University Computer Center Newsletter

SPECIAL ISSUE ON SMALL COMPUTERS

UNIVERSITY COMPUTER CENTER   UNIVERSITY OF MINNESOTA-TWIN CITIES   MINNEAPOLIS, MINNESOTA 55455

CONTENTS

BULLETINS

If you are not a regular reader of this newsletter, please see page 16.

If you are now using, or are planning to use a small computer system, please see page 16.

UCC newsletter

Special Issue — May, 1979

Director: Peter C. Patton
Editor : Amy Koepke, Mary Boyd

Comments about the content of this newsletter, or suggestions for changes may be directed to the editor, 235a Experimental Engineering, 612/373-7744.

introduction

Welcome to our first special issue of the University Computer Center Newsletter: an issue for current or potential small computer users. In this and future special issues, we will be discussing microcomputers, mini-computers, and any devices that may appear to fall on the fuzzy line between the two. To give unity to our special issues, we shall call this publication the Small Computer Newsletter. In this particular issue, we shall be talking about those computers popularly called microcomputers.

We have no publishing schedule for the special issues. When we have information to give you, we will publish it. Eventually, we will probably incorporate the small computer information into the regular Newsletter. In the same way, we are hoping to integrate expertise in the use of a limited variety of small computers across our regular professional staff.

We are now, or soon will be, providing support to users of small computers in the following ways:

1. We are now maintaining cross-processors (assemblers and compilers) on the Cyber 74 for the most popular microprocessors, such as the Intel 8080, M6800, and PDP-11.

2. The staff of the Special Interactive Computations Laboratory (SICL) and the CAI group will assist users of the TERAK's Pascal Operating System. SICL has trained staff members who can assist users of the TERAK's RT-11 operating system. Other UCC staff members are working with APPLE II microcomputers so that we can help you later in the year when more APPLEs are available at the University.

3. We are developing a catalog of software for small computers. This software is accessible on one or more of the large computers and can be copied electronically to the small computers ("down-loaded" is the jargon for this process).

4. In the UCC Reference Room (235a Exp Eng) we have subscriptions to the small computer journals requested most frequently, and information about books and journals that are available in the University's libraries.

In the following pages, each of these services is discussed and described in detail with emphasis, in this issue, on services to TERAK users.

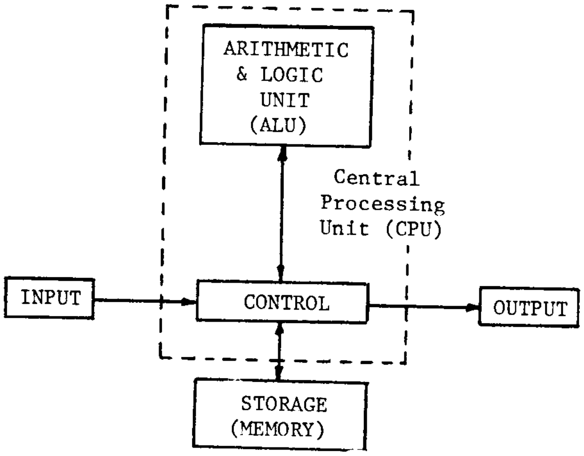
T.D. Hodge, 373-4599

WHOM TO CALL

Service	Phone
Classes/demonstrations (L. Fetcher)	376-1637
Consulting (HELP-line)	376-5592
Documentation (Reference Room)	373-7744
Newsletter (Reference Room)	373-7744
User groups/meetings (T. Hodge)	373-4599
Development, TERAK (W. Franta)	373-9751
Development, APPLE, IBM (M. Skow)	373-7745
Libraries/program acquisition (M. Skow)	373-7745
Hardware maintenance (A. Franck)	376-7291
Advice, TERAK (W. Franta)	373-9751
Advice, all systems (M. Skow)	373-7745
TERAK (SICL)	376-2975
Who is doing what (M. Skow)	373-7745
TERAK (CAI)	376-7267
APPLE II (R. Williams)	373-4573
APPLE II (M. Skow)	373-7745
IBM 5100 (W. Sackett)	373-4573

MICROCOMPUTERS?

The microcomputer is functionally no different than any other kind of computer: large computers, minicomputers, and microcomputers all perform the same type of function and have the same logical structure. Therefore, a complete computer consists of a unit for performing control functions, a unit for doing arithmetic and logic operations, a memory unit for storing data, and devices for input and output.



A COMPUTER

What is different about the microcomputer is the technology used to implement the computer's functions. Advances in solid state physics (or semi-conductor technology) have made it possible to reduce the cost and to increase the power of a computer. The current results of these technological advances have been utilized in the manufacture of microcomputers.

Since the beginning of computer development, the concept of computing has basically remained unchanged; all computing uses some type of bi-modal logic device that can represent an "either-or," "on-off," "yes-no," or a "one-zero" situation. These logic devices are grouped in such a way that the "grouping" implements a Boolean logic function, or "gate." Vacuum tubes were originally used to implement the gates.

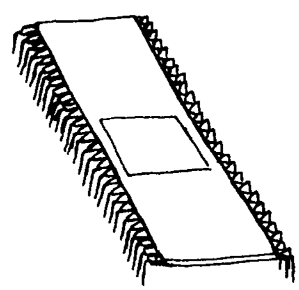
One of the first large scale commercial computers was the UNIVAC 1 (c1947) that was built from thousands of vacuum tubes. It was big enough to fill a room, and had less computing power than most of today's microcomputers. This type of computer was used for limited mathematical and pioneering business data processing applications; those which would otherwise have been impossible to solve.

Later in the 1950's the transistor (which had been invented in 1948) took the place of the vacuum tube. A transistor was originally a small piece of germanium (a semi-conductor), suitably doped with impurities; it is now always constructed of silicon. Where the vacuum tube is a bulky device with expensive internal elements, the transistor is a smaller, more reliable element that serves the same function as the vacuum tube, that is, as the active element of a logic gate. With the replacement of the vacuum tube, computers could now be built from transistors, which could be mounted on printed circuit boards, and obviously required much less space. A typical circuit board was five inches square, contained a dozen transistors, and was equivalent to a single flip-flop (basic memory element for digital computers; composed of two NOR or NAND gates). The cost of a board was $100, and there could be thousands of them in one computer. By using transistors on printed circuit boards, computers became so cheap that, by 1960, they were widely used for general data processing.

In the early 1960's, a new technology was developed; dozens of transistors could be fabricated on a single silicon chip, using the same technology as for a single transistor. This fabricated chip was called an integrated circuit; it was about 1/2 inch square and several could be mounted on one circuit board. The board still cost $100 but it now could contain all the circuitry for the arithmetic/logic unit in a computer. This early technology, which incorporates less than 100 gates per chip, is called small-scale integration (SSI).

By the mid-1960's, technological improvements had made it possible to develop chips with 100-1000 gates. This development is called medium-scale integration (MSI). Many large computers today are built with MSI chips. The minicomputer is also a product of MSI technology. In 1965 the DEC PDP-8, at $50,000 each, brought computers into the research laboratory. Many minicomputers are manufactured on a single printed circuit board with a hundred or so MSI circuits. Minicomputers are as cheap as $1000 today and are widely used.

By the very late 1960's, additional progress in semi-conductor technology resulted in large-scale integration (LSI) with chips containing more than 1000 gates.



