sharing Keyboard - Computer Communications"

PDP-5 - "Compact Computer Speeds Strength Test Calculations"

PDP-4 - "A Personalized Letter Answering and Typing System"

"GRAPHpad - an experimental computer-aided design program" - GRAPHpad is a general-purpose computer program being developed by Digital Equipment Corp. to demonstrate the versatility, speed, and ease of communicating with a running system program through the display subsystem.

Reprints

"A Multiuser Computation Facility For Education and Research," by Jack B. Dennis of the Massachusetts Institute of Technology. Reprinted from Communications of the ACM.

"The PDP-6 - time-sharing hardware" by R. P. Harris, Digital Equipment Corporation. Reprinted from DATAMATION, November 1964.

"Data-Dial: Two-Way Communication with Computers from Ordinary Dial Telephones" by T. Marill, D. Edwards and W. Feurzeig, Bolt Beranek and Newman, Inc. Reprinted from Communications of the ACM.

"Computer Processes Tandem Van De Graaff Data" by J. Leng, Digital Equipment Corp. Reprinted from Canadian Nuclear Technology Winter 1964.

Computer Application Notes

PDP-1 - "General-Purpose Computer Gives Harvard Teaching Machine Professorial Logic and Rhetoric"

PDP-4 - "KEYDATA On-Line Facility Handles Business Applications"

"NABISCO'S Computer Controlled Batching System - An Industry First"

PDP-5 -"Experimental Computer System Monitors and Controls NRU Reactor"

Revised Bulletin - F53(153), Type 153 Automatic Multiply & Divide a standard option for the PDP-5 has, now been revised and is available from DEC Technical Publications Department.

*****

Copies of literature mentioned may be obtained by contacting Angela Cossette, DECUS, Maynard, Massachusetts.

```
LOOP:    SOSG PTBUF+2           ;more data
         INPUT 1,               ;no, get more
         STATZ 1, 1ØØØØ         ;end of data test
         JRST DONE              ;yes, end of data
         ILDB 1Ø, PTRBUF+1      ;get a character
         SOSG CDPBUF+2          ;output buffer full
         OUTPUT 2,              ;yes, output it
         IDPB 1Ø, CDPBUF+1      ;store character in output buffer
         JRST LOOP              ;continue

DONE:    CLOSE 2,               ;close output
         CALL [SIXBIT /EXIT/]

PTRBUF:  Ø                      ;buffer address word
         Ø                      ;byte pointer for input buffer
         Ø                      ;item count for input buffer

CDPBUF:  Ø                      ;buffer address
         Ø                      ;byte pointer for output buffer
         Ø                      ;available count for output buffer
              END GO
```

PDP-6 SYSTEM APPLICATIONS

With the delivery of the first PDP-6 system, a new class of computer user is found in DECUS. The PDP-6 is a medium to large scale computing system designed for computation center and large scale real-time use. Its basic spacifications boast 16 accumulators, 36 bit word, 16K to 262K directly addressable 2µs memory, 363 instructions and unique hardware to implement time-sharing effectively. The software includes MACRO-6 assembly program, FORTRAN II, FORTRAN IV, DDT-6, editor, loader, and complete time-shared monitor.

A quick look at the first eight "6" systems gives some insight into the uses of PDP-6.

System 1 is the prototype located at Maynard, Massachusetts. It is being used to develop more advanced time-sharing systems, and has been operational since Spring of 1964. Further use of the PDP-6 in Maynard is discussed by Mr. Harrison Morse in his article on Page 1.

System No. 2 has been installed at M.I.T.'s Project MAC since November 1964. They are planning to use the PDP-6 as an integral time-sharing component interfaced to the 7094 for the purpose of studying visual communication methods between man and machine. Dr. Minsky of the institute plans to use it also for development of programming languages and information retrieval.

System No. 3 is scheduled to be installed at Brookhaven National Laboratory this month. To be used in high energy physics work, this unique system is mounted in a large trailer and is completely mobile. The PDP-6 replaces the Merlin System for the purpose of on-line reduction of data derived from scintillation counter hodoscopes, sonic spark chambers and wire spark chambers.

The University of Western Australia, Nedlands, Western Australia will be receiving System No. 4 early in 1965. This will be the first computation center installation where the central processor will be time shared via remote teletype stations. The scientific computation center will be used in crystalography research and time sharing will aid student training in computer technology. The PDP-6 will also be used for accounting and information retrieval by the Administration Department.

The largest PDP-6 configuration being built to date is System No. 6. Keydata Corporation, a subsidiary of Charles W. Adams Associates, will install the system in Technology Square, Cambridge, Massachusetts. The Keydata facility, connected to terminals located on subscribers' premises, will function on-line in real-time to provide computer services to scientific and business users.

(Continued on Page 8)

Reprinted with permission from the November 1964 issue of DATAMATION
magazine by Digital Equipment Corporation, Maynard, Massachusetts.

