DEC VAX-11/780

MANAGEMENT SUMMARY

The VAX-11/780 is a multi-user, multiprogramming system using 32-bit processor and bus structure with an extensive instruction set and a virtual memory operating system. DEC states that the VAX-11/780 processor and the operating system were designed together, giving the 11/780 over a billion words of addressing space and the ability to simultaneously process time-sharing, time-critical, and batch applications plus on-line program development.

The VAX-11/780 is DEC's first offering using 32-bit architecture, described as an upward extension of the PDP-11 family architecture. (The larger DECsystem-10 and DECsystem-20 computers are 36-bit machines.) The major components of the VAX-11/780 system are the:

- Processor—including the basic CPU, synchronous system bus, intelligent microcomputer console, interval and time-of-year clocks, and 8K bytes of cache memory. Up to 8 million bytes (two million words) of MOS memory, two kinds of peripheral buses, and a floating-point accelerator can be included with the processor.
- Peripherals—including a range of small- and large-capacity disk drives, magnetic tape systems, hard-copy and video terminals, line printers, and card readers.
- Operating System—including a virtual memory manager, swapper, system services, device drivers, file system, record management services, command language, and operator's and system manager's tools. The system can support up to 96 interactive users simultaneously.
- Languages—including the VAX-11 MACRO assembly language and, optionally, VAX-11 FORTRAN IV-PLUS, PDP-11 BASIC-PLUS-2/VAX, and PDP-11 COBOL-74/VAX. Development tools for both native-mode and PDP-11 compatibility-mode programs include editors, linkers, librarians, and debuggers.

▷

The VAX-11/780 is Digital's first computer system using 32-bit architecture. The system supports up to 64 interactive users as well as multi-stream batch processing, and can have up to 2 million words of real, error-correcting MOS memory. The VAX/VMS virtual memory operating system features one-billion word virtual addressing and a full demand-paging operation. Prices for the basic VAX-11/780 system packages range from $134,600 to $207,000.

CHARACTERISTICS

MANUFACTURER: Digital Equipment Corporation (DEC), 146 Main Street, Maynard, Massachusetts 01754. Telephone (617) 897-5111.

DEC is a worldwide corporation and the world's largest manufacturer of minicomputer systems. The company employs about 23,000 persons and maintains sales and service offices in all major U.S. cities and in major cities throughout Canada and the Western world.

MODEL: VAX-11/780.

DATE ANNOUNCED: October 25, 1977.

DATA FORMATS

BASIC UNIT: 32-bit word.

FIXED-POINT OPERANDS: Integers can be 8-bit bytes, 16-bit words, 32-bit longwords, and 64-bit quadwords. All have the same general format, with the high-order bit used as the sign. Negative numbers are represented in two's complement form.

FLOATING-POINT OPERANDS: Two floating-point formats are available: single-precision (called floating) that uses a 4-byte format, and double-precision (called double floating) that uses an 8-byte format. In both formats, the high-order bit is used as a sign and the next seven bits for the exponent. Single-precision fractions are 24 bits long, while double-precision fractions are 56 bits long. The 4-byte

►

The VAX-11/780 32-bit computer system supports up to 96 interactive users as well as multi-stream batch processing. The VAX/VMS virtual memory operating system features a full demand-paging operation that allows addressing space of up to one billion words. The 32-bit word length system supports up to 2 million words of real, error-correcting MOS memory.

- Network Services—including the DECnet/VAX network software and the DMC11 interprocessor communications link.

The VAX-11/780 processor provides 32-bit addressing, sixteen 32-bit general registers, and 32 interrupt priority levels. The instruction set operates on integer and floating-point operands, character and packed decimal strings, and bit fields, and supports nine fundamental addressing modes. The processor includes an 8K-byte write-through cache memory that results in an effective 290-nanosecond memory access time, and also a 12K byte Writable Diagnostic Control Store for diagnostic software. The processor's memory management includes four hierarchical processor access modes that are used by the operating system to provide read/write page protection between user software and system software.

Error-correcting code (ECC) MOS memory is connected to the main control and data transfer path (called the Synchronous Backplane Interconnect or SBI) via a memory controller. Physical memory is built using 4K-bit MOS RAM chips and is organized in 72-bit words. (64 bits for data and 8 for ECC). Each memory controller includes a request buffer that substantially increases overall system throughput and eliminates the need for interleaving in most applications.

The processor uses two standard clocks—a programmable real-time clock used by the operating system and diagnostics, and a time-of-year clock used for system operations. The time-of-year clock includes battery backup for automatic system restart operations.