CHIP

From LSI technology then, comes the "microprocessor" in which the computer's central processing unit is implemented in a single, or a very small number of, integrated circuits. This leads us to the "microcomputer," where the major component is a microprocessor, together with attendant memory and output/output components.

Hand in hand with these advances in technology came the revolution in our uses and views of computers. The small pocket calculator, at $30-$40, has become a standard tool for many people. The TV game is a popular item in family activity. Personal and hobby computing has become commonplace for a whole group of people. Educational toys and games in calculator-sized packages are fascinating and fun for elementary school students. Each of these activities and items is possible because of the development of LSI technology and the microprocessor. The functions that make up the calculator, the TV game, the hobby computer, and the educational toy can be packaged compactly and inexpensively; putting these items within the financial reach of most people.

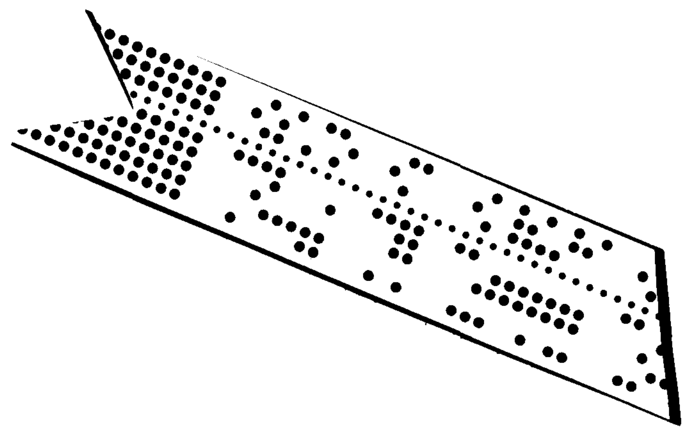
Now that we have seen the technological advances in hardware that produced the microprocessor and microcomputer and the ways in which our uses of computers have changed, let us look at the components that make up an entire microcomputer system.

The microcomputer system is a functionally complete computer, with a central processing unit, input/output devices, memory devices, and accompanying software.

Through the use of appropriate software, the input/output components of a microcomputer system allow you to have a dialog with the microcomputer. Most microcomputers will "talk" to you by displaying messages on a video display screen. Sometimes this video display is incorporated into the housing that contains the microprocessor, sometimes it is a separate unit. The video display is really much like an ordinary television set; however, it usually has better resolution. This means that you can display smaller characters on the screen without having the display become too fuzzy to read. If you don't need high resolution, you can use a TV set as a video display. if you need printed material from the microcomputer, you can attach a printer that will print up to 600 lines per minute, although this can be an expensive option.

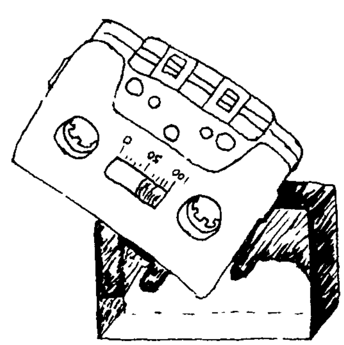
It is usually necessary for you to be able to "talk" to the microcomputer. You do this by typing your messages on a keyboard that is connected to a microcomputer. This keyboard looks very much like a typewriter keyboard and may be attached to the display unit. It is possible that, in the future, you will even be able to speak through a microphone to the microcomputer, and bypass the keyboard entirely.

The storage components of a microcomputer are used to store instructions (programs) and data that the computer will process. In order to process instructions and data rapidly, microcomputers generally have a fast access storage (memory). This memory can be characterized in two ways. There is a small amount of read-only-memory (ROM) that usually contains information that is not changed, e.g., a compiler. There is a larger amount of fast random-access-memory (RAM) that contains information that will be changed, e.g., programs and data. Both of these memory units are usually inside the microcomputer housing. Instructions and data that are not in use are usually stored on some form of slower, cheaper, bulk storage medium. There are several options available when choosing a slower, secondary memory medium. The slowest (and cheapest) medium is paper tape. Paper tape requires a special paper tape reader/punch device, and a primary disadvantage of paper tape is that you will need enormous quantities of it. You will use approximately 2000 feet of paper tape to record the same data that can be contained on a 90 minute cassette or on one side of a single density floppy disk. It also takes a long time to record or read data on paper tape.



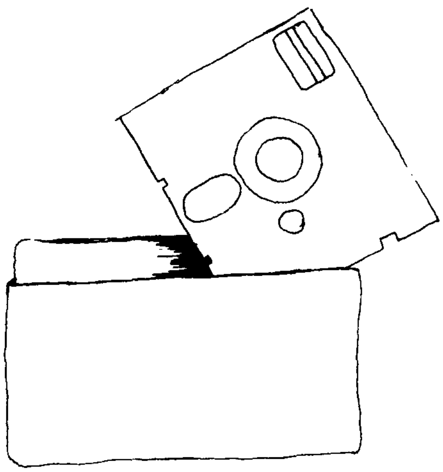
P A P E R   T A P E

An equally inexpensive but faster bulk memory medium is the cassette unit. This is exactly the same as the cassette tape that you play on your stereo system. Data is stored sequentially on cassette tape. To access data at the end of the tape, you must first pass over all the preceding data.



C A S S E T T E

For those who can afford it, the fastest bulk memory medium is the random access floppy disk. A floppy disk looks like a small, smooth record. It comes in two sizes: regular (8 inches) and mini (5 1/4 inches). Information is stored so that it can be directly (not sequentially) accessed. A regular floppy disk can hold up to 250,000 characters of information. You can buy both pre-recorded cassettes and floppy disks complete with programs, or you can buy blank cassettes and disks and record your own programs. These recorded programs can be stored in your library and re-used at any time.



F L O P P Y   D I S K

Microcomputers have a short history that began about 10 years ago. Datapoint had designed a very elementary computer and contracted with Intel and Texas Instruments to implement it on a single chip. Intel succeeded, but the computer executed instructions so slowly that Datapoint rejected it. Intel chose to market the device, and in 1971 the Intel 4004 microcomputer was born. All logic for the central processing unit was put on a single silicon chip that was less than 1/4 inch square and cost $5.00. Shortly thereafter, the Intel 8008 was introduced, with 8-bit registers and 16K bytes of memory. The Intel 8008 created a market where none had existed before; the programmable logic device was now available to a very wide and general audience. These early microcomputers were very slow and limited, but they were found to be very useful in many areas. In fact, the past few years have seen a large market develop, not only in business applications, but in the hobby and personal use areas as well.

Since the microcomputer, the minicomputer, and the large scale computer are all functionally equivalent devices, what is it that would entice you to use a microcomputer? Why would you choose a microcomputer over a minicomputer or a large scale computer?

One item to consider is cost. You should remember that the LSI processor in a microcomputer is cheap. The peripheral devices (visual display unit, keyboard, printer, memory devices, etc.) are the significant part of the hardware cost. Therefore, the cost of computation can be very small if few peripheral devices are needed; for example, using microprocessors in thermostats, microwave ovens, telephones, gasoline pumps, and computer games. Many traditional applications can be handled less expensively and more reliably. For example, transaction terminals, process control interfaces, peripheral device controllers, and front-end communication tasks are all possible microprocessor applications. In addition, there are certain types of applications that can lend themselves to microcomputer processing, for example, computer assisted instruction (CAI), graphics development, small business applications, and controlling parts of some manufacturing and refining processes.

