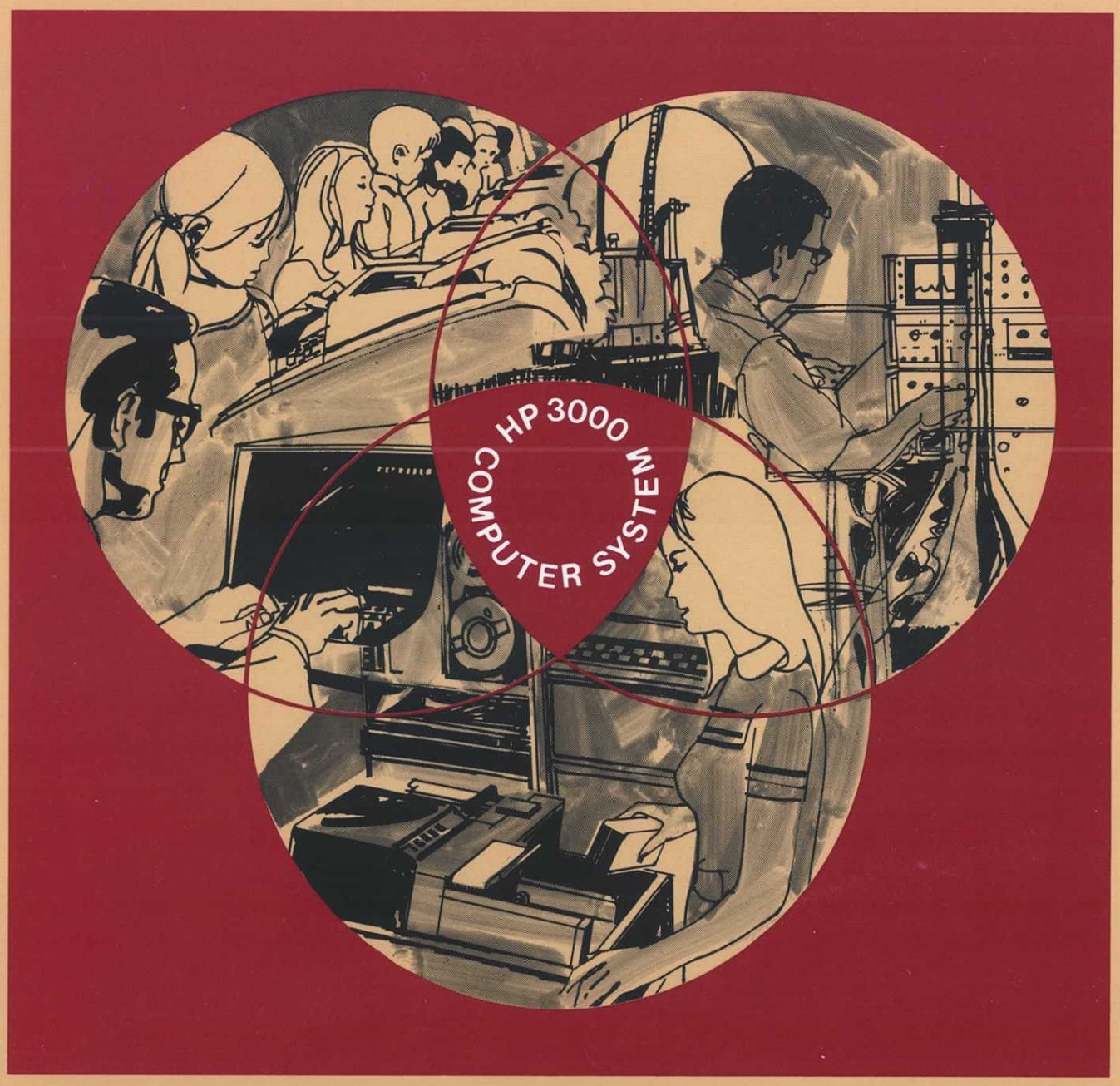
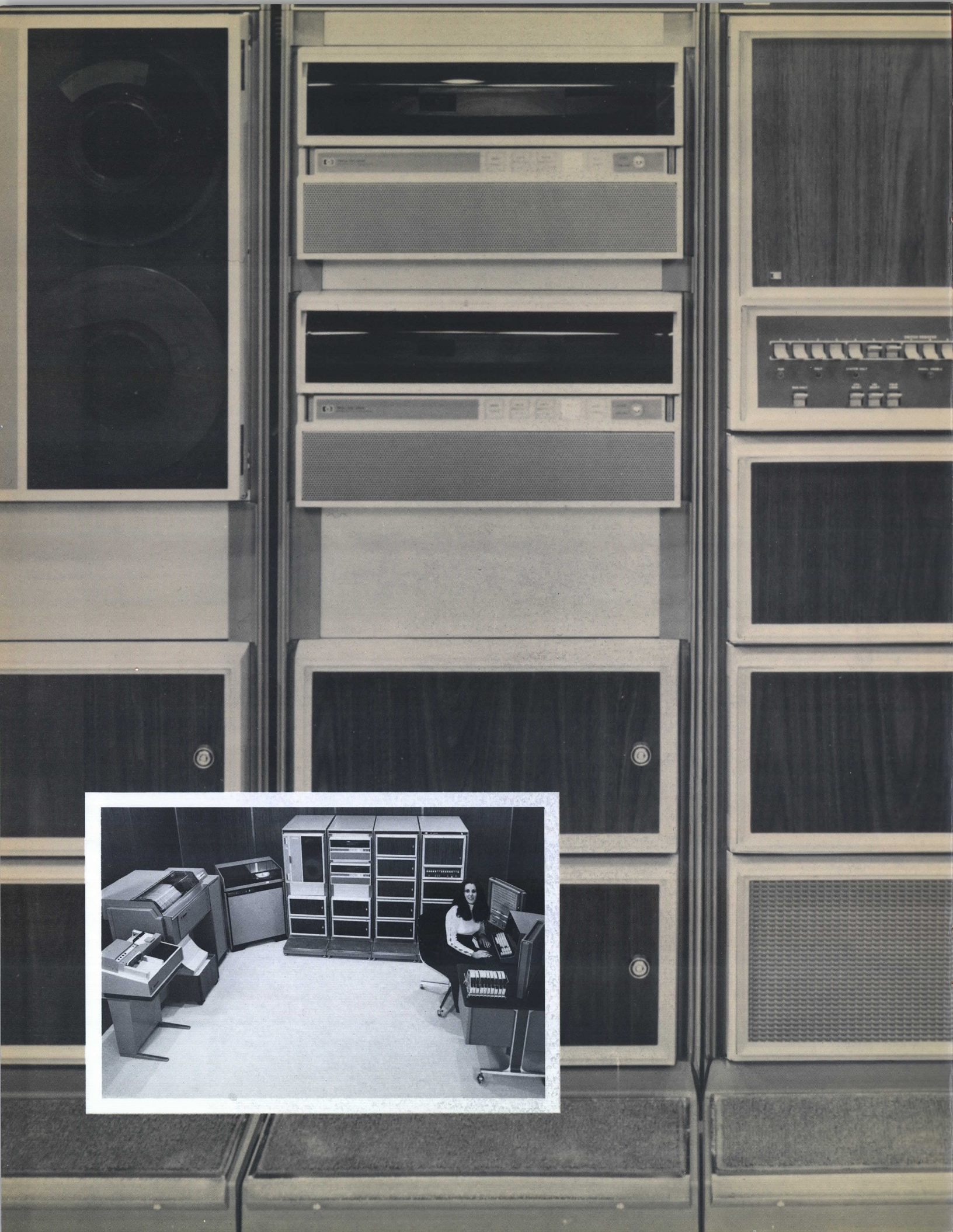
HP 3000 COMPUTER SYSTEM

