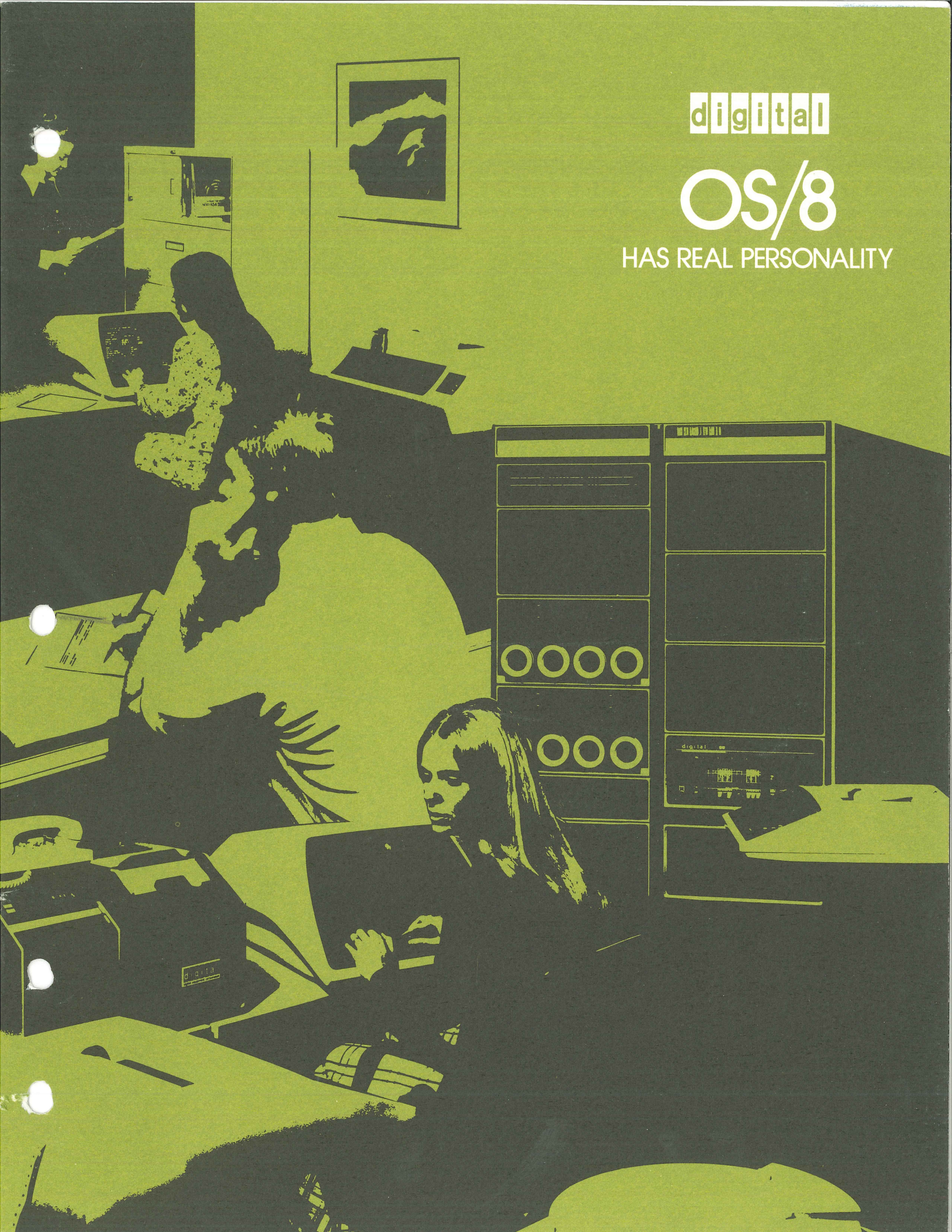
digital

OS/8

HAS REAL PERSONALITY




digital pdp8/e
digital equipment corporation maynard massachusetts
decpack
decassette
digital pdp8e

OS/8 SOFTWARE-SOME INSIGHTS

WHY OS/8?

With the dramatic decline of computer equipment prices over the last five years, an equally dramatic increase in product development expenditures has occurred. Chief among these higher costs is the labor intensive computer program development phase. What this means to you as a computer user is that the attractiveness of various computer applications is still dimmed by program development costs. The answer to this problem is provided by an inexpensive, yet powerful software product development system called OS/8.

WHAT IS OS/8?

OS/8 (Operating System/8) is a comprehensive library of system programs operating under the supervision of an integrated executive. This executive, through the keyboard monitor, controls the operation of the computer and handles all the input/output devices. The OS/8 operating system represents a major advance in small computer software development. Continually evolving, OS/8 has grown from previous PDP-8 software systems until it now offers features that were formerly available only on such powerful machines as the 36-bit DECsystem-10.

OS/8 thus fulfills the two primary functions of an Operating System:

- To organize a collection of hardware into a consistent and coherent whole and control the hardware so that it can be used for the development and running of programs.
- To assist users and allow their programs to make the fullest possible use of the hardware with the least possible programming effort, by the provision of software to supplement and enhance the hardware facilities available.

And it does this at the lowest possible cost because OS/8 is a total hardware-software package combining the best efforts of DIGITAL's hardware and software engineering expertise. With a small investment the computer user can purchase a complete Standard-8 System consisting of a powerful PDP-8/E minicomputer with 8 or 16K of core memory, a cartridge disk system, a dual drive magnetic tape cassette system, a hardware bootstrap loader, a hard copy I/O terminal, a large cabinet system enclosure and, of course, the OS/8 software package. The users who already have a PDP-8 configured with DECtape or Disk and a minimum of 8K of core need only purchase the software package. This is very easy to do since OS/8 supports more than fifteen peripherals of DIGITAL design and has the additional capability for the users to write handlers for their own devices. OS/8 is distributed on several media: DECtape, LINCtape, Cassette, and Paper Tape.