In the University environment, microcomputers are being used for a variety of projects: computer aided instruction (CAI), command control projects, gaming projects, classroom instruction, and laboratory data collection, for example. There is even a student club dedicated to using, building, and investigating microcomputers.

Microcomputers are also used to extend the life of the central mainframe system. It is possible to do some things in a more cost-effective manner on the microcomputers; for example, inter-terminal communication is more cost-effective on three TERAKs than on three Cyber terminals, and some graphics development is more reasonably done on a microcomputer.

Thus, with microcomputer capability doubling and the price dropping each year, it is reasonable to expect microcomputers to handle more and more of the peripheral computing tasks at the same or reduced prices. We should see microcomputers moving into any area where their use is more efficient and more cost-effective than are traditional processes. This, then, becomes the rationale for our involvement with the microcomputer.

M.C. Boyd, 373-2522

micros at the UofM

Although there has been a large amount of student and faculty microcomputer research activity at the University, the first professionally packaged microcomputers were introduced to the University of Minnesota in December, 1977 when several TERAK microcomputer systems were acquired. The TERAK was the first reasonably flexible and capable microcomputer that became available, at a signficant discount, to the University of Minnesota. The University was able to purchase, through EDUCOM, TERAK computers for $5500 to $7000. Currently, there are 34 TERAKs at the University, with more on order. There are 7 TERAK microcomputers available in a hands-on environment at the Special Interactive Computations Laboratory in the Space Science Center.

Almost a year later, in September 1978, IBM made several 5100 microcomputers available to the University under a special, very attractive lease arrangment. IBM is interested in re-entering the educational market and in having the academic community become acquainted with the features of their microcomputer system. There are 68 IBM 5100 microcomputers at sites around the state; e.g., UofM-Duluth, UofM-Morris, UofM-Twin Cities.

Also in 1978, the Minnesota Educational Computing Consortium (MECC) decided to provide a stable microcomputer system for its members; one capable of being used in an instructional environment. MECC evaluated several microcomputer systems and awarded a contract to Apple Computer, Inc., whereby APPLE II microcomputers could be purchased by state educational organizations through MECC. This contract was awarded in December, 1978. MECC needed a computer that was easy to program, required little development, supported BASIC programs, and was reasonably cheap. The APPLE II seems to fit these requirements. At this time, 100 APPLE II microcomputers have been delivered throughout the state, and 200 additional ones are on order.

In some respects, the TERAK is probably the most capable microcomputer of the three. It supports many languages and systems: Pascal, FORTRAN IV, BASIC, and "C" (with UNIX). It can accomodate three operating systems: the UNIX system, a Pascal system (UCSD), and the DEC RT-11 system. RT-11 provides a reservoir of DEC software. The TERAK has general purpose hardware that supports most superimposed software, which means, for example, that you could write your own compiler if you wished. The TERAK has 28K 16-bit words of memory, uses a direct access floppy disk of 250,000 characters per disk and has a visual display unit and keyboard. The TERAK, at $5500, provides a computer resource for applications that do not require the sophistication, power, and variety of the large Cyber 74 interactive computer system. These applications typically include classroom instructional use, language studies, research project analysis, and maintenance of student grade information.

The IBM 5100, which sells for $10,000 to $15,000, supports both the BASIC and APL languages. It is IBM's contention that APL, and therefore the 5100, will appeal to the academic community. Both BASIC and APL are supported in the hardware of the microcomputer in its read-only memory (ROM). There is much software available on cassettes (the 5100 uses cassettes for its auxiliary memory). The use of cassettes, of course, provides a sequential access memory. The software cassettes were not provided in the original lease agreement with IBM, but IBM has made some software available on request. The software covers topics such as mathematics-statistics, graphics, and econometrics. A casette provides 250,000 characters of sequential access memory. The 5100 is a capable machine at an attractive lease price.

The APPLE II microcomputer has been designed to serve an instructional audience. It offers the BASIC programming language with integer and floating point features. The Pascal language will be available sometime in the future. The APPLE II is noteworthy because it is the only one of the three microcomputers at UCC to provide color graphics, and with a proper interface it can even be attached to a home color television set. The APPLE also comes with a set of hand controls that can be used in a game setting. To date, the University has ordered approximately 40 of these microcomputers. The supply of available software is growing because the APPLE is one of the important hobby computers. The APPLE provides both cassette and floppy disk storage with 112,000 characters per disk. A 32K byte system with disk, communications controller, and color monitor is available, through the MECC contract, for approximately $1750.00.

These three, the TERAK, the IBM 5100, and the APPLE II, are the microcomputers that UCC is most familiar with now. Each has its own strengths and seems designed to serve its own audience. If we had to rank these three computers in decreasing order of capability, the TERAK is probably first, with the IBM 5100 following closely behind. The APPLE II is substantially farther down the list. In terms of purchase price, however, the APPLE II at $1750 would lead the list, with the TERAK next at $5500, and the IBM 5100 last at $10,000 to $15,000.

In all events, the choice comes back to that which we stated earlier: the cost and special features of a particular microcomputer must suit your needs and your pocketbook.

At UCC, we are continuing to explore the microcomputer field. New uses and applications are being investigated. Software and services are being developed. SICL is even building its own microcomputer.

As we learn and gain experience with microcomputers, it is our intention to share our knowledge with you, to keep you informed about the mushrooming microcomputer activity here at the University. This we will do through special newsletter issues.

In later sections of this special issue, we will discuss the TERAK microcomputer: its physical characteristics, what hardware features it has and what it looks like, and the topics of software and support.

M.C. Boyd, 373-2522

ucc services

The potential University user of a small computer will probably make a decision about the relative usefulness of a particular device on the basis of one or both of the following factors:

1. Strength of personal interest in the challenge of making the small computer perform against any odds.

2. The quantity and quality of the services easily available for a particular computer.

The second factor falls within our mission. We hope to offer good services for a limited number of specific small computers.

Why a limited number? Because we have limited resources in terms of staff, while there are dozens of potentially interesting and useful small computers. A discussion of the decision process appears elsewhere in this issue. In this article, we want to describe the services we can offer now in an obviously fluid situation. The service model will follow very closely the pattern associated with our big computers. Services for instructional use of small computers will be included in the central UCC budget. Communications costs and other charges will be borne by the department using the service. Services to research users will be supported by the researcher's budget.

Services for the use of specific small computers can be defined as follows:

- .Short courses and demonstrations
- .Consulting
- .Documentation, either produced by the vendor or by UCC
- .Newsletters
- .User groups and meetings
- .User program libraries and program acquisition
- .Communications systems development, both software and hardware
- .Program development
- .Hardware maintenance

Some of these services are well known. A few may need further clarification. The general guide will continue to be that development, whether in hardware or software, is generally undertaken by the researcher or the researcher's department. In some cases, however, the work might better be done by our Professional Services Divison under contract to the researcher or department. On the other hand, there are those services of general benefit to students and departments alike where the developmental costs could be supported by UCC. In any event, please contact UCC if you have questions concerning the support of services.

We hope by means of these services to encourage reasonable, wise, and constructive use of particular small computers which we feel have demonstrated proven strengths for the major University mission of education.

T.D. Hodge, 373-4599

short course

We plan to offer a new course on microcomputers this quarter. It is scheduled to begin on May 7 in Room 5 Architecture, from 3:15 - 5:00 PM, and will continue for five or six sessions, as follows:

Mon May 7	Introduction	Prof. R. Halvorsen
Wed May 9	The APPLE II	M. Skow
Fri May 11	TERAK-POS	R. Gonzales
Mon May 14	IBM 5100	W. Sackett
Wed May 16	TERAK RT-11	Prof. D. Riley
Fri May 18		To be arranged

There will be a hand-out at the May 7 meeting, which will provide background for the May 14 session on the IBM 5100. The IBM 5100 session will primarily be an answer period for the questions prompted by the handout.

