RSTS/E

RESOURCE SHARING TIMESHARING SYSTEM/EXTENDED

digital

RSTS/E

"A High-Performance 32-user Timesharing and Data Management System"

System Description

RSTS/E is a total computer system designed to allow multiple users to interactively process large amounts of data easily and efficiently. RSTS/E stands for "Resource Sharing Timesharing System/Extended." The system is built around the DIGITAL medium-scale PDP-11/40 and PDP-11/45 computers.

Sophisticated and Powerful Hardware

State-of-the-art PDP-11 family architecture combines the systems-level power of stack processing with the proven speed and efficiency of a general register machine. The complex addressing modes of systems data structures are built right into the PDP-11 hardware. So too are the multi-priority interrupt handling capabilities that drive the large complement of peripherals at their fully rated speeds.

The high performance design philosophy of the PDP-11 starts with memory systems that expand to 248K bytes. Bipolar solid-state memories operate at a speed of 300 nanoseconds; core memory at 900 nanoseconds. Processing speed, too, is exceptional with the processor capable of performing 3,000,000 operations per second. Floating point hardware performs arithmetic operations with 17 decimal digits of accuracy, typically in less than 12 microseconds. A memory management system assures efficient use of all available memory while permitting complete protection of programs and data.

Many Users, Large Programs

RSTS/E puts the exceptional hardware power of the PDP-11 into the hands of the users. RSTS/E provides versatile and rapid system access to 32 users simultaneously with jobs ranging from small "desk calculator" tasks to large 32K byte programs.

Extensive Data Base Support

The data file system of RSTS/E provides a wide range of on-line processing capabilities. Files may be random or sequential, numeric or alphanumeric. Files may be created, updated, extended and deleted interactively from the user's terminal or under program control. Files can be protected from access on an individual, group, or universal basis. Data may be stored on removable disk cartridges, disk packs, or magnetic tape. Industry compatible magnetic tape files prepared on another computer may be read on RSTS/E, and tape files generated on RSTS/E are readable by other computer equipment. Files may be accessed by many terminal users simultaneously and updated on-line.

Effective and Efficient User Interaction

Users interact with RSTS/E using BASIC-PLUS. The language is easy to learn and work with. Its immediate mode of operation enables the terminal to be used as a desk calculator. It also simplifies debugging since programs may be interrupted at any point, checked, corrected, and operation resumed. Also, built into BASIC-PLUS are the means to automatically recover from input/output errors.

Resource Sharing

With BASIC-PLUS, every terminal user has access to all the system peripherals and resources. Line printer, card reader, disks, tapes—all are available to any terminal user on-line. Enhancing the use of these resources is the ability to provide output in the most meaningful format for the recipient.

Operating Efficiency

RSTS/E installations can expect exceptional efficiency of operation because the operating system continuously and dynamically allocates processor time, memory space, file space and peripherals on a best-fit/best-throughout basis. Efficiency results too from the system manager and users having on-line access in BASIC-PLUS to 34 system management commands and operations, many of which are normally a manual management function.

Comprehensive Security

The ability for total or selective backup of programs and files is provided in RSTS/E. Backup may be done on-line without disrupting system users or it may be done off-line. In the unlikely event of a power brown-out or other malfunction, the basic system will automatically recover while all programs and data files are maintained intact and generally may be easily restarted from the point of failure.

Digital Equipment Corporation

DIGITAL is the world's largest and most experienced manufacturer of small computer systems, with well over 20,000 computers installed and in use. Over 5,000 installations are of PDP-11 family systems.

In 1963, DIGITAL began work on the PDP-6 timesharing system, the first timesharing system to be delivered with manufacturer-supported software. The follow-on 128-user DECsystem-10 is one of the most sophisticated, highly developed timesharing systems in the world with well over 200 installations. In 1968 DIGITAL introduced Timeshared-8, a versatile 16-user time-sharing system based on the popular PDP-8 computer.

RSTS-11, forerunner to RSTS/E, was introduced in June 1971. Over 150 RSTS-11 systems are installed today performing such jobs as on-line order entry, production control, accounts receivable and payable, inventory management, and invoicing. Schools and colleges use RSTS for on-line student registration and school management as well as for research, simulations, algorithmic problem solving and computer assisted instruction.

RSTS/E—The Total System

RSTS/E is surprisingly economical. Complete systems range in cost from $85,000 to $200,000. Lease costs, including maintenance, range from $2,500 to $6,000 per month.

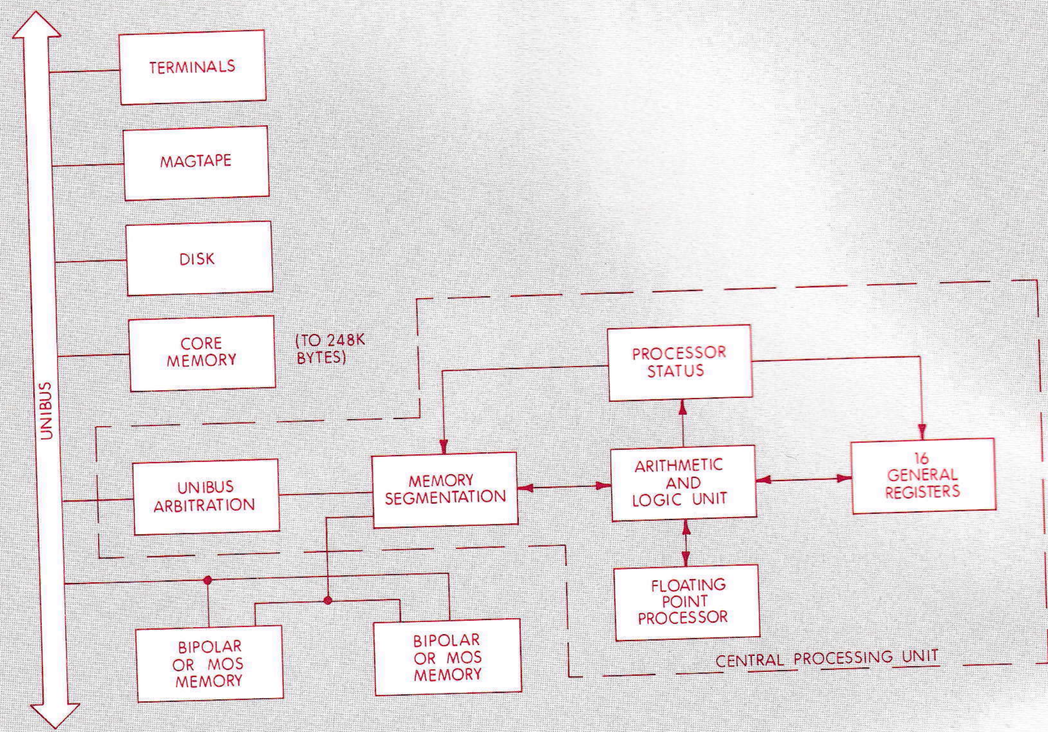
RSTS/E. Unparalleled performance. Complete data management. Easy to use. Highly efficient. Proven capability.

Contents | **Page**

RSTS/E FEATURES AT A GLANCE

- Up to 32 users—30 cps DECwriters, 240 cps CRT DECterminals, Teletypes or Remote Modems.
- Up to 248K-character (byte) memory—300 nanosecond bipolar memory (to 16K), 450 nanosecond MOS memory (to 64K), 900 nanosecond core memory (to 248K).
- Up to 32K-character programs for each user—Data need not intrude into program space.
- Up to 12 files simultaneously accessible to each user—An unlimited number of files may be opened and closed by one program. Maximum file size: 33.5 million characters.
- Resource Sharing—All peripheral devices may be used by any user at any terminal directly or under program control.
- Disk storage up to 343 million characters—Moving-head, 10-platter disk pack (up to 8); Moving-head cartridge disk (up to 8); Fixed-head swapping disk (up to 8).
- Industry-compatible Magnetic Tape—Programs may be written to read or write in format of *any* other computer for interchangeable use.
- Three types of variables and variable arrays—Floating point with up to 17 digits of accuracy, integers for indices and counters, character strings of unlimited length.
- Virtual memory for automatic-extension of in-core data to the disk(s).
- File Protection—Three levels of read or write access: Owner (programmer or user), Group (all with same project number), Universal.
- Multiple User Access to Common Files—Up to 32 users may read or write the same program or data file.
- Immediate Mode—Use terminal as a calculator or to interrupt a program, check and correct it, and resume execution.
- BASIC-PLUS—Powerful extension of Dartmouth BASIC. 50 program statements, 34 system commands, 24 operators, 3 types of variables, 40 functions.
- Character String Operations—14 relational and logical operators, 11 string functions. Strings of unlimited length.
- Matrix Operations—4 arithmetic operators, 8 matrix operators, 3 matrix functions. Unlimited array size.
- Programmable Timing Control—SLEEP, WAIT, 5 time functions and 2 date functions.
- Formatted Output—PRINT USING for floating dollar sign, asterisk fill, comma insertion, decimal point alignment, and trailing minus sign.
- Error Recovery—Automatic program recovery from input/output errors.
- System Manager Control—34 system management commands and operations, all usable directly or under program control.
- Dynamic Scheduling Algorithm—Continuous allocation of processor time, memory space, file space, peripheral access for best-fit/best-throughput basis.
- System Usage Accounting—On-line reporting of job status, memory usage, disk and peripheral usage, run time, connect time and log-ins.
- Commercial Data Processing Facilities—Sort/merge program, indexed file access method, decimal arithmetic capability, line printer spooling.

SYSTEM PERFORMANCE



PDP-11/45 PROCESSOR ORGANIZATION

System Versatility

RSTS/E, Resource Sharing Timesharing System/Extended, provides complete system access to 32 timesharing users simultaneously. Terminal response to even the largest "compute-bound" jobs is generally under 2 seconds. Even more important, the system is designed for optimum job execution speed. Jobs may range from a small "desk calculator" task to a large 32K byte program, even larger if program segments are chained together.

Providing system and application versatility are the wide variety of terminals and peripherals supported by RSTS/E. User terminals can range from 10 character-per-second Teletypes® to high-speed 240 cps CRT terminals, local or remote. Data may be entered interactively from user terminals, from high-speed paper tape or punched cards. Data files may be stored on fixed head disks, removable cartridge or stacked disks, or magnetic tape. The output may be to a user terminal, line printer, magnetic tape, or directly to another computer.

A summary of the hardware supported by RSTS/E is included as Appendix A.

UNIBUS® Architecture

Heart of RSTS/E is a PDP-11/40 or PDP-11/45 computer. The PDP-11 features UNIBUS architecture, over 400 instructions, 8-bit byte handling, 16 general registers, memory management, and hardware push-down stacks.

Key to the many strengths of the PDP-11 family is the fact that all system elements—processor, memory, peripherals, terminals—plug into a single asynchronous high-speed bus. Known as a UNIBUS, this bidirectional bus provides easy interfacing and simplifies the construction of multiprocessor or shared peripheral configurations such as RSTS/E.

UNIBUS architecture keeps PDP-11 systems from becoming outmoded. Due to its asynchronous nature, the UNIBUS is compatible with devices that operate over a wide range of speeds. Therefore, faster devices, terminals, or memory can be easily added or replace older versions without obsoleting the system.

Contributing to system reliability and speed, with the UNIBUS, fast devices have direct access to the memory—no multiplexers or synchronizing DMA hardware are required. These devices can send, receive, or exchange data without processor intervention and without intermediate buffering in memory.

Central Processing Unit

The central processing unit of the PDP-11 was designed as a high-performance component for large systems. Hardware push-down stacks provide fast temporary storage for frequently used data and for storage of program information during interrupts and subroutine calls. Stacks not only simplify interrupt and subroutine handling, but, by automatically nesting interrupts and subroutines, they facilitate the use of re-entrant and recursive subroutines. Software bookkeeping and overhead in RSTS/E are thus reduced.

The basic PDP-11 is capable of performing more than 3,000,000 operations per second. Combined with the large instruction set of 413 hardwired instructions, its overall execution speed makes the PDP-11 the fastest machine in its class.

Contributing further to the speed and precision of RSTS/E is the built-in hardware Floating Point Processor. The Floating Point Processor operates with single and double precision numbers to provide 7 and 17 decimal digits of accuracy. Single-precision multiplications are performed in 5.6 microseconds and double in 9.3 microseconds.

Memory Versatility and Protection

Memory systems of the PDP-11 also reflect its high performance design philosophy. Memory expands to 248K bytes (124K words) and is both byte and word addressable. Memory may be core, solid state, or the optimum combination of the two.

Bipolar solid-state memories operate at 300 nanoseconds. MOS solid-state memories at 450 nanoseconds. And traditional core memories at 900 nanoseconds. In RSTS/E, the run-time system software can be resident in solid-state memory, thus allowing high-speed access to commonly used routines while large user programs can utilize more economical core memory.

In the PDP-11, a memory management subsystem relocates, partitions, protects, and allocates the use of all memories in up to 48 variable-sized segments. Thus, programs are broken down into manageable pieces and stored wherever memory space is available. This assures efficient use of all available memory yet provides complete protection of user programs and the system monitor.

DATA BASE SUPPORT

Resource Sharing

RSTS/E users have on-line access to a wide range of program and data files. Files may be created, updated, extended, and deleted from the user's terminal or under program control. Up to 12 files may be open and accessible from a single program at any one time. Since files may be opened and closed during the running of a program, the actual number referenced in a program may be far greater than 12. The total number of files a user may have stored in his disk library is bounded only by the total system disk capacity and the library demands of other users.

RSTS/E files are not limited to disk files. Data may be read in from a card reader and printed out on a high-speed printer. The on-line user can assign devices and even other terminals for input and output functions through his programs. Thus, individual users get exclusive use of these devices for as long as required, then release them for others to use. This is known as "resource sharing."

Private data files may be stored on removable disk cartridges, disk packs, DECtape, magnetic tape, or paper tape. Confidential files may be dismounted when not in use and kept under lock and key. These stored files may be as large as 33.5 million bytes, yet accessible on a completely random basis.

Resource sharing is discussed further in the next section, "User Interaction."

Three Types of Data

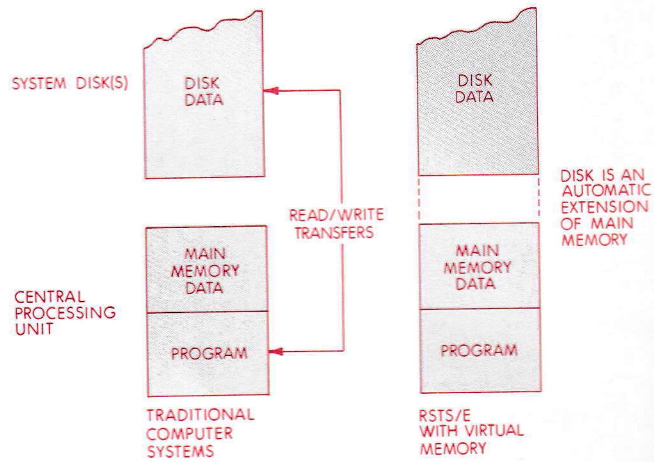
RSTS/E has the capability to handle three types of data—floating point, integer, and character string. Floating point numbers are used for most numeric representation and may be one of two levels of precision: 7 decimal digits (two computer words) or 17 digits (four words). Number size may vary from approximately 10^{38} to 10^{-38}.

Integers may be used for greater processing efficiency as indices, counters, and subscripts. They are whole numbers in the range −32,768 to 32,767.

Character strings are available for powerful processing of non-numeric data. Strings may be as short as a single character or unlimited in length. Strings used in virtual memory are limited to 512 characters. Groups of strings, a list of names and addresses for example, may be organized into tables or arrays just like numeric information. Since strings can be read from or written to external files in a sequential or random manner, whole files of textual data may be built up and updated on-line.

Virtual Memory Arrays

The concept of virtual memory essentially makes the system disks an extension of main memory. This permits the user to manipulate large arrays or tables of data without cutting into his program size and indeed, process larger masses of data than will fit in the entire main memory of the system. Furthermore, the user can access large amounts of data without the need for explicit read/write programming.



Virtual memory is easy to use. Two statements at the beginning of a program are required—one to open a disk file for virtual memory use and the second to dimension the size of the virtual memory arrays desired. From then on, variables and data in virtual memory are treated precisely as though they were in actual memory. For example, in the statement:

A = B (3, I)

the user need not be concerned whether the B array is stored in main memory or virtual memory.