summary description

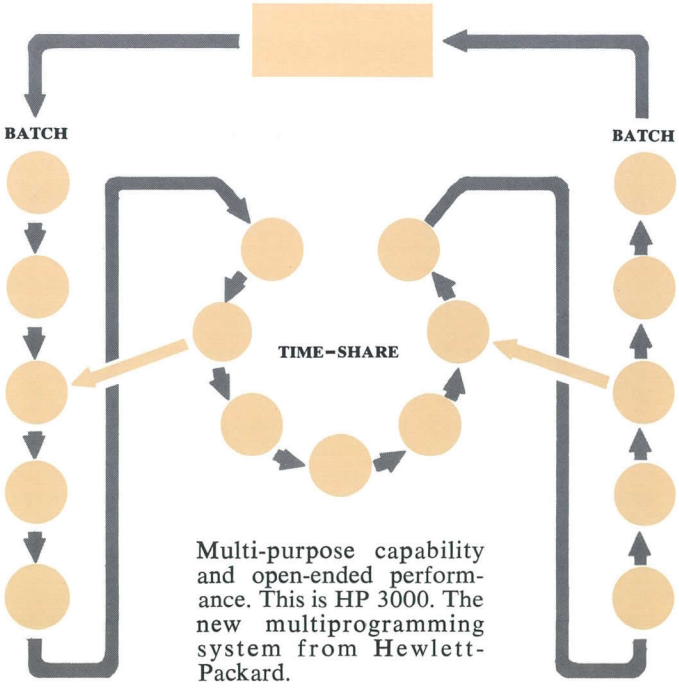


HEWLETT hp PACKARD



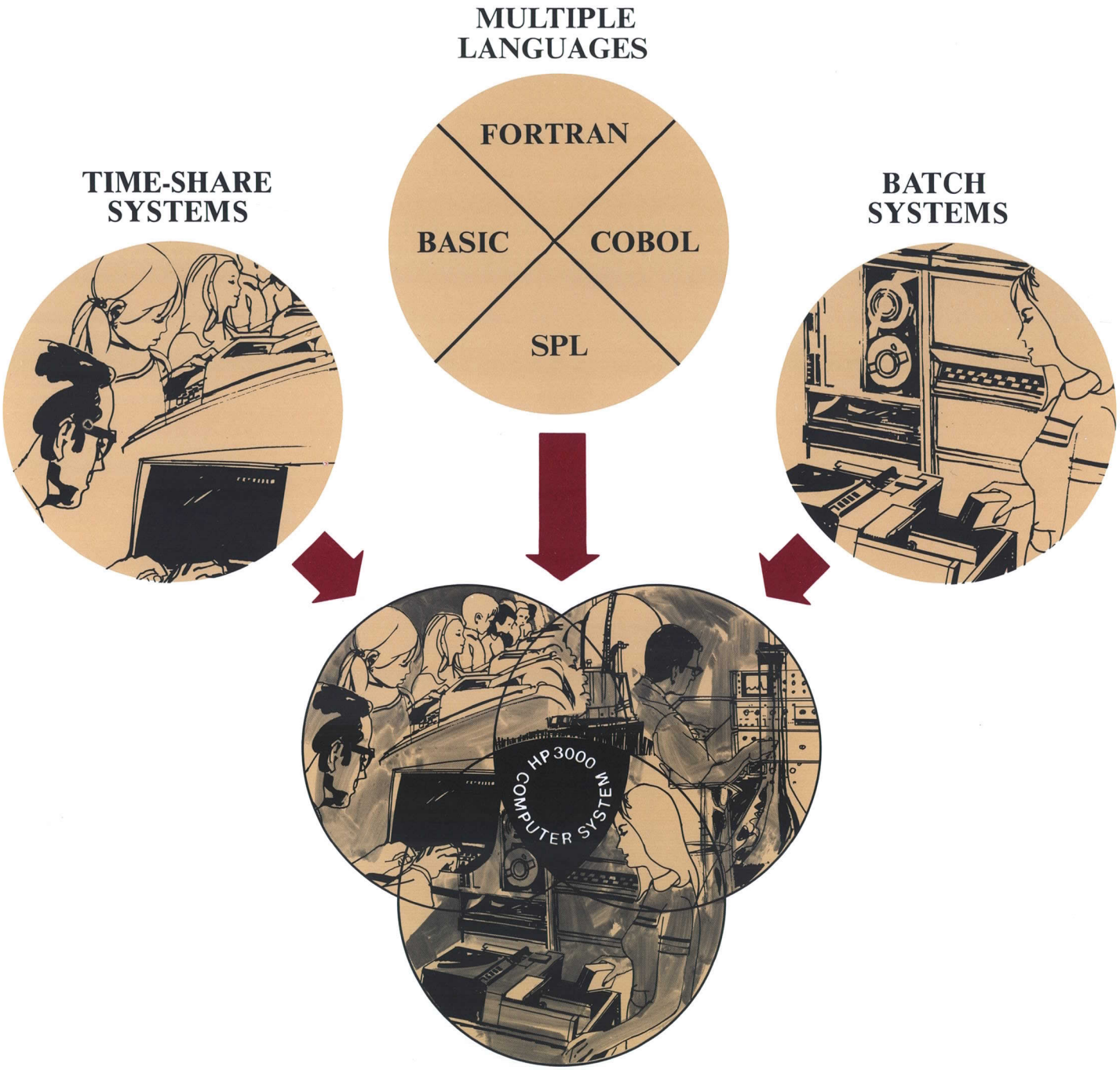
HP 3000
the multi-purpose computer system

The HP 3000 represents a new approach to multi-purpose computer systems. It combines unprecedented capabilities into one small-scale system. Sophisticated operating software opens a new level for solutions of your tasks through multiprogramming and multilingual capabilities. Terminal-oriented processing can be run concurrently in different languages. Multiple users can be accommodated in both multilingual time-sharing and data entry/retrieval modes. For day-to-day data processing, multiprogramming batch (multiple concurrent batch programming) is yet another capability. Additionally, multilingual time-sharing and multiprogramming batch activities can take place concurrently. The specific combination of capabilities is up to you. The flexibility of the HP 3000 hardware/software lets you tailor the system to your needs, whether in educational, scientific, industrial or commercial applications.



Multi-purpose capability and open-ended performance. This is HP 3000. The new multiprogramming system from Hewlett-Packard.

A NEW CONCEPT
in multi-programming systems



HP MULTI-PROGRAMMING PUTS IT ALL TOGETHER

HEWLETT-PACKARD is an industry leader in the design and manufacture of general-purpose computer systems. These systems are being used to solve a diverse range of everyday problems in science, engineering, industry, and education. In fact, HP is one of the largest suppliers of dedicated time-share systems.

This experience has shown that many computer users often require the use of more than one computational mode, including the following applications:

TIME-SHARED ACTIVITIES
Time-sharing • Data Entry • Data Retrieval • Data Base Management

BATCH PROCESSING
Scientific Programs • Industrial Data Processing • Educational Programs • Commercial Data Processing

In addition, most system and user needs change or expand rapidly over a period of time. This results in a need for multi-lingual programming and ease of hardware expansion.

The new HP 3000 combines all these characteristics into a powerful, low-cost multiprogramming system. It represents a total hardware/software solution to the need for a machine that can simultaneously handle a variety of tasks. These capabilities are implemented in a manner that provides all the features necessary for an efficient multiprogramming system. Processor, peripherals and software design permit many different users and functions to share system resources simultaneously.

Central processor hardware allows quick changes of environment with a maximum of protection. Features such as re-entrant code, relative addressing, memory protection and variable-length code segmentation (virtual memory) are standard. All are directed toward increasing multiprogramming efficiency. The HP 3000 optimizes multiprogrammed operations through MPE/3000 a powerful operating system. System resources such as storage, processor time and peripherals are efficiently allocated to each user.

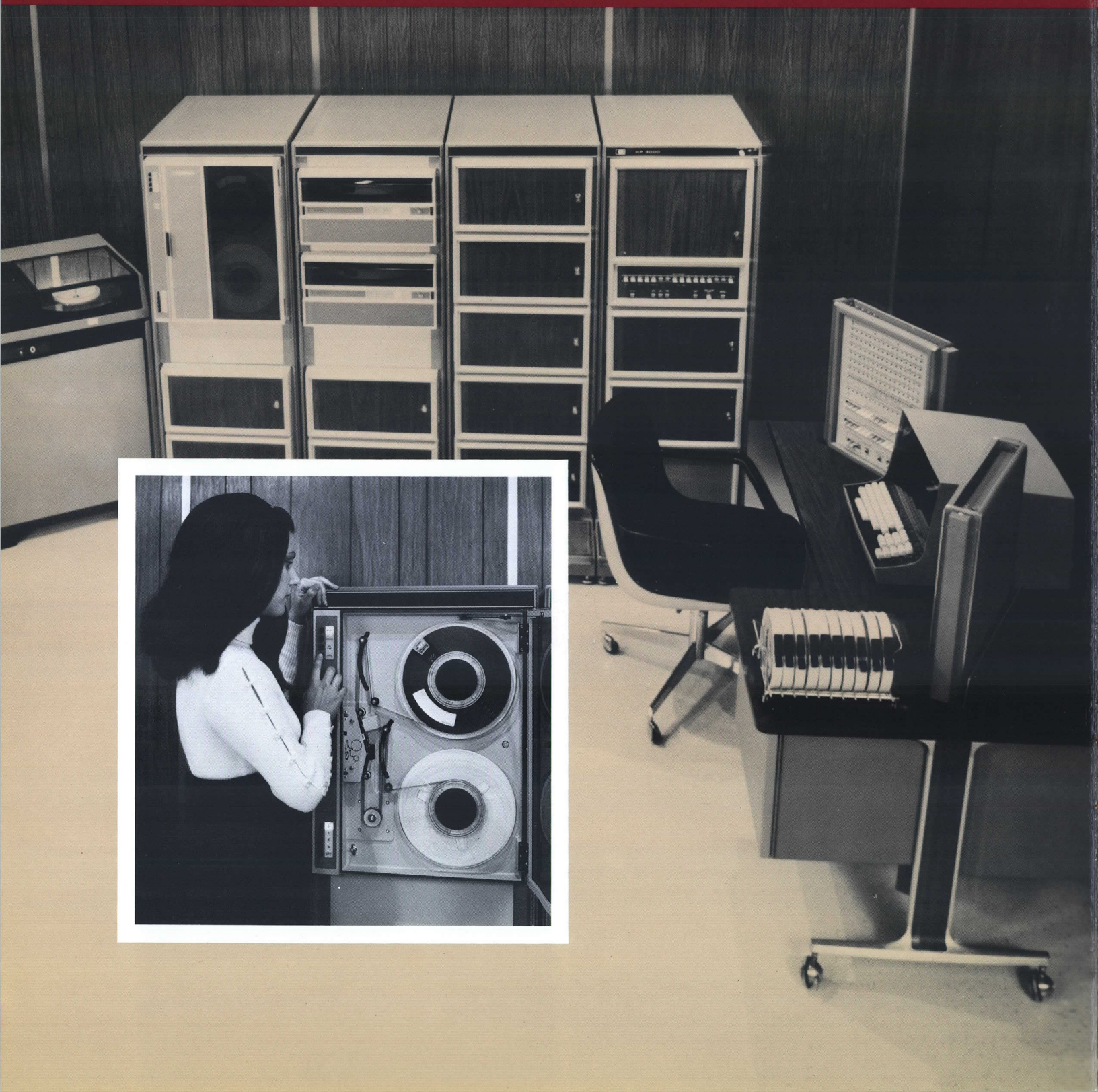
MPE/3000 is a general purpose, disc-based software system that supervises the processing of user programs submitted to the computer system. MPE relieves the user from many program control, input/output, and other housekeeping responsibilities by monitoring and controlling the input, compilation, run preparation, loading, execution and output of user programs. MPE also controls the order in which programs are executed, and allocates the hardware and software resources they require.

Hardware is organized on a truly modular basis. Communication between modules occurs over a high-speed, synchronous demand-response data bus. The input/output system is designed to provide maximum flexibility in communicating with peripheral devices. The modularity of HP 3000 permits incremental system expansion.

Powerful and modular hardware, concurrent operating modes and multi-lingual capability. These features make the HP 3000 the new standard of performance in low-cost computer systems.

HP 3000 FEATURES

for low-cost problem solving



GENERAL

- Multi-purpose computer performance for multi-terminal, and general-purpose tasks.
- Central processor design optimizes multiprogramming operations.
- Concurrent I/O and CPU operations for maximum throughput.
- Multi-lingual capability.
- Built-in protection mechanisms for programs and files.
- Modular hardware and software design for open-ended system expansion.
- Medium-scale system performance and capabilities at a small-scale system price.
- Proven reliability of Hewlett-Packard computers.

HARDWARE

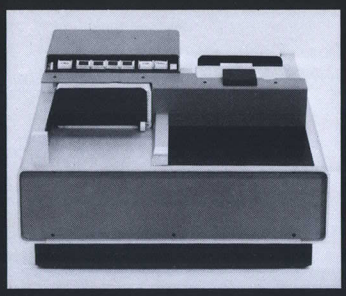
- Stack architecture for ease of compilation and fast execution.
- Microprocessing implemented via read-only-memories.
- 175 nanosecond microinstruction time.
- High-speed central data bus for modular expansion and technology-independent memories.
- 170 powerful instructions including floating point instructions, single, double and triple word shift instructions, bit test, byte string, loop control, move and list search instructions.
- Dynamic relocatability of user programs.
- Re-entrant code and stack architecture permit minimum memory overhead and recursive subroutine capability.
- Variable length code segmentation for virtual memory operation.
- Core memory expandable from 64 Kbytes to 128 Kbytes.
- Rapid context switching optimizes system resources.
- Built-in features include hardware floating point arithmetic, automatic memory protection, parity checking and power-fail auto restart.
- All instructions 8 or 16 bits in length.
- Direct memory access in I/O processor system permits device independent I/O program execution.
- Input/output capabilities provide concurrent data transfer at cumulative rates to 2.8 million bytes per second.
- Asynchronous multiplexers available for terminal access.
- 16 and 32-bit fixed-point and 32-bit floating point hardware.
- Triple word instructions aid 48-bit floating point arithmetic operations.
- Up to 253 interrupt levels with automatic hardware identification and priority control.
- Versatile storage units—Fixed head and removable disc files for on-line storage—plus 7 or 9 track magnetic tape units.
- A complete range of peripherals, including low speed and high speed terminals, card readers, card punches, line printers and paper tape equipment.