For additional information, call Thea Hodge, 373-4599.

what will it cost?

There are many microprocessors that are in wide use today; among them the Intel 8080, M6800, Z80, 650X, 1800, F8, 9900, PACE, and MICRONOVA. Other microprocessors have been introduced that are destined to come into wide use soon, including the Microflame, Intel 8086, Z8000, and 68000.

A respectable microcomputer system could be based on virtually any of these, but probably the most popular of all microprocessors in the laboratory, the school, and the office is the DEC LSI-11. The widespread popularity of the LSI-11 is due to the large selection of compatible circuit boards and its software compatibility with the familiar PDP-11. The LSI-11 system is typical of many microcomputers used in a wide variety of applications.

A complete microcomputer system might include the following:

system component	price range
central processing unit	\$ 450-\$ 650
56K bytes of read/write memory	\$ 800-\$2400
parallel input/output interface	\$ 185-\$ 700
serial input/output interface	\$ 180-\$ 250
floppy disk controller	\$ 500-\$1400
two floppy disk drives	\$1050-\$2000
8-slot backplane and enclosure	\$ 675-\$ 800
power supply	\$ 250-\$ 750
CRT	\$ 800-\$2500
operating system/high level language	\$ 200-\$3640

In addition to the features described above, a desirable system may include some of the following items:

system component	price range
printer	\$1600-\$ 3100
hard disk controller	\$1500-\$ 7950
hard disk drive	\$3500-\$ 5270
graphics terminal	\$1000-\$40000
magnetic tape subsystem	\$2500-\$11000

The purchase price of a completely assembled system such as described above will range between \$9500 and \$14,000 (not taking into account volume or educational discounts). These prices depend upon the supplies and the additional features provided. You must, of course, have the expertise to integrate the components into a computer system.

Standard pre-packaged microcomputer systems that do not supply all of the resources listed above (for example, omit the second floppy disk drive or reduce the number of card slots) can be purchased for considerably less money. One such system can be purchased for as little as \$5500 (TERAK) at the price available to educational institutions.

Costs and information on the items listed below below can be obtained from SICL:

- Processor chips
- 32K word memory boards
- Parallel input/output boards
- Serial input/output boards
- Floppy disk controller
- Floppy disk drives
- Floppy disk subsystem (packaged with controller and drives)
- Enclosures and backplanes

Power supplies
CRTs, printers
Hard disk controllers
Cartridge disk drives
Fixed head disk drives
Language processors and operating systems

C. Covey, 376-7007
W. Franta, 373-9751

pascal & micros

Because Pascal is a general-purpose language, its wide availability enables users of different computer systems to transport and share programs easily from one machine to another. Microcomputers have not been overlooked, and Pascal implementations, both compilers and interpreters, exist for such microcomputers as the LSI-11, Intel 8080, Z80, M6800, 9900, and the 6502. Newer 16-bit microcomputers such as the DEC 11/23 and 11/44, Intel 8086, Z8000, MC68000, and the Western Digital Pascal Microengine (R) either have, or nearly have, Pascal.

Pascal has been touted over the last year in various personal computing journals such as "Byte," "People's Computers," and "Creative Computing." Paul Helmers, editor of "Byte," tells me that he sees Pascal eventually supplanting BASIC on personal computers. The reasons he cites are its intrinsic superiority, and its being in the public domain.

As mentioned elsewhere in this newsletter, the University of Minnesota has many TERAK microcomputers that run the most widespread micro implementation of Pascal: the one from the University of California at San Diego (UCSD). This system comes with its own one-user monitor, and such software tools as a screen-oriented text editor, a CAI course set, and a graphics package. The entire system has been transported without change to the Intel 8080 and Z80 microcomputers. The Microcomputer Group (MCG), a student club on campus, has a copy of the UCSD system for use by its members.

Pascal offers a greater degree of portability than most other highly popular languages for several reasons:

1. The definition of Pascal clearly distinguishes between the language and its implementation on a particular machine; such aspects as character set and size of integers are defined by the implementation.

2. The inclusion of features in the language such as machine-independent data packing (such as character string).

3. The existence of the low-cost and highly reliable portable Pascal compiler, Pascal-P. Pascal-P's syntax analyzer accepts the same language regardless of what machine it runs on. An article appearing in the December issue of "Pascal News" describes a project to move a significant Pascal program (3000 lines) from an LSI-11 to the Cray-1 supercomputer (a Pascal speed ratio of 1 to 150).

The Pascal-P compiler (P stands for portable) has evolved through four versions over the last five years. It can be obtained as a kit configured to run on your machine, but because machine-dependent implementations of Pascal are now easily available, there is less need to order the kit. Pascal-P is a compiler written as a 3500-line Pascal program that generates code, called P-code, for a hypothetical "stack machine," called the P-machine. The kit also contains the same compiler expressed in its P-code equivalent, configured to your machine.

To make Pascal-P run on a real machine, an intepreter must be written for P-code in another language (such as assembler) already existing on the real machine. This typically takes a good programmer a few weeks. Then the copy of the Pascal-P compiler written in P-code will immediately run. You now have a Pascal interpreter. To produce a true compiler, the Pascal-P compiler written in Pascal must be edited to change the code-generation parts to generate code for your machine instead of P-code. The work required to do this runs about six months for a good programmer. After compiling this version twice you have a real Pascal compiler for your machine.

A frequent criticism of Pascal on microcomputers is that, by contrast with a limited language like BASIC, a full implementation requires more memory to run: on the order of 48K bytes of random access memory (RAM). This is true, in part, because BASIC is often implemented as an interpreter and not as a compiler. Most Pascal compilers, because they are written in Pascal, are not as compact in terms of memory as they could be if they were written in assembler. However, the problem really exists with the microcomputers themselves. Their architectures (general hardware design including instruction set) are so primitive that P-code instructions expand into too much actual machine code. For that reason, the UCSD system remains a Pascal-P interpreter in order to keep it small. This makes it run even slower, although the average compile speed on a TERAK is still an acceptable 600 lines per minute.

The solution to the hardware gap is to build a machine with a P-code instruction set! Then the code space for a Pascal implementation is greatly reduced, and it can compete favorably against other languages. A few new microprocessors being built now have P-code instruction sets. P-code, by the way, is also general enough to support other languages like FORTRAN. It actually has a much simpler structure than an ideal Pascal-oriented machine language.

The future, then, is very exciting because low-cost machines with modern languages like Pascal and MODULA are on the horizon. (MODULA is a simple Pascal-like language for systems programming.) We are making an effort to stay informed of these fast-paced events. As you may know, the University of Minnesota is the headquarters of the Pascal User's Group (PUG), an international organization with over 3300 members in 47 countries and 49 states. We therefore have the advantage of getting first-hand information quickly about Pascal on all computer systems including personal computers. We produce the quarterly "Pascal News" for PUG; subscribing to it is the easiest way to follow events surrounding the development of Pascal. Just call or write me for more information.

Andy Mickel, 376-7290

documentation

The UCC Reference Room in 235a ExpEng contains a small, non-circulating collection of professional books and journals for use by the Center's staff and others. The hours are 8:00 AM - 4:30 PM Monday through Friday. A photocopier is available and, within reason, materials may be copied.