The processor's console consists of an LSI-11 microcomputer with 16K bytes of read/write memory and 8K bytes of ROM, a floppy disk, and a terminal for local operations and an LSI-11 microcomputer for remote diagnosis. The console operator uses keyboard commands for diagnosis, bootstrapping, and incorporating software maintenance modifications.

Medium-capacity disk drives, unit record devices, terminals, interprocessor communications links, and user-specific devices are Unibus peripherals. The Unibus adapter provides the hardware pathways for data and control information to move between the Unibus and the SBI. The maximum aggregate throughput rate is 13.3 million bytes per second.

High-performance Massbus mass storage peripherals are connected to the SBI via a buffered Massbus adapter. The Massbus adapter provides the hardware path for data and control information to move between a Massbus peripheral controller and the SBI, and allows high-speed data transfers at a maximum aggregate throughput rate of 2 million bytes per second for each adapter. The Massbus adapter performs parity checking on both data and control information.

The processor includes 12K bytes of writable diagnostic control storage for updating the instruction set micro-

format provides approximately 7 decimal digits of precision, while the 8-byte format provides approximately 16 decimal digits of precision.

INSTRUCTIONS: The native instruction set is an extension of the PDP-11 instruction set that consists of 244 basic instructions, most of which can be applied to any one of several types of data, which can in turn be addressed in any one of nine ways. The native instruction set provides 32-bit addressing, 32-bit I/O operations, and 32-bit arithmetic. The instructions can be grouped into related classes based on their function and use: instructions to manipulate arithmetic and logical data types, instructions to manipulate special kinds of data, instructions to provide basic program flow control, instructions to perform special operating system functions, and instructions provided specifically for high-level language constructs.

Instructions and data are available in length. They need not be aligned on longword (32-bit) boundaries in physical memory, but may begin at any byte address (odd or even). Thus, instructions that do not require arguments use only one byte, while other instructions may be two, three, or up to 30 bytes in length, depending on the number of arguments and their addressing modes.

In addition to its 32-bit native instruction set, the processor can concurrently execute a compatibility-mode instruction set, which is a subset of the DEC PDP-11 instruction set. This is not done by emulation or simulation; both instruction sets are built into the microcode and logic of the processor. The compatibility-mode instruction set contains all the PDP-11 instructions except those which perform the following functions:

- **Execution of floating-point instructions.**
- **Use of both instruction space and data space.**
- **Execution of privileged functions such as: 1) HALT, RESET and special instructions, such as traps and WAIT, which are normally reserved for operating system usage; 2) direct access to internal processor registers such as the Processor Status Word and the Console Switch Register; 3) direct access to the trap and interrupt vectors, which must be initialized for interrupt servicing; and 4) execution in any mode other than user mode, along with the corresponding access to the alternate general register set.**

INTERNAL CODE: ASCII for text-oriented data; binary for calculations.

MAIN STORAGE

GENERAL: The main memory system is connected to the Synchronous Backplane Interconnect (SBI) via the memory controller. Physical memory is built using 4K MOS RAM chips. (The newer MS 780 memory uses 16K-bit chips.) Both memory types are organized in quadwords (64 bits) plus an 8-bit error-correcting code (ECC), which allows the correction of all single-bit errors and the detection of all double-bit errors and approximately 70 percent of errors which exceed 2 bits. Interleaving is possible with two controllers and equal amounts of memory on each. Interleaving is enabled/disabled under program control. It is performed at the quadword level because of the memory organization. The memory controllers allow the writing of data in full 32- and 64-bit units. Also, upon command from an SBI device, individual bytes (or a single byte) may be written.

Each memory controller buffers up to four memory access requests. This "request buffer" substantially increases memory throughput and overall system throughput and decreases the need for interleaving for most configurations. With this