Data in virtual memory arrays may also be processed using MATRIX statements. These statements perform operations on multiple elements of virtual memory arrays with a single statement.

Virtual memory may be used to store any type of data —floating point, integer, or character string. Floating point virtual memory might be used by an industrial distributor to store customer account balances on a daily basis. Character string virtual memory could be used to store names and course preferences for a college on-line registration system.

RSTS/E uses a system of in-core 256-word buffers when processing virtual memory arrays. With this system, a disk transfer is not necessarily made every time a virtual core variable is referenced. Consequently, virtual memory is as mindful of processor efficiency as it is of programming ease.

Two programs demonstrating the use of virtual memory are included as Appendices F and G.

Disk Files

RSTS/E provides users with a high degree of flexibility for handling quantities of data, both large and small. Three fundamental types of disk units are supported by RSTS/E.

TYPE	*CAPACITY (Characters*	*AVG. ACCESS TIME (Milli-seconds*
High-speed fixed-head disk	512K	17
Moving-head single-disk cartridge	2.4M	70
Moving-head, 10-platter disk pack	40M	41

Up to 8 of each of these devices are supported by RSTS/E, giving maximum disk storage of 343 million characters; individual disk files may be as large as 33.5 million characters. Thus, the installation has wide flexibility to meet its specific requirements.

Programs may be stored on the disk(s) in source language form or compiled form. Storage in the source language is used during program creation or when modifications will be made. After completion, compiled form would normally be used. While this requires somewhat more disk space, access is considerably faster since it need not be recompiled. Also, in compiled form, the program cannot be altered by unauthorized persons.

When a data file is established or opened in a program, the system automatically creates a buffer space in the user's memory space to buffer all input and output to and from the file. Typically, the amount of space reserved is determined by simply specifying the device—e.g., card reader (CR) sets up a buffer in memory of 82 characters (the contents of one punched card). However, using the RECORDSIZE option, the user program can specify the allocation of more buffer space than would be normally provided. On a disk file, total throughput can often be improved by using a larger buffer size as this permits a single disk transfer to read or write a large quantity of data.

If many records wlil be frequently used together, it is desirable for handling efficiency that they be stored in a contiguous area on the disk. The user may specify this by means of the CLUSTERSIZE option.

Larger data files are generally stored on one of the moving-head disk units. If the information is of a sensitive nature, the disk cartridge or disk pack may be removed from the system and locked up when it is not in use.

Files on Tape and Other Devices

RSTS/E supports both file-structured and non-file-structured devices. File-structured devices include disks, DECtape and industry-compatible magnetic tape. When a structured file is opened in a program, it is possible and often desirable to specify several pieces of information. In particular, a file name can be specified. Also, an extension to the name specifying whether the program is in source code (for ease of modifying the program) or compiled code (for efficiency in running—compilation is not necessary).

On structured files, a project and programmer number may also be specified. These form an integral part of the file protection scheme. Combined with a protection code, they determine who may access a file. Files can be read and/or write accessible to three classes of users:

1. Owner (programmer or user)
2. Group (all those with a specific identification number)
3. Universal (all users)

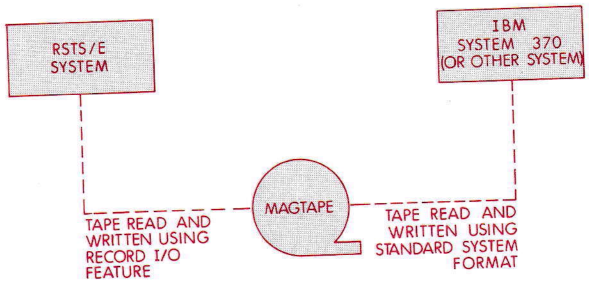
DECtape is generally used to store relatively short files such as programs and data files under 300K bytes in size. DECtape is economical and has found favor in the school and university environment with students, professors, and researchers each having their own private DECtape libraries on one or more four-inch reels.

Like DECtape, industry-compatible magnetic tape is used for sequential access to data, but it is capable of handling large masses of data. Large disk packs can be backed up on magtape, for example. (See "Disk File Integrity" section under "System Management".) Also magtape can be used to provide an interchange between RSTS/E and computer systems of other manufacturers. (See next section on Record I/O for further discussion.)

Non-file-structured devices include the card reader, line printer, high-speed paper tape reader and punch, and other user terminals. In using these devices, only the name of the device need be specified in the program I/O statement and the data file automatically is read in from or written out to the device. Customer invoices, for example, could be printed out on the line printer with the programmer merely specifying the name of the device (LP), the data to be printed, and the format. (More about formats in the "Formatted Output" section under "User Interaction.")

Record Input/Output

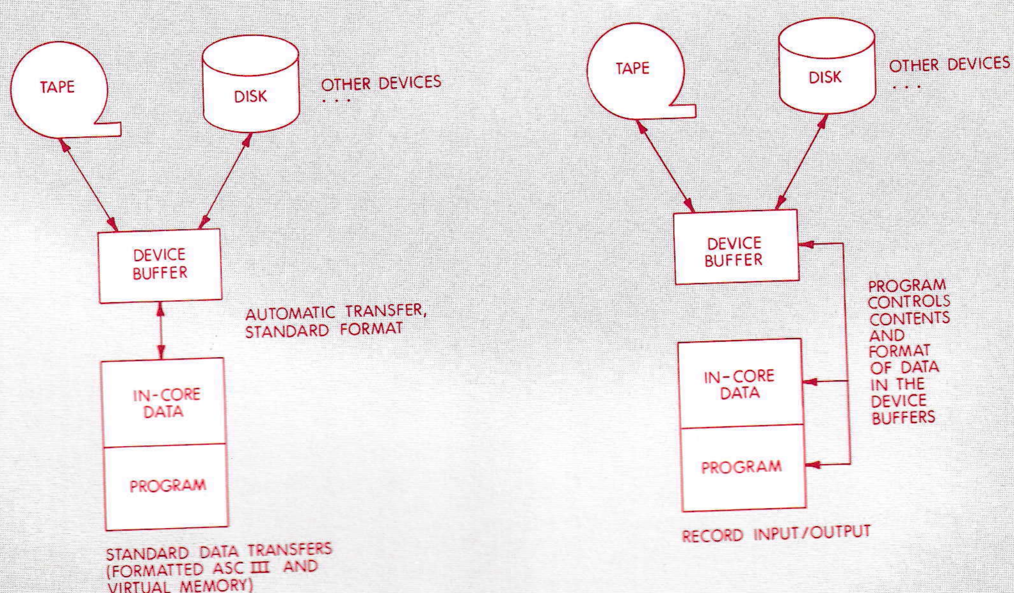
Record I/O files are files which are accessed by fixed-length data transfers with no conversion between data formats and no regard to any special characters within the records being transferred. Record I/O transfers are possible on all devices but are especially significant with respect to disk and magtape.



RECORD I/O PERMITS MAGTAPE TO BE PROCESSED ON OTHER COMPUTERS

It is this technique that permits industry-compatible magnetic tape to be written and accessed in a format compatible with the computer systems of other manufacturers. The key benefit of using Record I/O strictly within RSTS/E for disk files is its efficiency with respect to speed and minimization of disk space. These advantages are obtained at the cost of the simplicity of standard formatted ASCII and virtual array I/O, both of which are practically transparent to the programmer.

Another way of looking at Record I/O is a method which permits the user not only to manipulate the data to be input and output but also to manipulate how that data enters and leaves the I/O buffers going to and from the actual storage device (disk, tape, etc.)



DATA TRANSFERS BY CONVENTIONAL METHOD VERSUS RECORD I/O

As with other file accessing techniques, Record I/O requires that a file on a physical device be opened for use by means of an OPEN statement. Input and output to Record I/O files are performed directly between the device and the I/O buffer created by the OPEN statement. All I/O is specified in terms of single records, using the GET and PUT statements. Using sequential files such as magtape, the GET statement reads the next sequential record from the file open on the designated channel. For example:

GET #3

reads the next record from file channel 3.

Similarly, the PUT statement writes the contents of the I/O buffer for the specified channel onto the next sequential record of the file.

In the case of disk files, Record I/O transfers may be random as well as sequential. In this case, the user can specify which record in the file he wishes to read or write by using the RECORD statement. For example, to fetch the Jth record from file channel I would be done with this statement:

GET #I, RECORD J

The GET and PUT statements determine how data are moved between the I/O buffer and the device. Record I/O also provides for moving data between the I/O buffer and program. This is accomplished by means of the FIELD, LSET and RSET statements. The FIELD statement is used to dynamically associate string variable names with all or part of an I/O buffer. For each I/O file buffer, FIELD specifies the names of the string variables (or data) and the length, in characters, of each string variable, similar to a COBOL data-division file descriptor.

Once the string variable names have been defined as part of the I/O buffer with the FIELD statement, it is necessary to be able to store values in the string variables without moving them from the I/O buffer. The LSET and RSET statements store values in a string variable without redefining the string position; LSET causes the value to be left-justified in the string variable field, RSET causes it to be right-justified.

While Record I/O uses string variable names, the efficiency of Record I/O is not limited to the use of alphanumeric string data; floating point and integer values may also be stored using some simple conversion functions. This technique allows the optimization of file storage space compared to the standard RSTS/E integer and floating point variable storage.

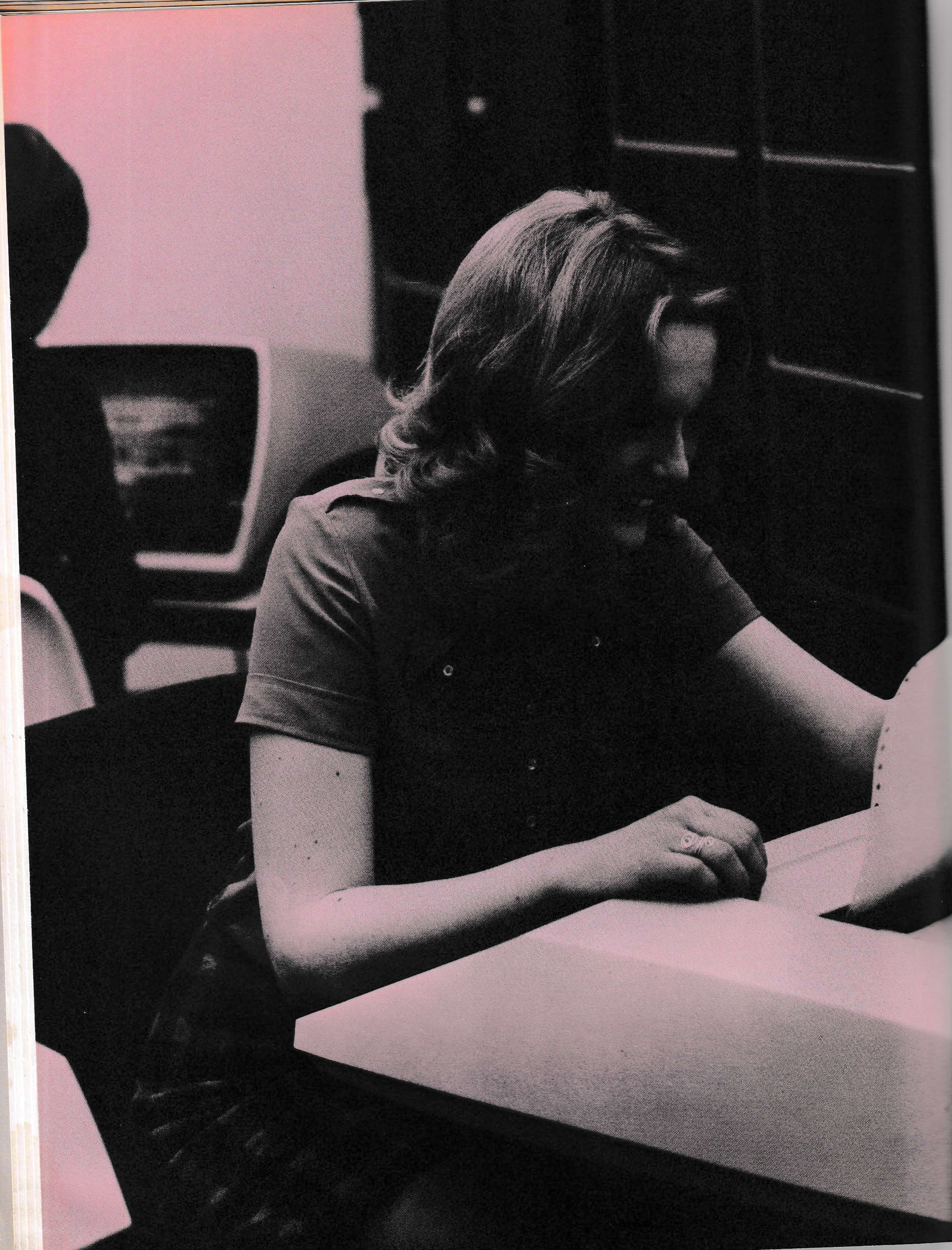
Multiple User Access to Common Files

It is often desirable to have one or more on-line disk files simultaneously accessible to more than one user. For example, in an order entry/inventory control system several clerks might be entering orders and each one must have access to the same customer master file and inventory control file. Or in a college on-line registration system, students would register and enter their course preferences at a number of terminals simultaneously.

Under RSTS/E any number of users may read data from the same file simultaneously. Typically, only one user at a time may write on the file. However, when multiple user updating is desirable as described above, the UPDATE feature permits this to be handled safely by locking out a physical disk record from other users while one user is in the process of updating the record. While the record is locked out, other users are temporarily prevented from accessing it, although they can read or write any other record in the file not currently locked out. When the locked-out record is updated, it then once again becomes accessible to other users. In this way, all users are guaranteed to have access only to current, valid records instead of records that are not up-to-date because they are in the process of being altered by another user.

File Security

Each terminal user has full control over the degree of privacy he desires for each file he creates. Access may be limited to one user, to those in the same group (or project), or to all system users. Access may be read only, write only or read/write. System and file security are discussed at greater length in the section on "System Management."



USER INTERACTION

Interactive Orientation

RSTS/E is oriented to an interactive rather than a batch mode of operation. Users interact with the system through a variety of terminals—high-speed cathode ray tube terminals for extremely rapid response where hard copy is not required; the 30 cps DECwriter for medium speed and printed copy; the industry workhorse Teletype for lowest initial cost; and a variety of remote modems and communications equipment. Although users control their activity through such terminals, they also, through resource sharing, have access to and control over system peripherals such as card readers, line printers, paper tape devices, disks, and magtape units. Thus, RSTS/E provides interactive terminals for ease of use and operational efficiency along with resource sharing for high performance and large system capability.

BASIC-PLUS—An Expanded Language

Timesharing users interact with RSTS/E using BASIC-PLUS. The language is easy to learn and work with, yet puts the enormous power of the system at the users' fingertips. The immediate mode of operation enables the terminal to be used for simple calculations. Dynamic debugging is faster since programs may be interrupted at any point, checked, corrected, and operation resumed.

BASIC-PLUS automatically checks all program commands for accuracy when they are entered. Errors are reported immediately. Since each program line is compiled as it is entered, there are no frustrating delays, even on the RUN command.

BASIC-PLUS is a significant extension of Dartmouth BASIC to increase its utility and make RSTS/E the ideal tool to solve a broad range of problems. For example, administrative applications such as on-line order entry, inventory control and payroll may be implemented efficiently by using language features suited for data processing. Text-processing applications such as Computer Assisted Instruction (CAI), automatic letter or document editing and production may utilize the set of character string handling functions. The utility of BASIC for computational applications such as structural design and simulation is extended with language features which allow more concise, and therefore, more efficient programming and program execution. BASIC-PLUS eliminates the constraints of BASIC for a variety of applications programming tasks.

Calculations in BASIC-PLUS are generally executed using floating point variables. The magnitude range of numbers lies between $.14 \times 10^{-38}$ and 1.7×10^{38}. Two levels of precision are available: 7 decimal digits (2 computer words) or 17 decimal digits (4 computer words). The degree of precision used is a system generation parameter. Whichever is chosen applies to all users of the system unless the system is regenerated for a different degree of precision.

BASIC-PLUS also allows the use of integers. These are whole numbers in the range −32,768 to 32,767. The most common uses of integers are in counting, indexing, and subscript operations. Since integers only occupy one computer word, their use often increases the execution efficiency of programs.