SOFTWARE

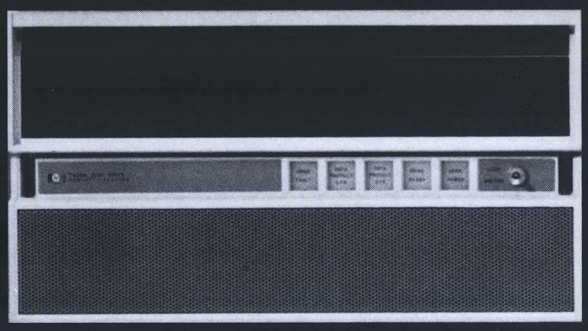
- HP's Multiprogramming Executive (MPE) executes many different user and system functions concurrently at the priority appropriate for the job.
- Large machine operating system characteristics at a small machine price.
- Absolute user protection by use of automatic hardware delimiters and software lockwords.
- Quick response with minimum software overhead.
- Extended ANSI FORTRAN IV, ANSI COBOL, BASIC, and Systems Programming Language (SPL).
- Full set of system utilities.
- Symbol trace (debug), text editor, scientific routines, statistical analysis routines, SORT.
- Hardware diagnostics for on-line or stand-alone check-out.
- Builds on HP's extensive experience in time- share and disc-operating systems.
- Complete facilities for system accounting and logging.

THE HP 3000 incorporates the low-cost and efficiency of the latest hardware and software technology into a multi-programmed computer system. With the HP 3000 you can incorporate your terminal-oriented, and batch processing needs into one convenient package. The system combines a new central processor design with a multiprogramming operating system and multi-lingual capability.

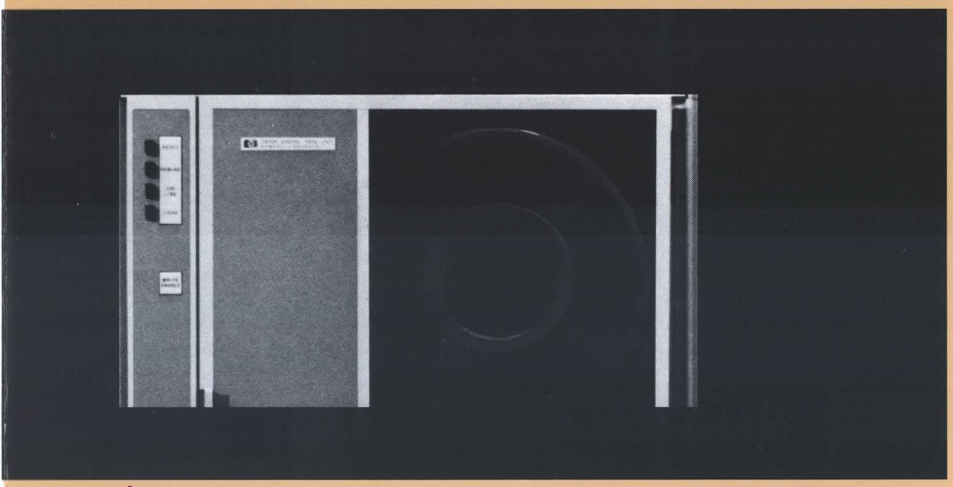
LOCAL BATCH JOBS



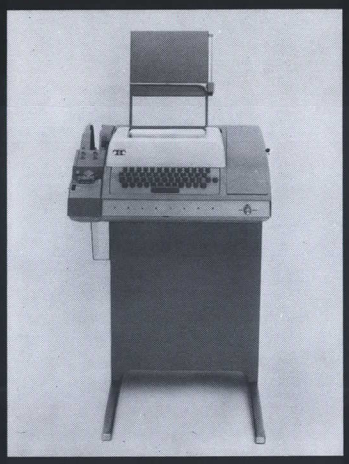
SYSTEM STORAGE

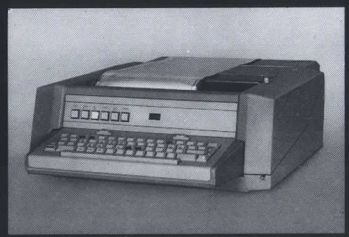


CENTRAL



USER TERMINALS





PROCESSOR

MULTIPLEXER

MULTI-PURPOSE OPERATION
to maximize your computer resources

GENERAL-PURPOSE BATCH

Consistency and compatibility are emphasized between HP 3000 operating modes. Batch processing activities and time-share terminal users access the same software. Programs developed in the time-share mode may be utilized under batch to take advantage of available system peripherals.

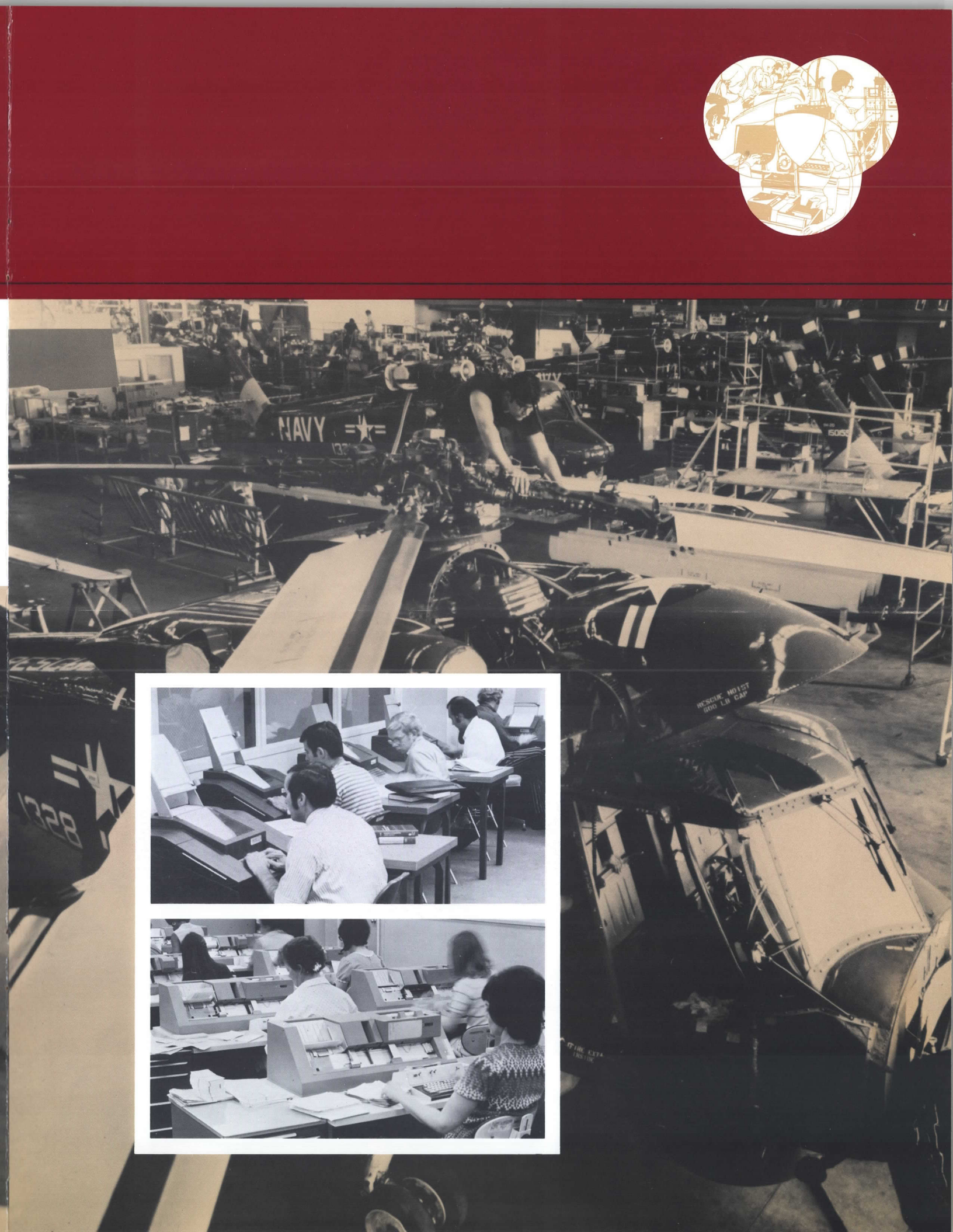
Software for general-purpose activities includes fast, efficient compilers, symbol trace (debug) and utility programs. Through efficient encoding, 170 powerful instructions have been implemented. HP 3000 hardware permits both fixed-point and floating-point arithmetic operations. These elements in combination permit shorter programs that execute at extremely high speed.

Additional features for HP 3000 batch operations include:

- Programs can be developed from terminal or batch devices.
- Drivers, new languages and libraries may be added on-line.
- Shared file domains permitting the same file to be utilized either under the time-sharing or batch mode.
- Batch jobs may be scheduled from sessions (a time-sharing terminal).
- Symbol trace (debug), utilities and scientific routines.
- Full range of low-cost, batch-oriented peripherals.

TIME-SHARED TASKS

When operated as a terminal-oriented system, the HP 3000 can be configured to handle multiple interactive users. Remote users can utilize the system for problem-solving, or data base management. Languages available to terminal users include HP extended FORTRAN IV, ANSI COBOL, HP extended BASIC and Systems Programming Language (SPL). The user may also enter batch jobs from his remote terminal. For example, a user may wish to do a FORTRAN compilation using a previously prepared source file. After the FORTRAN job has been submitted, the user may enter into a time-share session using BASIC. Status of the FORTRAN compilation can be periodically checked from within the time-share session.


NAVY
RESCUE HOIST
600 LB CAP

MULTI-PROGRAMMING EXECUTIVE
insures dynamic allocation of resources

HEWLETT-PACKARD'S Multiprogramming Executive (MPE) is designed to take maximum advantage of the HP 3000 CPU architecture for multiprogramming operations. MPE executes multiple user and system functions concurrently. Users are completely protected from each other and appear to have their own private machine.

Because MPE is a multi-purpose system, it can handle multiple modes of operation. These consist of time-shared activities, and multiprogrammed batch processing. Each user and function interfaces with MPE at a level of sophistication appropriate to the task.

Consistency and compatibility are applied throughout the MPE design. Batch and terminal users access the same software; users can be switched from one type of activity (time-share) to another (batch), on-line. Separate code and data segments permit sharing of code while maintaining the integrity of each user's results. Each user operates in a protected environment free from interference by other users. Program protection is supplied by hardware, and file security is provided by software.