PERIPHERALS/TERMINALS

DEVICE	DESCRIPTION	MANUFACTURER
MAGNETIC TAPE		
TEE16-A	Magnetic tape transport and Massbus adapter; expandable to a total of eight TE16-A transports; 800 or 1600 bpi (program selectable), 9-track, 45 ips, industry compatible; 36 or 72 KBS	DEC
TEU45/77	Magnetic tape transport and Massbus adapter; expandable to a total of eight transports; 800 or 1600 bpi (program selectable), 9-track 75/125 ips, industry compatible	DEC
LINE PRINTERS		
LA11-PA	Serial printer with controller, 132-position, 128-character, 7 x 7 dot matrix; 180 cps	DEC
LP11-CA	132-position, 64-character line printer, controller; 900 lpm	—
LP11-DA	132-position, 96-character line printer, controller; 660 lpm	—
LP11-RA	132-position, 64-character line printer, controller; 1250 lpm	Dataproducts
LP11-SA	132-position, 96-character line printer, controller; 925 lpm	Dataproducts
LP11-VA/ LPR11-VA	132-position, 64-character line printer, controller; 300 lpm/250 lpm	Dataproducts
LP11-WA	132-position, 96-character line printer, controller; 240 lpm	—
LP11-YA/ LPR11-YA	132-position, 64-character line printer, controller; 600 lpm/525 lpm	—
LP11-ZA/ LPR11-ZA	132-position, 96-character line printer, controller; 436 lpm/420 lpm	—
CARD UNITS		
CR11	Reader, 80-column, tabletop, controller; 300 cpm	Documation
CR11-B	Reader, 80-column, controller; 600 cpm	Documation
TERMINALS		
LA36	DECwriter II, 132-position, 96-character, 7 x 7 dot matrix; 30 cps	DEC
LA120	DECwriter III; 132 columns; 96-character ASCII keyboard; DEC 180 cps	DEC
LA38-GA	DECwriter IV; 132 columns; 96-character ASCII keyboard; 4 character sizes, six line spacings	DEC
VT52	DECscope, 24 lines x 80 characters, 128-character set; 75 to 9600 bps	DEC
VT100	DECscope, 24 lines x 80 characters or 14 lines x 132 characters, 96-character set	DEC

▷ code. The control storage is also used for executing microcode diagnostics, which can be loaded from the console's floppy disk.

The instruction set that the processor executes is selected under operating system control as either the native-mode or compatibility-mode set. The native-mode instruction set includes over 200 different opcodes, which can be grouped into classes based on their function and use. Instructions used to manipulate the general data types include integer and floating point-instructions, packed decimal instructions, character string instructions, and bit field instructions. Instructions used to manipulate special kinds of data include queue manipulation, address manipulation, and user-programmed general register control instructions. Instructions that provide basic program flow control and call procedures include branch, jump, and case instructions, subroutine call instructions, and procedure call instructions.

The compatibility mode provides the PDP-11 instruction set, with the exception of privileged and floating-point instructions. Under control of the operating system, the ▷

➤ buffer, memory bandwidth essentially matches that of the SBI—13.3 million bytes/second, including time for refresh cycles. This is because a number of transactions can occur concurrently. For example, the memory controller can accept a WRITE command from a MASSBUS adapter while it is reading previously requested data by the processor for increased throughput. Were it not for the request buffer, there would be about a 50 percent degradation in memory bandwidth, making interleaving necessary to approach the bandwidth.

TYPE: Error-correcting MOS.

CYCLE TIME: 600 nanoseconds. The VAX-11/780 processor includes an 8K-byte write-through memory cache that results in an effective 290-nanosecond memory access time.

CAPACITY: The minimum memory requirement is 128K bytes. Memory may be added in increments of 128K bytes to a maximum of one million bytes per controller. Two memory controllers may be connected to a system, for a total of two million bytes of physical memory.

CONTROL STORAGE: 12K bytes (plus parity) of Writable Diagnostic Control Store (WDCS) are provided to allow the Diagnostic Console Microcomputer to verify the integrity of crucial parts of the CPU, the intelligent console, the ➤

processor can execute PDP-11 instruction streams within the context of any process. When executing in compatibility mode, the processor interprets the instruction stream as a subset of PDP-11 code that does not include floating-point hardware instructions or privileged instructions. The compatibility mode enables the VAX/VMS operating system to provide an environment for executing most user-mode programs written for a PDP-11 (except stand-alone software). The processor expects all compatibility-mode software to rely on the services of the native operating system for I/O processing, interrupt and exception handling, and memory management. There are some restrictions, however, on the environment that the native operating system can provide for a PDP-11 program. For example, certain PDP-11 memory management instructions cannot be simulated by the operating system since they do not trap to native-mode software.

Like the PDP-11, the VAX-11/780 uses both the DCL and MCR command languages and implements the same FORTRAN-IV-PLUS, BASIC-PLUS-2, and COBOL languages. The FORTRAN compiler generates native 32-bit code on the VAX-11/780, and can concurrently execute a subset of the PDP-11 instruction set in its compatibility mode. The COBOL and BASIC language processors produce compatibility-mode code.

The VAX system can also be used as a host development system for RSX-11M and RSX-11S operating systems running on PDP-11 minicomputers. Like the PDP-11, the new system uses a Unibus for connecting peripherals; and like the PDP-11/70, it uses integrated Massbus adapters for interfacing high-speed peripherals. The disk structure is the same as that of the PDP-11 RSX-11 Real-Time Operating Systems and IAC (Interactive Application System), and the file access methods used Datatrieve/VAX are the same as those available with Datatrieve 11, the PDP-11 inquiry and report writing system.