WHAT CAN OS/8 DO FOR YOU?

OS/8 was specifically designed to shorten the time required for program development, increase throughput at dedicated data processing installations, and facilitate system management. What this means to you is a highly significant reduction in programming expense accompanied by increased work performance. This is made possible because OS/8 allows programmers to:

- Store data files or executable programs in one or more system libraries where they may be accessed for loading, modification or execution by simple keyboard commands.
- Perform convenient program chaining so that a problem may be divided into a set of smaller programs, each written in the language which is best suited to it. In a similar manner, very large programs may be coded in small segments that can be overlaid during execution to conserve memory storage.
- Write programs coded in a format that allows complete I/O device independence. Program I/O is performed by standardized calls to system device handlers and a comprehensive I/O supervisor called the User Service Routine. This feature permits programs to be written without regard for the characteristics of a particular I/O device.
- Assign logical names to devices within the OS/8 system. This permits symbolic referencing of peripheral devices and makes certain classes of devices fully interchangeable from a programming standpoint. User programs retain full control over the length of I/O buffers to ensure optimum use of available storage and fast, efficient block data transfers.
- Easily communicate with the OS/8 executive via the keyboard monitor which uses only seven monitor-level commands. The keyboard monitor accepts commands from the console terminal to assign logical device names, load, run, and save system or user programs, and execute the "invisible" debugging routine. This is so designated because it appears to the user as though it does not occupy any memory.
- Decode and execute additional monitor level commands using the Concise Command Language. The CCL functions as an extension of the keyboard monitor and is loaded automatically when needed. Most CCL commands instruct the system to perform a sequence of common operations, often requiring execution of several utility programs. More than forty such commands are provided, covering the most common operations. The comprehensive CCL vocabulary is easy to learn and its application reduces tedious keyboard I/O to an absolute minimum.

The operation of the PDP-8 and its various configurations is enhanced by OS/8 software.

- Call the OS/8 Command Decoder during execution of a system program or a device independent user program. It accepts a command line from the console terminal and decodes the command to determine what combination of peripherals, input files, output files and runtime options will be used during the current execution of the program. Because all programs use the same command decoder, I/O specification commands are highly standardized, and the time required to become familiar with the system command structure is greatly reduced.
- Control all device independent I/O and file directory operations under OS/8 through the User Service Routine or USR. Any system or user program may access the USR by executing a standard calling sequence. Functions performed by the USR include loading device handlers; searching file directories; creating, opening and closing files; calling the command decoder; and program chaining.
- Obtain maximum utilization of available storage for programs or data because the resident portion of OS/8 is limited to only 256 words of memory. Non-resident portions of the system are swapped into memory from the system device automatically, as required. Although OS/8 runs in only 8K of memory, it is self-expanding to use up to 32K.
- Do programming in the language best suited to the task. Also the higher level languages, BASIC, FORTRAN II, and an ANSI compatible FORTRAN IV, can be supported under OS/8. The highly efficient assembler language PAL8 can be used when speed or program size is critical.
- BATCH a series of OS/8 actions. The program BATCH can take a series of commands from a file and pass them to the Keyboard Monitor, Command Decoder, and CCL. This allows the OS/8 system to run any number of jobs without operator intervention.

What all this means to you is that OS/8 can save you time and money. It also allows you to maximize your programming resources in another important respect.

With the introduction of RTS-8 (Real Time System-8) a new dimension has been added to OS/8. RTS-8 allows the processing of multiple tasks on a PDP-8 in a real time mode. Tasks are written, assembled and loaded under OS/8 to create the independent RTS-8 system. This means the user can now monitor and control a number of processes concurrently. Additionally, the system offers a foreground-background mode which enhances the versatility of the PDP-8, by permitting a number of real-time tasks to be executed in the foreground, while the OS/8 system runs in the background for program development.

SOFTWARE CAPABILITIES

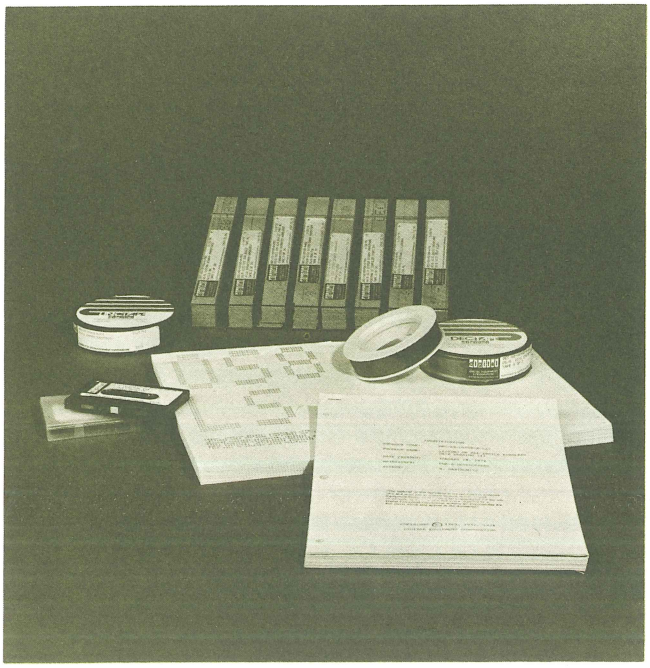
Software allows a computer to do useful work by means of carefully thought-out instructions understood by both people and computers.