BASIC-PLUS provides a comprehensive set of mathematical functions to the user—trigonometric, logarithmic, absolute value, truncation, pi, random number generator, and square root. Logical and relational operators are also available. (See Appendix D for a complete language summary.)

Immediate Mode of Execution

Normal timesharing use of RSTS/E consists of typing program text using a keyboard terminal and at the end of the program typing a RUN command at which time the program executes. A second mode of using RSTS/E, called immediate mode, consists of typing program statements on the keyboard and having them executed immediately. Program statements are identical in either case except that, in immediate mode, they are typed without line numbers.

Two uses of immediate mode might be 1) performance of simple calculations in situations which do not occur with sufficient frequency to justify writing a program and 2) program debugging. To debug a program a user can place STOP statements liberally throughout the program. Each STOP statement causes the program to halt and prints the line number at which the STOP occurred, at which time the user can examine and change various data values in immediate mode and give a command to continue program execution.

String Operations

Many RSTS/E applications, such as Computer Assisted Instruction, text editing and business data processing, require efficient processing of alphabetic data such as names, addresses and, indeed, entire sentences and paragraphs. BASIC-PLUS provides for the processing of character strings of various lengths, the maximum length being limited only by the available memory. When used with virtual memory, character strings are a maximum of 512 characters in length.

A comprehensive group of string operations is provided in BASIC-PLUS. Strings may be appended to one another. Strings may be compared to one another to see, for example, if a keyboard response is correct or to alphabetize a list of names.

Functions are available to extract, examine, or search for a string of characters contained within a larger string. The use of these powerful functions can be illustrated by a CAI example. Let us say the answer being sought is "MARINE." The program could be written so the following student responses would be acceptable:

MARINE
THE ANSWER IS MARINE!
I THINK IT'S MARINE

Whereas, the following responses would not be correct:

MARINA
SUBMARINE
THE ANSWER IS NOT MARINE (program recognized a negative word preceding the answer)

Further enhancing the utility of string variables is the capability of using string arrays or matrices. With this feature, an entire list of alphabetic data, say a list of names, could be read in with a single statement, processed and output with another statement. In standard BASIC, without string arrays, a separate READ and WRITE statement would be required for each and every name in the list.

Matrix Operations

The user of RSTS/E may improve processing and programming efficiency by organizing his numeric data into one- and two-dimensional arrays or matrices. The BASIC-PLUS matrix commands add, subtract, multiply, and invert entire data matrices in a single operation. Commands are also available to initialize a matrix to zeroes, ones, or the identity matrix.

Both numeric and character string matrices may be input, read, and printed with single commands. If the matrices won't fit in main memory, the BASIC-PLUS virtual memory facility can be used as an extension of main memory as needed. Thus, array size never restricts program size, or vice versa; RSTS/E offers unlimited array capability even with the largest programs.

Extended Program Statement Coding

The effectiveness of RSTS/E in solving problems in a broad variety of application areas is significantly increased with the addition of numerous extensions to the structure (syntax) of the BASIC program statements. These highly flexible program statements, previously found only in advanced scientific languages like ALGOL, permit more concise expression of complex program steps.

Consider the following examples:

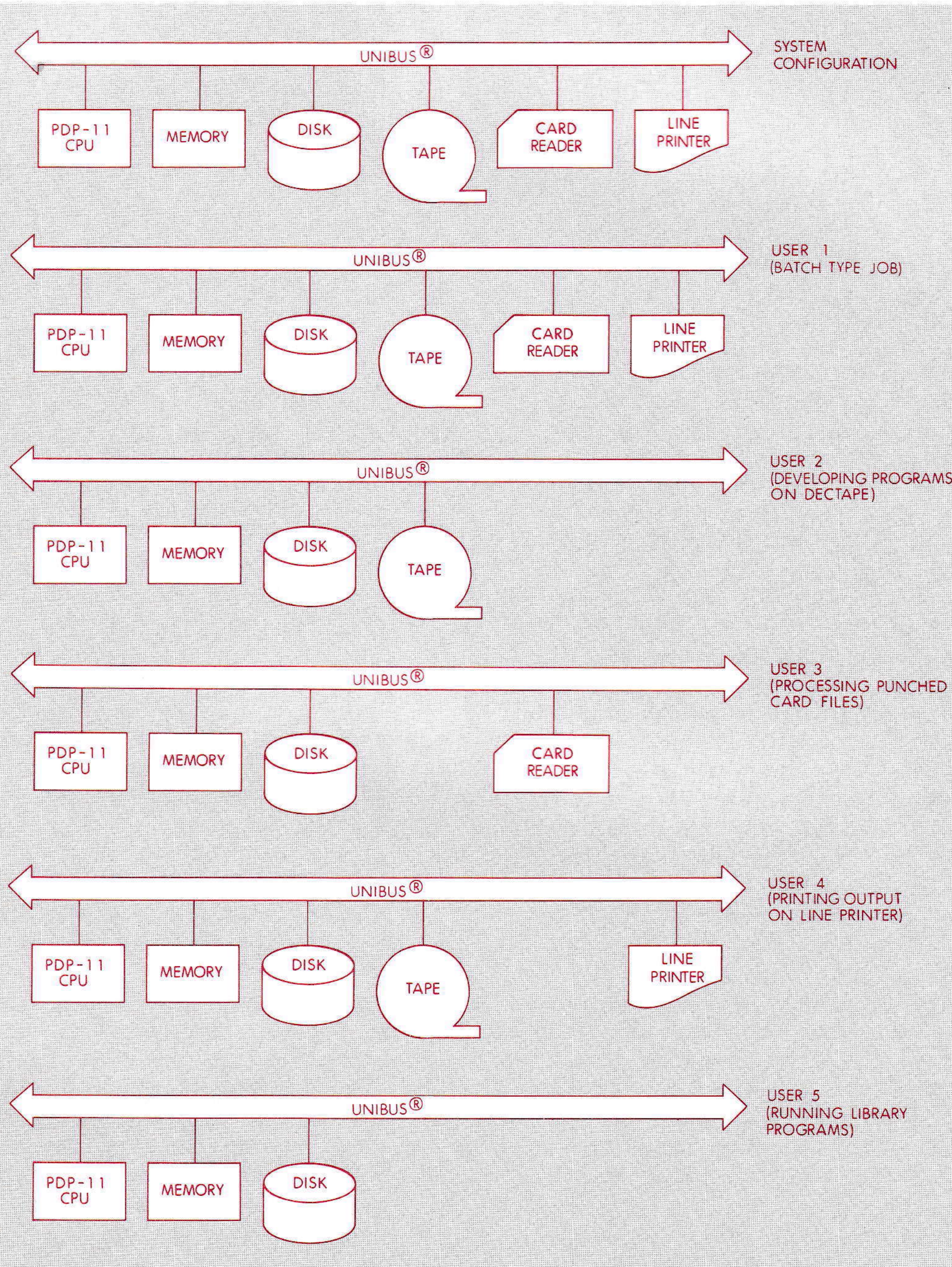
BASIC-PLUS	*Standard BASIC*
10 IF X>Y AND Y>Z THEN Z = X ELSE Z = Y	10 IF X<=Y THEN 50 20 IF Y<=Z THEN 50 30 LET Z=X 40 GOTO 60 50 LET Z=Y
10 A(Y1, Z1) = Z1+3 FOR Z1 = 1 TO L	10 FOR Z1 = 1 TO L 20 LET A(Y1, Z1) = Z1+3 30 NEXT Z1
10 A1 = PI*R1 IF R1 = 5.2 OR R1 = 0.6	10 IF R1 = 5.2 THEN 30 20 IF R1<>0.6 THEN 50 30 LET PI = 3.14159 40 LET A1 = PI*R1
10 GOTO 5530 UNLESS X1$>Y1$ AND Z1$ = "ABC"	(2 statements)
10 ON X(2, 5) GOTO 100, 150, 200, 250, 300	(5 statements)
10 A1, B, C = SIN (2*X*PI) IF R1> .5	(5 statements)

Resource Sharing

RSTS/E terminal users may have exclusive use of any peripheral on the system (except the public disk(s) which is a shared device). Not only can all devices be accessed by users at any time, but also any device can be accessed from a BASIC-PLUS program. Users may use a device as long as needed, and then return it for assignment to another user or program. The ability to enter, store, and retrieve programs and data files using high-speed peripheral devices makes RSTS/E a true general-purpose, problem-solving and data management system.

Examples of the value of the resource sharing concept are: one user may use the card reader, line printer, magtape and disk for performing a "batch" administrative data processing task; another terminal user may use a DECtape unit for retrieving or creating a tape file intended for off-line storage; and, when the card reader is free, yet another terminal user may read in a punched card file which contains a BASIC program created at an off-line card punch.

Access privileges to the system devices are controlled by the system manager during system initialization. However, once access has been granted, RSTS/E automatically allocates the use of the devices among those users having access privileges.



RSTS/E RESOURCE SHARING

Programmable Timing Control

BASIC-PLUS gives the user the ability to control certain operations in actual time. The SLEEP function allows the user to suspend his program from execution for a specified number of seconds. When the time specified has elapsed, execution resumes. Let us say a RSTS/E installation has a substantial number of users trying to print things on the line printer—invoices for example. Rather than each one of these users getting in a queue and inserting a SLEEP command in his program to wait a few seconds if the line printer is busy and then trying to access it again, consider this more elegant approach with BASIC-PLUS. Each user writes his line printer output into a specified disk file. Then a program running at, say, the system manager's terminal examines that disk file periodically and, if it has something in it, prints it out on the line printer. If the disk file is empty, the program sleeps a few seconds and examines it again. Optimum throughput with never a delay to the user!

In some applications, the length of time that a terminal user takes in responding to a message printed at his terminal is a significant variable. The WAIT function provides an interval timer feature which may be used for signaling the program that the terminal user has not responded within some predetermined length of time. One example of the use of the WAIT function is in Computer Assisted Instruction applications where one measure of student performance may be his "think time." If the student takes more than 5 seconds, for example, to respond to a question, the computer can restate the question in another manner, and record the delay as one element of the student's overall performance.

An additional real-time feature provides year, month, day, and time-of-day information to RSTS/E programs.

Formatted Output

Many applications, such as business data processing, require more flexible control of the printing format than Dartmouth BASIC allows. BASIC-PLUS includes a PRINT USING statement which may be used to achieve precise definition of printed data format. PRINT USING allows character, decimal, and exponential data field lengths and positions to be defined, and mixed, in a line of output. In addition, leading dollar sign or asterisk symbols may be "floated" to automatically precede the most significant digit of decimal fields. Also, trailing minus signs may be specified for compatibility with accounting report standards.

Format	*BASIC-PLUS*	*Standard BASIC*
Floating dollar sign	$95.20 $4,382.69 $0.43	$ 95.2 $ 4382.69 $ 0.43
Asterik fill	$***20.32 $**120.48	Not available
Comma insertion, decimal point alignment	4,832,684.15 1,497.00	4832684.15 1497
Trailing minus sign	572.83—	—572.83

Error Recovery

One of the more frustrating situations for a time-sharing terminal user is having his program cancelled because an input/output error condition occurs (perhaps temporary), therefore causing all results created (in a file, for example) to that point to be lost. This situation, though rare, may be eliminated in RSTS/E by use of the ON ERROR GOTO statement. This subroutine call statement is triggered by a variety of input-output operation errors. The called subroutine is passed a value which identifies the error type, and attempts to recover from the error condition. If the subroutine is successful, normal execution of the application program resumes.

Occasionally, problems will occur within the telephone system causing an unexpected disconnect for a remote user. In this event, the remote terminal will be detached from the job (system), but the program will continue processing. The user can then re-dial the computer system, re-attach to his job, and then continue interaction with his program.

In all cases, on hardware or software error, the file system is kept intact and secure. In the unlikely event of a system "crash," users will merely have to perform a simple determination of the status of their file processing at the time of the crash, and then continue.

SYSTEM MANAGEMENT

Efficient Scheduling Algorithm

RSTS/E installations can expect exceptional efficiency of operation because the operating system continuously and dynamically allocates processor time, memory space, file space and peripheral access on a best-fit/best throughput basis. The RSTS/E operating system automatically and dynamically assigns one of 255 job priority levels to each timesharing job. These priority levels are based on such criteria as job size, computing requirements, current time since last quantum of run time for the job, and input/output requirements. They may also be altered by the System Manager.

Disk allocation is made dynamically as users require. Users do not have to plan ahead for their use of disk space; however, additional efficiencies may be realized if they do. Specifying contiguous disk segments, as discussed previously in the "Record I/O" section, can decrease the number of disk accesses required for reading and writing large files.

Control of User Access and Resources

RSTS/E provides facilities to the System Manager for his accurate and effective control of system use. The System Manager may specify for each user his programmer and project number, his password, his maximum logged-out disk space, and his maximum number of files.

If desired, user access to the system can be controlled by the System Manager. In fact, if he wished, he could control access automatically through a program, thus relieving the tedium of system administration. For example, in a school, certain users could be automatically limited to 30 minutes of log-in time per day or to two log-ins per day. Should users not log-off at a designated time, the System Manager can force a log-off of the user's terminal which will preserve his files, but terminate his job execution.

Facilities are available for the System Manager to send messages to all terminal users. Also an automatic shutdown system is provided which periodically warns users that the system will shut down at a designated time. Any users still active at the designated time are logged off in an orderly fashion, with full integrity of all active files.

Access to peripheral devices is generally open to all users under the resource sharing concept on a first-come, first-serve basis. However, the capability is available to the System Manager to intervene in peripheral assignment and permit assignment as he sees fit. In addition, the System Manager can specify how the space on the system disks is to be allocated.

System Usage Accounting

The System Manager, as well as any terminal user, can determine the status of the RSTS/E system through use of the SYSTAT program. This program gives information on:

Status of all jobs
Disk structure and status
Status of other peripheral devices
Run time to date

A more detailed accounting of a specific user, or all users, is possible using the program MONEY. For each unique account, this yields information on:

CPU run time
Connect time of the user's terminal
Memory usage
Peripheral device usage
Number of log-ins and log-outs
Disk storage allocation

System and File Security

As mentioned above, to gain access to a RSTS/E system, a user must first have a programmer number assigned by the System Manager. After that, whenever he uses the system, he must identify himself correctly by entering his number and password (non-printing). Either the user or System Manager has the capability of changing this password at any time. This facility, when combined with the individual file access protection codes, (discussed in the section, "File Security") provides an effective means of safeguarding user data.

Additional protection can be provided by "private," removable disk packs and cartridges. A private disk is one upon which only authorized users (by the System Manager) may create files. Other users may access these files only if protection codes permit. Private disks may be mounted or dismounted from the on-line system at any time. When not in use, they may be kept under lock and key.

Disk File Integrity

An essential requirement of any computer system is to be able to restore on-line disk files to a known state in the event of system or program failure. RSTS/E provides two facilities to "back up" disk files—one facility is called "ROLLIN/ROLLOUT" and the other is called "BACKUP."

ROLLIN/ROLLOUT provides a means for transferring the entire contents of one or more mass storage devices to one or more other mass storage devices. Usually a disk would be copied to a magtape, one or more DECtapes, or another disk (as in the case of removable packs).

The program is run "off-line," that is, when RSTS is not in use. ROLLIN/ROLLOUT provides a very fast, non-file-structured copy; that is, copies between mixed media do not allow individual files to be restored without restoring the entire volume, which must also be done off-line.

The BACKUP program, on the other hand, is an on-line RSTS system program that allows selective file-structured copying on a file-by-file basis. An individual user may back up any or all of his files with a simple command.

The System Manager may specify that any or all files of any or all accounts be backed up. Restoration of the files is also accomplished on-line and provides the user with the same flexibility as in the backing up of his files—any or all files can be restored with a single command. Although the BACKUP program is slower than ROLLIN/ROLLOUT for the entire volume backup, it has the following advantages:

1. It is an on-line operation; other users continue to execute their programs undisturbed.
2. It provides a verification of the disk pack file structure and provides a list of all files backed up.
3. It can optionally delete inactive files at the operator's discretion.
4. It is selective—only those files requiring backup need be transferred.
5. It restores files in a manner which reduces the overhead required for accessing those files.