THE SDC TIME-SHARING SYSTEM

facilities, services & potential

by JULES I. SCHWARTZ

In spite of all the power available in present day computers, a feeling of uneasiness, even disappointment seems to prevail among those concerned with paying for and using computers. These feelings stem primarily from the relative inaccessibility and inflexibility of the computer. The ability of users to produce computer systems that can be readily modified is frequently much less than supposed, and the ability of computers to afford quick and accurate solutions to a variety of even simple problems is surprisingly limited. Part of the reason for this dilemma is the formidable wall between users and computers in most installations. This wall is faced not only by managers and customers, but also by programmers.

One of the major reasons for the wall is the persistence of traditional techniques for applying computers. These techniques require that programs be prepared (in a language that is not usually oriented to application), be sent to the computer (where they are processed and executed with no intervention by the user), and, when processed, be sent back hours (or days) later. Aside from the fact that this approach can lead to numerous delays and wasted computer runs, there are many applications for which this detached operation is completely unsatisfactory. Thus the concept of continuous interaction by the user with his computer (*on-line* computer operation) promises to become an essential part of current techniques in computer application.

On-line use of a computer is not a new concept, for it has appeared in a variety of situations in the past; but these applications have been special purpose (e.g., SAGE) or quite limited (e.g., use of small computers such as the Bendix G-15 or the Digital Equipment Corporation's PDP-1). Complete user interaction may be available on small computers, but because of their properties, many desirable features of large on-line systems cannot be made available, such as access to many general- and special-purpose languages. Therefore, it seems, use of a large computer on-line would be desirable. Unfortunately, such use would introduce certain inefficiencies, without major changes in technique. For one person to preempt the total capacity of a large machine for long periods of time would be highly uneconomical if he could not keep the computer occupied all the time it was assigned to him. Time-sharing permits on-line use of the computer simultaneously by a large number of people by giving each user time when he requires it. This kind of system provides a direct and continuous working relationship between users and computer and keeps the computer busy most of the time by limiting the amount of idle time due to human thought or output from on-line devices.

In the Command Research Laboratory at SDC, a large percentage of the problems run are of the on-line, man-machine, interactive variety. Since these applications are virtually impossible to run in a serial fashion (one-at-a-time), the requirement to produce a system that would permit parallel running was a necessity. For this, and for the sake of further study of the time-sharing process itself, a time-sharing system was developed as the main program vehicle with which to run the laboratory.

characteristics of time-sharing systems

Given the general requirements for time-sharing systems, it may now be worthwhile to examine the properties (and hence the definition) of such systems. Four characteristics of time-sharing systems encompass most of their distinctive features; such a system is:

- Simultaneous—A number of people use the computer at the same time.
- Instantaneous—All users receive responses from their

Mr. Schwartz is head of the time-sharing project at System Development Corp., Santa Monica, Calif., with whom he has been associated since prior to its spin-off from RAND. His software development projects have included PACT, the Lincoln utility system, and JOVIAL (Jules' Own Version of the International Algebraic Language). He holds a BS in math from Rutgers and an MA in mathematical statistics from Columbia Univ.

programs or the system within seconds—or fractions of a second—of the completed computation.

- Independent—Different programs, services, and devices can be in use separately or in combination during any given period of time.
- General-purpose—No restriction is placed on the kind of program or the application under the time-sharing system.

Various on-line systems have some, but not all of the properties just described. Others, such as the SDC system, do have all of these characteristics.

equipment configuration of the SDC system

Since on-line operation of a computer requires the use of input-output devices in addition to tapes, card-readers, printers, etc., on-line devices available to users of the system will be reviewed first. These devices include teletypes, typewriters, and cathode-ray tube (CRT) display consoles. The teletypes have been installed in various locations within SDC and are also used from numerous remote locations, as far away as Pittsburgh, Pa. Within SDC, there are eight model 28 Teletypes, 16 model 33's and three Soroban typewriters. There also exists capacity for eight remote teletype stations operating simultaneously. In addition to three types of keyboard devices, there are six display consoles currently available. These are all located within the Command Research Laboratory.

The SDC time-sharing system may also be used remotely via a 2kc line, the terminal to which may be a computer. That computer may in turn service displays and keyboards. This linkage permits the remote computer to communicate with a program in the SDC computer, thus permitting computer-computer-human interaction under time-sharing. Currently one such line is operating between the AN/FSQ-32 in Santa Monica and a CDC 160-A computer at the Stanford Research Institute, Palo Alto, Calif.

The basic characteristics of the time-shared AN/FSQ-32 system are listed in Table I. Assisting the Q-32 under time-sharing is the PDP-1, which serves as the major interface between the on-line devices and the Q-32. A simplified diagram of the complex is shown in Fig. 2. The local Q-32 configuration is shown in Fig. 3.