But just to do useful work is not the goal. Certain software is selected because of its ability to accomplish a specific function with the least amount of effort. Software efficiency is crucial to saving programming time, computer time and operating costs. Whether or not you get the most from your computer also depends a lot on in-depth understanding and use of computers and their capabilities.

FEATURES OF THE OS/8 OPERATING SYSTEM

The ability of OS/8 to make the most of existing computer hardware is its outstanding feature. The people at DIGITAL have had many years of experience designing and producing computers, and they understand the requirements of a high-performance system. OS/8, designed for the PDP-8/E computer, is a breakthrough in small computer software development, offering all the aspects of a powerful operating system, yet requiring a minimum of programming know-how.

OS/8 shortens program development time, increases throughput, and facilitates system management. Its applications are in education, industry, data processing, research, and engineering.



OS/8-
THE PERSONALITY SYSTEM

SOPHISTICATED
A range of languages is available from basic to assembler. Either of two powerful text editors can be used for program development. Every OS/8 system can be easily expanded to include virtually any peripheral device.

POWERFUL
From 8K to 32K of memory.

VERSATILE
Complete I/O device independence so that when a system is expanded, no rewriting or reassembly is required. The user-service routine may be accessed by any system or user program executing a standard calling sequence.

EFFICIENT
Easy access to stored data files or programs simplifies the application and shortens the time for program development, debugging, and execution.

Because of the compactness of the OS/8 monitor, users can obtain maximum usage of the memory available for their application programs.

EASY TO WORK WITH
Easy to work with, yet comprehensive Concise Command Language (CCL), with more than 40 commands, requests common operations. CCL keeps tedious keyboard I/O to a minimum.

THE MAIN SOFTWARE COMPONENTS OF THE OS/8 SYSTEM ARE
- Keyboard monitor
- Command decoder
- Device handlers
- User-service routine (USR)
- Library of system programs.

Keyboard Monitor—Provides communication between the user and the OS/8 executive routines by accepting commands from the terminal keyboard.

Command Decoder—Allows the user to communicate with the system library program by accepting a command string from the keyboard, indicating input/output devices and files.

Device Handlers—Transfer data to and from peripheral devices. Device handlers are an integral part of the system that allow OS/8 to perform device independent I/O on as many as 15 different peripherals at a time.

User-Service Routine (USR)—Controls the directory operations for the OS/8 system. A program can use the USR by means of standard subroutine calls. The utility functions do not need to be coded into a user's program, a feature saving development time and memory.

Library of System Programs
BATCH processing
FORTRAN
BASIC
Assemblers
Editing Programs
Debugging Programs
File Maintenance Routines
Laboratory Application Programs
Industrial BASIC

OS/8 KIT CONTENTS AND ORDERING CODES

	OS/8 System Kit*		OS/8 Extension Kit	OS/8 FORTRAN IV Kit	OS/8 LAB-8/E Kit
Kit Contents	MONITOR ABSLDR PIP PAL8 SABR CREF BUILD EPIC FOTP DIRECT CAMP BOOT TDCOPY DTCOPY EDIT	FORT LIB8 LOADER BITMAP SRCCOM CCL RESORC PIP10 MCPIP DTFRMT TDFRMT RK8FMT RKEFMT ODT	BATCH MSBAT BASIC TECO	F4 FRTS RALF FORLIB LIBRA LOAD	AABG BASAV PTS TIH CORD DAQD DAFFT PAFFT CON
Papertape	QF015-AB		QF006-AB	QF008-AB	QF009-AB
DECtape	QF015-AC		QF006-AC	QF008-AC	QF009-AC
LINCtape	QK015-AA		QK006-AA	QK008-AA	—
Cassette	QF015-AN		QF006-AN	QF008-AN	—

* Purchase of the OS/8 System Kit is a prerequisite for the other OS/8 kits or individual OS/8 programs.

OS/8 SYSTEM PROGRAMS CAN DO MORE FOR YOU

BATCH PROCESSING

Batch processing monitor—Lengthy jobs can be run on the computer during off hours and can be executed unattended.

Optional spooling of output files—Execution is faster, less paper is consumed, and output can be dumped selectively from the spool files to any hard-copy device at a later time.

Full-monitor command set—With BATCH, only a single command is needed to carry out a complicated procedure ordinarily requiring several monitor calls, extensive I/O specifications, and numerous commands. Operator intervention is not required.

All the commands for Keyboard Monitor, Concise Command Language, and Command Decoder are available in BATCH.

FORTRAN II

Uses ANSI Standard FORTRAN—Programmers can write their programs in a widely known language which conforms to industry standards.

A-Format specification and Hollerith literals—Programs can be written to process alphanumeric data. Hollerith data can be included in format statements, and characters do not have to be counted, a feature which saves programming time.

Hollerith constants— No need to remember numeric values for alphanumeric data. Constants are automatically read and translated by the program.

Mixed-assembly language and FORTRAN code allowed—Provides programming flexibility because FORTRAN and assembler code can be efficiently combined to meet unusual program requirements.

Program chaining—Large programs can be handled in limited memory.

OS/8 FORTRAN IV

Full, ANSI-standard FORTRAN IV— Existing programs run with little or no modification. New software is not machine-dependent. Requires little learning time.

Hardware independence —The run time system can use from 8K to 32K, and automatically configures itself to take advantage of an extended arithmetic element or floating point processor.