APPENDICES

APPENDIX A

HARDWARE SUMMARY

PDP-11/40 and PDP-11/45 Central Processing Unit

Memory		
Bipolar (300 nanosecond) (PDP-11/45 only)	0 to 16K	characters (bytes)
MOS (450 nanosecond) (PDP-11/45 only)	0 to 64K	characters (bytes)
Core (900 nanosecond interleaved)	80K to 248K	characters (bytes)
Floating Point Processor (PDP-11/45 only)		
Single-precision multiplication	5.6 microseconds	
Double-precision multiplication	9.3 microseconds	
Floating Point Unit (PDP-11/40 only)		
Single-precision multiplication	29.0 microseconds	
UNIBUS Transfer Rate	2.5M words/second	
Instruction Set	413	
Device Controllers	128	

Peripherals	Number Supported
Rotating Memory (Disks)	
High-speed, fixed-head, 512K characters, 17 millisecond access	1 to 8
Moving head, single cartridge, 2.4M, 70ms access	1 to 8
Moving head, 10-platter pack, 40M, 41ms access	1 to 8
Magnetic Tape	
DECtape unit	1 to 8
Magnetic tape, 7 or 9 track	1 to 8
Line Printer	
65 lines per minute, 132 column, 64 characters	1 to 4
300 lines per minute, 80 or 132 column 64 or 96 characters	1 to 2
1200 lines per minute, 132 column, 64 or 96 characters	1
Card Reader—300 cards per minute	1
High-speed paper tape reader and punch	1
Terminals	
DECwriter, 30 characters/second; DECterminal, 240 cps, CRT; Teletype, 10 cps; Remote modems	1 to 32

APPENDIX B

RSTS/E CONFIGURATION GUIDELINES

Four major configuration items must be determined for RSTS/E systems:

1. Memory (MOS and core)
2. Disks (swapping and storage)
3. Peripherals
4. Terminals and communications

Before configuring, the following factors should be estimated:

1. Number of simultaneous users (N).
2. Average program size in thousands of bytes (A). Experience has shown that each line of BASIC-PLUS code is approximately 54 bytes. A 60-line program is considered small (3,240 bytes), while a 500-line program is quite large (27,000 bytes). Important note: typically one BASIC-PLUS *line* contains two or more BASIC statements. In other words, a 500-line program could conceivably contain 1300 or more BASIC statements.

   This is a sample *line* of BASIC-PLUS code:

   ```
   10 A = B\PRINT A\IF A = C THEN 100
   ```

   These are BASIC-PLUS statements:

   ```
   A = B
   PRINT A
   IF A = C THEN 100
   ```
3. Estimated size of largest two programs (L1, L2) that will be run simultaneously. Maximum program size is 32,000 bytes.

Memory

The RSTS/E run time system which does terminal and peripheral scheduling and handling is 20K bytes in size. The BASIC-PLUS compiler is 28K bytes. This is all resident in main memory. For fewer than 20 users this can be resident in core memory; more than 20 users requires 32K bytes of high-speed MOS memory.

One-fourth of all active programs should be core resident and/or the two largest programs should fit in core.

Total bytes of main memory (MOS plus core) required can be determined from the larger of the following formulae:

1. $\text{Memory} = 48K + \frac{N \times A}{4}$

2. $\text{Memory} = 48K + L1 + L2$

Minimum memory required is 80K bytes. Memory should always be rounded up in 16K byte increments.

Disks

System code and utilities require 512K bytes (or characters) of disk space. This would normally be resident on a moving head storage disk (RK05 or RP03).

All simultaneous user programs must be able to reside on the swapping disk(s). Swapping disk space in bytes (or characters) may be determined from the following formula (minimum value for L1 in this fomula is 16K):

$$\text{Disk Space (characters)} = N \times L1$$

The RP03 may be used as a swapping and storage disk for up to 12 users. For more than 12 users, one or more RS11 disks (512K bytes each) must be used. The RK05 is used only for a storage disk on RSTS/E.

Storage disk space required is determined from the estimated size of data files that will be kept on line.

Peripherals

Addition of a Floating Point Processor (FP11-B) is strongly recommended for all systems serving more than 16 users.

Magnetic tape is highly recommended for backup on all systems with one or more RK05 or RP03 disks. Seven- or nine-track tape should be determined by compatibility requirements (if any) with other computer systems.

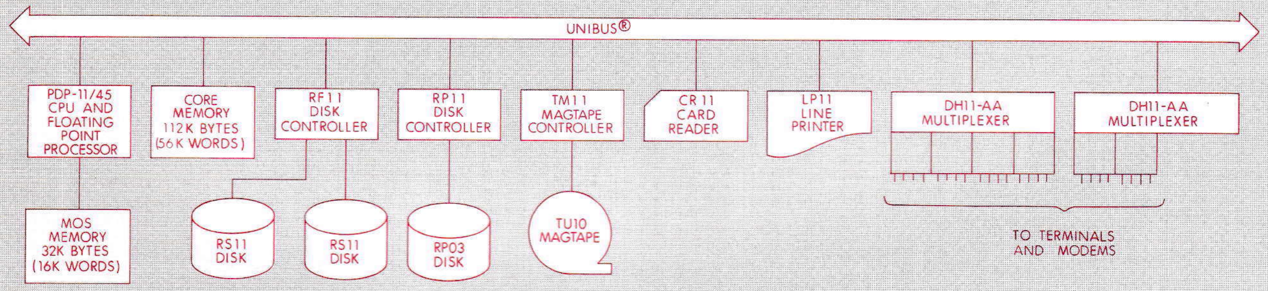
A line printer is normally recommended. All other peripherals are optional and depend entirely upon the applications for which the system is intended.

Terminals

RSTS/E handles up to 32 terminals using DH11/DM11 multiplexer equipment as outlined in Appendix C. Terminal speeds and distances from the CPU are defined in the table below.

	Maximum Speed (cps)	Maximum Distance (ft)
Local Terminals		
20 mA current loop type		
Teletype (LT33, LT35)	10	1,000
DECwriter (LA30-C)	30	1,000
DECterminal (VT05-B)	30	1,000
EIA/CCITT		
DECwriter (LA30-E)	30	25
DECterminal (VT05-B)	240	25
Remote Terminals		
EIA/CCITT		
DECwriter (LA30-E)	30	n/a
DECterminal (VT05-B)	30	n/a

SAMPLE 24-USER RSTS/E CONFIGURATION



Summary

- Typical configuration for data management application with 24 timesharing users
- File storage of 40 million characters (20 million words) on removable disk pack
- Fixed-head disk capacity of 1 million characters (512K words) for swapping and file storage
- PDP-11/45 CPU includes Floating Point Processor, Memory Management unit and 32K bytes of MOS memory to store RSTS/E run time system software.
- Industry-compatible magnetic tape
- 300 cpm punched card reader
- 300 lpm line printer
- RSTS/E software expandable to 32 users
- Approximate purchase price: $184,000. Approximate lease price including maintenance: $5,200/month.

APPENDIX C

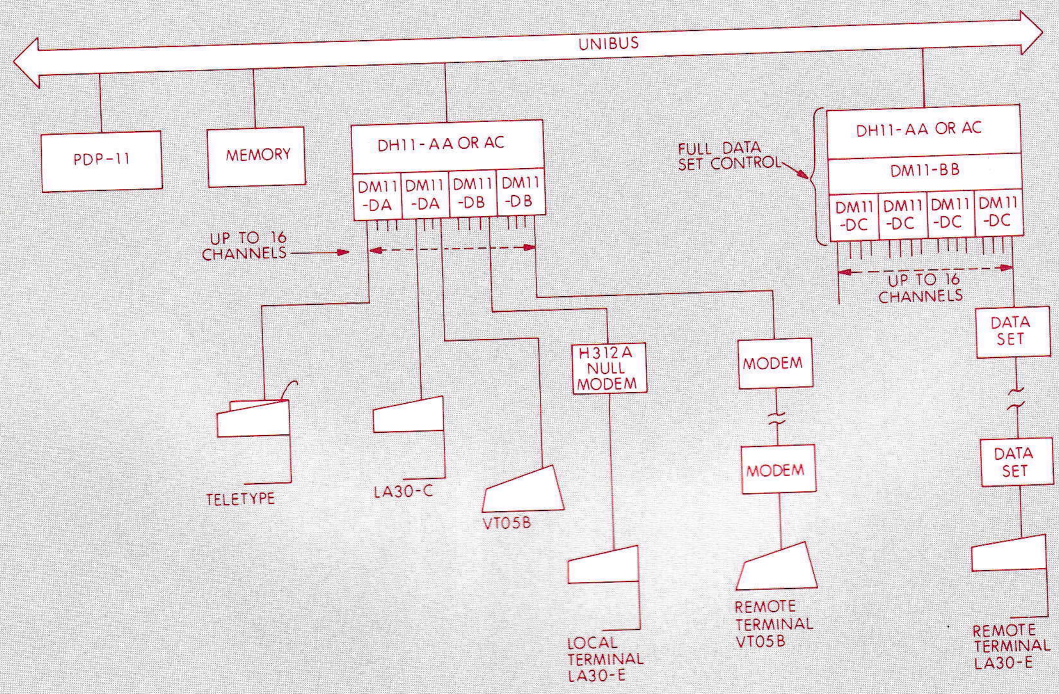
COMMUNICATIONS WITH RSTS/E

RSTS/E communicates with a wide variety of terminals. A program called TTYSET permits the System Manager or individual user to define the characteristics of a terminal, in particular:

1. Echoplex or full duplex operation
2. Terminal speed (local terminals up to 240 cps, remote terminals to 30 cps)
3. Upper case or upper and lower case characters
4. Hardware tab control
5. Form witdth control (to 254 columns)
6. Enable/disable paper tape reader/punch

Terminals are conected to RSTS/E via 16-line multiplexers and 4-line adapters. Some sample arrangements are shown below.

Line Adapter	*Type of Terminal*
DM11-DA	Local Terminal 20 mA current loop type Teletype (LT33, LT35) DECwriter (LA30-C) DECterminal (VTO5-B)
DM11-DB	Local EIA/CCITT DECwriter (LA30-E) DECterminal (VTO5-B) Private Line Modems
DM11-DC	Remote EIA/CCITT Bell 103, 202 or equivalent modems with data set control



APPENDIX D

RSTS/E (BASIC-PLUS) LANGUAGE SUMMARY

• Variables

Type	*Variable Name*	*Examples*
Floating Point	single letter optionally followed by a single digit	A I X3
Integer	any floating point variable name followed by a % character	B% D7%
Character String	any floating point variable name followed by a $ character	M$ R1$
Floating Point Matrix	any floating point variable name followed by one or two subscripts in parentheses	S(4) E(5,1) N2(8) V8(3,3)
Integer Matrix	any integer variable name followed by one or two subscripts in parentheses	A%(2) 1%(3,5) E3%(4) R2%(2,1)
Character String Matrix	any character string variable name followed by one or two subscripts in parentheses	C$(1) S$(8,5) A2$(8) V1$(4,2)

• Operators

Type	*Operator*		*Operates On*
Arithmetic	↑	exponentiation	numeric variables and constants
	−	unary minus	
	*,/	multiplication, division	
	+,−	addition, subtraction	
Relational	=	equals	string or numeric variables and constants
	<	less than	
	<=	less than or equal to	
	>	greater than	
	>=	greater than or equal to	
	<>	not equal to	
	==	approximately equal to	
Logical	NOT	logical negation	relational expressions composed of string or numeric elements with relational operators
	AND	logical product	
	OR	logical sum	
	XOR	logical exclusive or	
	IMP	logical implication	
	EQV	logical equivalence	
String	+	concatenation	string constants and variables
Matrix	+,−	addition and subtraction of matrices of equal dimensions, one operator per statement	dimensioned variables
	*	multiplication of conformable matrices	
	*	scalar multiplication of a matrix	

- **Functions**

Type	*Function*	*Explanation*
Mathematical	ABS(X)	returns the absolute value of X
	ATN(X)	returns the arctangent in radians of X
	COS(X)	returns the cosine of X in radians
	EXP(X)	returns the value of e^x, where e=2.71828
	FIX(X)	returns the truncated value of X, SGN(X)*INT(ABS(X))
	INT(X)	returns the greatest integer in X which is less than or equal to X
	LOG(X)	returns the natural logarithm of X, $\log_e X$
	LOG10(X)	returns the common logarithm of X, $\log_{10} X$
	PI	has a constant value of 3.1415926
	RND(X) or RND	returns a random number between 0 and 1
	SGN(X)	returns the sign function of X, a value of 0, or 1 preceded by the sign of X
	SIN(X)	returns the sine of X in radians
	SQR(X)	returns the square root of X
	TAN(X)	returns the tangent of X in radians
Print	POS(X)	returns the current position of the print head for the device X, 0 is the user's teletype. (This value is imaginary for disk files.)
	TAB(X)	moves print head to position X in the current or next print record, regardless of current position.
String	CHR$(X)	returns a character string having the ASCII value of X. Only one character is generated.
	ASCII(A$)	returns the ASCII value of the first character in the string A$.
	LEFT(A$,N)	returns a substring of the string A$ from the first character to the Nth character (the leftmost N characters).
	RIGHT(A$,N)	returns a substring of the string A$ from the Nth to the last character (the rightmost characters of the string starting with the Nth character).
	MID(A$,N1,N2)	returns a substring of the string A$ starting with character N1 and being N2 characters long (the characters between and including the N1 to N1+N2—1 characters).
	INSTR(N1,A$,B$)	indicates a search for the substring B$ within the string A$ beginning at character position N1. Returns a value of 0 if B$ is not in A$, and the character position of B$ if B$ is found to be in A$ (character position is indicated by the start of the string).
	VAL(A$)	computes the numeric value of the string of numeric characters A$. If A$ contains any character not acceptable as numeric input with the INPUT statement, an error results. For example: VAL("15") = 15
	NUM$(N)	returns a string of numeric characters representing the value of N as it would be output by a PRINT statement. For example: NUM$(1.0000) = 1, NUM$(n) = (space)n if n>0 and NUM$(n) = -n if n<0.
	SWAP(I%)	causes a "swap byte" operation to occur on the integer variable; returns the value with the bytes reversed.
	RAD$(I%)	converts an integer to a three character string. Used to convert a value, expressed in Radix 50 code, back to a string.
	LEN(A$)	returns the number of characters in the string A$, including trailing blanks.

Type	Function	Explanation
System	DATE$(0)	returns the current date.
	DATE$(N)	returns a character string corresponding to the Julian date N after 1970.
	TIME$(0)	returns the current time of day as a character string as follows: TIME$(O) = "05:30 PM"
	TIME$(N)	returns a string corresponding to the time at N minutes before midnight, for example: TIME$(1) = "11:59 PM" TIME$ (1400) = "12:40 AM" TIME$ (721) = "11:59 AM"
	TIME(0)	returns the clock time in seconds since midnight, as a floating point number.
	TIME(1)	returns the central processor time used by the current job in seconds.
	TIME(2)	returns the connect time (during which the user is logged into the system) for the current job in seconds.
	SLEEP X	dismiss this job for X seconds
	WAIT X	causes the currently running program to be dismissed for either X seconds or until a line is typed at the user terminal, whichever comes first. Generates error condition #15 if wait is exhausted.
	WAIT 0	causes the system to wait for input from the user terminal with no time limit.
Matrix	TRN(X)	returns the transpose of the matrix X.
	INV(X)	returns the inverse of the matrix X.
	DET	folowing an INV(X) function evaluation, the variable DET is equivalent to the determinant of X.

BASIC INSTRUCTION SUMMARY

REMARK
LET
DIM
RANDOMIZE
IF THEN
IF GOTO
FOR
NEXT
DEF
FNEND
INPUT
STOP
END
GOTO
ON-GOTO
GOSUB
ON-GOSUB
RETURN
CHANGE
READ
DATA
RESTORE
PRINT

BASIC-PLUS PROGRAM STATEMENTS*

IF . . . THEN-ELSE
FOR-WHILE
FOR-UNTIL
FOR-UNLESS
CHAIN
ON ERROR GOTO
RESUME
MAT READ
MAT PRINT
MAT INPUT
MAT ZER
MAT CON
MAT IDN
MAT TRN
MAT INV
PRINT USING

BASIC-PLUS INPUT/OUTPUT PROGRAM STATEMENTS

OPEN (name) AS FILE (number)
CLOSE
DIM #1, (virtual array)
INPUT #n
PRINT #n
INPUT LINE #n
FIELD #n
Get #n, RECORD
PUT #n, RECORD
NAME—AS
KILL

* Matrix Addition, Subtraction and Multiplication also possible.