This collection contains microcomputer information. For cataloging and indexing the collection, a data base is maintained on the University's computing facilities. Since microcomputers are such a recent development and changes are so rapid, the best source for information will be the journal collection. For each journal entered in the library data base, the appropriate bibliographic data is entered along with article titles, page numbers, authors, and several descriptors.

The library data base can be searched in either batch mode or interactive mode. We give examples for both. For more detailed information on searching the library data base, call me and ask for a copy of a small booklet describing our Reference Room data base.

Note: in the examples below, all user responses are given in lower case. All system responses appear in upper case.

SAMPLE INTERACTIVE SEARCH

Log on to the Cyber 172 with your user number and password. When the system responds with:

```
    RECOVER/SYSTEM:
```

enter:

```
    get,dunham/un=yze6075
    call,dunham
```

"dunham" is a procedure file that gives you access to the library data base. The system will respond with:

```
    BEGIN SYSTEM 2000
    ----------------
    COMMAND FILE IS INPUT:
```

Three dashes (---) and a question mark (?) will appear to ask you for a command. A note of caution: ALL commands must end with a colon (:). You may want to enter the command

```
    echo off:
```

to keep the system from re-printing each command before it responds.

Sample commands that you may use:

```
    describe:
```

This command will write out the data base definition with component numbers and names. When searching the data base, you may call items by component name (e.g., article title) or component number (e.g., c26). Components listed as non-key may not be used for searching purposes. Component numbers of repeating groups (RG) may not be used to search, but they may be used to print out those components included within the repeating group.

```
    print:
```

The print command is used to print out the data contained within a specified component number or name. You may limit this to one component or you may specify as many as needed. For example, to find out what journals the Reference Room has, enter this command:

```
    print title where document type eq journal:
```

or

```
    pr c2 wh cl eq journal:
```

Both commands will provide you with a listing of the periodicals that can be found in our Reference Room. The second example uses valid abbreviations.

```
    print c2,c3,c26 wh c32 eq microcomputers:
```

will list journal titles, dates, and article titles where the descriptor is "microcomputers." If you want the citations listed in some order, use the ORDER BY command. Here is the same command as above, but ordered by date:

```
    print c2,c3,c26, ob c3,c2,c26
    wh c32 eq microcomputers:
```

Since a command this long is a bother to type, we have saved lengthy, frequently-used commands, known as strings, under code names. To see the strings, enter:

```
    describe strings:
```

To use a string, enter:

```
    *c36(microcomputers)
```

where *C36 is the code name of a string that will produce a count of journal articles with "microcomputers" used as the descriptor. The colon is contained within the string so it is unnecessary when using any of the string commands. It is always advisable to take a count of the number of data sets before having them printed out.

You may combine descriptions with the c40 string.

```
    *c40(microcomputers,microprocessors)
```

When you finish your search, enter:

```
    exit:
```

SAMPLE BATCH SEARCH

```
NAME(MI72)
BIN card
USER(usernum,passwor)
GET(DUNHAM/UN=YZE6075)
CALL(DUNHAM)
7-8-9 card
DESCRIBE:
DESCRIBE STRINGS:
*C36(MICROCOMPUTERS)
PRINT C2,C3,C36 wh C32 EQ MICROCOMPUTERS:
EXIT:
6-7-8-9 card
```

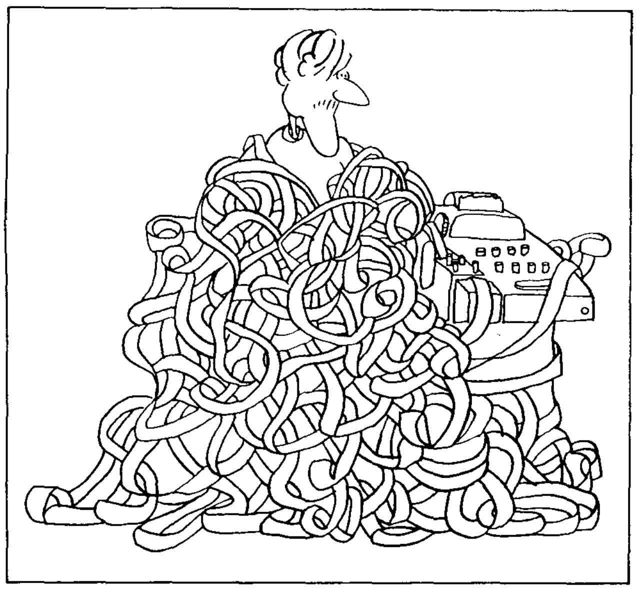
This sample deck will provide you with the data base definition, the string commands, a count of the articles that have microcomputers as a descriptor, and a list of journal titles, dates, and article titles where microcomputers is used as the descriptor.

* * *

If you need more information than we have in the Reference Room, contact Crystal Clift in the Engineering Library in Lind Hall. Ms. Clift will conduct a search of Lockheed's network of approximately 100 data banks (DIALOG) and suggest ways of conducting and limiting a search in order to provide the most useful citations. Recently, I had her do an experimental search of two data bases, COMPENDEX and INSPEC. By limiting the search to the past six months and combining 2 descriptors (microcomputers and microprocessors), I obtained 50 of the most recent citations from COMPENDEX and 63 from INSPEC. By the way, this search is not free. The search we made of COMPENDEX and INSPEC cost $50.00.

Following you will find a list of items taken from the Engineering Library card catalog; call numbers are included.

M. Dunham 373-7744



CITATIONS FOR THE PAST MONTH

BIBLIOGRAPHY

Barden, William T. (510.79, qB235)
How to buy and use minicomputers and microcomputers.

Katzan, H. (510.79, K159i)
The IBM 5100 Portable Computer.

Barna, Arpad (510.79, B252)
Introduction to microcomputers and microprocessors.

Bibbero, Robert J. (510.79, B47)
Microprocessors in instruments and control.

Didday, Richard L. (510.79, D561)
Home computers: 2**10 questions and answers.

Bryan Norris (editor) (621.38, E1257)
Electronic power control and digital techniques.

Hilburn, John L. (510.79, H543)
Microcomputers/microprocessors: hardware, software, and applications.

Grimes, Jack D. (Pam 2326)
Microprocessors and microcomputers.

Adam Osborne and Associates, Inc. (Pam 1122)
Introduction to microcomputers

Klingman, Edwin E. (510.79, K687)
Microprocessor systems design.

Korn, Granino Arthur (510.79, K842m)
Microprocessors and small digital computer systems for engineers and scientists.

Leahy, William F. (510.79, 1471)
Microprocessor architecture and programming.

McGlynn, Daniel R. (510.79, M177)
Microprocessors: technology, architecture, and applications.

Samuel C. Lee (editor) (510.79, M5826)
Microcomputer design and application.

W. C. Lin (editor) (510.79, qM5833)
Microprocessors: fundamentals and applications.

Motorola Semiconductor Products (510.79, qM857)
Microprocessors applications manual.

Peatman, John B. (510.79, P329)
Microcomputer-based design.

Osborne, Adam (510.79, Os103)
An introduction to microcomputers.

Sawin, Dwight H. (510.79, Sa96)
Microprocessors and microcomputer systems.

Sippl, Charles J. (620.351079, Si75m)
Microcomputer dictionary and guide.