Interrupt-driven run-time system—The system can perform specialized background functions, such as refreshing a scope display, while the processor is executing an I/O transfer.

Dynamic run-time overlays—Provides 10 times the existing size of memory, so that programs requiring over 300K words of core storage can be structured to run on a 32K PDP-8.

Direct access I/O—Random blocks in very large files can be accessed and updated independently.

Generalized array subscripting—Data is more accessible; further processing is easier.

Every program is fully I/O-device independent—Standard device specifications can be used, or unit specifications can be freely assigned at load-time.

Error diagnostics at compile and run-time—Makes the job of debugging easier and quicker.

Mixed-mode arithmetic—Eliminates mixed-mode errors.

Logical variables and operators—Expedites routine decision processing.

Text manipulation—Enables FORTRAN IV to accept and store words of text, as well as numeric values, such as tabular data.

OS/8 FORTRAN IV PLOTTER

Drives an incremental plotter under control of OS/8 FORTRAN IV—Permits optimum use of core.

Complete user control over pen state (up/down), angle of plotting, and overall size of plot—Diagrams and other drawings can be adapted to the user's specific needs.

OS/8 BASIC

Similar to Dartmouth College BASIC—Compatible with, and as easily learned and applied as versions of Dartmouth BASIC. Offers extended operations and functions for the experienced programmer.

8K-32K of memory—Provides in 8K plenty of space for computational code or data storage. If more than 8K is available, it can be used to increase program size and/or data storage capability.

Program chaining—Program size is virtually unlimited.

Two-processor configurations—OS/8 BASIC runs with or without the Extended Arithmetic Element (EAE). Programs execute faster with EAE.

Interactive editing—Provides for instant modification of a program.

String operations—Expands the applications into such areas as text processing, formatting, and command language interpretation.

Core-image files—Compiled BASIC programs may be stored as save images and loaded into another OS/8 system. Programs created on one OS/8 configuration will run on any other OS/8 system having sufficient memory.

Dynamic file I/O—No limit to the total number of files a program can access, as long as only four remain active simultaneously. Files and devices may be specified during program execution.

OS/8 INDUSTRIAL BASIC

A modification of OS/8 BASIC that supports real-time industrial data acquisition and control. It permits interfacing with functions or software modules written in PAL assembler, offering flexibility to the user.

Industrial BASIC supports the standard UDC-8 options (Universal Digital Controller) including digital input and output, analog input and output, sense switches, flip flops and clock.

LAB-8/E FUNCTIONS—The LAB-8/E functions included in OS/8 BASIC enable the user to solve a range of real-time and pseudo real-time problems using a higher-level language. This program contains a set of 12 functions which enable a user or OS/8 to utilize the A/D converter, display control, real-time programmable clock and 12 channel buffered digital I/O.

SABR ASSEMBLER

SABR (Symbolic Assembler for Binary Relocatable programs) is an advanced, one-pass assembler producing relocatable binary code with automatically generated page and field linkages. It supports an extensive list of pseudo-operations which provide, among other facilities, external subroutine calling with argument passing and conditional assembly.

A SABR program may call routines from a large library of subroutines and functions. These are loaded together with the SABR program by the Linking Loader. In an optional second pass, SABR produces an octal/symbolic listing of assembled programs.

The relocatable binary tape produced by a SABR assembly is loaded into core for execution with the 8K Linking Loader.

Binary relocatable output—Programs can be divided into logical entities. Each division can be written, assembled, and loaded independently.

Extensive library of mathematical, utility, and input-output subroutines—Provides optimized code and saves programming, debugging, and execution time.

Extensive list of pseudo-ops—You have a powerful assembler.

Generates links to off-page references—Saves time and effort, as programs can be written without concern for page boundaries.

From 8K to 32K of memory—Programs are easily expandable.

SABR is fully integrated into the OS/8 operating system—Saves programming, debugging, and execution time.

PAL-8 ASSEMBLER

Extended symbolic assembler—Takes advantage of all the OS/8 features: disk, DECtape, line printer, and more.

Conditional assembly—Enables the same source program to produce different binaries for different purposes.

Paginated listings, page headings, and numbered pages—Improves program documentation.

Binary-symbol table search—Fast assemblies.

ABSLDR (Absolute Binary Loader)

The absolute loader program reads a binary program into memory and creates the resident core image. Remembers and saves core locations that have been used thus conserving mass storage space.

EDITING PROGRAMS:
EDIT—SYMBOLIC PROGRAM EDITOR

Interactive—Programs can be easily modified.

Many editing commands available—Programs can be typed, modified, and checked quickly and conveniently using simple keyboard commands.

Character search command—Part of a line can be changed without retyping the entire line.

Character string search—A specific part of a very long program can be easily located. Saves time.

TECO

Powerful editor that will handle any form of ASCII text, including program files or listings, manuscripts, and data files—The OS/8 user needs to know only a few simple commands to accomplish all editing.

Character-oriented rather than line-oriented—There are no extra line numbers. It is not necessary to replace an entire line of text in order to change one character.

Editing Macro commands—Permit the operator to make repetitive changes throughout a large program or data file with only a single command.

Command strings may be saved—Saves time on subsequent executions.

DEBUGGING PROGRAMS:
BITMAP

Maps the memory requirements of any binary file—Vacant memory areas are easily identified, eliminating time spent on scanning source listings page-by-page to locate free core areas.

Locates double- or multiple-memory register allocations—If a memory register is accidentally allocated to more than one binary data word, BITMAP flags the register and indicates how many times its contents were specified.