The operating system for the VAX-11/780 is VAX/VMS, a general-purpose operating system that provides for the concurrent execution of multi-user time-sharing, batch, and time-critical applications. VAX/VMS provides: 1) virtual memory management for the execution of large programs; 2) event-driven priority scheduling; 3) shared memory, file, and interprocess communication data protection based on ownership and application groups; and 4) programmed system services for process and subprocess control and interprocess communication.

VAX/VMS performs process-oriented paging, which allows the execution of programs larger than the physical memory allocated to them. Four billion bytes of addressing space are provided. Paging is handled automatically by the system, freeing the user from any need to structure the program. In the VAX/VMS operating system, a process pages only against itself; thus, individual processes cannot significantly degrade the performance of other processes.

The memory management facilities provided by VAX/VMS can be controlled by the user. Any program, with

SBI, and the memory controller. In addition, the WDCS can be used to implement updates to the system microcode.

STORAGE PROTECTION: The system's memory management logic divides memory into 512-byte pages. Each page is assigned a protection code specifying which, if any, access modes are to be permitted read or write access to the page.

CENTRAL PROCESSOR

GENERAL: The VAX-11/780 processor has a 32-bit architecture based on the DEC PDP-11 family of 16-bit minicomputers. While using address modes and stack structures similar to those of the PDP-11, the VAX-11/780 provides 32-bit addressing for a large program address space, and 32-bit arithmetic and data paths for increased processing speed and accuracy. The processor includes the basic CPU, synchronous system bus, intelligent microcomputer console, interval and time-of-year clocks, and 8K bytes of cache memory. Up to two million bytes of memory, up to four Massbuses, a Unibus, and a floating-point accelerator can be included with the processor. The processor provides 32-bit addressing, sixteen 32-bit general registers, and 32 interrupt priority levels. The instruction set operates on integer and floating-point operands, character and packed decimal strings, and bit field data. The instruction set supports nine fundamental addressing modes.

The processor's memory management includes four hierarchical processor access modes that are used by the system to provide read/write page protection between user software and system software. Memory is connected to the main control and data transfer path (the SBI) via a memory controller. Each memory controller includes a request buffer that substantially increases overall system throughput and eliminates the need for interleaving in most applications.

The processor uses two standard clocks: a programmable real-time clock used by the operating system and by diagnostics, and a time-of-year clock used for system operations. The time-of-year clock includes battery backup for automatic system restart operations.

The "intelligent" console consists of an LSI-11 microcomputer with 16K bytes of read/write memory and 8K bytes of ROM, a floppy disk unit, a terminal for local operations, and an optional port for remote diagnosis. The console operator uses keyboard commands for diagnosis, bootstrapping, and incorporating software maintenance modifications.

Medium-capacity disks, unit record devices, terminals, interprocessor communications links, and user-specific devices are Unibus peripherals. The maximum aggregate throughput rate for Unibus peripherals is 13.3 million bytes per second.

High-performance mass storage peripherals are connected to the SBI via up to 4 buffered Massbus adapters. The Massbus adapters provide the hardware pathways for data and control information to move between Massbus peripheral controllers and the SBI, and allow high-speed data transfers at a maximum aggregate throughput rate of two million bytes per second for each adapter. The Massbus adapter does parity checking for both data and control information.

REGISTERS: The VAX-11/780 provides sixteen 32-bit general registers that can be used for temporary storage, as accumulators, as index registers, and as base registers. The processor offers a variety of addressing modes that use the general registers to identify instruction operand locations, including an indexed addressing mode that provides a true post-indexing capability.

DEC VAX-11/780

sufficient privilege, can prevent pages from being swapped out, or prevent the entire working set from being swapped out, to optimize program performance in time-critical or interactive environments. Sharing and protection are provided for individual 512-byte pages. Four hierarchical modes (kernel, executive, supervisor, and user) provide page protection.

VAX/VMS schedules CPU time and memory residency on a preemptive priority basis. Thus, time-critical processes do not have to compete with lower-priority processes. The scheduler adjusts the priorities of processes assigned one of the low 16 priorities to overlap I/O and computation. Time-critical processes can be placed in one of the top 16 scheduling priorities, in which case the scheduler does not alter their priorities, but they can still be altered by the system manager or an appropriately privileged user.

The operating system also includes system services to control processes and process execution, control time-critical response, control scheduling, and obtain information. Process control services allow the creation of subprocesses as well as independent detached processes. Processes can communicate and synchronize using mailboxes, shared areas of memory, or shared files. A group of processes can also communicate and synchronize using multiple common-event flag clusters.