Sippl, Charles J. (510.79, Si75)
Microcomputer handbook.

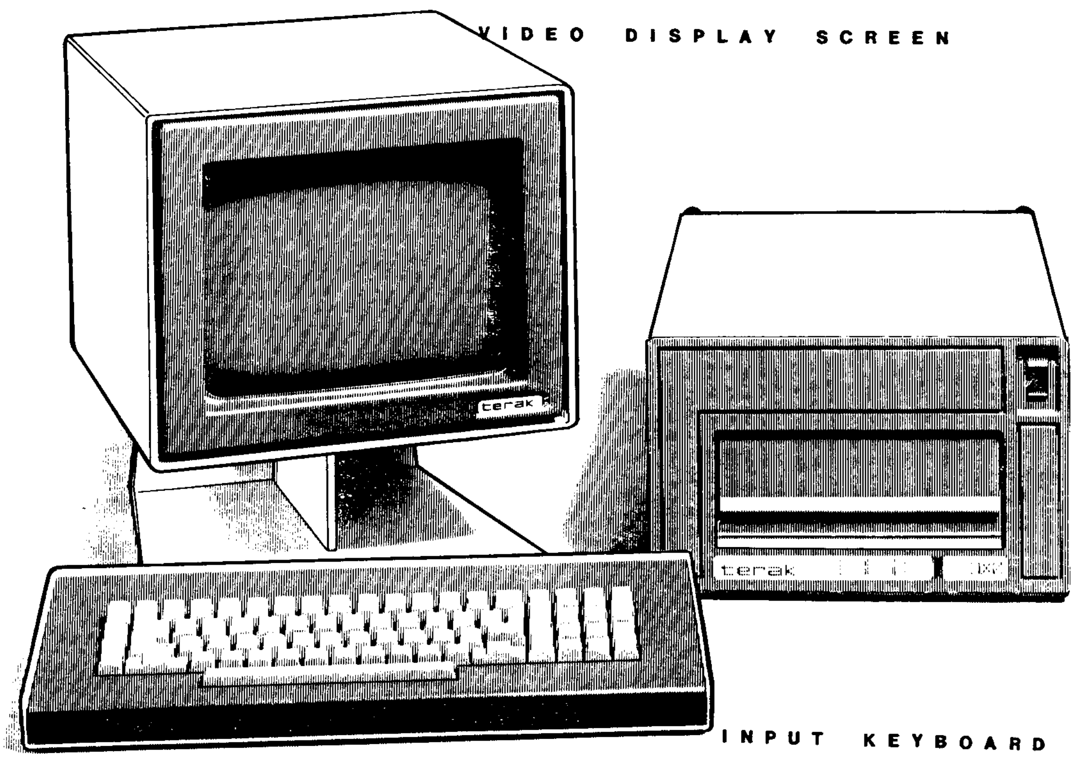
Soucek, Branko (510,79, So81)
Microprocessors and microcomputers.

THE TERAK

The TERAK 8510/A Graphics Computer system consists of a model 8510 data processor, memory, a video display screen, and an input keyboard.

The 8510 data processor incorporates a DEC LSI-11 microprocessor, a single floppy disk drive, a disk controller that can handle up to four additional floppy disk drives, circuitry and power supplies, and 4K 16-bit words of metal-oxide semiconductor (MOS) read/write memory. The video display controller and 24K words of main (RAM) memory are contained on 8.5 inch x 10 inch printed circuit board. This board is also housed in the 8510 data processor cabinet. The 8510 is approximately 7 1/2 inches x 12 inches x 18 inches and weights less than 40 pounds.

Input/output to the TERAK is via the model 8532 keyboard/display. The keyboard is free-standing with a full 128-character ASCII set and function keys for cursor control, space, carriage return, etc. The video display (CRT) has a 12 inch black and white screen and provides high resolution graphics (320 dots wide x 240 dots high), alternate character sets, and simultaneous character and graphics display. The character matrix is a 640 wide x 240 high matrix of dots, with each character displayed within a block 8 dots wide x 10 dots high. The keyboard and display unit are housed in separate cabinets. The video display unit cabinet is 10.5 inches x 13 inches x 12 inches and the keyboard housing is 3.5 inches x 16 inches x 6.5 inches. Their combined weight is 40 pounds.



want a terak?

Question: If someone offers you a TERAK, should you

a. accept it,
b. decline it, or
c. run the other way as fast as you can?

If you do not know the answer to this question, you are certainly part of the majority of people both on this campus and in the world. The TERAK however, can be quite useful for both research and instruction, since it is a portable, but reasonably powerful computer system, that costs between $5500 and $7000, depending on the model that you buy. It can execute programs in FORTRAN, BASIC, and Pascal, collect laboratory data, record grades for large courses, and turn out "personalized" letters. It can draw reasonably detailed pictures and graphs (and even make them appear to move). It can prompt telephone interviewers on questions, and record the replies for later analysis. It can even act as a "number cruncher," provided you have a whole afternoon or weekend to let it nibble away.

If the above description stimulates your curiosity, you will probably find the remainder of this article of interest; I will describe the principal features of the TERAK system.

The TERAK Model 8510 is a packaged version of the LSI-11 microprocessor, manufactured by the Digital Equipment Corporation (DEC). (DEC is the largest manufacturer of minicomputers in the world.) TERAK, a rather small company in Arizona, purchases the LSI-11 central processor from DEC and packages it, together with memory, a television-type terminal, a keyboard for input, and a floppy-disk drive, into a couple of small boxes that will easily fit on the top of a desk. If you buy a TERAK, you will receive a reasonably complete computer system that should permit you to solve problems within a few hours after you turn it on.

The central processor of the TERAK can do all the usual things: run programs, add, multiply, and divide both integer and floating point numbers. It can execute most of the instructions of the larger PDP-11 computers manufactured by DEC, so if you have some programs written for that larger machine, they will run quite well on the TERAK, but probably more slowly. Programs written for the Cyber computer can also be made to run on the TERAK, provided they are written in a high-level language such as FORTRAN, BASIC, or Pascal. Some modification may be necessary, and the programs will certainly take more time. (Some crude measurements have shown that the TERAK executes FORTRAN programs at about one percent of the speed of the Cyber.) The TERAK can be made to talk to the Cyber computers over the telephone, so it will not be necessary for you to type in all your old programs again.

The memory of the TERAK may be a limiting factor if you usually run large programs. The TERAK can store approximately 28,000 16-bit words: both programs and data. In determining the amount of memory required for a program, you should remember that the TERAK uses two words to store a non-integer or floating point number and even this space gives you considerably less precision than the Cyber. Another problem with the memory is that about 4000 words of the memory are lost to the program if the graphics display feature is used.

The television-type display on the TERAK is quite nice and is certainly the unique feature of this microcomputer. It can display 24 lines of 80 characters each on the screen at one time. You can define your own characters, so the TERAK can display messages in Greek, Hebrew, or Cyrillic letters as well as in Latin characters. On the same screen (and even at the same time), the TERAK can draw graphs or displays on a matrix of 240 vertical by 320 horizontal points. The resolution of such displays is not superb (close inspection of straight lines reveals a certain jaggedness) but graphs that are not too complicated look quite reasonable. A connector on the back of the TERAK can be used to reproduce the display on any number of video monitors; this can be very useful in classroom instruction. The big disadvantage of this display terminal is that it does not produce a hard copy that you can take away from the computer. There is a plug on the back of the TERAK for connecting an auxiliary printer, but such a device will set you back another $1000.