Maps multiple files simultaneously—Overlays may be mapped onto the program that they will load over. Modules may be mapped in sets, reflecting the manner in which they will be loaded. Programs can be compared, register-by-register, on the basis of storage allocations.

CREF (Cross-Reference Utility Program)

Alphabetical cross-reference table—Produces a numbered assembly listing, and then alphabetically lists each symbol and literal in the program and line number of every reference to it.

Optional two-pass operation—Doubles the number of symbols that can be accommodated in a program.

ODT (Octal Debugging Technique)

Invisibly co-resident with the user program—No need to allocate memory for a debugging package during development.

Breakpoints can be set anywhere in a program—The user can trace the execution of a program step-by-step or even instruction-by-instruction. Execution proceeds normally up to and then after the breakpoints.

Binary memory search mechanism—Specified areas of memory may be searched.

FILE MAINTENANCE ROUTINES:

PIP (Peripheral Interchange Program)

Will transfer ASCII, core image, or binary files from one I/O device to another—Offers great flexibility in programming and saves time.

Can merge and delete files, as well as list, zero, or compress file directories—Leaves the user more time to concentrate on problem solving.

FOTP (File-Oriented Transfer Program)

Can transfer an entire class of files in one operation—Eliminates the time spent in specifying large numbers of files one at a time.

Wild-card and wild-character constructions—Useful for transferring a large class of files with a given name, or given part of a name. FOTP allows users to quickly and efficiently back their systems and/or application program. Suppose a disk containing 200 files, of which 50 were FORTRAN source programs, needed to be transferred to DECtape. This task could be done with a single command.

OS/8 TSAR

Powerful, modular FORTRAN IV—Designed to acquire, manipulate, display, plot, and report experimental data in a form that is easy to use, yet provides all the power needed to solve many scientific and engineering problems. Special-purpose functions useful for a specific application can be easily added.

Modular design—Easy for the user to add special-purpose functions needed for a specific application.

Powerful commands (for Fourier Analysis, data I/O, mathematic and program control)—Commands make each procedure essentially a high-level language program.

LABORATORY APPLICATION PROGRAMS: MINIMUM SYSTEM REQUIREMENTS

The following components are required to operate the Laboratory Application programs.

- PDP-8/E central processor with an ASCII compatible terminal.
- 8K of main frame memory.
- Auxiliary storage devices:
  a. Dual-drive DECtape and bootstrap.
  b. Single-drive, removable-cartridge DECpack with a dual-drive cassette or a high-speed paper tape reader/punch.
  c. Fixed-head disk with at least 128K words of storage plus a dual-drive cassette or a high-speed paper tape reader/punch.
- A/D converter and multiplexer.
- Programmable real-time clock with Schmitt triggers.
- Point plot display controller.
- OS/8 binary license.

SUPPORTED OPTIONS

- Extended arithmetic element.
- Additional DECtape or disk drives.
- XY analog plotter.

OS/8 LAB-8/E SOFTWARE SYSTEM

The LAB-8/E software is a wide-ranging, easy-to-use laboratory application package.

Software systems for the LAB-8/E carry the researchers from controlling their experiments through data acquisition, to reduction and analysis.

The software is interactive, communicating with the scientist through the display scope and console terminal.

The LAB-8/E system software has provisions for hard-copy plots on a standard X-Y analog recorder, giving the researcher familiar outputs. All programs provide numerical output for further analysis by higher level languages. The following sections describe the LAB-8/E application software package that runs under the OS/8 operating system.

BASIC SIGNAL AVERAGER

The Basic Signal Averaging software allows an effective high-speed technique for improving the signal-to-noise ratio of a repetitive analog signal. Data is acquired via an analog-to-digital converter (ADC). The raw data is stored in a current buffer after the sweep and is added to the sum buffer after the sweep is over. During the sweep, control may be exercised to display the average or the last sweep.

Start-up of the program is conversational, allowing specification of the number of sweeps, sampling rate, and delay from the starting sync pulse.


digital

Features of the Basic Signal Averaging Software

- 1,024 points (expandable).
- 24-bit word length storage of the sum buffer.
- 30 kHz to 250 Hz data sampling rates; 40 kHz for single channel averaging.
- 1 to 16 inputs.
- Sampling rate, number of sweeps, and delay are changeable using simple keyboard commands.
- Display-the-input-signal or display-the-averaged-data modes.
- Accumulation of the average may be paused at any time.
- Displayed average may be contracted or expanded for visual clarity.
- Print-out of data between any two selected points.
- Plot of waveform on X-Y analog recorder.

ADVANCED SIGNAL AVERAGER

The Advanced Signal Averager is distinguished by the capabilities to: back (presync) average; sort or edit averages contingent on some external event; sample at two different rates; and calculate the raw statistics needed for standard deviation confidence limits, and trends. It provides statistical confidence limits around the average, standard deviation for each data point in the average, and provides first order trend read out.

Features and Options:

- 65 to 1,024 points averaged in double precision (24 bits).
- 5.7 kHz to 0.5 Hz data sampling rates.
- 1 to 16 inputs.
- Conversational-mode scope display asks questions about averaging parameters.
- View-Input or View-Average Mode. Accumulation of average may be paused at any time.
- Dual resolution sweeps simultaneously.
- Data can be sorted or edited using the digital I/O option.
- Both negative and positive delays from the sync signal.
- Confidence limits (or standard deviation) may be calculated on each point while the average is being taken.
- Trend analysis on each data point.
- Data is automatically normalized before output.
- Data may be typed out between movable cursors.
- X-Y analog plotter calibration routine.
- Supports up to 32K of memory.