MPE is modular both functionally and physically. Each system function utilizes a procedure or routine. The system can be organized on this basis because of hardware efficiency for procedure calls. This organization also permits an open-ended system that can be easily expanded and maintained. The major functional units of the system are briefly discussed below.

COMMAND INTERPRETER

System users interact with the HP 3000 through the Command Interpreter. Command Interpreter user commands are entered via the user at his terminal, or from punched cards or paper tape. Commands access the appropriate portions of MPE for file creation, file access, program loading, program execution, etc. They also access subsytems such as language processors and utilities. Certain commands for the system operator, real-time programmers and proprietary subsystems are limited to specific privileged users.

INTRINSICS

Programming access to HP 3000 hardware is made through system routines called intrinsics. The user accesses a set of callable intrinsics which are appropriate for the particular problem. Actual interaction with hardware is accomplished through user uncallable intrinsics to maintain system integrity.

FILE SYSTEM

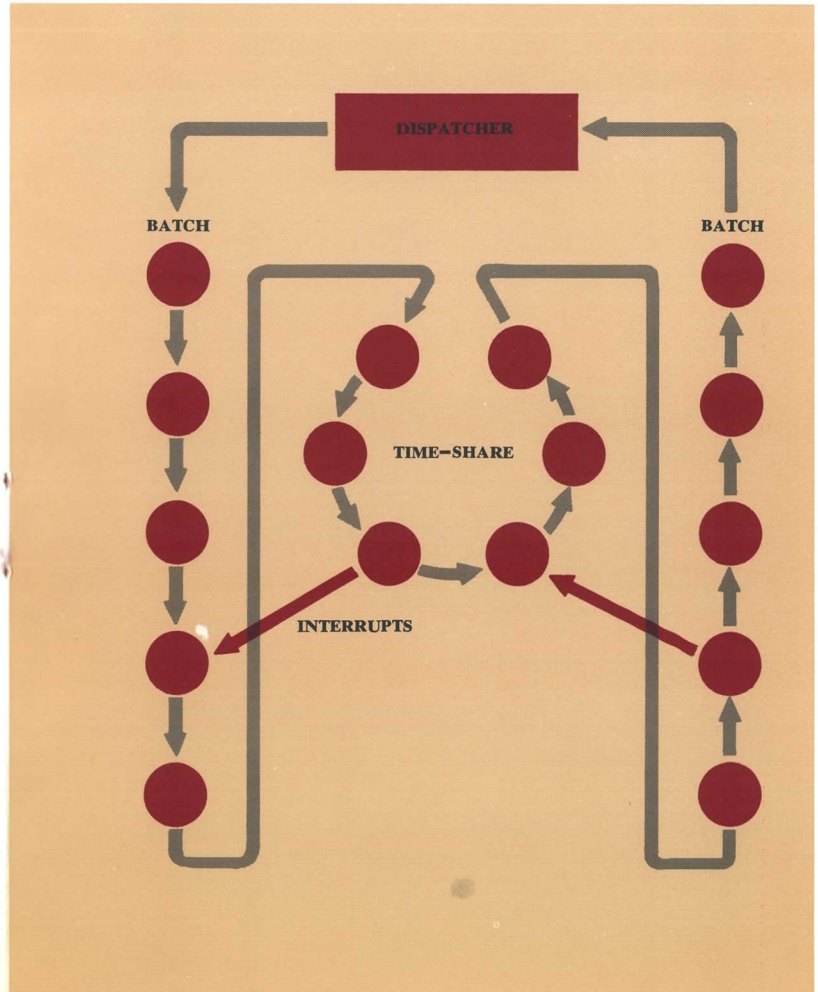
Uniform access to disc files and standard input/output devices is accomplished through the MPE File System. Files are accessed in two modes: sequential, which can have fixed or variable record lengths, or direct. Files are opened, accessed and closed programmatically. In addition, file commands allow programs to reference files without specific knowledge of their actual names or characteristics. File commands also allow file specifications to be altered at run-time. Three classes of file security are available to the user.

INPUT/OUTPUT SYSTEM

The input/output system queues, initiates, monitors and completes all I/O requests for standard devices. All I/O devices can be operated concurrently (within system bandwidth). The system is normally invisible to the user since he accesses I/O devices through the file system. Devices are specified either implicitly or explicitly. For example, when a program requests data from a particular file, it implicitly specifies a logical I/O device; the file system then specifies the explicit I/O device address.

DISPATCHER

The dispatching function allocates CPU time among programs in execution. All processes are entered into a master queue according to their priority. When execution has been interrupted (I/O, internal interrupt, time interrupt, etc.) CPU control is granted to the highest priority process ready to execute in core. The master queue is divided into five standard areas (additional areas are system manager definable). These are called sub-queues and are either linear or circular. Linear sub-queues are dispatched in a linear fashion. In a circular sub-queue all processes are at the same priority and are dispatched in "round-robin" fashion according to a time-slice. To provide maximum system throughput, the system manager can turn on and off sub-queues to limit certain types of activity at certain times.



MEMORY MANAGEMENT

Main memory is dynamically allocated on a priority basis among contending users. Several programs can be active in memory concurrently. When a higher priority program must be serviced, the executing program is interrupted or overwritten (data is saved). Programs may be relocated anywhere in main memory and continue executing from the point of interruption. Available main memory is located through list link search CPU instructions. In addition, available core memory and a portion of the system disc constitute the virtual memory area. All process code and data segments are allocated disc space in the overlay area. As processes are dispatched, at least the initial code segment and the entire data segment are allocated core space. As the procedure executes, other code segments are automatically made present when needed. Since code segments are unalterable during execution, they are simply overlayed when more memory is needed.

MULTI-PROGRAMMING HARDWARE
for built-in system performance

THE HP 3000 central processor was designed to optimize multiprogramming operations. This has been achieved by using such features as variable length code segmentation, special CPU instructions, dynamic memory allocation, built-in memory protection and in general replacing as much software overhead as possible by hardware capabilities. Some of the significant hardware features for multiprogramming operation are discussed below.

Variable Length Code Segmentation — Descriptors in the Code Segment Table provide the mechanism for virtual program memory and enable the memory management system to allocate available main memory dynamically. In addition to Code Segmentation, MPE provides software-managed Data Segmentation controlled by descriptors residing in a Data Segment Table. This provides the complementary function of a virtual memory mechanism for data segments. Thus, the operating system has complete control of segment allocation, and there is minimal "checkerboard" waste of memory resources.

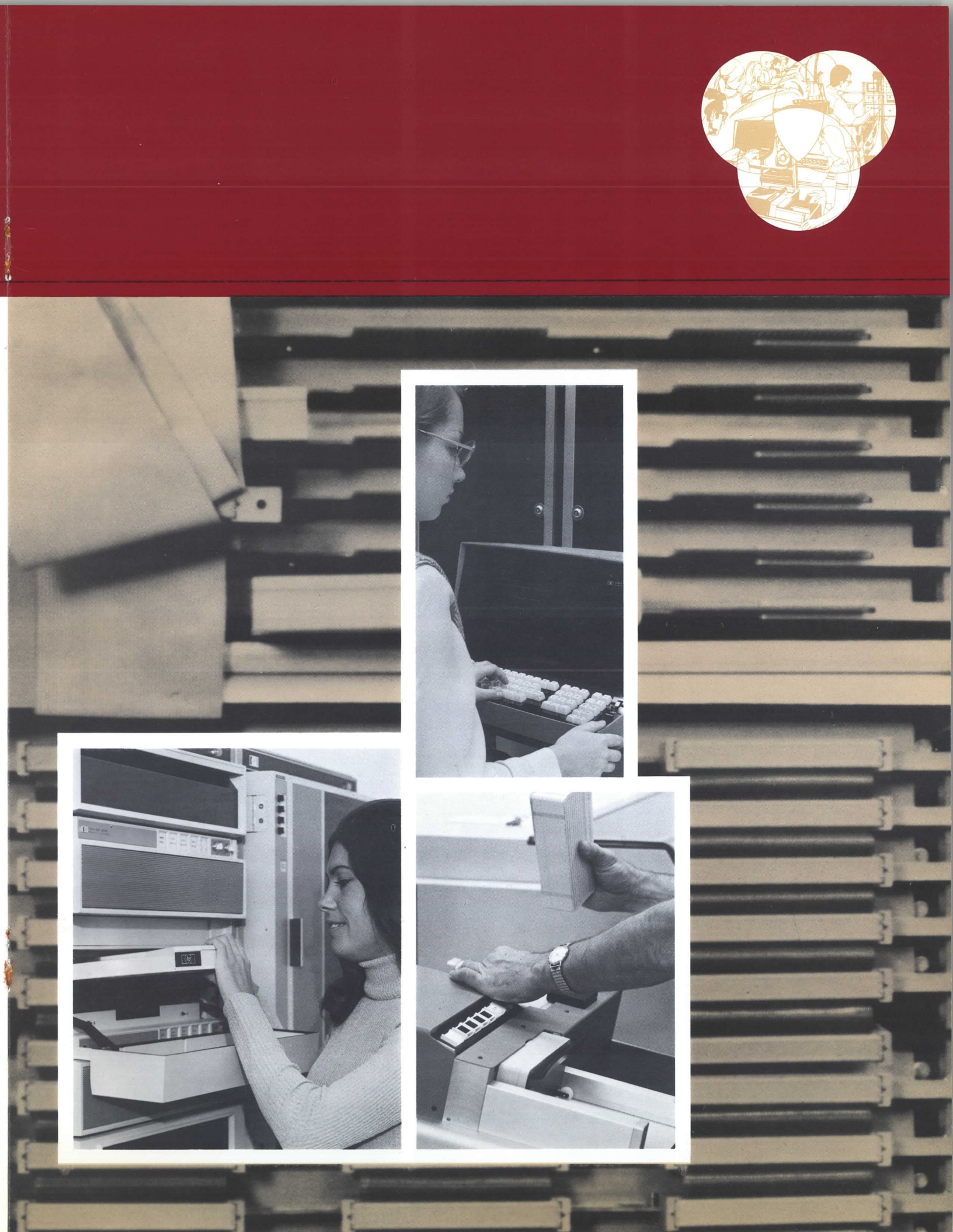
Protection Mechanisms — System hardware registers restrict non-privileged users from accessing any region of memory outside their environment. This satisfies the requirements of providing necessary protection between individual users and between users and the operating system.