Fig. ?

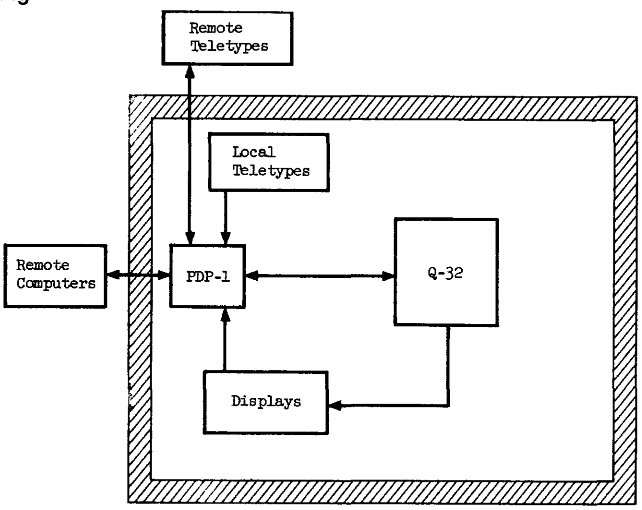


Fig. 3

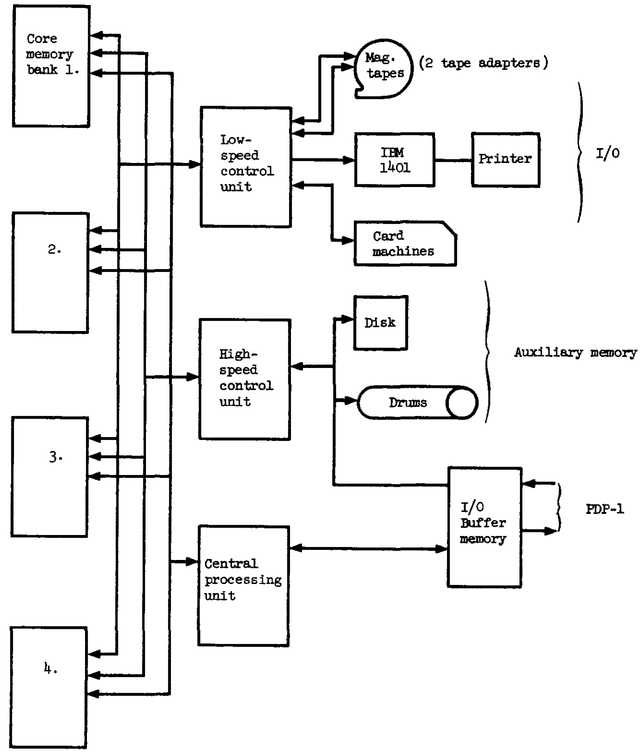


Fig. 1 Characteristics of the AN/FSQ-32 Storage Devices

DEVICE	SIZE	WORD RATE	AVERAGE ACCESS TIME
Core Memory	65K	2.5 usec/wd.	—
Input/Output Core Memory	16K	2.5 usec/wd.	—
Magnetic Drums	400K	2.74 usec/wd.	10 msec
Disc File	4000K	11.75 usec/wd.	225 msec
Magnetic Tapes	16 Drives	128 usec/wd. (High density)	5 to 30 msec (no positioning), depending on whether the tape is at load point, and whether it is being read or written.

the time-sharing system

To understand the services available to a user of the time-sharing system, a review of the system itself would be useful. A simplified description of the system is as follows:

All programs requested by users are stored on an assigned section of the drum until the user quits.

A program is given one "quantum" of time to operate, which represents the maximum time for one operating cycle. The program's turn ends when it requests an input-output transfer, when the quantum of time ends, or when an error condition arises. When the turn ends, the executive system determines if another program is ready to operate. If not, the program will reside in core memory until another program is ready. When another program is ready to operate (and it occupies an area of core which overlaps that of the first), the executive writes the first back to its place on the drum, and brings in the new program to core.

The process of exchanging programs from core to drum is called *swapping*.

The basic cycle of an execution followed by a swap is fundamental to the time-sharing technique of operation. Controlling this process is a program called the *executive system*.

The executive system resides permanently in 16,000 registers of core. It has the following major functions:

- Responds to all computer interrupts and takes the required action.
- Interprets inputs and commands from teletypes and typewriters.
- Allocates both core and peripheral storage.
- Performs all input-output.
- Schedules the running of object programs (as in the above example).
- Performs a number of on-line debugging services.

Object programs, which can contain up to approximately 47,000 words of storage, are placed in core storage only when they are active (ready to compute). When they are not active (stopped or waiting for input or output), they reside on the drums, placed there by the executive. As shown in Figs. 4 and 5, they are put on drums by the system when given the LOAD command (see below - Basic Commands) via teletype.