VAX/VMS provides a program development capability that includes editors, language processors, and a symbolic debugger. The BLISS-32 implementation language was designed for the development of operating systems, data base file systems, communications software, and utilities. The VAX-11 FORTRAN IV-PLUS and BLISS-32 language processors produce native code, whereas the PDP-11 COBOL-74/VAX and PDP-11 BASIC-PLUS-2/VAX language processors produce compatibility-mode code.

The VAX/VMS operating system provides a file and record management facility that allows the user to create, access, and maintain data files and records within the files with full protection. The record management services handle sequential and relative file organizations, sequential and random record access, and fixed and variable-length records. Indexed files with sequential and random record access are available to compatibility-mode programs, such as those written in PDP-11 COBOL-74/VAX or PDP-11 BASIC-PLUS-2/VAX.

Data communications capabilities for the VAX-11/780 are provided by DECnet, a family of network products developed by DEC that add networking capabilities to all of the company's computer families and operating systems. Using DECnet, various kinds of computer networks can be constructed to facilitate remote communications, resource sharing, and distributed computation.

DEC's Digital Network Architecture (DNA) provides the common network structure upon which all DECnet products are built. The architecture is designed to handle

Four registers have special significance: the Program Counter contains the address of the next instruction to be executed; the Stack Pointer contains the address of the base (or top) of a stack maintained for subroutine and procedure calls; the Frame Pointer contains the address of the base of a software data structure stored on the stack and called the stack frame, which is maintained for procedure calls; and the Argument Pointer contains the address of the base of a software data structure called the argument list, which is maintained for procedure calls.

In addition, the first six registers have special significance for character and packed decimal string instructions and the Polynomial Evaluation instruction. These instructions use the first six registers to store temporary results and, upon completion, leave results in the registers that a program can use as the operands of subsequent instructions.

A register's special significance does not preclude its use for other purposes, except for the Program Counter. The Program Counter cannot be used as an accumulator, as a temporary register, or as an index register. In general, however, most users do not use the Stack Pointer, Argument Pointer, or Frame Pointer for purposes other than those designated.

Registers can be used for temporary storage, accumulators, base registers, and index registers. A base register contains the address of the base of a software data structure such as a table or queue, and an index register contains a logical offset into a data structure. Whenever a register is used to contain data, the data is stored in the register in the same format as it would appear in memory. If a quadword or double floating operand is stored in a register, it is actually stored in two adjacent registers.

ADDRESSING: The processor's addressing modes allow almost any operand to be in a register or in memory, or used as an immediate constant. There are seven basic addressing modes that use the general registers to identify the operand location, including:

- **Register Mode, in which the register contains the operand.**
- **Register Deferred Mode, in which the register contains the address of the operand.**
- **Autodecrement Mode, in which the contents of the register are first decremented by the size of the operand, and then used as the address of the operand. The size of the operand (in bytes) is given by the data type of the instruction operand, and depends on the instruction.**
- **Autoincrement Mode, in which the contents of the register are used as the address of the operand, and then incremented by the size of the operand. If the Program Counter is the specified register, the mode is called the Immediate mode.**
- **Autoincrement Deferred Mode, in which the contents of the register are used as the address of a location in memory containing the address of the operand, and then are incremented by four (the size of an address). If the Program Counter is the specified register, the mode is called the Absolute mode.**
- **Displacement Mode, in which the value stored in the register is used as a base address. A byte, word, or longword signed constant is added to the base address, and the resulting sum is the effective address of the operand.**
- **Displacement Deferred Mode, in which the value stored in the register is used as the base address of a table of addresses. A byte, word, or longword signed constant is added to the base address, and the resulting sum is**

a broad range of application requirements because all the functions of the network—from the user interface to physical link control—are completely modular. DNA allows nodes to operate as switches, front ends, terminal concentrators, or hosts.

DECnet/VAX is designed to:

- Provide an interprocess communication facility that is highly transparent and easy to use.
- Provide a high-level language programming interface.
- Allow programs to access files at other systems.
- Allow users and programs to transfer files between systems;
- Allow users to transmit command files to be executed in other systems; and
- Allow an operator to down-line load RSX-11S system images into other systems.

The VAX-11/780 clearly is an impressive system, and in view of the fact that there are more than 50,000 PDP-11's installed and each user is a potential VAX customer, there is no doubt that DEC will be able to sell it. As DEC has done in the past, it will initially direct the VAX system into scientific, industrial, and educational markets where time-critical, computational, and control applications are important. There, DEC will compete with such established producers of 32-bit systems as Interdata and Systems Engineering Laboratories.

USER REACTION

Early returns from Datapro's 1980 user survey include responses from five DEC VAX-11/780 users with 11 systems. The average system has been installed for 16 months. Ten were purchased and one was leased. All five users listed engineering and scientific applications with education, word processing, distributed processing, and interactive graphics also mentioned. In-house programmers were the primary source of applications programs. Memory capacities averaged 1.4 megabytes and disk storage averaged 270 megabytes. The number of work stations per system ranged from 12 to 32.