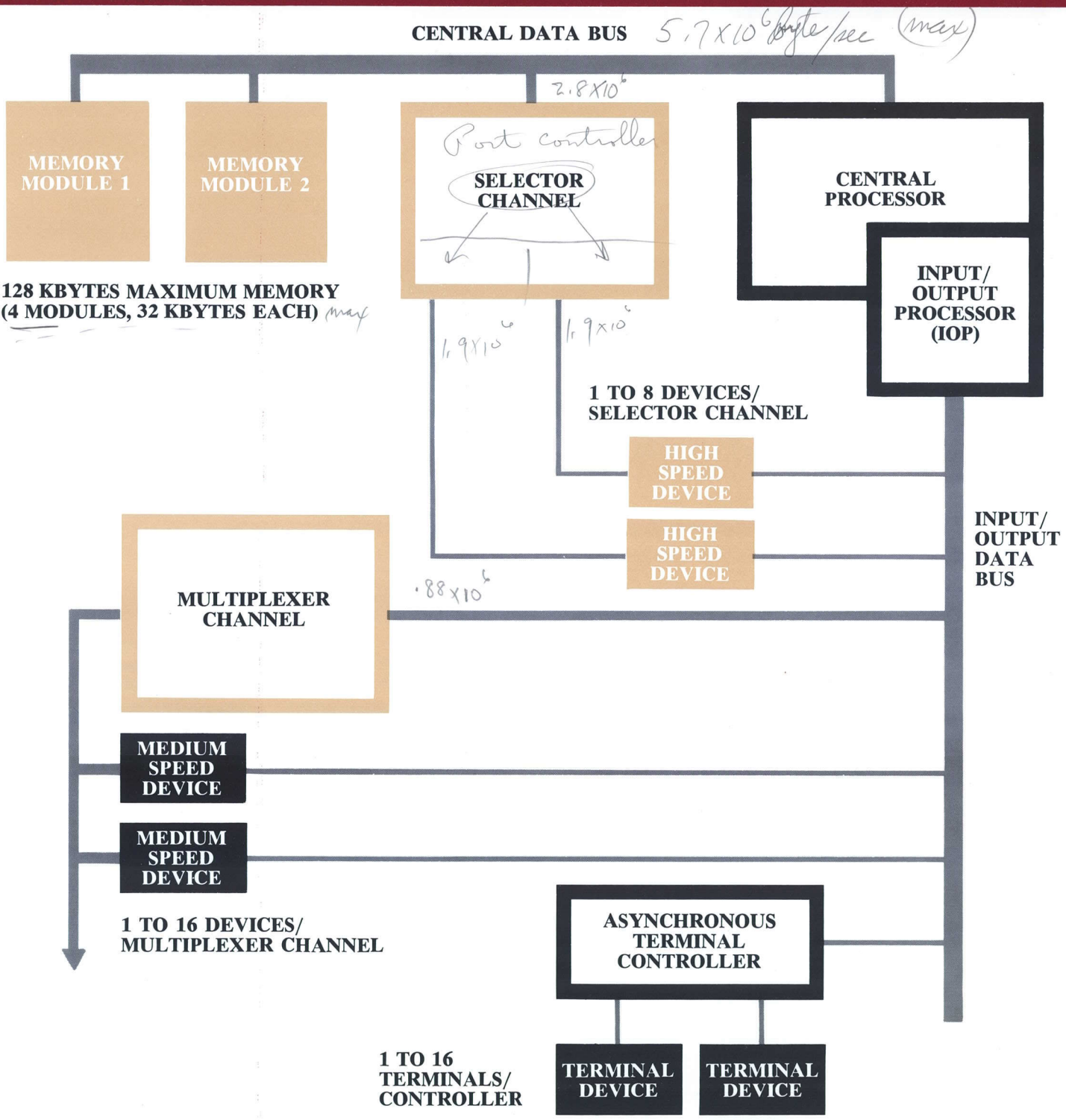
Microprogrammed Instruction Set — By placing the burden on the CPU microprocessor, many special functions in the instruction set greatly reduce the amount of code required to perform recurring system operations. MOVE and BYTE instructions assist the File System and I/O System. Hardware stack markers and registers allow saving and restoring linkages and passing parameters to and from procedures. A list link search instruction provides fast searches of a linked list. Stack-oriented and single-address instructions provide maximum flexibility in programming.

Relocatable Code and Data Segments — All addressing is relative to hardware registers resulting in simple and efficient relocatability.

Separation of Code and Data — By separating programs into separate code and data segments, a number of significant operational benefits are made possible. These include:

- Re-entrant Code — Since code segments are unmodifiable during execution, they can be entered and exited at any point. This permits code to be shared by several users at the same time, eliminating the need to provide a separate copy of an executing program for each user. Alterable data is maintained in each users separate data area.
- Procedures and Subroutines are naturally recursive.
- Since code is not modifiable, it can be overlaid rather than swapped out.
- Program constants may be stored within the code segment allowing them to be shared, thus reducing data stack size.




CENTRAL DATA BUS
MEMORY MODULE 1
MEMORY MODULE 2
128 KBYTES MAXIMUM MEMORY (4 MODULES, 32 KBYTES EACH)
SELECTOR CHANNEL
CENTRAL PROCESSOR
INPUT/ OUTPUT PROCESSOR (IOP)
1 TO 8 DEVICES/ SELECTOR CHANNEL
HIGH SPEED DEVICE
HIGH SPEED DEVICE
INPUT/ OUTPUT DATA BUS
MULTIPLEXER CHANNEL
MEDIUM SPEED DEVICE
MEDIUM SPEED DEVICE
1 TO 16 DEVICES/ MULTIPLEXER CHANNEL
ASYNCHRONOUS TERMINAL CONTROLLER
1 TO 16 TERMINALS/ CONTROLLER
TERMINAL DEVICE
TERMINAL DEVICE

POWERFUL HARDWARE
for maximum throughput

THE HP 3000 hardware contains built-in high-performance features that maximize system throughput. Characteristics such as hardware floating-point arithmetic, 170 one-word instructions, stack architecture, automatic memory protection, parity error detection and power-fail automatic restart are implemented as part of basic system design. Core memory is optimized to deliver submicrosecond cycle time. The entire logic utilizes the latest IC devices. Long-life components and refined packaging result in a system that is less costly, more reliable and easier to service.

Modular Structure—HP 3000 utilizes independent hardware modules organized around a very high-speed central data bus. This structure results in cumulative data rates between memory and central processor, input/output processor, and selector channel of up to 5.7 million bytes per second. Input/output is optimized by using a separate multiplexed asynchronous I/O Data Path. Data from the Multiplexer Channel is applied directly to the Input/Output Processor at cumulative rates up to 880 thousand bytes per second. For very high-speed I/O, a Selector Channel can also be added. Selector Channel inputs completely bypass the Input/Output Processor permitting concurrent data transfer with the Multiplexer Channel.

Central Processor—Many features that are extra cost in other computers are standard in HP 3000. Implementation of these features is made possible by the efficiency of stack architecture and the use of the latest semiconductor technology. HP 3000's powerful, microprogrammed instructions include add, subtract, multiply, and divide arithmetic operations on 16 and 32-bit fixed-point quantities. These same functions are also provided for 32-bit floating-point quantities. Additionally, many instructions have been implemented that simplify 48-bit floating-point operations.

A total of 170 specific instructions are provided—a very powerful instruction set for a computer of this class. (See pages 26 and 27 for the complete listing.)

Input/Output—Flexible and effective input/output implemented in three ways. This results in use of data paths suited for the capabilities of individual peripheral devices.

- Selector Channel: Provides for very high speed data transfer or additional I/O bandwidth. Selector channel inputs bypass the Input/Output Processor completely. This permits single device transfer rates of up to 2.8 million bytes per second. The selector channel module accepts up to four channels on a multiplexed basis. Each channel in turn interfaces up to eight devices which communicate on a one-at-a-time basis.
- Multiplexer Channel: Used for fast input/output device servicing without burdening the Central Processor. Block transfers are carried out without CPU "cycle-stealing." By multiplexing device inputs, cumulative data rates of 880 thousand bytes per second are possible. Each multiplexer handles up to 16 devices. For terminal-oriented applications additional multiplexers are added to the system.
- Direct I/O: Used for low speed asynchronous devices and single word data transfer. Data transfer occurs through the CPU registers.

Interrupt System—The HP 3000 provides for up to 253 priority interrupts, with 16 levels of interrupt masking. The microprogrammed interrupt processor automatically identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved by the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine. This microprogrammed context switching is performed in less than 18 microseconds.

Memory—System modularity permits the use of four asynchronous main memory modules. Each module has its own independent port. Two or four-way module interleaving is available to increase effective data rates in and out of main memory. Any main memory module configuration is allowed providing no single module is larger than 64 Kbytes. Total main memory capacity is 128 Kbytes (16-bit words with 17th parity bit). Core memories are available in 16-Kbyte increments. Cycle time is 960 nanoseconds.

MULTI-LINGUAL SOLUTIONS
to minimize expensive programming

SYSTEMS PROGRAMMING LANGUAGE

The choice of programming languages on most computers lies in selecting one of two mutually exclusive groups; a high-level machine-independent language such as FORTRAN, on the one hand, and a completely machine dependent assembly language on the other. The high-level language enables the overall program to be written clearly and concisely. But high-level languages do not offer the scope required for explicit optimization, interrupt handling, bit manipulation or control of peripherals. An assembly language can cope with all of these, but usually at the expense of lengthy development, obscure coding and a requirement for extensive comments. HP Systems Programming Language (SPL) is machine dependent, but offers facilities for laying out a program in a clear fashion so that it becomes almost self-documenting.

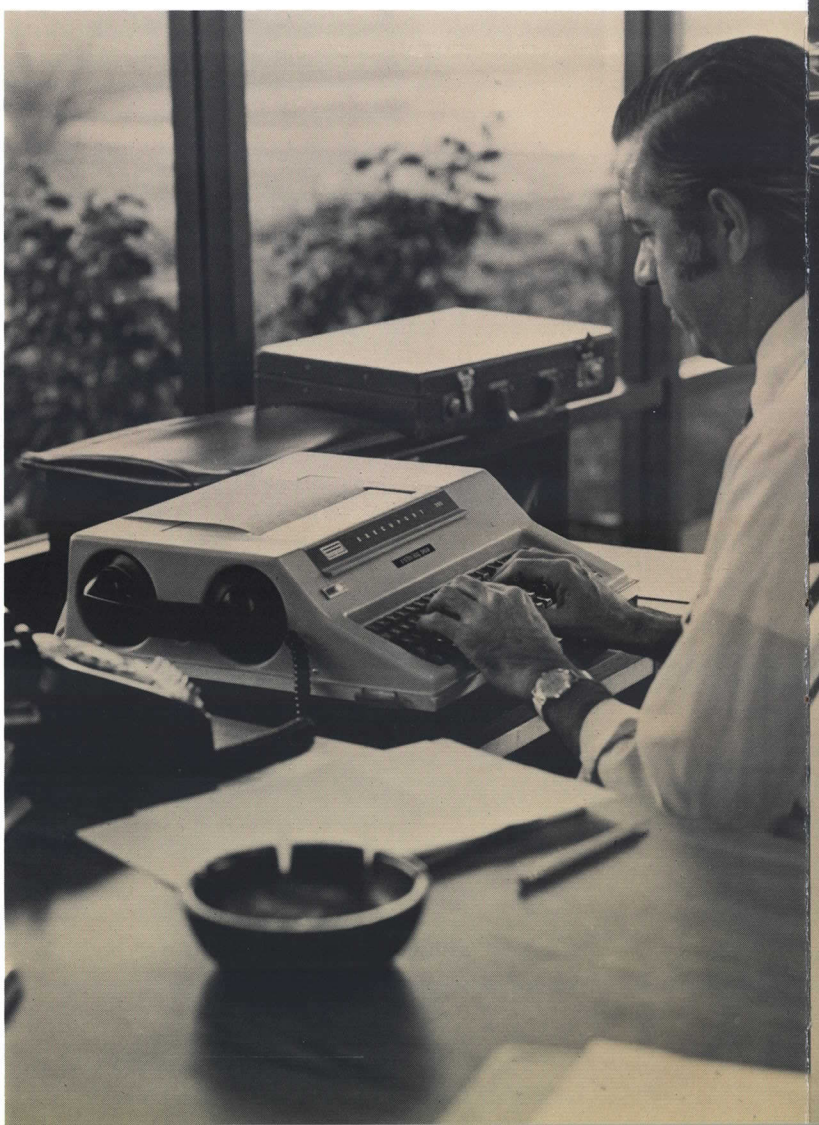
The ideal systems programming language is transparent to the programmer; that is, each line of syntax produces an expected sequence of assembly code. This results in compiled code that is as efficient as the assembly code produced by a top-rated systems programmer. This capability is found in HP's System Programming Language.