The Fix Program, which occupies approximately 2,000 words of permanent storage, reacts to all computer errors, logs the error on a typewriter at the maintenance console, attempts to fix the error, then returns to the executive system with a description of the error condition. The system then isolates the user(s) affected by the error, notifies the affected user(s), concludes the affected program(s), and then continues the normal cycle.

A major function of the executive system is to interpret and respond to the various teletyped commands made by a user. It is essential that most users know these basic commands. They are used to sign in, load, and run object programs, stop programs, and sign out. These commands are:

- LOGIN: The user begins a run. With this command he gives his identification and a "job number."
- LOAD: The user requests a program to be loaded (from either tape or disc). Once this command is executed, the program is an object program to the system.
- GO: The user starts the operation of an object program or restarts the operation of an object program that has been stopped. Once the user gives this command, he can send teletype messages to his object program or the time-sharing system.
- STOP: The user stops the operation of an object program.
- QUIT: The user finishes a particular job. Upon receipt of the command QUIT, the time-sharing system punches accounting information into a card and removes the object program from the system.

Figs. 4 and 5 are examples of this sequence as it would appear on a user's teletype. It will be noted that to each command from a user there is a response on the teletype from the system. Until this response is received, the user cannot assume that the system has reacted to the command. Normally, this response is immediate.

In the brief description of the LOAD command above, it was stated that the requested object program is loaded from tape or from disc. It is assumed, of course, that the program exists in binary form—expected by the executive to perform the LOAD function. There are a number of ways in which a program can be put into this form. Some of the more common will be explained in the section on service routines.

The sequence of actions taken after the LOAD command is given is as follows:

The Executive examines the name of the program, which follows immediately after the command LOAD. If it finds that this program has been stored previously on the disc, it brings it in from the disc, places it on the drum, and responds with $LOAD OK as in Fig. 4. If the program is not on the disc, it types out instructions at the computer console for the computer operators to load this program, or the specified tape reel (optional) for this teletype. It also types $WAIT on the user's teletype. When this operation is completed by the operator, the program is stored on drum and disc (for subsequent loads), and the response $LOAD OK is typed on the teletype. The time between the $WAIT and $LOAD OK can be a matter of minutes.

Fig. 4. Load From Disc

```
LOGIN 0050 JCX.25
 $OK LOG ON 10
LOAD PRG2             Object program on disc
$LOAD OK              Object program on drum and disc
GO
 $MSG IN.
!STOP
 $MSG IN.
QUIT
 $MSG IN.
```

Fig. 5. Loading a Program From Tape

```
LOGIN 1123 JBX.30
 $OK LOG ON 10
LOAD TPRD 1852
$WAIT.                Object program on tape
$LOAD OK              Object program on drum and disc
GO
 $MSG IN.
!STOP
$MSG IN.
QUIT
 $MSG IN.
```

Fig. 6. User Talking to Object Program

```
LOGIN 0050 JDX.25             User to executive
 $OK LOG ON 10                Executive response
LOAD CALC                     User to executive
$WAIT.                        Executive response
$LOAD OK                      Executive response
GO                            User to executive
 $MSG IN.                     Executive response
 CALC READY                   Object program response
+ 2,2                         User to program
 RESULT +4.00000000           Program response
* 3,3,3                       User to program
 RESULT +27.0000000           Program response
!STOP                         User to executive
 $MSG IN.                     Executive response
QUIT                          User to executive
 $MSG IN.                     Executive response
```

"talking" on a teletype

It is quite clear that once the GO command is issued, it is necessary for the user to communicate over the teletype with the object program. Therefore, until otherwise notified, the executive assumes that any input on the teletype is to the object program, not the executive, and therefore simply passes the input to the program, which reacts in some way, probably with an output on the teletype. Fig. 6 shows a typical sequence of steps taken when a user communicates with an object program.

There usually comes a time when the user must start talking to the executive after he has communicated with his object program—to give the STOP command, for example. The user signals this intent by typing an exclamation point. This action and its consequences are demonstrated in Fig. 6. Correspondingly, it is frequently necessary to recommence talking to the object program, having once used the exclamation point. This is done by typing quotation marks, as shown in Fig. 7.