BASIC-PLUS USER COMMANDS

Command	Explanation
HELLO	Indicates to RSTS that a user wishes to log onto the system. Allows the user to input project-programmer number and password.
BYE	Indicates to RSTS that a user wishes to leave the terminal. Closes and saves any files remaining open for that user.
NEW	Clear the user's area in core and allows the user to input a new program from the terminal. A program name can be indicated following the word NEW or when the system requests it.
OLD	Clears the user's area in core and allows the user to recall a saved program from a storage device. The user can indicate a program name following the word OLD or when the system requests it. If no device name is given, the file is assumed to be on the system disk.
APPEND	Allows the user to recall a saved program from a storage device and merge its statements into the current program in core.
DELETE	Allows the user to remove one or more lines from the program currently in core. Following the word DELETE the user types the line number of the single line to be deleted or two line numbers separated by a hyphen indicating the first and last line of the section of code to be removed.
LIST	Allows the user to obtain a printed listing at the user terminal of the program currently in core, or one or more lines of that program. The word LIST by itself will cause the listing of the entire user program. LIST followed by one line number will list that line; and LIST followed by two line numbers separated by a hyphen will list the lines between and including the lines indicated.
LISTNH	Same as LIST, but does not print header containing the program name and current date.
CONT	Allows the user to continue execution of the program currently in core following the execution of a STOP statement or the typing of a CTRL/C.
RUN	Allows the user to begin execution of the program currently in core. The word RUN can be followed by a file name in which case the file is loaded from the system disk, compiled, and run; alternatively, the device and file name can be indicated if the file is not on the system disk. A device specification without a file name will cause a program to be read from an input only device (such as high-speed reader, card reader, etc.).
RUNNH	Same as RUN, but does not print header containing the program name and current date.
SAVE	Causes the program currently in core to be saved on the system disk under its current file name with the extension .BAS. When the word SAVE is followed by a file name or a device and a file name, the program in core is saved under the name given and on the device specified. A device specification without a file name will cause the program to be output to an output only device (line printer, high-speed punch, etc.).
UNSAVE	The word UNSAVE is followed by the file name and extension of the file to be removed. If no device is specified, the disk is assumed.
RENAME	Causes the name of the program currently in core to be changed to the name specified after the word RENAME.
REPLACE	Same as SAVE, but allows the user to substitute a new program with the same name for an old program, erasing the old program.
COMPILE	Allows the user to store a compiled version of his BASIC program. The file is stored on disk with the current name and the extension .BAC. Or, a new file name can be indicated and the extension .BAC will still be appended.
LENGTH	Returns the length of the user's current program in core.
CATALOG	Returns the user's file directory. Unless another device is specified folowing the word CATALOG, the disk is the assumed device.
TAPE	Used to disable the echo feature on the terminal while reading paper tape via low-speed reader.
KEY	Used to re-enable the echo feature on the user terminal following the issue of a TAPE command.
ASSIGN	Used to reserve an I/O device for the use of the individual issuing the command. The specified device can then be given commands only from the terminals which issued the ASSIGN.
DEASSIGN	Used to release the specified device for use by others. If no particular device is specified, all devices assigned to that terminal are released. An automatic DEASSIGN is performed when the BYE command is given.

BASIC-PLUS SPECIAL CONTROL CHARACTERS

Command	*Explanation*
RETURN Key	Enters a typed line to the system; results in a carriage return/line feed operation at the user terminal.
ESCape or ALT MODE Key	Enters a typed line to the system, echoes on the user terminal as a $ character and does not cause a carriage return/line feed.
LINE FEED Key	Used to continue the current logical line on an additional physical line. Performs a carriage return/line feed operation.
RUBOUT Key	Deletes the last character typed on that physical line. Erased characters are indicated on the terminal between back slashes.
CTRL/C	Causes the system to return to BASIC command mode to allow for issuing of further commands or editing. Echoes on terminal as ↑C.
CTRL/U	Deletes the current typed line, echoes as ↑U and performs a carriage return/line feed.
CTRL/O	Used as a switch to suppress/enable output to the user terminal.
TAB or CTRL/I	Performs a tabulation to the next of nine tab stops (eight spaces apart) which form the terminal printing line.
CTRL/Z	Used as an end-of-file character.

APPENDIX E

MAJOR DIFFERENCES BETWEEN RSTS/E AND RSTS-11

RSTS/E	RSTS-11
PDP-11/40 or PDP-11/45	PDP-11/20, PDP-11/40, or PDP-11/45
Memory 80K to 248K characters (bytes)	Memory 48K to 56K
Requires parity memory	Parity memory not required
Swapping disk RS11 or RP03	Swapping disk RS11 or RK05
32 users with PDP-11/45; 16 users with PDP-11/40 unless average job size is under 8K bytes	16 users maximum
RSTS/E run time system can be stored in high-speed MOS memory	Not available
High-speed hardware floating point processor	Not available
32K byte maximum program size	16K byte maximum program size
Terminal communication via 16-line multiplexers or individual interfaces	Terminal communication via individual interfaces
Supports 40M character RP03 disk	Not supported
Supports 1200 lpm LP11 line printer	Not supported
Supports multiple line printers	Not supported
Comprehensive error logging for easy system maintenance	Not supported

APPENDIX F

SAMPLE ORDER ENTRY/INVENTORY CONTROL PROGRAM

Included below is a sample BASIC-PLUS program that is part of a hypothetical order entry/inventory control system. The entire system might include programs to maintain the inventory and customer master files, keep track of accounts receivable, produce purchase orders for those inventory items below minimum stock levels, keep track of accounts payable, etc. This sample program would be the part of the system utilized for entering new orders and producing invoices (a second copy of which might be used for warehouse picking slips). Clerks would utilize RSTS/E terminals to enter orders specifying an order number, a customer number, and an item number and quantity for each item on the order. The order entry program would check the inventory level for that item, back-ordering any items with insufficient stock. In addition the system would perform all price extensions and invoice totalling, applying any discount to which that customer is entitled. Finally, the program would update the customer's year-to-date total sales volume and his current receivable, signaling any order for which the customer has exceeded his line of credit.

Please note that this program is intended as an example only, with its primary purpose being to demonstrate BASIC-PLUS language features. It does not constitute part of a "recommended" procedure for implementing a complex order entry system.

The following documentation is included with the program:

1. A layout of the inventory and customer master files
2. A flowchart of the program
3. A list of the program variables
4. A listing of the program
5. The terminal printout from a sample run of the program
6. The invoices produced from a sample run of the program

ORDER ENTRY FILE LAYOUT

Inventory File

Name: INVENT.ORY
Type: Virtual Memory Array
Index By: Item No.
Layout of Array Items:

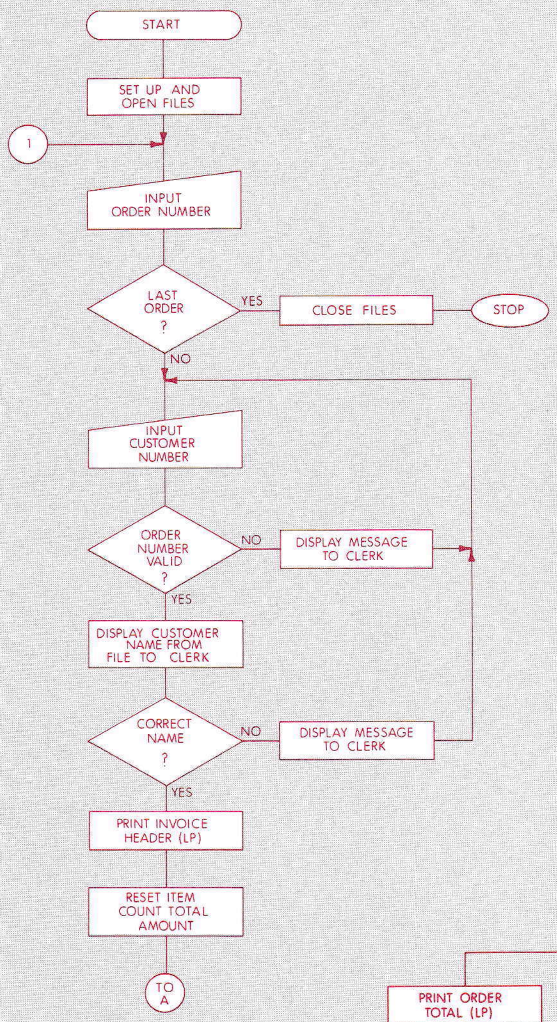
Field	*Type*	*Contents*
I$ (N) =	32 character string	— Item description
I (N, 0) —	Floating point	— Item Price
I (N, 1) —	Floating point	— Quantity on hand (if negative indicates backlog)

Customer File

Name: CUSTOM.ER
Type: Virtual Memory Array
Indexed by: Customer No.
Layout of Array Items:

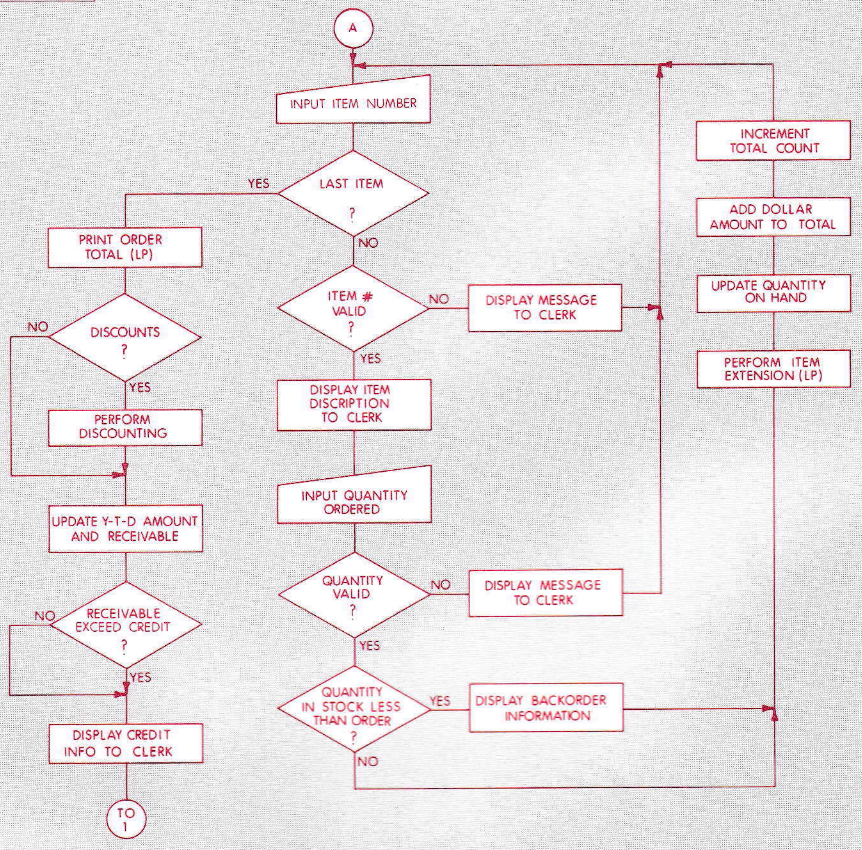
Field	*Type*	*Contents*
C$ (N, 0) =	32 character string	— Customer Name
C$ (N, 1)	"	— Billing Address
C$ (N, 2)	"	— Shipping Address
C (N, 0) —	floating point	— Y-T-D Total Order and Volume
C (N, 1)	"	— Current Receivable and Amount
C (N, 2)	"	— Maximum Credit Line ($)
C (N, 3)	"	— Discount %

PROGRAM FLOW



List of Program Variables

Variable	Contents
D =	Discounted Total Amount of Order
I =	Inventory Item No.
N =	Customer No.
O =	Order No.
P =	Discount Percentage
Q =	Item Quantity Ordered
S =	Serial Line Item No. on Invoice
T =	Total Amount of Order



```
10 REM  PROGRAM NAME: ORDER
        SAMPLE PROGRAM TO PERFORM ORDER ENTRY AND
        INVOICE GENERATION

        USES TWO FILES:
        1)AN INVENTORY MASTER FILE
        2)A CUSTOMER MASTER FILE

        PRODUCES INVOICES ON THE LINE PRINTER

20 REM  FILE SETUP SECTION

25 INPUT "PRINT INVOICES ON WHICH DEVICE";P$     !NORMALLY LP:
30 OPEN "INVENT.ORY" AS FILE 1
        :OPEN "CUSTOM.ER" AS FILE 2
        :OPEN P$ AS FILE 3
40 DIM #1,I$(200)=32,I(200,1)
   :DIM #2,C$(100,2)=32,C(100,3)

100 REM          PROCESS AN ORDER

110 PRINT : INPUT "ORDER NUMBER";O
        :IF O=0 THEN CLOSE 1,2,3 : GOTO 600 !EXIT IF ORDER # = 0
120 INPUT "CUSTOMER NUMBER";N
        :IF N > 0  AND  N < 101 THEN 130 ELSE
           PRINT "INVALID CUSTOMER NUMBER" : GOTO 120
130 PRINT "CUSTOMER IS " C$(N,0);", RIGHT"; : INPUT A$
        :IF LEFT (A$,1) <> "Y" GOTO 120          !VERIFY CUSTOMER NAME

                 PRINT INVOICE HEADER

140 PRINT #3 CHR$(12)+"INVOICE FOR ORDER NO.";O
        :PRINT #3                         !TOP OF FORM AND HEADING
150 PRINT #3 "SOLD TO:",C$(N,0)
        :PRINT #3 ,,C$(N,1) : PRINT #3
        :PRINT #3 "SHIP TO:",C$(N,2) : PRINT #3 !PRINT NAME & ADDRESS
160 PRINT #3 "ITEM  QUANTITY  DISCRIPTION",TAB(50);"UNIT COST  "+
        "COST OF ITEM"
        :PRINT #3 "----  --------  ---------------------------------"+
        "  ----------  -------------" : PRINT #3
170 S=1 : T=0 !RESET SERIAL LINE ITEM NO., TOTAL DOLLAR AMOUNT

                 ACCEPT LINE ITEMS

200 INPUT "ITEM NUMBER";I : IF I=0 THEN 500 ELSE
        IF I > 0  AND  I < 201 THEN 210 ELSE
           PRINT "NO ITEM FOUND WITH THAT NUMBER" : GOTO 200

210 PRINT "ITEM" I; "IS "I$(I)
        :INPUT "QUANTITY DESIRED";Q
        :IF Q<=0 GOTO 200          !CHECK THAT THIS IS WHAT HE WANTS
220 PRINT #3 USING " ##     ######    ",S,Q; !PRINT ITEM SERIAL & QUANTITY
230 PRINT #3 I$(I);TAB(50);                 !PRINT DESCRIPTION
235 PRINT #3 USING "#,###.##      #,###.##",I(I,0), Q*I(I,0);
                                      !PRINT UNIT PRICE & EXTENSION

                 CHECK FOR ENOUGH INVENTORY
                 - IF INSUFFICIENT, SHOW BACKORDERED

240 IF Q <= I(I,1) THEN 260 ELSE
        IF I(I,1) > 0 THEN 250 ELSE
           PRINT #3 " BACKORDERED!"; :
```

```
        PRINT "ALL BACKORDERED!" : GOTO 260
250 PRINT Q-I(I,1);"OF THESE MUST BE BACKORDERED!"
      :PRINT #3 Q-I(I,1);"BACKORDERED!";
260 PRINT #3                      !FINISH THE LP LINE

      DO THE EXTENSION, UPDATE THE QUANTITY ON HAND,
            AND KEEP TRACK OF THE ORDER TOTAL AMOUNT

300 PRINT USING "#### ITEMS SOLD AT #,###.## EACH FOR A TOTAL OF"+
      "$$#,###.##",Q, I(I,0), Q*I(I,0)
310 I(I,1)=I(I,1)-Q : T=T+Q*I(I,0) : S=S+1
      : GOTO 200


500 REM          END OF ORDER
                 PRINT TOTALS AND DO CREDIT CHECK

510 PRINT S-1;"ITEMS THIS ORDER FOR A TOTAL OF";
      :PRINT USING "$$##,###.##",T

      !CHECK FOR DISCOUNTING

520 PRINT #3 TAB(61);"------------"
      :PRINT #3 TAB(54);"TOTAL";
      :PRINT #3 USING "$$##,###.##",T

            !CHECK FOR DISCOUNTING

530 P=C(N,3) : IF P=0 GO TO 560
540 D=T-P*T                       !CLACULATE DISCOUNTED TOTAL
550 PRINT "DISCOUNTED TOTAL AT"P*100;"% DISCOUNT LEVEL IS";
      :PRINT USING "$$##,###.##",D
      :PRINT #3 : PRINT #3 TAB(43);"DISCOUNTED TOTAL";
      :PRINT #3 USING "$$##,###.##",D

560 REM          UPDATE YEAR-TO-DATE AND CURRENT RECEIVABLE

565 C(N,0)=C(N,0)+D : C(N,1)=C(N,1)+D

      !CHECK FOR CREDIT LINE EXCEEDED.
      IF NOT, PROCESS NEXT ORDER

570 IF C(N,1) <= C(N,2) GOTO 110
580 PRINT USING "CUSTOMER HAS A CURRENT RECEIVABLE OF $$##,###.##"
      C(N,1)
      :PRINT USING "YET HE HAS A CREDIT LINE OF ONLY $$##,###.##"
      C(N,2)
      :PRINT USING "WHICH IS $$##,###.## OVER HIS LIMIT.",C(N,1)-C(N,2)
590 PRINT USING "THIS RECEIVABLE REFLECTS ###.#% OF HIS BUSINESS THIS YEAR"
      C(N,1)*100/C(N,0)
      :GOTO 110
600 END
```