Some immediate advantages of SPL/3000 are:

- Clarity—The language enables the programmer to clearly state operations he wants performed. Finished programs are clearly read both by the writer and other users.
- Coding Efficiency—Programs such as operating system modules and application packages can be written and documented with SPL several times faster than with a standard assembly language.
- Self-Documenting—Even involved SPL programming techniques can be clearly analyzed. Code is set out in blocks that contain long and meaningful identifiers.
- Run Time Efficiency—SPL execution times are within 5% of comparable assembly language coding for a majority of applications.
- Simple and Flexible Language Design—SPL is easy to use and results in rapid compilation and the production of efficient code.

FORTRAN/3000

HP 3000 FORTRAN conforms to the American National Standards Institute (ANSI) standard for FORTRAN (x3.9—1966). Many extensions have been added by HP to make the language adaptable to a wider range of problems and programming styles. To support multi-terminal capabilities, FORTRAN/3000 has been extended to allow free-form program input from a terminal device. In addition, ACCEPT and DISPLAY statements are available for free field input/output of data from a user's program being executed at a terminal. Free-form input allows the terminal user to prepare programs without regard for the position-dependent fixed-format representation FORTRAN normally demands. Statement sequence numbers are optional.



FORTRAN/3000 supports exceptional capabilities in the areas of character string manipulation and data file I/O. A new data type "character" allows the FORTRAN/3000 user to directly manipulate strings up to 255 characters in length. Subscript notation may be used to access substrings of a character string directly. In addition, string arrays of any number of dimensions may be declared.

Up to 99 data files may be opened simultaneously during execution of a FORTRAN program. The structure of the file and method of access can be declared by the programmer or left to default values.

COBOL/3000

HP 3000 COBOL is based on the ANSI Standard COBOL* (USAS x3.23—1968) at a level upward compatible with the highest level of the Federal Government Standard. COBOL/3000 is an extremely powerful and versatile computer language. It is ideal for administrative, financial, accounting, agency, inventory, warehousing, distribution and other commercial EDP applications.

COBOL/3000 consists of a basic nucleus and functional processing modules that provide the following capabilities:

Table Handling—for defining tables of contiguous data items and accessing an item relative to its position in the table.

Sequential Access—to access records of a file in an established sequence. Sharing memory area among files is also provided.

Random Access—to access records of a mass storage file according to a programmer-supplied key. Sharing memory area among files is also provided.

Sort—to order a file of records according to a set of user-specified keys within each record. Special processing of addition, deletion, creation, altering, editing, etc., is also provided.

Segmentation—to specify object program segmentation requirements.

Library—for specifying text that is to be copied from a library. Library text is available to a source program at compile time and need not be actually written as part of the source program.

An additional functional processing module for interprogram communication provides the capability to call subprograms written in COBOL/3000 or other HP 3000 languages from COBOL/3000 programs.

Additional COBOL/3000 features include:

- Direct communication with SORT/3000 via the SORT verb.
- Compilation time editing.
- Data segmentation through DYNAMIC-type subprograms.

BASIC/3000

Hewlett-Packard is one of the world's leading manufacturer's of BASIC-oriented computer systems for instrumentation, time-sharing and batch utilization. Applications for computer assisted instruction (CAI), instrumentation, text editing, circuit analysis, and simulation have already been implemented in HP BASIC. By building on this extensive experience, HP has extended HP 3000 BASIC language capabilities even further.

BASIC/3000 operates in all multiprogramming modes. Programs or data files created in one mode can be operated upon in any other. The availability of extended precision arithmetic and access to a wide range of system peripherals permits implementation of BASIC for many new business, scientific and educational applications.

**ECMA COBOL conforms with ANSI COBOL.*

Additional information is presented in data sheets, available from all HP sales offices.

SOFTWARE TOOLS
to speed your problem solving...

SCIENTIFIC SOFTWARE

For the scientific user, a collection of powerful support programs is available. These provide many common scientific functions in professionally programmed form, thus eliminating the need for the user to create them. This software falls into two categories, the Scientific Library and Statistical Analysis Routines (STAR/3000).

SCIENTIFIC LIBRARY

The Scientific Library is a collection of procedures that perform the scientific functions required most often. These procedures can be called by user programs written in FORTRAN/3000, COBOL/3000, SPL/3000, or BASIC/3000 and provide the following:
Error and Gamma Functions • Exponential, Sine-cosine, and Fresnel Integrals • Elliptic Integrals and Functions • Bessel Functions • Elementary Statistics • One-way Frequency Distribution • Correlation • Multiple Linear Regression

STAR/3000

Statistical Analysis Routines (STAR) allow simplified access to statistical functions of the Scientific Library through a keyboard terminal or batch job.

The user need not learn a programming language to use STAR since all communication is done via commands (in batch mode) or questions and answers (keyboard). STAR permits the user to input data or use data from an integrated data base, perform statistical operations upon it, and output the results in an easily readable form. Operations provided include:

- Elementary statistics
  Mean · Standard Error of the Mean · Standard Deviation · Variance · Kurtosis · Skewness · Range · Minimum · Maximum
- One-way frequency distribution
- Product-moment correlation
- Multiple linear regression
- Transformations
  Square Root · Exponential · Natural Logarithm · Common Logarithm · Inverse · Round
- Scatter diagramming
- Histogram generation

All analyses are performed on single precision, floating-point data, using single precision arithmetic.

SORT/3000

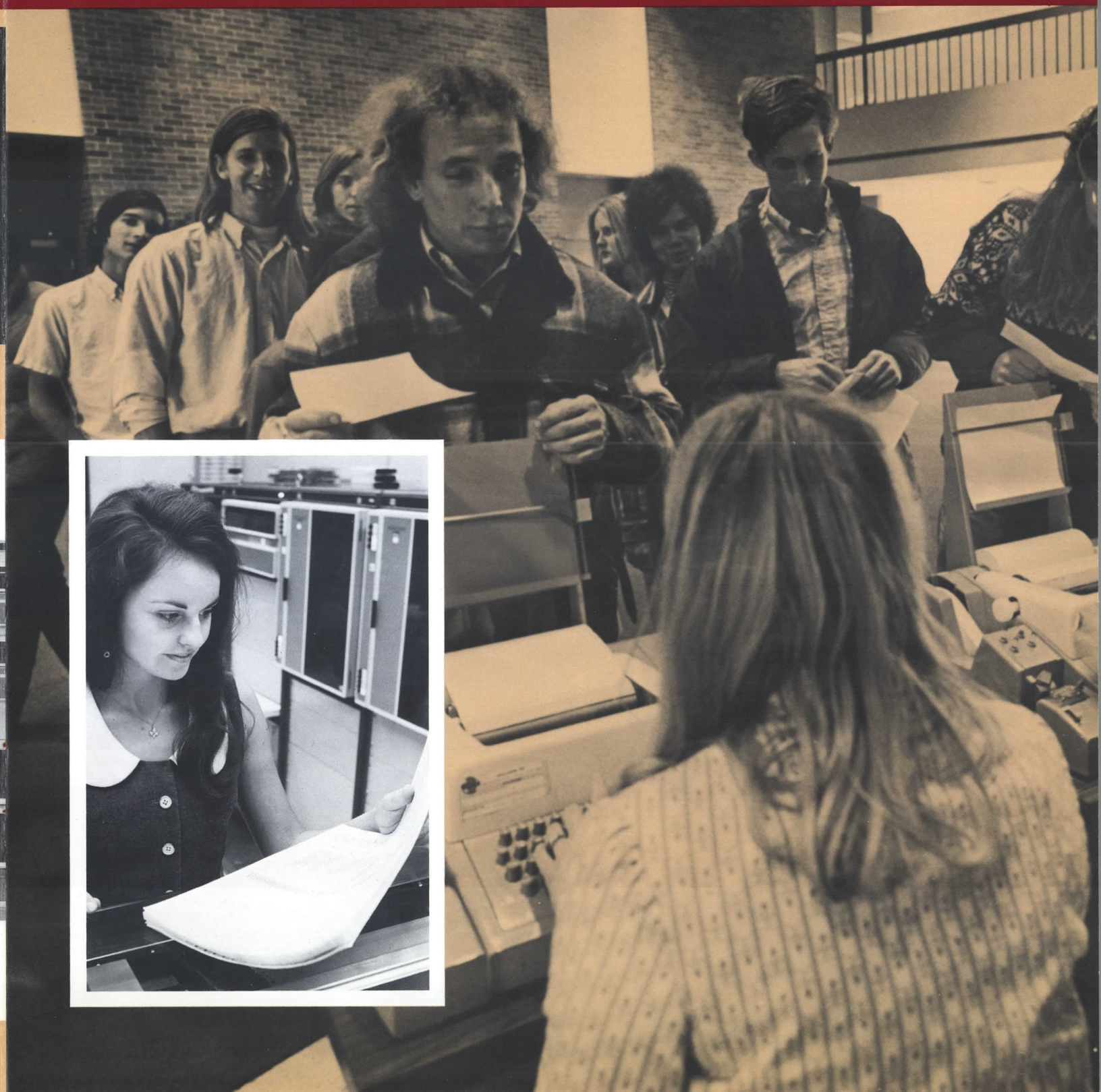
The SORT/3000 Subsystem provides the capability to *sort and/or merge* multiple files of sequential records into a sequential file. This permits users of the HP 3000 Computer System to arrange large quantities of records (a file) into a prescribed order. Sorting is based on keys (values of one or more data fields). Merging forms one sorted sequence of records by combining one or more previously sorted sequences of records.

The program may be employed in a variety of applications:

- As a free standing subsystem, SORT/3000 can be activated through commands in Batch or Session mode.
- As a number of procedures, SORT/3000 provides a set of procedures callable by user programs written in SPL, FORTRAN, BASIC, or COBOL, also via the SORT verb in COBOL.