Fig. 7. User Talking to Both Program and Executive

LOGIN 0050 JDX.25	
$OK LOG ON 10	
LOAD CALC	
$LOAD OK	
GO	
$MSG IN.	
CALC READY	Program to user
+2,3,4	User to program
RESULT +9.00000000	Program to user
*4,4	
RESULT +16.0000000	
!DIAL 28 HELLO	User to executive
$MSG IN.	
"*3,3,3	User to program
RESULT +27.0000000	
!STOP	User to executive
$MSG IN.	
GO	
$MSG IN.	
—4,2	User to program
RESULT +2.00000000	
!STOP	User to executive
$MSG IN.	
QUIT	User to executive
$MSG IN.	

other executive services

In addition to responses to basic commands, the executive performs a number of other services for users. Included among these is a set of functions necessary to check out (debug) object programs on-line. The kinds of debugging commands available through the executive include the following:

- *Open*: Displays the contents of the given memory or machine register and uses this as a base address for other debugging commands.
- *Modify Open-Register Address*: Changes the address of the opened register by the given increment or decrement.
- *Insert*: Inserts the given values into the opened register.
- *Mask*: Inserts values by the given mask.
- *Mode*: Displays values according to specified mode (floating, decimal, octal, Hollerith, symbolic machine language).
- *Search*: Finds the register containing a specified value.
- *Breakpoint*: When a specified point in the program is reached, notifies the user, (on options) displays registers, and stops or continues the program. As many as five breakpoints are allowed simultaneously.
- *Dump:* Dumps a given set of registers, either on teletype or tape (to be printed off-line).

The actual commands to perform these functions usually include a symbol or address with one or two unique teletype characters.

Figs. 8 and 9 show examples of on-line debugging operations.

Fig. 8. Examining and Changing Words in Octal and Floating Point

40000'/	DBUG command
40000' = 0140000000040052'	DBUG response
DATAWORD/	DBUG command
DATAWORD = +4.56289757E+002	DBUG response
2.5 E-3 *	DBUG command
$IN	DBUG response

Fig. 9. Examining and Changing Words in Hollerith and Integer

MESGWORD (5) /	DBUG command
MESGWORD = CONTINUE	DBUG response
BETA/	DBUG command
BETA = —197	DBUG response
—495 *	DBUG command
$IN	DBUG response

Several commands are available to enable users to communicate with each other. These commands and their functions include:

- DIAL: Permits sending messages to any other user in the system.
- LINK: Initiates linkage of any two teletypes so that they act as one. Both teletypes can input and output to and from the same program or the systems. Also, both teletypes can type all the information being input or output on each other.

Commands to query the system about its status are available through the executive. Examples of these are:

- USERS: The response to this command is the number of currently active users in the system.
- TAPES: The response to this command is the number of tapes available for use by object programs.
- DRUMS: The response to this command is the amount of drum space available for object programs.

Quite clearly, object programs must have access to the on-line input-output devices that are part of the system. In addition to these, the system at SDC permits access to disc files and tapes as completely as is possible with the computer. Since a "traffic" problem exists with these devices (as well as with the on-line devices), all input-output units are assigned by the system. No object program makes an absolute (machine) reference to an input-output unit. In the object program, requests for input-output "files" include arbitrary names, for which assignment of a particular unit is required. The executive then assigns an actual (machine) unit or an area of a unit such as disc (if one is available). For subsequent reads, writes, or positionings of the unit, the object program requests the executive to move the file by using the name given in the initial request. Any attempt by an object program (in machine language) to read or write a unit directly will result in a computer interrupt that will stop the program.

A mechanism has been provided to permit the multiple use of one program by several users. An object program may in its input-output file declarations request more than one teletype or display console (in this case, specific units). Thus a single program can act as a recorder, monitor, situation generator, or play some other central role to a group of users in a game or other team effort. *(To be continued).* ■

A FAST SENDOFF FOR THE PDP-8

Decimal to Binary Conversion by Radix Deflation on PDP-8 (DECUS No. 5/8-7)

By: Donald V. Weaver, Consultant, New York City, New York

e usual view of a packed BCD number in 8-4-2-1 code sees it as a data pattern, a string of decimal digits stored in order of magnitude. It is a useful view, but incomplete. Let us look at it here in light of the parallel fact that whatever data is placed in the accumulator can be processed by arithmetic logic. It is all 'numberger' to the digital computer.

The BCD code for any three-digit decimal number ABC from 000 to 999, stored in the accumulator of the PDP-8 computer, can be taken by the computer literally as a 12-bit binary number with an over-inflated value, $A(16)^2 + B(16) + C$, since the decimal digits are positioned there with radix 16. The following program provides a short, fast method of reducing the inflated radix 16 to the true radix 10 by scale-right $10(10A + B) + C$, stored in the accumulator as a pure binary number. The commented code details this transformation:

```
DBIN, 0         /ABC in BCD code in accumulator
      DCA T     /Save
      TAD T     /A(16)^2+B(16)+C ≡ 16(16A)+16B+C
      AND A     /Erase digits B and C leaving 16(16A)

      CLL RTR   /Divide by 4 leaving 4(16A)
      DCA T2
      TAD T2
      RAR       /Divide by 2 leaving 2(16A)
      TAD T2
      CIA       /Negate the sum obtaining -6(16A)
      TAD T     /Add the original BCD code to get 10(16A)+16B+C

      DCA T     /Save
      TAD T     /10(16A)+16B+C ≡ 16(10A+B)+C
      AND AB    /Erase digit C leaving 16(10A+B)

      CLL RTR
      DCA T2
      TAD T2
      RAR
      TAD T2
      CIA
      TAD T     /Result is 10(10A+B)+C
      JMP I DBIN /True binary value in accumulator

T,    0
T2,   0
A,    7400
AB,   7760
```

Execution time of 57 to 58 microseconds is faster than by other methods for the PDP-8. It is also more compact, occupying only 26 memory registers in all. A possible space/time tradeoff is noted in the program. The radix-deflating sequence of seven lines at DBIN+4 can be placed in a subroutine and reused on the second radix deflation at a cost of 8 additional microseconds for each linkage, reducing the memory requirement to 23 registers.