FORTRAN was the primary programming language at all of the sites, with PASCAL and BASIC also mentioned. All of the users plan to expand their data communications during the current year. There was only one specific complaint, and three advantages—good response time, system easy to expand and reconfigure, and good productivity aids—were mentioned by all five users. Hardware and software compatibility with other DEC systems was also mentioned as an advantage.

The table below shows the ratings given for the VAX-11/780. One user declined to give specific ratings since the system had only been installed October 1979.

the address of the location that contains the actual address of the operand.

Of these seven basic modes, all except Register Mode can be modified by an index register. When an index register is used with a basic mode to identify an operand, the addressing mode is the name of the basic mode with the suffix "Indexed." Therefore, in addition to the seven basic addressing modes that use registers, the processor recognizes six indexed addressing modes.

The processor also provides Literal Mode addressing, in which an unsigned 6-bit field in the instruction is interpreted as an integer or floating-point constant.

INSTRUCTION REPERTOIRE: **The native-mode instruction set that the processor executes is based on over 200 different opcodes. The opcodes can be grouped into classes based on their function and use. Instructions used to manipulate the general data types include:**

- **Integer and floating-point instructions.**
- **Packed decimal instructions.**
- **Character string instructions.**
- **Bit field instructions.**

Instructions that are used to manipulate special kinds of data include:

- **Queue manipulation instructions.**
- **Address manipulation instructions.**
- **User-programmed general register control instructions.**

Instructions that provide basic program flow control and permit the calling of procedures are:

- **Branch, jump, and case instructions. (The processor provides a branch instruction, CASE, that implements higher-level language computed GO TO statements. For CASE, the user supplies a list of displacements that generate different branch addresses indexed by the value obtained as a selector. The branch falls through if the selector does not fall within the limits of the list.)**
- **Subroutine call instructions.**
- **Procedure call instructions.**

INSTRUCTION TIMINGS: **All times shown are for longword (32-bit) operands, in microseconds. Floating-point instruction times are for systems that include the Floating-Point Accelerator.**

	Fixed Point	Floating Point
Add/Subtract	0.4	0.8
Multiply	1.6	1.2
Divide	9.8	4.2

INTERRUPTS: **The processor recognizes 32 interrupt priority levels. The highest 16 interrupt priority levels are reserved for interrupts generated by hardware, and the lowest 16 levels are reserved for interrupts requested by software. Normal user software runs at the process level, which is interrupt priority level zero.**

To handle interrupt requests, the processor enters a special system-wide context. In the system-wide context, the processor executes in kernel mode, using a special stack called the interrupt stack. The interrupt stack cannot be

	Excellent	Good	Fair	Poor	WA*
Ease of operation	1	3	0	0	3.3
Reliability of mainframe	2	2	0	0	3.5
Reliability of peripherals	0	3	0	0	3.0
Maintenance service:					
Responsiveness	1	3	0	0	3.3
Effectiveness	1	2	1	0	3.0
Technical support:					
Trouble-shooting	0	2	2	0	2.5
Education	0	1	2	0	2.3
Documentation	1	3	0	0	3.3
Manufacturer's Software:					
Operating system	2	2	0	0	3.5
Compilers & assemblers	2	2	0	0	3.5
Applications programs	0	3	0	0	3.0
Ease of programming	2	2	0	0	3.5
Ease of conversion	2	2	0	0	3.5
Overall satisfaction	1	3	0	0	3.3

*Weighted Average on a scale of 4.0 for Excellent.

These users were obviously pleased with most aspects of their VAX-11/780 systems, the exception being DEC's technical support, specifically their trouble shooting and education programs. When asked to cite specific system advantages, all four of these users said that they were happy with the system's response time and that it was easy to expand or reconfigure. Three users were impressed with the programming aids available with the VAX, and felt that they helped to keep programming costs down, and two were impressed with the fact that they could use terminals and other peripherals previously used on other systems. One user also indicated that he felt the system was power/energy efficient.

When asked about any problems they encountered with the VAX system, only two users responded. One said that some equipment he ordered was late in being delivered, and another said that the program/data compatibility was not what DEC had promised.

Based on the reaction of these users, it appears that DEC did its homework relative to the VAX-11/780. Perhaps the most impressive aspect of the survey is that all four of the users said that they would recommend the VAX-11/780 to other users in their situation.□

referenced by any user-mode software because the processor selects the interrupt stack only after an interrupt, and all interrupts are trapped through system vectors.