The SORT Subsystem is capable of sorting or merging records on as many keys as are specified by the user. Individual keys may be contiguous, separated or overlapping and may appear anywhere in the record. Different key fields have their own sequences, thus indicating that different keys can be sorted in different order (ascending/descending) in the same run.

Length, type, relative position, number and priority of keys and type of sort, input and output files, are specified as parameters by the user.



...and for efficient data manipulation

EDIT/3000, TEXT EDITOR

The EDIT/3000 Text Editor permits the user to create and edit on-line/batch computer programs and ordinary manuscript text. It allows the user to manipulate files of upper and lower case ASCII characters with great ease. Lines, strings and characters can be inserted, deleted, replaced, searched for, etc. The files to be edited can be source language programs, such as FORTRAN, SPL, COBOL, etc., or textual material, such as reports.

EDIT/3000 interacts with the user through edit commands. The command language is so designed that a non-experienced user will find those commands that normally exist in all editors (e.g., DELETE, REPLACE, INSERT). Experienced users will find all commands necessary to write complex edit command sequences, where editing is based on conditions found within the text itself.

The Text Editor performs in three modes:

Session: The edit commands are called interactively at the terminal.

Batch: The editor locates the commands in the job input stream.

USE: The editor reads the commands from a file, but sends messages to a standard output device. Edit records are read from a standard input device. This enables implementation of special subsystems for text editing, where it's desirable to make the use of commands transparent to the user.

TRACE/3000, THE SYMBOL TRACE FACILITY

TRACE/3000 is a programmable debugging tool for high-level languages (FORTRAN/3000 and SPL/3000). It allows the programmer to monitor the execution of a program. The programmer can use TRACE/3000 to check the state of the program whenever a variable is changed or a label is passed. In addition, the programmer can specify selective conditions for output of information; e.g. print data only when a variable exceeds a certain value, or when a variable is changed a specific number of times.

HARDWARE DIAGNOSTICS

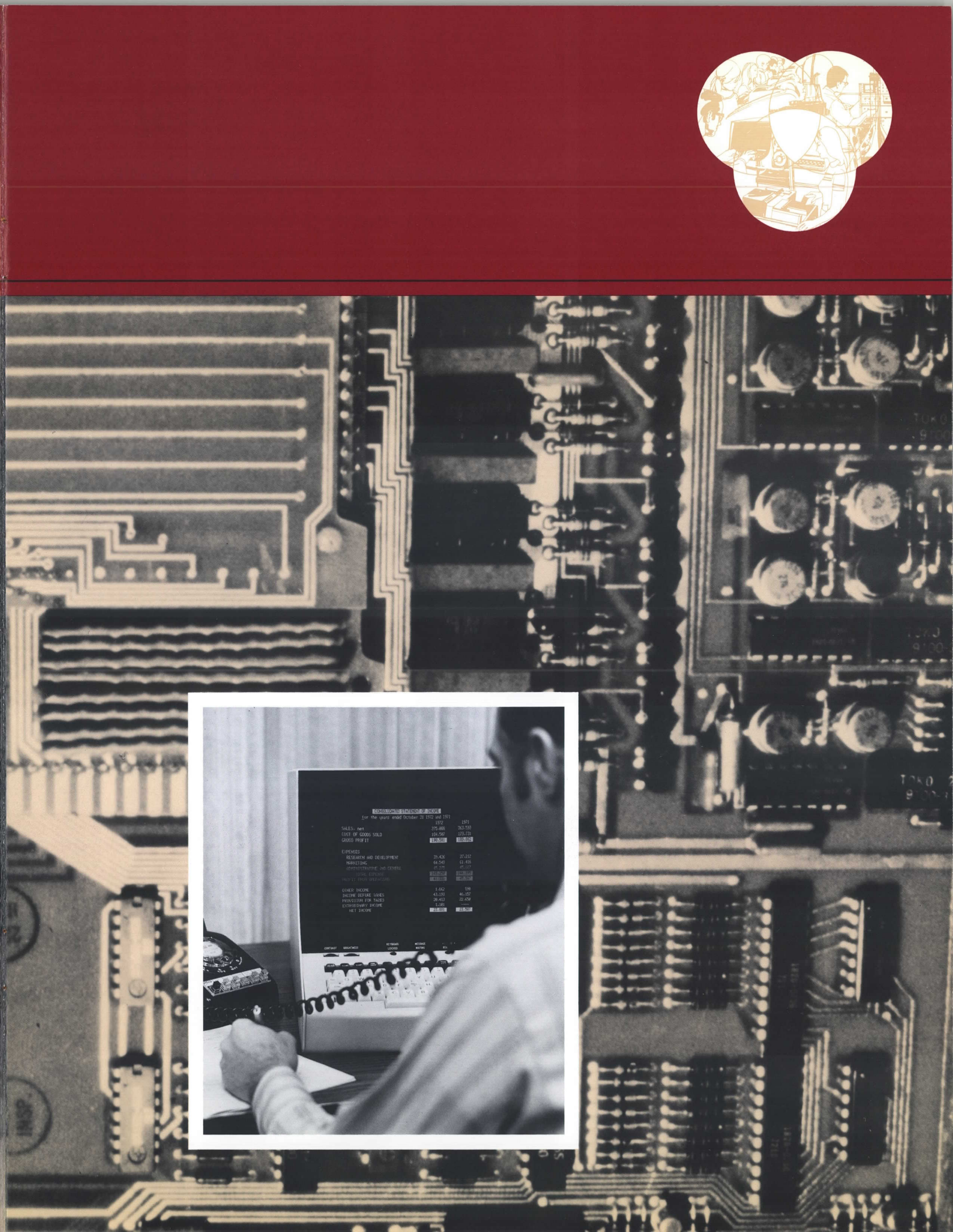
Three levels of diagnostic software are provided to help the computer operators identify, diagnose and correct equipment problems.

- SDM/3000, the System Diagnostic Monitor (performs on-line diagnostics)
- Stand-alone Diagnostics
- Microdiagnostics

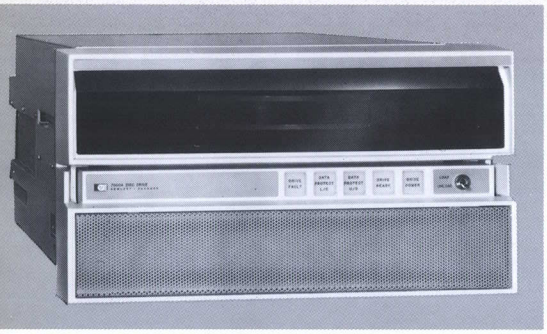
The System Diagnostic Monitor (SDM/3000) performs on-line diagnostics under control of the computer's operating system. SDM/3000 operates any series of diagnostics that exercise and test the hardware of the system (peripherals, memory, etc.). Because SDM/3000 functions under the computer's operating system, it is not always necessary to halt useful work on the system while on-line diagnostics are being performed.

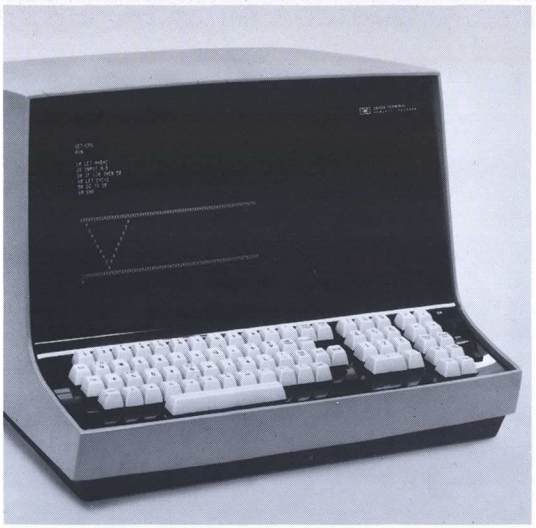
If SDM/3000 cannot function because sufficient hardware is not operable for the operating system of the computer, stand-alone diagnostics can be used. These perform the same functions as the on-line diagnostics of SDM/3000, but each is a stand-alone program that operates directly on the central processor without the need for an operating system.

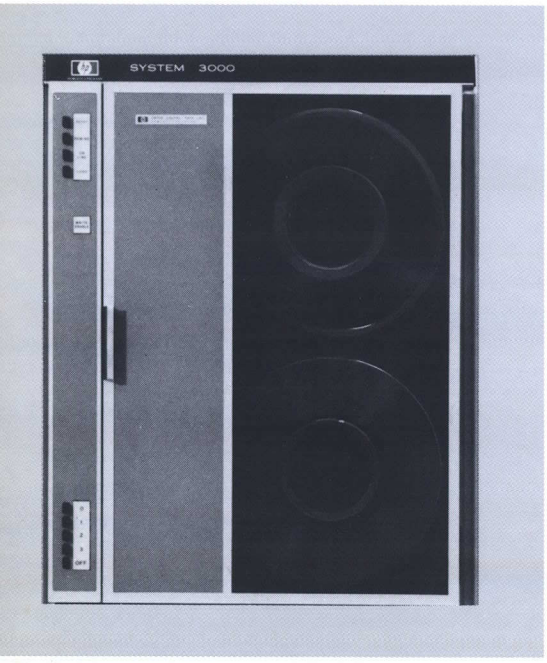
If the problem is such that the stand-alone diagnostics cannot be operated, the microdiagnostics can be used. These are microprograms that replace the instruction set microprogram of the central processor and check hardware functions.