The floppy-disk drive is one of the most expensive parts of the whole TERAK system. If you have never seen one, a floppy disk is a round piece of mylar, coated with a magnetic oxide and enclosed in a paper jacket for protection. The disk used on the TERAK can hold about 250,000 characters; this is the equivalent of about 100 typewritten pages or 3 completely punched boxes of cards. (You will certainly find that carrying floppy disks is much easier than carrying cards.) Despite this large information capacity, it is much more convenient to have two floppy disk drives on your TERAK system. The second drive, which costs about $1800, raises the price of the basic system to the higher price mentioned above. With only one drive, it becomes difficult, if not impossible, to work with large programs. It is also difficult to make backup copies of disks, which is absolutely necessary if you are as clumsy and disorganized as I am.

If you are at all familiar with the trends in the computer industry today, you must know that the cost of the electronics (hardware or "iron" as it is called in the trade) forms a small part of the total cost of using computers. The remaining costs are for the software or programming. This is an area in which the TERAK excels. There are large numbers of programs written for the PDP-11. Well-tested operating systems, compilers, and application programs in almost every field have been written and tested. It is not necessary to re-invent a program to do an efficient sort of random data, or do a time-series analysis, or calculate the parameters of an electronic circuit. Many of these programs are available free or for a minor charge from program libraries such as DECUS, the DEC Users' Society. Even programming systems that are generally sold may already be available on campus and may be available to you through a contract.

There are two operating systems that are in general use for TERAKs on this campus. The Pascal system was written at the University of California, San Diego and is a complete system for writing, modifying, copying, compiling and executing programs written in BASIC or assembler language. This system has a nice screen-oriented editor that moves characters around in front of your eyes and is well-suited for a TERAK with only one floppy disk. It has the disadvantage that you are limited to one language, Pascal. The other operating system is DEC's RT-11 that also has a collection of editors, copiers, compilers, etc.; an advantage of RT-11 is that it will accept programs in FORTRAN and BASIC. For sheer computing speed or real-time data acquisition, FORTRAN is certainly the language of choice as opposed to BASIC. RT-11 also has a program that will format and hyphenate text material, which is useful when writing those "personalized" letters.

With all this information, you should be able to answer the opening question. I would certainly take it. I use the TERAK for computing answers to problems, for teaching a class on methods of experimental physics, for composing lecture notes and scientific papers, and for revising those interminably revised committee reports. Other applications in the School of Physics include keeping grades for large lecture classes and recording data from research instruments. Other applications in various fields are described in this newsletter. The TERAK may not be the most powerful computer available, but it has one additional advantage--it is available on an easy payment plan.

M. Marshak, 373-0241

POS on the terak

POS, a Pascal-based, portable, one-user operating system from the University of California, San Diego, is intended primarily for microcomputers. POS is written in Pascal, but the compiler generates code for an ideal machine called the P-machine. If the system is to run in a host machine different from the P-machine (the usual case), the compiler is not rewritten to generate code; rather, the P-code is interpreted in the host machine.

Transporting POS to another machine is a relatively easy task since only the P-interpreter needs to be written. This represents about 4 percent of the total POS software; one good reason why moving the system to another computer system is so cost effective.

POS consists of a memory-resident portion and a non-resident portion. The resident portion includes global information about the status of the system, to be used by different non-resident modules. If the host machine is not the P-machine, the interpreter, written in assembly language, will be resident too.

The non-resident part consists of modules that can be called directly, one at a time, by the resident portion; the user just invokes the system functions. These functions are file handling, editing, compiling, assembling, linking, and debugging. For some of these functions, the user has several options and can reconfigure the system at will. Three different editors are available, Pascal and BASIC compilers are available, and the user can request execution of utility and user programs by specifying, to the resident portion, the file name containing such programs.

The system functions are:

File handling:

This is the module in charge of manipulation and maintenance of disk storage. A disk is referred to by "volume" name. A volume has a "directory" specifying the "files" stored in it. The file handler thus provides for manipulation of volumes, directories, and files.

Volumes can be initialized and reorganized; damaged areas on disk can be kept from being used; volumes can also be listed, copied, and set up for initialization of the POS.

Directories can be listed, copied, and duplicated.

Files can be created, allocated, copied, removed, saved, and listed. These operations can be performed on several files at the same time, either by giving a list of such files, or by giving a generic description of them.

Finally, the applications of the file handler are expanded by viewing each input/output device as a volume. This allows operations on volumes to be performed on any input/output unit. For example, a transfer of a file to a volume called PRINTER (assigned to the printer) will give a hard copy of the file.

Editing:

Three editors are provided. Two of them have powerful capabilities for screen handling; the other is a line-oriented editor.

In addition to the conventional functions of searching, inserting, changing, and deleting found in the line-oriented editor, the screen-oriented editors have facilities for indentation of block structured languages and paragraph-oriented text. A full screen or "page" of text (called window) is always displayed and directional cursor movement is provided by simply pressing one of the 4 directional keys.

The window moves with the cursor so that the position of the cursor in the file can always be seen through the window.

One of the editors is intended primarily for handling files of any size.

Another outstanding feature is that the editor permits the insertion of other text files, or portions thereof, into the file being edited.

Compiling:

Two languages, Pascal and BASIC, are provided. POS supports Pascal, but it differs in some areas from Standard Pascal.

Limitations:

Procedures cannot be passed as parameters; dynamic memory management is a constrained, stack-like operation; exit GO TO's are not implemented; the program header does not have file parameters; the DISPOSE, PACK, and UNPACK procedures are not implemented; Booleans cannot be written.

Extensions:
In case statements, execution falls through next statement when no case label is selected; overlaying is possible; strings and long integers are new types; several procedures can be exited at once.

A new text file type, "interactive," is defined, and differs from any other file type in that no character look ahead is performed when reading. Random access of file is possible, and I/O error detection is provided.

The class concept is introduced and is called "unit." A unit is a collection of inter-related objects such as constants, types, variables, and procedures; the outside user has constrained access to these objects via an interface definition clause within the unit.

Some features of the BASIC compiler include: arrays of more than two dimensions; recursive subroutines; Pascal-type string operations.

Assembling:
An adaptable assembler is provided and can be very easily modified to generate code for other machines.

Linking:
Linking is achieved through the use of libraries. Utility programs for creating and maintaining libraries are available.

POS is easy to use. Commands always prompt the user for the information needed, so one does not have to remember the sequence and type of parameters required. That is, a command is presented in the form of a question and an answer.

To save time for the user, the resident portion of the operating system coordinates the usage of the different modules. For instance, the compiler will compile, by default, the file just being edited, and if errors are found, the editor is called and locates the user at the place of the error. When a program is run, it will be compiled and linked if necessary.

At each moment of interaction with the POS the user will see, displayed on the screen, most of the commands available at that time.

In summary, POS has powerful features, such as the screen editors, file insertion when editing, grouped file specification for operations in the files, and the concept of treating any input/output device as a volume, which simplifies and makes more powerful and flexible both program writing and file handling. A sophisticated language, Pascal, is an outstanding feature never available in small computers before.

All of these make the UCSD Pascal operating system one of a kind in the competitive world of microcomputers.

R. Rivera, 373-7904

UNIX on the terak

UNIX is a general purpose, multi-user, interactive operating system for the Digital Equipment Corporation PDP 11/40, 11/45, and 11/70 computers. A version of the UNIX operating system has been written for the TERAK microcomputer.