SAMPLE PROGRAM DIALOGUE

```
RUN ORDER

PRINT INVOICES ON WHICH DEVICE? LP:

ORDER NUMBER? 45678
CUSTOMER NUMBER? 18
CUSTOMER IS BANANA BALLPOINT PEN CO. , RIGHT? YES
ITEM NUMBER? 55
ITEM 55 IS PDP-11 CERTIFIED DECTAPE
QUANTITY DESIRED? 24
 4 OF THESE MUST BE BACKORDERED!
  24 ITEMS SOLD AT    12.00 EACH FOR A TOTAL OF    $288.00
ITEM NUMBER? 190
ITEM 190 IS RP03 DISK PACK (10 HIGH)
QUANTITY DESIRED? 3
   3 ITEMS SOLD AT   475.00 EACH FOR A TOTAL OF $1,425.00
ITEM NUMBER? 116
ITEM 116 IS LINE PRINTER RIBBON (LS11)
QUANTITY DESIRED? 4
   4 ITEMS SOLD AT     7.50 EACH FOR A TOTAL OF     $30.00
ITEM NUMBER? 84
ITEM 84 IS RK05 DECPACK  2200 BPI
QUANTITY DESIRED? 5
   5 ITEMS SOLD AT    99.00 EACH FOR A TOTAL OF    $495.00
ITEM NUMBER? 0
 4 ITEMS THIS ORDER FOR A TOTAL OF    $2,238.00
DISCOUNTED TOTAL AT 10% DISCOUNT LEVEL IS    $2,014.20
CUSTOMER HAS CURRENT RECEIVABLES OF  $63,146.40
YET HE HAS A CREDIT LINE OF ONLY  $61,000.00
WHICH IS   $2,146.40 OVER HIS LIMIT.
THIS REFLECTS 60.4% OF HIS BUSINESS THIS YEAR

ORDER NUMBER? 0


READY
```

SAMPLE LINE PRINTER OUTPUT

```
SOLD TO:     BANANA BALLPOINT PEN CO.
             12 ASCOTT BLVD.
             INKVILLE, NEBR.

SHIP TO:     495 PARK AVE.
             NEW YORK, NY  10017

PURCHASE ORDER NO.   45678

ITEM  QUANTITY  DESCRIPTION                       UNIT COST  TOTAL COST
----  --------  -----------                       ---------  ----------

 1        24    PDP-11 CERTIFIED DECTAPE              12.00      288.00   4 BACKORDERED
 2         3    RP03 DISKPACK (10 HIGH)              475.00    1,425.00
 3         4    LINE PRINTER RIBBON (LS11)             7.50       30.00
 4         5    RK05 DECPACK  2200 BPI                99.00      495.00
                                                             ----------
                                                  TOTAL      $2,238.00

                                       DISCOUNTED TOTAL      $2,014.20
```

APPENDIX G

SAMPLE SORT PROGRAM

This sample program shows how an alphanumeric sort can be performed in BASIC-PLUS and compares this procedure to the same sort done in another "advanced" BASIC without virtual memory. The BASIC-PLUS program also demonstrates the use of integer variables for indices and subscripts and the use of character string arrays. Since sorts and merges are done frequently in all types of applications, it is important that the system be efficient in performing this operation.

In standard BASIC, as each name is read from the data list, it is explicitly written onto the disk (Statement 80); this is not necessary in BASIC-PLUS, as the list is simply stored in a character string array, A$, in virtual memory. This is automatically performed (Statement 70) and is transparent to the programmer.

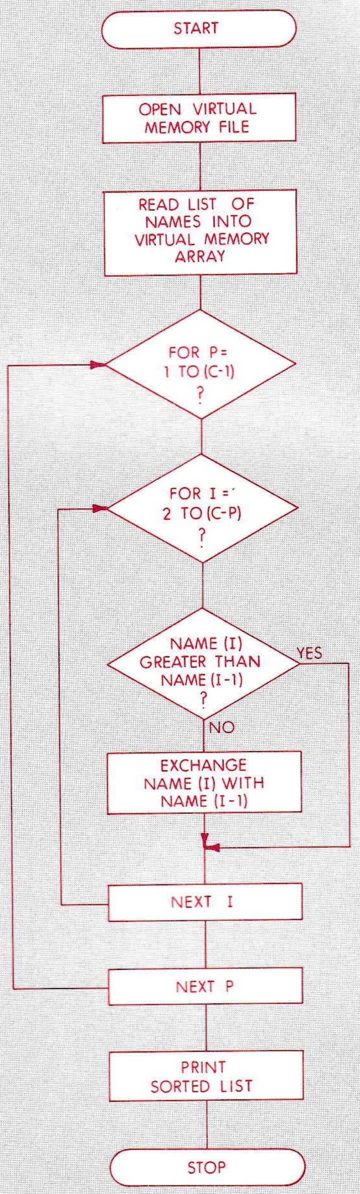
In the actual sort (Statements 700-820) in standard BASIC each pass through the list requires two explicit read and two explicit write statements (730, 740, 790, 800). BASIC-PLUS does not require these explicit read/write statements.

Again, when the list is printed out in standard BASIC, each name must be explicitly read from the disk (Statement 840); this is unnecessary in BASIC-PLUS.

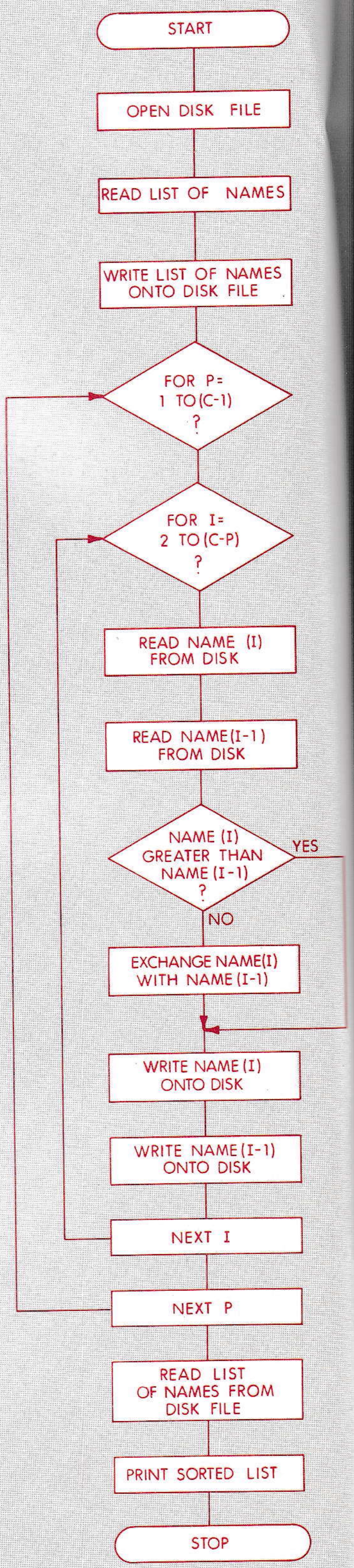
BASIC-PLUS also allows the use of integers as indices and subscripts (indicated by a "%" following the variable name or value). The computer system manipulates only one byte (character) when these variables are used in the program rather than 2 or 4 characters required by standard floating point variables in standard BASIC.

If the list of names is in random order, approximately 7320 disk accesses would be required in standard BASIC to sort a list of 60 names. With virtual memory in BASIC-PLUS, approximately 240 accesses are required, or only 1/30 that required by standard BASIC. In processing time, this program will run in about 30 seconds on RSTS/E versus over 10 minutes on a "comparable" system with standard BASIC.

BASIC-PLUS



STANDARD BASIC



SAMPLE SORT PROGRAM

BASIC-PLUS

```
10 DIM #1%, A$(100%)=32%
20 OPEN "FILE" AS FILE 1%
50 C%=60%
60 FOR I%=1% TO C%
70 READ A$(I%)

90 NEXT I%
100  }
     } DATA
690  }
695 C1%=C%-1%
700 FOR P%=1% TO C1%
710 X%=C%+1%-P%
720 FOR I%=2% TO X%
725 J%=I%-1%


750 IF A$(I%)>A(J%) THEN 810
760 C$=A$(I%)
770 A$(I%)=A$(J%)
780 A$(J%)=C$


810 NEXT I%
820 NEXT P%
830 FOR I%=1% TO C%

850 PRINT A$(I%)
860 NEXT I%
999 END
```

STANDARD BASIC

```
10 DIM A$(20), B$(20), C$(20)
20 FILES SORT1
50 C=60
60 FOR I=1 TO C
70 READ A$
80 PRINT #1, I; A$
90 NEXT I
100  }
     } DATA
690  }
695 C1=C-1
700 FOR P=1 TO C1
710 X=C+1-P
720 FOR I=2 TO X

730 READ #1, I; A$
740 READ #1, I-1; B$
750 IF A$>B$ THEN 810
760 C$=A$
770 A$=B$
780 B$=C$
790 PRINT #1, I; A$
800 PRINT #1, I-1; B$
810 NEXT I
820 NEXT P
830 FOR I=1 TO C
840 READ #1, I; A$
850 PRINT A$
860 NEXT I
999 END
```

APPENDIX H

```
LIST
RANGE    01:26 PM            06-APR-73
10 PRINT:PRINT "RADIUS","AREA":PRINT
20 A=.5
30 A=10*A
40 PRINT SQR(A),PI*A
50 GOTO 30
99 END

READY

RUNNH

RADIUS         AREA

 2.23607        15.708
 7.07107        157.08
 22.3607        1570.8
 70.7107        15708
 223.607        157080
 707.107        .15708E 7
 2236.07        .15708E 8
 7071.07        .15708E 9
 22360.7        .15708E 10
 70710.7        .15708E 11
 223607         .15708E 12
 707107         .15708E 13
 .223607E 7     .15708E 14
 .707107E 7     .15708E 15
 .223607E 8     .15708E 16
 .707107E 8     .15708E 17
 .223607E 9     .15708E 18
 .707107E 9     .15708E 19
 .223607E 10    .15708E 20
 .707107E 10    .15708E 21
 .223607E 11    .15708E 22
 .707107E 11    .15708E 23
 .223607E 12    .15708E 24
 .707107E 12    .15708E 25
 .223607E 13    .15708E 26
 .707107E 13    .15708E 27
 .223607E 14    .15708E 28
 .707107E 14    .15708E 29
 .223607E 15    .15708E 30
 .707107E 15    .15708E 31
 .223607E 16    .15708E 32
 .707107E 16    .15708E 33
 .223607E 17    .15708E 34
 .707107E 17    .15708E 35
 .223607E 18    .15708E 36
 .707107E 18    .15708E 37
 .223607E 19    .15708E 38
 .707107E 19    .15708E 39
FLOATING POINT ERROR AT LINE 20
0               0
```

SIZE AND PRECISION OF NUMBERS IN RSTS/E

The first program (RANGE) computes the area of a circle with its radius increasing in steps by the square root of 10. Note that the last area calculated (1.5708×10^{38}) is near the limit of BASIC-PLUS, yet precision is fully maintained.

The second program (DIGITS) calculates square roots to 17 decimal digits of accuracy, thereby demonstrating the high precision of BASIC-PLUS.

```
LIST
DIGITS  04:16 PM         10-MAY-73
5 PRINT "PROGRAM PRINTS SQUARE ROOTS TO 17 DECIMAL DIGITS"
6 PRINT "OF PRECISION":PRINT
10 PRINT "YOUR NUMBER";
15 INPUT N
18 PRINT "SQUARE ROOT IS ";
20 A=SQR(N)
25 I=0:N=1
30 I=I+1:N=N*10
35 IF A/N>1 THEN 30
40 A=A/N
50 FOR J=1TO 17
55 B=A*10
60 A=B-INT(B)
65 PRINT CHR$(48+INT(B));
70 IF J=I THEN 75 ELSE 80
75 PRINT ".";
80 NEXT J
90 PRINT:PRINT
95 GOTO 10
99 END

READY

RUNNH
PROGRAM PRINTS SQUARE ROOTS TO 17 DECIMAL DIGITS
OF PRECISION

YOUR NUMBER? 2
SQUARE ROOT IS 1.4142135623730950

YOUR NUMBER? 200
SQUARE ROOT IS 14.142135623730950

YOUR NUMBER? 20000
SQUARE ROOT IS 141.42135623730950

YOUR NUMBER? 2000000
SQUARE ROOT IS 1414.2135623730950

YOUR NUMBER? 625
SQUARE ROOT IS 25.000000000000000

YOUR NUMBER? 624.999999999999999
SQUARE ROOT IS 24.999999999999998

YOUR NUMBER? 9
SQUARE ROOT IS 3.0000000000000000

YOUR NUMBER? 9.000000000000001
SQUARE ROOT IS 3.0000000000000001

YOUR NUMBER? 473.9452741
SQUARE ROOT IS 21.770284198879903

YOUR NUMBER? ^C

READY
```

APPENDIX I

SAMPLE 32-USER SYSTEM STATUS REPORT FROM RSTS/E

```
RUN $ SYSTAT
OUTPUT STATUS TO?

RSTS V05-16 MARKETING 45 STATUS ON 12-APR-73 AT 01:10 PM UP: 1:44:36

JOB    WHO        WHERE    WHAT     SIZE    STATE      RUN-TIME
 1      1,77      KB0      BACKDK   8K      RN             3.5
 2    100,126     KB16     CAI2     15K     RN  SW        35.3
 3    100,105     KB21     LIST     3K      TT  SW        58.1
 4     **,**      KB29     LOGIN    4K      KB  SW         0.1
 5    100,107     KB23     CAI2     15K     TT  SW      1:13.4
 6    100,109     KB25     CAI2     15K     TT  SW      1:14.0
 7    100,104     KB20     LIST     3K      RN  SW        59.1
 8    100,110     KB26     CAI2     15K     TT  SW      1:15.4
 9    100,103     KB27     LIST     3K      TT            56.9
10    100,101     KB30     MATRIX   13K     RN          9:03.7
11    100,100     KB2      DIGITS   2K      KB  SW      1:14.2
12    100,106     KB22     LIST     3K      TT  SW        55.2
13    100,108     KB24     CAI2     15K     RN          1:10.7
14    100,111     KB31     CAI2     15K     RN  SW      1:07.7
15    100,118     KB12     STAT     7K      RN  SW      4:12.9
16    100,112     KB28     CAI1     7K      RN  SW      1:03.0
17    100,130     KB1      SYSTAT   6K      RN            45.7
18    100,115     KB3      MATRIX   13K     RN  SW      7:18.2
19    100,124     KB15     CAI3     6K      TT  SW        22.8
20      1,7       KB8      XQWIK    9K      RN  SW      3:15.0
21    100,121     KB11     EDITCH   8K      RN  SW         2.0
22    100,114     KB7      1LINE    3K      KB  SW      4:16.5
23    100,119     KB4      STAT     7K      RN  SW      3:33.0
24    100,116     KB5      STAT     7K      RN  SW      4:04.3
25    100,115     KB6      ABC      2K      TT  SW      1:39.1
26    100,120     KB10     ROCKET   5K      KB  SW         8.5
27    100,117     KB13     STAT     7K      RN  SW      4:17.7
28    100,123     KB18     LIST     3K      RN  SW        30.3
29    100,124     KB14     WEKDAY   5K      TT  SW      1:39.4
30    100,124     KB19     CRAPS    5K      KB  SW      1:23.5
31    100,113     KB9      SMASH    6K      RN  SW      2:38.0
32    100,122     KB17     LIST     3K      RN  SW        43.4

BUSY DEVICES:
DEVICE   JOB      WHY
DT0       11      AS

DISK STRUCTURE:
DISK     OPEN     FREE    CLUSTER ERRORS  COMMENTS
DP0       26      67598     2       0     PUBLIC

SMALL    LARGE
 82       2

READY
```