HISTOGRAM APPLICATION PROGRAM

This program provides a powerful tool for the neurophysiologist (and others) to investigate both spontaneous and stimulated spike train activity through the generation of three histogram types: time-interval, post-stimulus, and latency. The time-interval histogram (TIH) program yields a frequency distribution of interspike intervals. An additional histogram is available which indicates the overall rate for intervals during the TIH acquisition. The post-stimulus (PST) histogram indicates distribution of activity over a range of times following a stimulus; while the latency (LAT) histogram indicates the time distribution of the first "M" spikes following a stimulus. Both the post-stimulus and latency program have a second histogram which indicates total activity following each stimulus.

Features:

- Minimum time may be specified (e.g. in order to ignore stimulus artifacts).
- Output data displayed on the CRT may be smoothed, expanded, contracted, plotted or photographed.
- Variable resolution of frequency distribution.
- Overflow and underflow channels.

AUTO- AND CROSS-CORRELATION APPLICATION PROGRAMS

The Auto- and Cross-Correlation Package is designed to correlate data at sampling rates ranging from 1 to 204.7 milliseconds, on-line, with the user controlling all parameters from the terminal. It displays and scales data while computing and provides output that can be used with other DIGITAL programs.

Correlation, as it applies to waveforms, can be used to detect periodic signals buried in noise or provide a measure of similarity between two waveforms. Autocorrelation measures the similarity of a signal to a time-delayed version of itself, while cross-correlation measures the degree of similarity of one source or input to a second source. No synchronizing events, such as the trigger required in signal averaging, need be available for the application of correlation techniques.

Features:

- Both Auto- and Cross-Correlation on-line.
- Up to 512 point correlograms.
- Display of the input or the correlogram.
- Cursors are provided for selective type out.
- Type out of correlation coefficients (+1 to −1) or raw data.
- Convenient analog plotting routine with plotter calibration.
- Baseline adjustment.
- Parameters may be re-specified at any time.
- Data taking may be paused at any time.

DAQUAN

DAQUAN is used for data acquisition in the time domain by boxcar, multisweep signal averaging, and for general-purpose data reduction. After the data is acquired, a wide variety of processing techniques are used to analyze the data interactively. A special feature of DAQUAN is its ability to determine peaks in a complex spectrum. Once the peaks have been defined, a report may be printed containing individual peak information consisting of peak minima, peak maxima, maxima as a percentage of the largest peak, and the percent area. The command structure of DAQUAN is simple to use and flexible, with each operation requiring only a single command to operate on one or two spectra.

DAQUAN has commands for:
Spectra comparison
Spectra stripping
Baseline corrections
Gaussian fitting
Lorentzian fitting
Integration
Differentiation
Axis inversion
Multiplication
Scaling
Plotting (Standard X-Y analog recorder)
Data printout or punching
Deconvolution of Fused Peaks (by Gaussian, Lorentzian or mixed technique)

Fast Fourier Transform (FFT) Processing Using DAQUAN (requires EAE)

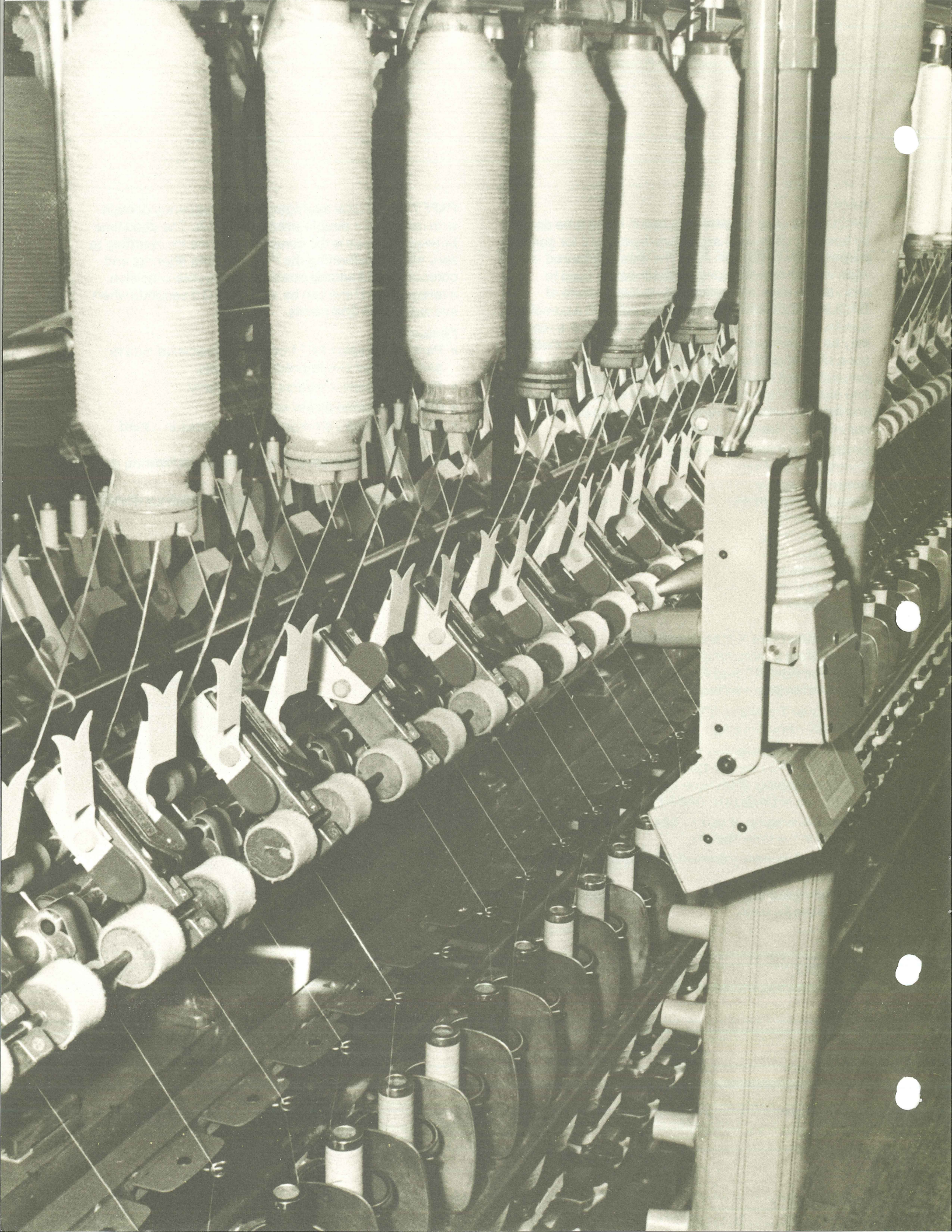
DAFFT (Data Acquisition and FFT) and PAFFT (Power Average by FFT) are two DAQUAN-based programs for signal processing. Both have most of the functional capabilities of DAQUAN; DAFFT has provisions for:

- Signal averaging of up to 1,024 double-precision points in the time domain.
- FFT into frequency domain; up to 1,024 complex coefficients.
- Power versus frequency spectrum as sum of squares of the coefficients.

PAFFT does signal averaging in the frequency domain with up to 1,024 double-precision points. The dead time between sweeps is the computation time for resetting to zero mean, performing the FFT, Hanning filtering, and computing and adding power to the double-precision average. Dead time can be as little as 0.5 seconds when averaging with 256 points.

Features:

- Perform complex FFT from single precision data to give complex single-precision results.
- Perform complex inverse FFT similarly.
- Compute scaled power spectrum.
- Allow temporary storage of 1,024 values in a third buffer area.
- Have option of 3-point or 11-point digital filtering in time domain.
- Have option of Hanning filtering in frequency domain.



HOW OS/8 FITS INTO AN INDUSTRIAL OPERATION

OS/8 is the basis of a new management information system designed for the textile industry by the Parks-Cramer Company of Fitchburg, Massachusetts.

In textile production, processed fibers (cotton, wool, synthetics) must be spun into yarn before being woven into fabric. The spinning process is very much like the old spinning-wheel principle. Fibers are stretched and twisted on a spinning frame, then wound around a spindle at the bottom of the frame. Spinning-frame operation is around-the-clock, with up to 400 spindles producing yarn on a single frame. It is not unusual to have several hundred frames in a single room. Nor is it unusual for a yarn break to occur at any individual spindle. These breaks can cause significant loss. The system developed by Parks-Cramer minimizes losses by shutting down material feed to an inoperative spindle, and keeping track (via computer) of any and all conditions in the spinning room.

A solid-state photoelectric sensor moves along the spinning frame, looking for broken yarn. When the detector locates a broken end, it directs a blast of air at a "feed-stop" device, terminating any additional fiber feed to the process and raising a bright red flag to indicate repair is needed. A patrolling person called a "spinner" spots the flag, corrects the problem, and puts the spindle back into operation.

A six-bit signal is sent to the computer at every spindle location, designating whether or not the spindle is functioning. Up to 16 detectors may be travelling through the spinning room at once, all sending information back to the computer to be stored and tabulated. At the end of each eight-hour shift, the computer prints out several reports. The Spinning Room Performance Report indicates the number of pounds of yarn produced, percent of time running for each frame, average number of ends down on each frame on each pass, overall frame efficiency, and employee efficiency at correcting spindle problems.

Other reports designate frames operating below standard, provide print-out data on special test frames, and indicate which individual spindles are down with high frequency.

System software was developed under OS/8, using the PAL-8 assembler, EDIT, CREF, and various other system programs. The program, including report tables and memory-resident data, operates in less than 12K of core, with a disk used for storage of data collected on-line.

In addition to continuously collecting and tabulating incoming spindle data, the system scans each spinning frame every six seconds to determine whether it is operating or not, and keeps a record of how long the frame is down. (Since the frame must be deactivated to manually replace full spools with empty ones, this report also measures employee efficiency.)

Textile industry management has found the system to be valuable in providing continuous information on the efficiency of their operation, in helping to pinpoint problem areas, and in helping to optimize all phases of production.

OS/8—THE ADAPTABLE OPERATING SYSTEM

OS/8 supports small, low-budget as well as full-scale systems. All products are carefully designed, reliable, and easy to use. The sophistication of the system is a result of the installation of large numbers of PDP-8 systems.

The efficiency and versatility of the OS/8 programs save development time and dollars. Simple commands can transfer files, list files, access data, compile programs, and bootstrap systems. Programs and data can be tested and modified on-line. One program, LAB-8/E, comprises an entire laboratory program.

The OS/8 system is powerful, sophisticated and easy to learn. It allows the laboratory worker, business data processor, or university researcher to concentrate on the major problem-solving aspects of the job at hand, leaving the routine work to the computer and its carefully defined software.

A wide variety of help is available to the OS/8 user through DIGITAL's Software Services, which offers such resources as consulting, publications, and program updates. This back-up group is dedicated to supporting the user's needs and standing by DIGITAL's systems. An enormous program library is available through DECUS, the Digital Equipment Computer Users Society. This worldwide organization provides unlimited opportunity for application and other information exchange. Customer training courses and the expertise of skilled Field Service engineers and technicians are among the other DIGITAL services provided.