The interrupt service routine executes at the interrupt priority level of the interrupt request. When the processor receives an interrupt request at a level higher than that of the currently executing software, the processor honors the request and services the new interrupt at its priority level. When the interrupt service routine issues the REI (Return from Exception or Interrupt) instruction, the processor returns control to the previous level.

PHYSICAL SPECIFICATIONS: Nominal operating environment for the VAX-11/780 processor is 59 to 90 degrees Fahrenheit (15 to 32 degrees Centigrade), at 20 to 80 percent relative humidity within specified wet-bulb and dew-point limits. These are *processor* specifications; electromechanical peripherals may be more sensitive to their environments. The CPU is housed in a five-foot-tall, four-foot-wide, double-width highboy cabinet.

INPUT/OUTPUT CONTROL

UNIBUS: General-purpose and customer-developed devices are connected to the VAX-11/780 system via the Unibus. Since the SBI deals in 30-bit addresses, 18-bit Unibus addresses must be translated to 30-bit SBI addresses. This mapping function is performed by the Unibus adapter, a special interface between the SBI and the Unibus, which translates Unibus addresses, data, and interrupt requests to their SBI equivalents, and vice versa. The Unibus adapter does priority arbitration among devices on the Unibus—a function that is handled by logic in the PDP-11 CPU's. The address translation map permits contiguous disk transfers to and from noncontiguous pages of physical memory. (These are called scatter/gather operations.) The Unibus adapter allows two kinds of data transfers: program interrupt and direct memory access.

To make the most efficient use of the SBI bandwidth, the Unibus adapter facilitates high-speed DMA transfers by providing buffered DMA data paths for up to 15 high-speed devices. Each of these channels has a 64-bit buffer (plus byte parity) for holding four 16-bit transfers to and from Unibus devices. The result is that only one SBI transfer (64 bits) is required for every four Unibus transfers. The maximum aggregate data transfer rate through the buffered data paths is 1.5 million bytes/second. In addition, on SBI-to-Unibus transfers, the Unibus adapter anticipates upcoming Unibus requests by pre-fetching the next 64-bit quadword from memory as the last 16-bit word is transferred from the buffer to the Unibus. The result is increased performance. By the time the Unibus device requests the next word, the Unibus adapter has it ready to transfer.

Any number of unbuffered DMA transfers are handled by one direct DMA data path. Every 8- or 16-bit transfer on the Unibus requires a 32-bit transfer on the SBI (although only 16 bits are used). The maximum transfer rate through the direct data path is 750 thousand bytes/second.

The Unibus adapter permits concurrent program interrupt, unbuffered and buffered data transfers. The aggregate throughput rate of the direct data path plus the 15 buffered data paths is 13.3 million bytes/second.

MASSBUS: High-performance peripheral devices, such as disk and tape units, are connected to the VAX-11/780 system using a Massbus adapter. The Massbus adapter is the interface between the Massbus and the SBI and performs all control, arbitration, and buffering functions. Address mapping is similar to that performed by the Unibus adapter. Up to four Massbus adapters can be used in each VAX-11/780 system. Each adapter can accommodate data transfers of up to 128K bytes to and from noncontiguous pages in physical memory (scatter/gather). The VAX/VMS operating system supports transfers of 65K bytes maximum to be consistent with other devices.

Each Massbus adapter uses a 32-byte silo data buffer, which permits transfers at rates of up to 2 million bytes/second to and from physical memory (8 million bytes/second with all 4 Massbus adapters). As in the Unibus adapter, data is assembled in 64-bit quadwords (plus byte parity) to make maximum efficient use of the SBI bandwidth.

On memory-to-Massbus transfers, as on memory-to-Unibus transfers, the adapter anticipates upcoming Massbus data transfers by pre-fetching the next 64 bits of data from memory.

The combination of Unibus and Massbus transfer rates provide a maximum throughput of 9.5 million bytes/second to and from the SBI. Thus, there is ample bandwidth remaining (3.8 million bytes/second) to handle the CPU, which typically uses 1 million bytes/second.

CONFIGURATION RULES

The VAX-11/780 contains the CPU, memory management, bootstrap loader, standard instructions, 8K-byte parity bipolar cache memory, programmable real-time clock, time-of-year clock (with battery backup), and 12K bytes of writable diagnostic control store.

Also included as standard equipment is an integral diagnostic console subsystem, for use in both local and remote operations, which consists of a microcomputer (LSI-11 with 16K bytes of read/write memory and 8K bytes of read-only memory) to which an RX01 floppy disk unit and an LA36 DECwriter are connected.