The program could be adopted almost trivially [dac 7 = dca 8 tad 8 and cma, add T = cia, tad T] by a PDP-4 programmer for fast conversion of 3-digit decimal numbers on the PDP-7 computer in a program of 22 words and 49 microsecs. (alt. 20 words/67 microsecs.). The same program with a modified subsequence to deflate radix 64 could be used by the PDP-7 on 6-bit IBM decimal code packed three digits per 18-bit word in a program of 30 words/70 microsecs. (or 24 words/87 microsecs.). On 6-bit code, however, radix deflation is slightly less efficient than the best of other methods.

NOTICE that the single-register action of this method of RADIX DEFLATION indicates it might be advantageously implemented with hardware. DECUS members are granted permission to make a copy of the program for their use on the DEC PDP computers.

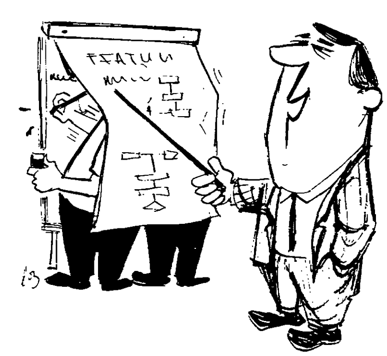
DIGITAL PROGRAM LIBRARY NOTE

The package, as described in the November issue of DECUSCOPE, consisting of FORTRAN, Floating Point System, and Multiply Subroutines was mailed to all PDP-5 users in November. A form was sent along with the package which requested pertinent installation information. All of these forms have not been returned. The information requested is needed to insure adequate program distribution.

On December 18, a new version of PAL will be distributed that eliminates the operating restrictions and adds the new feature of parameter assignments. Also at that time, it is expected to distribute Symbolic Tape Editor and the write-up for DEC-5-24 Binary Punch 33 which has been revised to correct the starting address error. The correct starting address is 1666 octal. DEC-5-25 Binary Punch 75A has been corrected and a revised tape and listing will be distributed soon.

DDT-5 is in the final stages of debugging and documentation.

A new FORTRAN Users Manual PDP-4/7 is in the process of completion. The new manual contains all the necessary information needed to program FORTRAN on the PDP-4 and 7. The summary description has been retained as the appendix. The DDT, Assembler, and Canute Manuals are being reprinted. These manuals will be distributed to all PDP-4 users sometime in January.



"...and here are some of our latest developments..."

NEW DECUS MEMBERS

Delegates

PDP-1

Mr. Lawrence Cole will replace Mr. Jack Tauber as delegate from ITT, Data and Information Systems Division, Paramus, New Jersey.

Mr. Robert Zavadil will replace Mr. Charles M. Haltom as delegate for Air Force Technical Applications Center, Washington, D. C.

PDP-5

Mrs. Ruth Kelly
Bell Telephone Laboratories Inc.
New York City, New York

Dr. Paul D. Coleman
University of Maryland
School Of Medicine
Baltimore 1, Maryland

Individual Members

Sanford C. Adler
George Lieberman
Dr. Mark S. Mayzner
New York University
Bronx, New York

Karl L. Zinn
University of Michigan
Center for Research on Learning
and Teaching, Ann Arbor, Michigan

William F. Holst
ARCON Corporation
Lexington, Massachusetts

George MacIlroy
The Foxboro Company
Natick, Massachusetts

Published Monthly
for
Digital Equipment Computer Users Society
(DECUS)
Maynard, Massachusetts

Contributions to DECUSCOPE should reach the Editor before the 7th day of the current month for publication in that issue. Material received after the 7th will be considered for publication in the months following.

DECUS PROGRAM LIBRARY

A Note from the Programming Chairman, Richard J. McQuillin

A 2-core version of DECAL-BBN is now available as a DECUS program. A description of the system is being made available as Maintenance Memorandum No. 2 to be appended to the DECAL-BBN Programming Manual. The new version uses the second core as symbol table storage. The work was supported by the Decision Sciences Laboratory, ESD, USAF.

In the Process of Completion

A Dice Game for the PDP-5 by Edward Steinberger, DEC

This program is as the title indicates a Dice Game for the PDP-5. No special hardware required.