UNIX was originally written by Ken Thompson and Dennis Ritchie at Bell Laboratories, and was described by them in the July 1974 issue of the "Communications of the ACM." UNIX offers a number of features including:

1. A hierarchical file system incorporating dismountable volumes.
2. The ability to initiate asynchronous processing.
3. Over 100 subsystems, including a dozen languages.

Because our current version of UNIX uses the disk quite extensively and inefficiently, response time is not very good and disk space is at a premium. A number of modifications can be made to UNIX to improve response time and disk utilization. SICL is working on the following modifications:

1. Rewriting the program swapping algorithm.
2. Implementing contiguous files.
3. Rewriting the floppy disk driver.
4. Allowing more than one user process to reside in memory at one time.

The UNIX system on the TERAK is reasonably fast for program editing and program execution. With the additional work noted above, the system can be made fast enough for program development.

A. Perandi, 373-7881

terak p-m dump

The UCSD Pascal system is generally very easy to use; with one notable exception: finding and fixing fatal run-time errors. When a program "blows up" with a fatal error, the system provides a cryptic message such as: "Value range error S#1, P#12,IPC#1218" plus an optional octal dump. This information is of little use to the average user; it doesn't illuminate the cause of the difficulty.

To solve this problem, the CAI Systems Group has developed and implemented a Post-Mortem Dump (PMD) enhancement for use with version 1.4g of the system on the TERAK machine. Instead of printing a cryptic series of numbers when a fatal run-time error occurs, PMD displays:

1) A general error message.
2) Any specific error information available.
3) The source statement which caused the error.
4) The name of the Procedure, Function, or Program which contains the error.
5) The values of (simple) program variables.

A typical error display might look like:

```
Value Range error
Invalid Value was 101, outside range of [ 1..100]
In source line: AGELIST[ I := AGE] ;
Variables in Procedure INDEX:

NAME    :="Jane Smith"  AGE     :=       25
SINGLE  :=      TRUE    TEMP    :=     98.6
SEX     :=   ORD( 1)    TITLE   :="Vice-President"
I       :=       101    J       :=       12
```

Thus the user has the essential information for diagnosing the problem: i.e., the value of the variable "I" is apparently outside the allowable range of indexes for array AGELIST.

A copy of the PMD enhancement and the 1.4g version of the system in which it is embedded may be obtained by contacting the UCC's CAI Group.

E. Schleske, 376-2975

development

Most of the development activity to date has been in software for the TERAK. Some of this work is described in other articles in this issue. We will attempt a brief summary here:

Professor Marvin Marshak, Physics Department

1. Plotting package, in FORTRAN
2. Network circuit analysis program, in BASIC (for instructional use)
3. 5-node network for control of laboratory experiments (TERAK plus four LSI-11's)
4. Grading program for large courses (produces histograms, final grades, etc.)
5. Word processing

Michael Skow, University Computer Center

Conversion of CAI programs for classes in Greek

Professor William Franta, UCC-SICL

1. Networking experiments
2. Communications software development for up- and down-loading (TERAK-Cyber)
3. Grading system, now being documented

Professor Donald Riley, Mechanical Engineering

1. Calcomp-type plotting package
2. Engineering graphics package, 3-D and rotation; working on hidden line problem
3. Design package for simple linkages in mechanisms

Also, Professor Franta and his group at SICL are continuing their projects in evaluation of TERAK hardware and software.

T.D. Hodge, 373-4599

classical greek

During Spring Quarter 1979, students taking Classical Greek will be using a TERAK microcomputer for laboratory sessions. This TERAK development is a result of the conversion of an existing course originally developed to run on the Control Data Corporation's PLATO system.

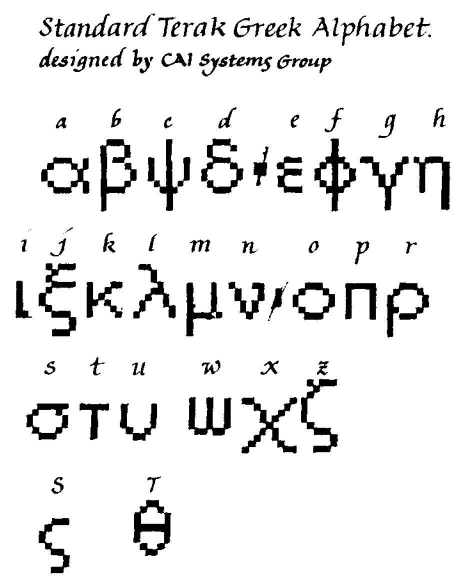
The PLATO system was originally chosen as the delivery system for the course because of its unique graphics capabilities. However, the TERAK development represents a much less expensive approach.

Four programmers in our CAI group worked part-time last summer and fall on a project to develop a text driven interpreter. Variable width, detailed Greek characters are displayed on the screen. The course, contained on six floppy disks, represents 30-50 hours of interaction. The figure shown below illustrates the character set developed for this project.

The TERAK was found to be somewhat limited in memory for this application, but has plenty of speed, good reliability, and an editor that is easy to use.

Efforts are now being directed to the conversion of the New Testament Greek course from PLATO to the TERAK.

M.M. Skow, 373-7745



IF YOU DO NOT RECEIVE THE UCC NEWSLETTER REGULARLY AND WISH TO DO SO, PLEASE FILL IN THE REQUESTED INFORMATION AND RETURN TO:

USER SERVICES
UNIVERSITY COMPUTER CENTER
227 EXPERIMENTAL ENGINEERING, UNIVERSITY OF MINNESOTA
208 UNION STREET SE
MINNEAPOLIS, MINNESOTA 55455

NAME: ______________________________ DEPARTMENT: ______________________

ROOM, BUILDING: ______________________ CAMPUS: ________________________

THE UNIVERSITY COMPUTER CENTER IS VERY INTERESTED IN COMPILING INFORMATION ON THE SMALL COMPUTER SYSTEMS BEING USED AT THE UNIVERSITY. WE FEEL THAT SUCH A COLLECTION OF INFORMATION WOULD BE OF HELP TO ANY CURRENT OR POTENTIAL SMALL COMPUTER USER; IN EFFECT, IT WOULD PREVENT THE PROBLEM OF HAVING TO RE-INVENT THE WHEEL.

WE WOULD LIKE TO KNOW WHAT HARDWARE YOU HAVE, WHAT SOFTWARE YOU HAVE (ACQUIRED OR DEVELOPED ON YOUR OWN), WHAT PROJECTS YOU ARE WORKING ON, AND WHAT PROJECTS YOU WOULD LIKE TO SEE DEVELOPED.

PLEASE BE ASSURED THAT PROVIDING US WITH THIS INFORMATION WOULD, IN NO WAY, OBLIGATE YOU TO SHARE EITHER YOUR EQUIPMENT OR YOUR SOFTWARE.

IF YOU ARE USING A SMALL COMPUTER SYSTEM AND ARE WILLING TO PROVIDE SUCH INFORMATION, PLEASE SEND THE DETAILS TO:

MICHAEL SKOW
UNIVERSITY COMPUTER CENTER
227 EXPERIMENTAL ENGINEERING
UNIVERSITY OF MINNESOTA
208 UNION STREET SE
MINNEAPOLIS, MINNESOTA 55455

RETURN TO:

User Services
University Computer Center
227 Experimental Engineering
University of Minnesota
208 Union Street SE
Minneapolis, Minnesota 55455