DESCRIPTION OF PROGRAMS

User Job	*Quantity*	*Description*
MATRIX	2	A 40x40 matrix is loaded with random numbers and copied into another 40x40 matrix. The second matrix is inverted and multiplied by the original, printing out the diagonal of the result. All data storage is in core. Heavy CPU load. Re-runs continually.
STAT	4	A program which teaches introductory statistics and goes compute bound for long periods of time. Re-runs continually.
LIST	6	Continually types the contents of a disk file on a terminal.
CAI 1, 2, 3	8	Interactive question and answer sequences with 2 sec. time out transfer if no terminal response is given. The program consists of 3 chained links, 2 of which require 15,000 words of main memory. Re-runs continually.
SMASH	1	Creates and moves files around on the disk. Simultaneously reads 6 and writes 6 files which are copies of input files. Files contain 100 records of five number fields each in alphanumeric mode. SMASH includes matrix storage of 40x40 floating point numbers. Re-runs continually.
XQWIK	1	SORT program (one of three modules) which is sorting a 243 record file, 24 characters/record. Sorting on 2 key types in 3 byte ascending ASCII followed by a 2 byte descending signed integer. Re-runs continually.
EDITCH	1	DOS compatible character editor.
BACKDK	1	On-line backup and recall utility program for saving and restoring disk files selectively by: file name or extension, similar names, by account or by creation date.
LOGIN	1	Login utility in process of bringing on a new user.
SYSTAT	1	System status utility program which produced the attached job profile. Re-runs continually.
ABC	1	User program.
1LINE	1	User program.
DIGITS	1	Game.
ROCKET	1	Moon landing simulation.
WEKDAY	1	Game.
CRAPS	1	Game.
	32	

APPENDIX J

Commercial Extensions to RSTS/E

In order to provide those system facilities common to most commercial data processing operations, many enhancements to the standard RSTS/E systems are provided by the DDS 500 Commercial Timesharing System software which is available as an extra-cost option. These enhancements are known collectively as the "Commercial Extensions to RSTS" and include the following:

- Disk sort/merge program
- Indexed file access method
- Decimal arithmetic capability
- Line printer spooling package.

These features are described below in greater detail.

Disk Sort Package

The RSTS Disk Sort Package is a series of programs allowing the user to sort the records of a disk file into some order, based upon the contents of those fields of the records designated as "SORT keys." Up to 15 separate key fields may be selected, with the sequencing on each key being specified as ascending or descending. Separate keys may be of different length and may be of alphanumeric, integer, floating point or decimal type. The input data file may contain up to 32,767 records of up to 512 characters each.

Records must be of fixed length from 5 to 512 characters, aligned on block boundaries for maximum sorting efficiency. Formatted ASCII files may also be sorted.

The technique used for the sort is as follows: The data file is read and the keys extracted. A record is made consisting of the extracted keys and the data record's disk address. These key-address records are then sorted. Since keys are usually much shorter than the records from which they are derived, this technique greatly reduces sort time and disk work file space. Output from the sort may be either the sorted key-address file or a fully-resequenced data file. If key-address file output is elected, use of a subroutine allows user program retrieval of the data file records in the sequence of the key-address file. As the data file can be associated with several different sorted key-address files, the same data file can be retrieved in several different sequences. The disadvantage of this technique is that, in its use of indirect reference of the data file through the key file, the time for record access increases. If a data file is to be accessed primarily in one sequence, it is most efficient to select output of a fully-resequenced data file.

The SORT Program may be called from a user program or may be initiated via interactive dialogue.

Indexed File Access Method

The Indexed Access Method (IAM) facility allows the user to access disk file data records randomly by specifying merely a record's key rather than its disk file address. In this way, a user can achieve fast, random access to his data records without having to be concerned with the intricacies of disk file organization. Sequential processing of these records is supported either directly (if there have been no records added to the file since the last file reorganization) or by means of a key-address file output from the SORT package.

In addition to specifying the key of the record to be accessed, the user program need merely specify the function to be performed on the record. The functions are: inquire, update, add (new record), and delete. Accessing of records is performed as follows: The record key is first converted to numeric form (by encoding all alphas). The numeric form is then hash coded, that is, it is converted into a disk block address within an index to the data file records. An entry in the index consists of the entire record key and a pointer to the associated data record. The index block is searched for a key matching the one specified in the function. When it is found, the associated data record is then read. (No duplicate record keys are permitted.)

Statistical and empirical analysis shows that this technique results in an average of 2.2 disk accesses to get to a data record. This assumes a key field of 15 characters maximum length and an index with at least 10% of the available key space unused (90% load factor). This is possible because of the large number of keys that can be fit into a single index block (at least 30), thus keeping the index overflow (to the next block) very low. It is also possible to specify that the file index be kept on a separate device from the data records, allowing for faster record retrieval.

A utility program is provided in addition to the file access method proper. This includes a file builder which organizes the data and index files from raw input data, and a file reorganizer for incorporating added records in order (for direct sequential access) and physically removing records which have been flagged for deletion.

Decimal Arithmetic

RSTS normally has two modes for dealing with numbers internally—integer and floating point. Integer counting is precise, but is limited to a maximum value of ± 32,767. Floating point numbers are stored in mantissa and exponent fashion, thus they have the capacity to hold very large or very small numbers. Seven decimal digits of accuracy are achieved with a 2-word floating point format; 17 decimal digits with the 4-word format. This representation, however, is usually only a close binary approximation of a decimal number. Moreover, over many repeated operations, cumulative errors can lead to a loss of accuracy.

The decimal arithmetic option replaces the standard floating point arithmetic with 4-word fixed-point arithmetic. This format achieves eighteen places of accuracy with twelve places to the left of the decimal point and six places to the right. As all numbers thus represented, including fractions, are true decimal numbers, there can be no cumulative error due to repeated operations. For this reason, this representation is normally preferred when performing any accounting functions.

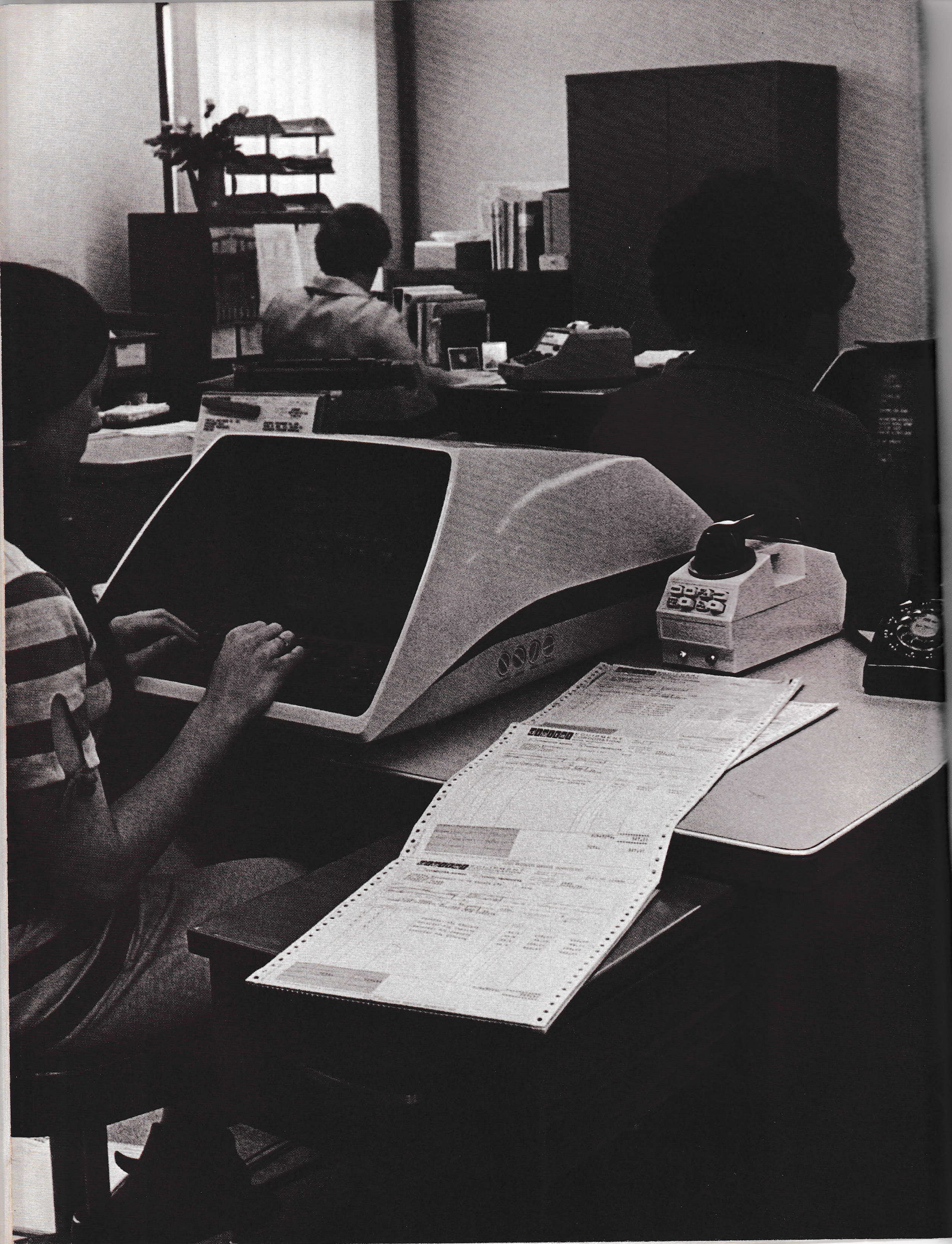
The decimal arithmetic option, if elected, is included at system generation. From that point on, any operations, variables or constants which would normally use a floating point format use the decimal format instead. The extended mathematical functions (SIN, COS, TAN, ATN, SQR, LOG, LOG10, ↑, and EXP) are not supported under the decimal arithmetic option.

Line Printer Spooling Package

The line printer spooling package is a series of BASIC-PLUS programs which allows the user to specify disk, DECtape, or magnetic tape files to be output to a system line printer (or any other device). To utilize the spooler, the user enters his request for output; the request is then queued and initiated when the output device becomes available. In this way possible contention conflicts for the system line printer are avoided. User programs can go on to perform other tasks and system throughput can often be increased by as much as 25%.

In addition to queuing and performing simultaneous output, the spooler also performs other services for the user, optionally, as the user directs. These include printing a two page "big letters" heading consisting of the file name, extension, and other information; printing multiple copies of the file; backing up to page boundaries on printer error (e.g. "out of paper") situations; and deleting the input file after it has been output.

Included also in the package is support for multiple line printers.



DIGITAL EQUIPMENT CORPORATION WORLDWIDE SALES AND SERVICE

MAIN OFFICE AND PLANT

146 Main Street, Maynard, Massachusetts, U.S.A. 01754 • Telephone: From Metropolitan Boston: 646-8600 • Elsewhere: (617)-897-5111
TWX: 710-347-0212 Cable: DIGITAL MAYN Telex: 94-8457

UNITED STATES

NORTHEAST

REGIONAL OFFICE:
275 Wyman Street, Waltham, Massachusetts 02154
Telephone: (617)-890-0320/0330 TWX: 710-324-6919
WALTHAM
15 Lunda Street, Waltham, Massachusetts 02154
Telephone: (617)-891-1030 TWX: 710-324-6919
CAMBRIDGE/BOSTON
899 Main Street, Cambridge, Massachusetts 02139
Telephone: (617)-491-6130 TWX: 710-320-1167
ROCHESTER
130 Allens Creek Road, Rochester, New York 14618
Telephone: (716)-461-1700 TWX: 710-253-3078
SYRACUSE
5858 East Molloy Road, Rm. 142, Picard Building
Syracuse, New York 13211
Telephone: (315)-455-5987/88
CONNECTICUT
240 Pomeroy Avenue, Meriden, Connecticut 06450
Telephone: (203)-237-8441/7466 TWX: 710-461-0054

MID-ATLANTIC — SOUTHEAST

REGIONAL OFFICE:
U.S. Route 1, Princeton, New Jersey 08540
Telephone: (609)-452-2940 TWX: 510-685-2338
MANHATTAN
810 7th Ave.
New York, N.Y. 10019
Telephone: (212)-582-1300
NEW YORK
95 Cedar Lane, Englewood, New Jersey 07631
Telephone: (201)-871-4984, (212)-594-6955, (212)-736-0447
TWX: 710-991-9721
NEW JERSEY
1259 Route 46, Parsippany, New Jersey 07054
Telephone: (201)-335-3300 TWX: 710-987-8319
PRINCETON
U.S. Route 1
Princeton, New Jersey 08540
Telephone: (609) 452-2940 TWX: 510-685-2338
METUCHEN
195 Main St.
Metuchen, N.J. 08840
Telephone: (201)-549-4100/2000

MID-ATLANTIC — SOUTHEAST (cont.)

LONG ISLAND
1 Huntington Quadrangle
Suite 1SO7 Huntington Station, New York 11746
Telephone: (516)-694-4131, (212)-895-8095
PHILADELPHIA
Station Square Three, Paoli, Pennsylvania 19301
Telephone: (215)-647-4900/4410 Telex: 510-668-8395
WASHINGTON
Executive Building
6811 Kenilworth Ave., Riverdale, Maryland 20840
Telephone: (301)-779-1600/752-8797 TWX: 710-826-9662
DURHAM/CHAPEL HILL
Executive Park
3700 Chapel Hill Blvd.
Durham, North Carolina 27707
Telephone: (919)-489-3347 TWX: 510-927-0912
ORLANDO
Suite 130, 7001 Lake Ellenor Drive, Orlando, Florida 32809
Telephone: (305)-851-4450 TWX: 810-850-0180
ATLANTA
2815 Clearview Place, Suite 100,
Atlanta, Georgia 30340
Telephone: (404)-458-3133/3134/3135 TWX: 810-757-4223
KNOXVILLE
6311 Kingston Pike, Suite 21E
Knoxville, Tennessee 37919
Telephone: (615)-588-6571 TWX: 810-583-0123

CENTRAL

REGIONAL OFFICE:
1850 Frontage Road, Northbrook, Illinois 60062
Telephone: (312)-498-2500 TWX: 910-686-0655
PITTSBURGH
400 Penn. Center Boulevard
Pittsburgh, Pennsylvania 15235
Telephone: (412)-243-9404 TWX: 710-797-3657
CHICAGO
1850 Frontage Road, Northbrook, Illinois 60062
Telephone: (312)-498-2500 TWX: 910-686-0655
ANN ARBOR
230 Huron View Boulevard, Ann Arbor, Michigan 48103
Telephone: (313)-761-1150 TWX: 810-223-6053
DETROIT
23777 Greenfield Road, Suite 189
Southfield, Michigan 48075
Telephone: (313)-559-6565

CENTRAL (cont.)