ASSIM from Lawrence Radiation Laboratory, Berkeley, California

A PDP-5 Simulator which runs on the IBM 7094 using FORTRAN IV.

ADDITIONS TO THE PDP-4 LIBRARY

DECUS No. 4-8

Short-Loader for the PDP-4 - Flynn Loader

By: J. Baldridge, and Jon Stedman, LRL, Berkeley, California

This loader, called the Read-In Loader, will load binary tapes of the following format:

```
dac A
c (A)
dac B
c (B)
 .
 .
jmp Y
 Ø
```

This program was described fully in the October DECUSCOPE. Tapes are now available.

DECUS No. 4-9

SQRT - Single Precision Square Root Subroutine

By: Jon Stedman, LRL, Berkeley, California

This square root subroutine performs the fixed point square root of the contents of the AC(0,17). The result is left in the AC. Calling Sequence: JMS SQRT
RETRUNS HERE. . . .

ADDITONS TO THE PDP-5 LIBRARY

DECUS No. 5-4

Octal Typeout of Memory Area with Format Option

By: Donald V. Weaver, New York City, New York

Symbolic Listing Only

DECUS No. 5-5

Expanded Adding Machine

By: Donald V. Weaver, New York City, New York

Expanded Adding Machine is a minimum-space version of Expensive Adding Machine (DEC-5-43-D) using a table lookup method and including an error escape facility.

This is a basic version to which additonal control functions can easily be added: Optional vertical or horizontal format, optional storage of intermediate result without reentry, fixed-point output of results within reason, and other features that can be had in little additional space under switch register control. Symbolic listing only.

DECUS No. 5-6

BCD to Binary Conversion of 3-Digit Numbers

By: Donald V. Weaver, New York City, New York

This program is based on DEC-5-4 and is intented to illustrate the use of alternative models in program construction.

While not the fastest possible, this program has one or two interesting features. It converts any 3-digit BCD-coded decimal number, $D_1D_2D_3$ into binary in the invariant time of 372 microseconds. Efficient use is made of BCD positional logic to work the conversion formula $(10D_1 + D_2)10 + D_3$ by right shifts in the accumulator. In special situations, it could be profitable to insert and initial test/exit on zero, adding 12 microseconds to the time for non-zero numbers. Symbolic Listing only.

DECUS No. 5/8-7

Decimal to Binary Conversion by Radix Deflation on PDP-8

By: Donald V. Weaver, New York City, New York

Full description on Page 5 .

DECUS No. 5-8

PDP-5 Floating Point Routines

By: Jerry Crawford, Culver City, California

Consists of:

1. Square Root - Tape and Symbolic Listing
2. Sine-Cosine - Tape
3. Exponential - Tape

The calling sequence for all routines is:

/argument in floating accumulator
/JMS to floating subroutine
/returns with answer in floating accumulator

DECUS No. 5/8-9

Analysis of Variance PDP-5/8

By: Henry Burkhardt III, Digital Equipment Corporation

The following is a brief description of an analysis of variance program written for the standard PDP-5/8 configuration (i.e. 4K memory, ASR33 teletype). The output consists of:

A. For each sample:

1. sample number
2. sample size
3. sample mean
4. sample variance
5. sample standard deviation

B. The grand mean

C. Analysis of Variance Table:

1. The grand mean.
2. The weighted sum of squares of class means about the grand mean.
3. The degrees of freedom between samples.
4. The variance between samples.
5. The pooled sum of squares of individual value about the means of their respective classes.
6. The degrees of freedom within samples.

(Continued on Page 8.)

PDP-5 COMPUTER OPTIONS

The Type 34 Oscilloscope Display

The Type 34 Oscilloscope Display is a point-plotting device which permits rapid conversion of digital computer data into graphic and tabular form upon the screen of an oscilloscope. It has been economically designed to meet the particular needs of those requiring an immediate qualitative display of information but who do not require the high degree of accuracy or size afforded by other DEC displays.

Information is displayed as a point of light whose location is specified by two 10-bit digital words. Location of any desired point may be specified by any of the 1024 X and 1024 Y coordinate addresses contained on the tube face. Discrete points may be plotted in any sequence at a rate of up to 45 kilocycles depending upon the computer being used.

The Type 34 may be purchased with either a Tektronix RM-503 Oscilloscope for normal work or a Tektronix RM-564 Storage Oscilloscope for displaying very large amounts of information without apparent flicker. The Type 34 may also be purchased independently for use with a customer supplied oscilloscope or plotter.

The 630 Data Communication System

The 630 Data Communication System, a real-time interface between Teletype stations and a central computer, is used in message switching, data collecting, and data processing multi-user applications. By performing serial-parallel conversion and certain control functions, the 630 System permits many Teletype operators to communicate with the computer. Because the computer handles communications so much faster than the Teletypes, operators experience no delay, even when communicating simultaneously.