The photoelectric detector (and associated logic) is incorporated into a moving, blowing, and vacuum system (designed and manufactured by the Parks-Cramer Company), which removes dust and lint from the spinning area. The system—cleaner, detector, and computer—is sold and installed by Parks-Cramer of Fitchburg, Massachusetts.

OS/8 AT A GLANCE

MINIMUM HARDWARE REQUIREMENTS

- PDP-8/e, PDP-8/f, PDP-8/m, PDP-8/a
- 8K of memory
- Mass storage: TU56 DECtape, RK8E, DF32, RF08 disks
- Console Device: LA30, TTY, LA36

OS/8 SOFTWARE FEATURES

- Store data and programs by name in files on any of a number of devices.
- Edit source programs and data with one of several fast and powerful editors.
- Write in the language best suited to a given task: BASIC, FORTRAN II, FORTRAN IV or PAL ASSEMBLER.
- Make full use of device independent I/O through simple, standard I/O calls.
- Interface to a number of devices through standard handlers, or write short handlers for unique devices.
- Develop programs on an OS/8 system, which can be saved as binaries or core images, then loaded into a smaller PDP-8 system.
- Control the system through a powerful and easy to use concise command language, which allows such combinations of actions as compilation, loading and executing to be specified in a single line.

PROGRAM	FUNCTION	BENEFIT
OS/8 SYSTEM PROGRAMS (Standard)		
BUILD (System Build)	Used for insertion and deletion of device handlers (system generation)	Allows fast convenient generation or reconfiguration of any OS/8 system.
PAL 8	8K, two pass, assembler	Expanded symbol table—allows for greater number of symbols to be defined and faster assemblies.
ABSLDR (Absolute Binary Loader)	Used to load PAL 8 binary output	Quickly loads binary program into memory for execution or storage of program in system library.
PIP (Peripheral Interchange Program)	Used to transfer files between devices	Data is available in a variety of media —flexibility.
EDIT (System Editor)	Used to create and modify source files	Reduces de-bug time.
DIRECT	Used to print extended directory listings	Provides information regarding the contents of files.
ODT	Octal debugging technique	Facilitates running prototype programs under controlled conditions.
OS/8 UTILITY PROGRAMS (Standard)		
BITMAP	Constructs table (MAP) showing memory locations used by files	Indicates number of times a particular memory location is referenced—aids debugging.
CREF	Cross-reference program pinpoints symbolic references	Aids writing, debugging and maintaining by providing an alphanumeric symbolic listing.
EPIC	Edit, punch and compare functions	Provides user with specified utility functions.
SRCCOM	Compares source files	Aids debugging by printing differences between source files.
SYSTEM PROGRAMS (Optional)		
BASIC	Interactive language	Exceptionally fast and easy to learn core efficient compiler.
BATCH	Used to provide BATCH processing monitor	Programs may be submitted during non-prime hours for execution in prescribed sequence.
TECO	Used to edit text and correct programs	Easily learned mnemonics afford full editing capabilities.
FORTRAN IV	Full ANSI FORTRAN IV	Programs can be written in industry standard language.

digital

DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts, Telephone: (617) 897-5111 • ARIZONA, Phoenix • CALIFORNIA, Sunnyvale, Santa Ana, Los Angeles, Oakland, San Diego and San Francisco (Mountain View) • COLORADO, Englewood • CONNECTICUT, Meriden, Fairfield • DISTRICT OF COLUMBIA, Washington (Lanham, Md.) • FLORIDA, Orlando • GEORGIA, Atlanta • ILLINOIS, Northbrook • INDIANA, Indianapolis • LOUISIANA, Metairie • MASSACHUSETTS, Marlborough and Waltham • MICHIGAN, Ann Arbor and Detroit (Southfield) • MINNESOTA, Minneapolis • MISSOURI, Kansas City and Maryland Heights • NEW JERSEY, Fairfield, Metuchen and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Huntington Station, Manhattan, New York, Syracuse and Rochester • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland, Dayton and Euclid • OKLAHOMA, Tulsa • OREGON, Portland • PENNSYLVANIA, Bluebell and Pittsburgh • TENNESSEE, Knoxville • TEXAS, Dallas and Houston • UTAH, Salt Lake City • WASHINGTON, Bellevue • WISCONSIN, Milwaukee • ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Canberra, Melbourne, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BRAZIL, Rio de Janeiro, Sao Paulo and Porto Alegre • CANADA, Calgary, Montreal, Ottawa, Toronto and Vancouver • CHILE, Santiago • DENMARK, Copenhagen • FINLAND, Helsinki • FRANCE, Grenoble and Paris • GERMANY, Berlin, Cologne, Hannover, Frankfurt, Munich and Stuttgart • INDIA, Bombay • ISRAEL, Tel Aviv • ITALY, Milan and Turin • JAPAN, Tokyo • MEXICO, Mexico City • NETHERLANDS, The Hague • NEW ZEALAND, Auckland • NORWAY, Oslo • PHILIPPINES, Manila • PUERTO RICO, Santurce • SPAIN, Barcelona and Madrid • SWEDEN, Stockholm • SWITZERLAND, Geneva and Zurich • UNITED KINGDOM, Birmingham, Bristol, Edinburgh, London, Manchester and Reading • VENEZUELA, Caracas

PRINTED IN USA 1901 00774 3871/Y 03 10