INDIANAPOLIS
21 Beachway Drive — Suite G
Indianapolis, Indiana 46224
Telephone: (317)-243-8341 TWX: 810-341-3436
MINNEAPOLIS
Suite 111, 8030 Cedar Avenue South,
Minneapolis, Minnesota 55420
Telephone: (612)-854-6562-3-4-5 TWX: 910-576-2818
CLEVELAND
25000 Euclid Ave.
Euclid, Ohio 44117
Telephone: (216)-946-8484 TWX: 810-427-2608
KANSAS CITY
532 East 42nd St., Independence, Missouri 64055
Telephone: (816)-461-3440 TWX: 816-461-3100
ST. LOUIS
Suite 110, 115 Progress Parkway, Maryland Heights,
Missouri 63043
Telephone: (314)-878-4310 TWX: 910-764-0831
DAYTON
3101 Kettering Boulevard, Dayton, Ohio 45439
Telephone: (513)-294-3323 TWX: 810-459-1676
MILWAUKEE
8531 W. Capitol Drive, Milwaukee, Wisconsin 53222
Telephone: (414)-463-9110 TWX: 910-262-1199
DALLAS
8855 North Stemmons Freeway, Dallas, Texas 75247
Telephone: (214)-638-4880 TWX: 910-861-4000
HOUSTON
3417 Milam Street, Suite A, Houston, Texas 77002
Telephone: (713)-524-2961 TWX: 910-881-1651
NEW ORLEANS
3100 Ridgelake Drive, Suite 108
Metairie, Louisiana 70002
Telephone: (504)-837-0257
ROCKFORD
500 South Wyma St.
Rockford, Illinois 61101
Telephone: (815)-965-5557
TULSA
3140 S. Winston
Winston Sq. Bldg.
Suite 4
Tulsa, Oklahoma 74135
Telephone: (918)-749-4476

WEST

REGIONAL OFFICE:
310 Soquel Way, Sunnyvale, California 94086
Telephone: (408)-735-9200
SANTA ANA
2110 S. Anne St.
Santa Ana, Calif. 92704
Telephone: (714)-979-2460
F.S. 714-979-2464 TWX 910-391-1189
WEST LOS ANGELES
1510 Cotner Avenue, Los Angeles, California 90025
Telephone: (213)-479-3791/4318 TWX: 910-342-6999
SAN DIEGO
6154 Mission Gorge Road, Suite 110
San Diego, California 92120
Telephone: (714)-280-7880, 7970 TWX: 910-335-1230
SAN FRANCISCO
1400 Terra Bella, Mountain View, California 94040
Telephone: (415)-964-6200 TWX: 910-373-1266
OAKLAND
7850 Edgewater Drive, Oakland, California 94621
Telephone: (415) 635-5453/7830 TWX: 910-366-7238
ALBUQUERQUE
6303 Indian School Road, N.E., Albuquerque, N.M. 87110
Telephone: (505)-296-5411/5428 TWX: 910-989-0614
DENVER
2305 South Colorado Boulevard, Suite #5
Denver, Colorado 80222
Telephone: (303)-757-3332/758-1656/758-1659
TWX: 910-931-2650
SEATTLE
1521 130th N.E., Bellevue, Washington 98005
Telephone: (206)-454-4058/455-5404 TWX: 910-443-2306
SALT LAKE CITY
431 South 3rd East, Salt Lake City, Utah 84111
Telephone: (801)-328-9838 TWX: 910-925-5834
PHOENIX
4358 East Broadway Road, Phoenix, Arizona 85040
Telephone: (602)-268-3488 TWX: 910-950-4691
PORTLAND
Suite 168
5319 S.W. Westgate Drive, Portland, Oregon 97221
Telephone: (503)-297-3761/3765

INTERNATIONAL

EUROPEAN HEADQUARTERS

Digital Equipment Corporation International Europe
81 Route de l'Aire
1211 Geneva 26, Switzerland
Telephone: 42 79 50 Telex: 22 683

FRANCE

Equipment Digital S.A.R.L.
PARIS
327 Rue de Charenton, 75 Paris 12 EME, France
Telephone: 344-76-07 Telex: 21339
GRENOBLE
10 rue Auguste Ravier, F-38 Grenoble, France
Telephone: (76) 87 56 01/02 Telex: 32 882 F (Code 212)

GERMAN FEDERAL REPUBLIC

Digital Equipment GmbH
MUNICH
8 Muenchen 13, Wallensteinplatz 2
Telephone: 0811-35031 Telex: 524-226
COLOGNE
5 Koein 41, Aachener Strasse 311
Telephone: 0221-40 44 95 Telex: 888-2269
Telegram: Flip Chip Koeln
FRANKFURT
6078 Neu-Isenburg 2
Am Forsthaus Gravenbruch 5-7
Telephone: 06102-5526 Telex: 41-76-82
HANNOVER
3 Hannover, Podbielskistrasse 102
Telephone: 0511-69-70-95 Telex: 922-952
STUTTGART
D-7301 Kemnat
Marco-Polo-Strasse 1
Telephone: (0711) 25 40 07 Telex: 725-3404

AUSTRIA

Digital Equipment Corporation Ges.m.b.H.
VIENNA
Mariahilferstrasse 136, 1150 Vienna 15, Austria
Telephone: 85 51 86

UNITED KINGDOM

Digital Equipment Co., Ltd.
U.K. HEADQUARTERS
Arkwright Road, Reading, Berks.
Telephone: 0734-583555 Telex: 84327
READING
The Evening Post Building, Tessa Road
Reading, Berks.
Fountain House
Butts Centre
Reading, RG1 7QN
Telephone: Reading 583555
Telex: 84328
BIRMINGHAM
29/31, Birmingham Road, Sutton Coldfield, Warwicks.
Telephone: (0044) 21-355 5501 Telex: 337 060
MANCHESTER
6 Upper Precinct, Worsley, Manchester M28 AZ
Telephone: 061-790-8411 Telex: 668666
LONDON
Bilton House, Uxbridge Road, Ealing, London W.5.
Telephone: 01-579-2334 Telex: 22371
EDINBURGH
Shiel House, Craigshill, Livingston,
West Lothian, Scotland
Telephone: 32705 Telex: 727113

NETHERLANDS

Digital Equipment N.V.
THE HAGUE
Sir Winston Churchilliaan 370
Rijswijk/The Hague, Netherlands
Telephone: 070-995-160 Telex: 32533

BELGIUM

Digital Equipment N.V./S.A.
BRUSSELS
108 Rue D'Arlon
1040 Brussels, Belgium
Telephone: 02-139256 Telex: 25297

SWEDEN

Digital Equipment AB
STOCKHOLM
Englundavagen 7, 171 41 Solna, Sweden
Telephone: 98 13 90 Telex: 170 50
Cable: Digital Stockholm

NORWAY

Digital Equipment Corp. A/S
OSLO
Trondheimsveien 47
Oslo 5, Norway
Telephone: 02/68 34 40 Telex: 19079 DEC N

DENMARK

Digital Equipment Aktiebolag
COPENHAGEN
Hellerupveg 66
2900 Hellerup, Denmark

FINLAND

Digital Equipment AB
HELSINKI
Tiitismaantie 6
SF-00710 Helsinki 71
Telephone: (090) 370133
Cable: Digital Helsinki

SWITZERLAND

Digital Equipment Corporation S.A.
GENEVA
81 Route de l'Aire
1211 Geneva 26, Switzerland
Telephone: 42 79 50 Telex: 22 683
ZURICH
Scheuchzerstrasse 21
CH-8006 Zurich, Switzerland
Telephone: 01/60 35 66 Telex: 56059

ITALY

Digital Equipment S.p.A.
MILAN
Corso Garibaldi 49, 20121 Milano, Italy
Telephone: 872 748 694 394 Telex: 33615

SPAIN

Digital Equipment Corporation Ltd.
MADRID
Ataio Ingenieros S.A., Enrigue Larreta 12, Madrid 16
Telephone: 215 35 43 Telex: 27249
BARCELONA
Ataio Ingenieros S.A., Ganduxer 76, Barcelona 6
Telephone: 221 44 66

CANADA

Digital Equipment of Canada, Ltd.
CANADIAN HEADQUARTERS
P.O. Box 11500
Ottawa, Ontario, Canada
K2H 8K8
Telephone: (613)-592-5111 TWX: 610-562-8732

CANADA (cont.)

OTTAWA
P.O. Box 11250
Ottawa, Ontario
K2H 7T9
Telephone: (613)-592-5120 TWX: 610-562-8907
TORONTO
230 Lakeshore Road East, Port Credit, Ontario
Telephone: (416)-274-1241 TWX: 610-492-4306
MONTREAL
9675 Cote de Liesse Road
Dorval, Quebec, Canada 760
Telephone: (514)-636-9393 TWX: 610-422-4124
CALGARY/Edmonton
Suite 140, 6940 Fisher Road S.E.
Calgary, Alberta, Canada
Telephone: (403)-435-4881 TWX: 403-255-7408
VANCOUVER
2210 West 12th Avenue
Vancouver 9, British Columbia, Canada
Telephone: (604)-736-5616 TWX: 610-929-2006

GENERAL INTERNATIONAL SALES

REGIONAL OFFICE
146 Main Street, Maynard Massachusetts 01754
Telephone: (617)-897-5111
From Metropolitan Boston 646-8600
TWX: 710-347-0217/0212
Cable: DIGITAL MAYN
Telex: 94-8457

AUSTRALIA

Digital Equipment Australia Pty. Ltd.
SYDNEY
P.O. Box 491, Crows Nest
N.S.W. Australia 3065
Telephone: 439-2566 Telex: AA20740
Cable: Digital, Sydney
MELBOURNE
60 Park Street, South Melbourne, Victoria, 3205
Telephone: 699-2888 Telex: AA40616
PERTH
643 Murray Street
West Perth, Western Australia 6005
Telephone: 21-4993 Telex: AA92140
BRISBANE
139 Merivale Street, South Brisbane
Queensland, Australia 4101
Telephone: 44-4047 Telex: AA40616
ADELAIDE
6 Montrose Avenue
Norwood, South Australia 5067
Telephone: 63-1339 Telex: AA82825

NEW ZEALAND

Digital Equipment Corporation Ltd.
AUCKLAND
Hilton House, 430 Queen Street, Box 2471
Auckland, New Zealand
Telephone: 75533

JAPAN

Digital Equipment Corporation International
TOKYO
Kowa Building No. 17, Second Floor
2-7 Nishi-Azabu 1-Chome
Minato-Ku, Tokyo, Japan
Telephone: 404-5894/6 Telex: TK-6428

JAPAN (cont.)

Rikei Trading Co., Ltd. (sales only)
Kozato-Kaikan Bldg.
No. 18-14, Nishishimbashi 1-chome
Minato-Ku, Tokyo, Japan
Telephone: 5915246 Telex: 781-4208

PUERTO RICO

Digital Equipment Corporation de Puerto Rico
MIRAMAR
American Airlines Bldg.
804 Ponce De Leon, Miramar, Puerto Rico
Telephone: 809-723-8068/67 Telex: 385-9056

ARGENTINA

BUENOS AIRES
Coasin S.A.
Virrey del Pino 4071, Buenos Aires
Telephone: 52-3185 Telex: 012-2284

BRAZIL

RIO DE JANEIRO — GB.
Ambriex S.A.
Rua Ceará, 104, 2.° e 3.° andares
Fones: 221-4560/44, 252-9873
Cable: RAIOCARDIO
SÃO PAULO — SP
Ambriex S.A.
Rua Tupi, 535
Fones: 51-0912; 52-0655; 52-7806
Cable: RAIOCARDIO
PORTO ALEGRE — RS
Ambriex S.A.
Rua Cel. Vicente, 421, 1.° andar
Fones: 24-7411; 24-7696

CHILE

SANTIAGO
Coasin Chile Ltda. (sales only)
Casilla 14588, Correo 15, Santiago
Telephone: 396713 Cable: COACHIL

INDIA

BOMBAY
Hinditron Computers Pvt. Ltd.
69/A, L. Jagmohandas Marg.
Bombay-6 (W.B.), India
Telephone: 38-1615; 36-5344 Telex: 011-2594 Plenty
Cable: Tekhind

MEXICO

MEXICO CITY
Mexitek, S.A.
Eugenia 408 Deptos. 1
Apdo. Postal 12-1012
Mexico 12, D.F.
Telephone: (905) 536-09-10

PHILIPPINES

MANILA
Stanford Computer Corporation
P.O. Box 1608
416 Dasmarinas St., Manila
Telephone: 49-68-96 Telex: 742-0352

VENEZUELA

CARACAS
Coasin, C.A.
Apartado 50939
Sabana Grande No. 1, Caracas 105
Telephone: 72-8662; 72-9637
Cable: INSTRUVEN

RETURN THIS CARD FOR MORE INFORMATION ON

RSTS/E

TO ALLOW US TO SERVE YOU BEST, PLEASE INDICATE YOUR INTERESTS BELOW AND ANSWER ALL QUESTIONS.

☐ Send a BASIC-PLUS Language Manual	$7.50
☐ Send a RSTS/E Users Guide	$7.00
☐ Send a RSTS/E System Managers Guide	$7.00
☐ Have a Salesman call	

(PAYMENT *MUST* BE ENCLOSED)

Type of Company or Institution?

☐ Consulting Engineer or Architect Firm
☐ Manufacturer
☐ Distributor or Warehouse
☐ Retailer
☐ Secondary School
☐ Junior or Community College
☐ College or Univ. Computer Center
☐ College or Univ. Department

Type of Application (Please describe)? ____________________

____

____

____

Current Computer Usage?
Manufacturer and Model ____________________

____

☐ On-site ☐ Lease or Rent
☐ Own ☐ Timesharing
☐ Remote Job Entry ☐ Service Bureau

Approx. Annual Budget $____________________

____
NAME

____
TITLE

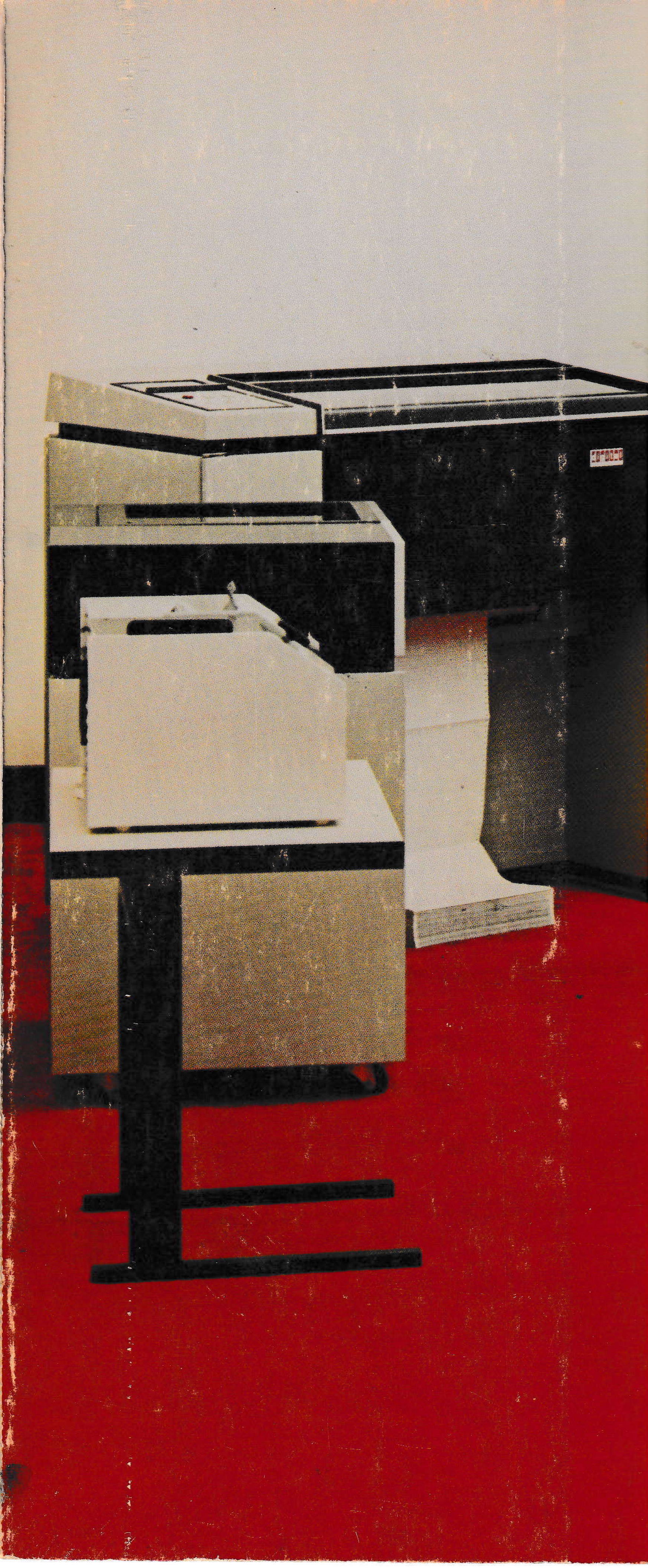
____
DEPARTMENT

____
COMPANY OR SCHOOL

____
STREET ADDRESS

____
CITY STATE ZIP

____
TELEPHONE: AREA CODE NUMBER



STAPLE HERE

FOLD HERE

First Class
Permit #33
Maynard, Mass.

Business Reply Mail. No postage necessary if mailed in United States

Postage will be paid by:

Digital Equipment Corporation
146 Main Street
Maynard, Mass. 01754

ATTN: PDP-11/45 Marketing 5-2

PRINTED IN U.S.A.