The system has four different operating modes: send-only (Teletype stations cannot send), receive-only (Teletype stations cannot receive), half duplex (Teletype stations can send and receive but not simultaneously), and full duplex (Teletype stations can send and receive simultaneously). The station handling capacity of a 630 depends on the mode of operation. In full duplex, for example, a maximum of 64 two-pair stations can be accommodated. By party-lining even more stations can be added.

(Continued on Page 8)

Magnetic Tape System Type 580

The Type 580 Magnetic Tape System is a semi-automatic data storage system for Programmed Data Processor-5. Data transmission in the 580 is under program control, while the timing of motion delays, end of record delays, write clock pulses, etc., is automatic. Densities are 200 and 556 bits per inch (selected by program), and maximum transfer rate is 13,334 characters per second. Format is compatible with IBM NRZ1 in either binary or BCD parity mode.

The Type 580 contains a 12-bit data buffer, which accumulates the data word in both reading and writing, and a 9-bit command register, which receives control information from the PDP-5 accumulator and device selector. All commands, data and status indications are transferred to or from PDP-5 through the accumulator.

Operation

The system performs eight functions, as follows:

Write
Read Forward
Read Reverse
Space Forward (one or N records)
Space Reverse (one or N records)
Rewind
Write Real Time (one word at a time)
Read Real Time

Type 154 Memory Extension Control

The Type 154 Memory Extension Control provides for the extension of core memory in Programmed Data Processor-5 to 32,768 words in increments, or fields, of 4,096 words. The Type 154, consisting of one mounting panel of logic, can be added to PDP-5 without modification of the central processor. Added to the console are lights to indicate the contents of the field address registers and switches to load them manually.

The most important functional elements of the 154 are two 3-bit registers, the instruction field register and the data field register, each of which is used as an extension of the memory address register to select the appropriate field. Switching circuits determine when one or the other of these registers is to be decoded along with the usual 12-bit memory address information for repeated use until it is changed by the program. Thus the Type 154 does not increase memory access time; the memory cycle time using extended memory is 6 μsec, the same as when using the basic memory of 4096 words. Six iots (input-output transfer instructions) are added to PDP-5 with the installation of the Type 154.

Analysis of Variance PDP-5/8 (Continued)

7. The variance within samples.
8. The total sum of squares of deviations from the grand mean.
9. The total degree of freedom.
10. The total variance.
11. The ratio of the variance between samples to the variance with sampl[illegible]

This is the standard analysis of variance table that can be used with the F test to determine the significance, if any, of the differences between sample means. The output is also useful as a first description of the data.

All arithmetic calculations are carried out by the Floating Point Interpretive Package (DEC-5-30-A).

DECUS No. 5-10

Paper Tape Reader Tester

By: Anthony Schaeffer, LRL, Berkeley, California

A test tape can be produced and will be continuously read as an endless tape. Five kinds of errors will be detected and printed out. The Read routine is in 6033-6040. Specifications: Binary with Parity Format - Length: registers in locations (octal): 10, 11, 40 through 67 (save 63, 64), and 6000-7777.

DECUS No. 5-11

PDP-5 Debug System

By: Anthony Schaeffer, and Don Zurlinden, LRL, Berkeley, California

Purpose of this program is to provide a system capable of:

1. Octal dump 1 word per line.
2. Octal dump 10_8 words per line.
3. Modifying memory using the typewriter keyboard.
4. Clearing to zero parts of memory.
5. Setting to HALT codes part of memory.
6. Entering breakpoints into a program.
7. Initiating jumps to any part of memory.
8. Punching leader on tape.
9. Punching memory on tape in RIM format.
10. Punching memory on tape in PARITY format.
11. Load memory from tape in PARITY format.

PDP-6 System Applications Continued

Lawrence Radiation Laboratory is scheduled for System No. 5. The laboratory plans to use the PDP-6 as the I/O processor in the OCTOPUS System. Its function will be to control all input/output transfers between each of the following components:

2	IBM 7094's
2	CDC 3600's
3	IBM 1402's
1	CDC 6600
1	IBM STRETCH
1	UNIVAC LARC
10	Tape Units
30	Displays
1	Radiation printer (30,000 lpm)

Rutgers University, Physics Department, will be using System No. 7 in their high energy physics work. Utilizing a film reader, they will perform nuclear particle analysis.

System No. 8 is scheduled for M.I.T.'s Laboratory for Nuclear Science. The [illegible]oratory will use the PDP-6 to control their Precision Encoding and Pattern Recognition (PEPR) System. The role of the PDP-6 in the system is to generate the scan pattern commands, store the coordinates of the detected tracks, and single out those tracks which are of interest to the investigators.

R.P. Harris and R. Handy, DEC