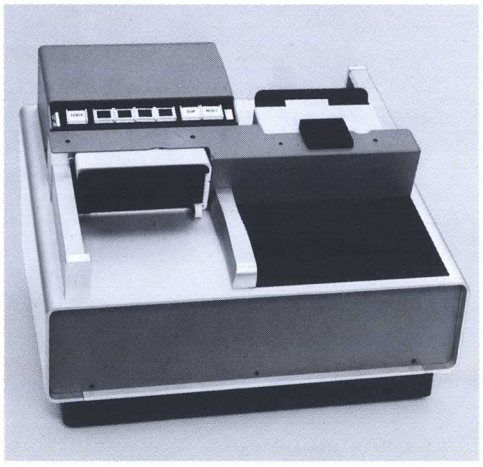

CONSOLIDATED STATEMENT OF INCOME
for the years ended October 31 1972 and 1971
1972
1971
SALES: net
275-088
243-533
COST OF GOODS SOLD
184-587
173-731
GROSS PROFIT
90-501
69-802
EXPENSES:
RESEARCH AND DEVELOPMENT
39.426
27.212
MARKETING
64.540
61.436
ADMINISTRATIVE AND GENERAL
45.329
41.127
TOTAL EXPENSE
149-295
129-775
PROFIT FROM OPERATIONS
(58-794)
(59-973)
OTHER INCOME
1.662
598
INCOME BEFORE TAXES
43.133
46.157
PROVISION FOR TAXES
20.413
22.450
EXTRAORDINARY INCOME
1.185
—
NET INCOME
23.905
23.707
CONTRAST
BRIGHTNESS
KEYBOARD LOCKED
MESSAGE WAITING
ESC

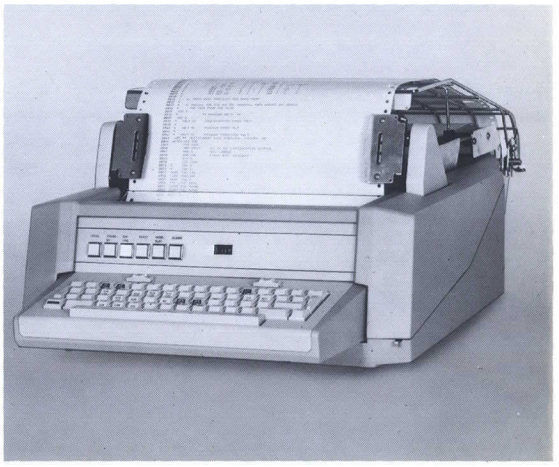
A WIDE CHOICE OF PERIPHERALS
to match your specific needs

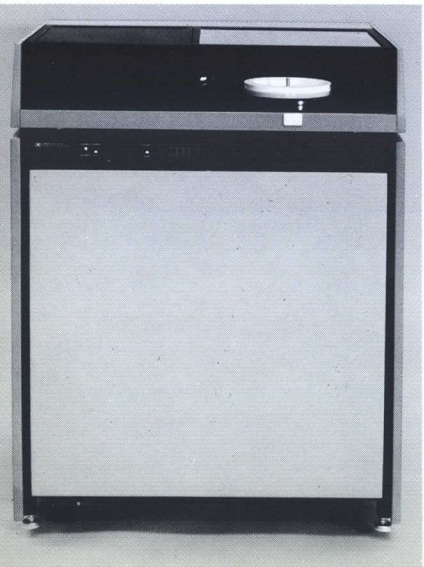












A complete selection of peripheral devices is available for HP 3000. Mass storage devices include both fixed and moving-head disc files. Input/Output devices are available for tab cards, punched tape, magnetic tape and line printout. Several types of system terminals and user terminals are also offered, including standard teleprinter terminals.

HIGH-SPEED DISC FILES (FIXED-HEAD)

These high-speed head-per-track disc files are available in increments of 1, 2 and 4 million bytes per unit. Average access time is only 8.7 milliseconds and data transfer rate is 470,000 bytes per second.

REMOVABLE DISC FILES

For maximum flexibility and storage capacity, two types of removable disc files are available. A two-disc unit provides one fixed and one removable disc with a total combined capacity of 5 million bytes. Average seek time is less than 30 milliseconds and data transfer rate is 246 thousand bytes per second. For maximum storage an 11-high disc file provides 46.8 million 8-bit bytes of storage per unit. This unit has an average seek time of 30 milliseconds and a data transfer rate of 235 thousand bytes per second. Both types of units can be used on the same system.

MAGNETIC TAPE TRANSPORTS

Low-cost magnetic tape storage is available on two types of digital magnetic tape units. Recording densities are either 800 or 1600 bpi at speeds up to 45 inches per second.

CARD EQUIPMENT

Two types of card readers operate at 600 and 1200 cards per minute.

CONSOLES/TERMINALS

Reliable high speed system communication is provided by either a 30 character per second (hard-copy output) terminal or a CRT display terminal. Standard ASR-33 equipment is also available for console and terminal use.

LINE PRINTERS

Line printer output is generated at either 200 or 600 lines per minute. Both units provide 132-column print lines, using either 64 or 96 characters.

PUNCHED TAPE EQUIPMENT

High-speed punched tape input equipment reads at 500 characters per second. Punched tape output is available as a separate unit at 75 characters per second. Either paper, plastic or mylar tape may be used with all units.

INSTRUCTION SET
the largest in its class

The HP 3000 instruction set is the most powerful set in a machine of its class. All 170 instructions require only one word and many stack operations can be packed two instructions per word.

MEMORY ADDRESS INSTRUCTIONS

LOAD	Load word onto stack
LDB	Load byte onto stack
LDD	Load double word onto stack
LDPP	Load double from program, positive
LDPN	Load double from program, negative
LDX	Load index register
LRA	Load relative address onto stack
STOR	Store TOS into memory
STB	Store byte on TOS into memory
STD	Store double on TOS into memory
CMPM	Compare TOS with memory
ADDM	Add memory to TOS
SUBM	Subtract memory from TOS
MPYM	Multiply TOS by memory
INCM	Increment memory
DECM	Decrement memory

BRANCH INSTRUCTIONS

BR	Branch
BCC	Branch on condition code
BCY	Branch on carry
BNCY	Branch on no carry
BOV	Branch on overflow
BNOV	Branch on no overflow
BRO	Branch on TOS odd
BRE	Branch on TOS even
IABZ	Increment TOS, branch if zero
IXBZ	Increment index register, branch if zero
DABZ	Decrement TOS, branch if zero
DXBZ	Decrement index register, branch if zero
CPRB	Compare range and branch

LOOP CONTROL INSTRUCTIONS

TBA	Test variable against limit, branch
MTBA	Modify variable, test against limit, branch
TBX	Test index register against limit, branch
MTBX	Modify index register, test against limit, branch

STACK OP INSTRUCTIONS

CMP	Integer compare top two items
ADD	Integer add top two items
SUB	Integer subtract top from second word
MPY	Integer multiply top two items
DIV	Integer divide top two items
NEG	Integer negate TOS
LCMP	Logical compare top two items
LADD	Logical add top two items
LSUB	Logical subtract top from second word
LMPY	Logical multiply top two items
LDIV	Logical divide top two items
NOT	Logical complement TOS
DCMP	Double integer compare
DADD	Double integer add
DSUB	Double integer subtract
MPYL	Multiply long
DIVL	Divide long
DNEG	Double integer negate
FCMP	Floating point compare
FADD	Floating point add
FSUB	Floating point subtract
FMPY	Floating point multiply
FDIV	Floating point divide
FNEG	Floating point negate
FLT	Float an integer
DFLT	Float a double integer
FIXT	Fix and truncate
FIXR	Fix and round
OR	Logical OR top two items
XOR	Logical Exclusive OR
AND	Logical AND
TEST	Test TOS
DTST	Test double word on TOS
BTST	Test byte on TOS
INCA	Increment TOS by one
INCB	Increment second word of stack by one
INCX	Increment index register
DECA	Decrement TOS by one
DECB	Decrement second word of stack by one
DECX	Decrement index register
ZERO	Push integer zero on TOS
DZRO	Push double integer zero
ZROB	Zero second word of stack
ZROX	Zero index register
DEL	Delete TOS
DDEL	Double delete TOS
DELB	Delete second word of stack
DUP	Duplicate TOS
DDUP	Double duplicate TOS
XCH	Exchange TOS and second word of stack
DXCH	Double exchange
XAX	Exchange TOS and index register
XBX	Exchange second word of stack and index register
CAB	Rotate TOS, second word of stack, and index register
LDXA	Load index register onto TOS
LDXB	Load index register into B
STAX	Store TOS into index register
STBX	Store second word of stack into index register
ADAX	Add TOS to index register
ADBX	Add second word of stack to index register
ADXA	Add index register to TOS
ADXB	Add index register to second word of stack
NOP	No operation

SINGLE WORD SHIFT INSTRUCTIONS

ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
LSL	Logical Shift Left
LSR	Logical Shift Right
CSL	Circular Shift Left
CSR	Circular Shift Right
SCAN	Scan bits

DOUBLE WORD SHIFT INSTRUCTIONS

DASL	Double Arithmetic Shift Left
DASR	Double Arithmetic Shift Right
DLSL	Double Arithmetic Shift Left
DLSR	Double Arithmetic Shift Right
DCSL	Double Circular Shift Left
DCSR	Double Circular Shift Right

TRIPLE WORD SHIFT INSTRUCTIONS

TASL	Triple Arithmetic Shift Left
TASR	Triple Arithmetic Shift Right
TNSL	Triple Normalizing Shift Left

BIT TEST INSTRUCTIONS

TSBM	Test and set bits in memory
TBC	Test bit and set condition code
TRBC	Test and reset bit, condition code
TSBC	Test and set bit, set condition code
TCBC	Test and complement bit, set condition code

FIELD INSTRUCTIONS

EXF	Extract bit field
DPF	Deposit bit field

IMMEDIATE INSTRUCTIONS

LDI	Load immediate to TOS
LDNI	Load negative immediate
CMPI	Compare immediate
CMPN	Compare negative immediate
ADDI	Add immediate
SUBI	Subtract immediate
MPYI	Multiply immediate
DIVI	Divide immediate
ORI	Logical OR immediate
XORI	Logical Exclusive OR immediate
ANDI	Logical AND immediate
LDXI	Load index register immediate
LDXN	Load index register negative immediate
ADXI	Add immediate to index register
SBXI	Subtract immediate from index register

PROGRAM CONTROL INSTRUCTIONS

PCAL	Procedure call
SCAL	Subroutine call
EXIT	Procedure and interrupt exit
SXIT	Subroutine exit
HALT*	Halt
PAUS*	Pause
XEQ	Execute instruction in stack

MOVE INSTRUCTIONS

MOVE	Move words
MVB	Move bytes
MVBW	Move bytes while
SCW	Scan bytes while
SCU	Scan bytes until
CMPB	Compare bytes

I/O AND INTERRUPT INSTRUCTIONS

SIO*	Start block transfer
RIO*	Direct read
WIO*	Direct write
TIO*	Direct test
CIO*	Direct control
SED*	Set enable/disable external interrupts
SIN*	Set interrupt
SIRF*	Set interrupt reference flag
SMSK*	Set mask
RMSK*	Read mask
CMD*	Send command to module

REGISTER CONTROL INSTRUCTIONS

PSHR	Push registers onto stack
SETR*	Set registers from stack
ADDS	Add to stack
SUBS	Subtract from stack
XCHD*	Exchange data base and TOS

SPECIAL INSTRUCTIONS

PLDA*	Privileged load from absolute address
PSTA*	Privileged store into absolute address
RSW	Read switch register
LLSH*	Linked list search
LLBL	Load label
MVBL*	Move from data base+ to data limit+
MVLB*	Move from data limit+ to data base+

NOTES:

TOS = Top of stack

** = Privileged instruction*

HEWLETT hp PACKARD

For more information, call your local HP Sales Office or East (201) 265-5000;
Midwest (312) 677-0400; South (404) 436-6181; West (213) 877-1282.
Or write: Hewlett-Packard, 1501 Page Mill Road, Palo Alto, California 94304.
In Canada: 275 Hymus Blvd., Pointe Claire, Quebec.
In Europe: Post Office Box 85, CH-1217 Meyrin 2, Geneva, Switzerland.

Printed in U.S.A. 2/73

5952—4431(22)