Optional expansions of the VAX-11/780 can be made through the use of the CPU, Unibus, or VAX-11/780 expansion cabinets. Options mounted in the CPU cabinet include a high-performance Floating-Point Accelerator with power supply, 12K bytes of writable control store, an additional 896K bytes of error-correcting memory, memory battery backup for up to one megabyte of memory, a serial line unit for remote diagnosis, and up to two Massbus adapters. The VAX-11/780 expansion cabinet offers expansion for up to one megabyte of memory with control, up to two more Massbus adapters, and one memory backup unit.

The VAX-11/780 system supports high-performance mass storage devices for on-line data retrieval, unit record equipment for data processing, terminals and line interfaces for interactive users, and a line interface for interprocessor communications.

As many as four Massbus adapters, each of which can support up to eight disk drives or magnetic tape controllers, can be connected to the system. In addition, up to eight medium-capacity disk drives can be connected to the system's Unibus.

Card readers and line printers can be spooled input and output devices managed by operator-controlled queues. The LP11 and LA11 series line printers provide a wide range of models to choose from. Up to two CR11 card readers, four LP11 line printers, and 16 LA11 printers can be attached to the system.

The system supports full-duplex communications with both hardcopy and video terminals, and can support up to 96 terminal lines. The DMC-11 serial synchronous communications line provides high-performance point-to-point interprocessor connection using the DEC Data Communications Message Protocol (DDCMP).

All equipment is integrated with the systems software and is supported by both on-line error logging and diagnostics. Each component includes error checking and correction features. The software provides power failure and error recovery algorithms.

MASS STORAGE

RK611 CARTRIDGE DISK SUBSYSTEM: Includes single- or dual-access 14-million-byte RK06 cartridge disk drive and control unit. Expandable to 8 RK06 disk drives. These drives use a removable data cartridge with three recording surfaces and 411 tracks per surface. Average rotational delay is 12.5 milliseconds, and average access time is 38 milliseconds. Data transfer rate is 538,000 bytes per second.

RK711 CARTRIDGE DISK SUBSYSTEM: Includes single- or dual-access 28-million-byte RK07 cartridge disk drive and control unit. Expandable to 8 RK06 or RK07 cartridge disk drives. Average access time is 49 milliseconds. Data transfer rate is 538,000 bytes per second.

REM03 MOVING-HEAD DISK SUBSYSTEM: Includes single- or dual-access 67-million-byte RM03 removable disk pack drive and Massbus adapter. Expandable to a total of 8 RM03 drives. Average rotational delay is 12.5 milliseconds, and average access time is 38.3 milliseconds. Data transfer rate is 1.2 million bytes per second.

REP06 MOVING-HEAD DISK SUBSYSTEMS: Includes single- or dual-access 176-million-byte RP06 removable disk pack drive and Massbus adapter. Expandable to a total of 8 drives. The RP06 drives use IBM 3336-type disk packs, recording data on 19 or 20 surfaces. Data in the RP06 is organized on 815 tracks per surface, 22 sectors per track, and 512 bytes per sector. The RP06 is a double-density version of the older RP05 drive and uses the IBM 3336 Model 11-type disk pack.

Average rotational delay is 8.3 milliseconds, and average head positioning time is 28 milliseconds. Data transfer rate is 860K bytes per second (2.5 milliseconds per 16-bit word).

INPUT/OUTPUT UNITS

Please refer to the Peripherals/Terminals table on page M11-384-403 for information on the DEC peripheral equipment for the VAX-11/780 system.

DATA COMMUNICATIONS

DEC offers a number of interface controllers for the VAX-11/780 system. A VAX-11/780 can be connected to almost any type of communications channel (private phone, dial-up phone, 20-ma line, telegraph line), terminal, or modem. Supplementing these interfaces is additional data communications hardware to provide flexibility in unique situations.

There are six variations of the *DZ11 Asynchronous Multiplexer* available with VAX-11/780 systems. Three variations are designed for EIA/CCITT terminals or lines, and the other three are for 20-ma current loop terminals or lines. The DZ11 provides control for up to 16 asynchronous terminal devices or 16 full- or half-duplex lines. Each line can be individually programmed through software control for one of 15 line speeds between 50 and 9600 bps. The DZ11 includes modem controls to operate a Bell 103, 113, or equivalent 300-bps data set, or a Bell 212 at up to 1200 bps. Interrupts can be programmed to occur for each character or after 16 characters. The DZ11 is generally transparent to data, but can report parity errors and framing errors. Input characters are buffered with identification hardware in a first-in/first-out (FIFO) buffer, or "silo" (in DEC terms). Up to 16 DZ11's can be used in a system.

The DMC11 *Network Link* is designed for high-performance point-to-point interprocessor connection based on the Digital Data Communications Message Protocol (DDCMP). The DMC11 provides local or remote interconnection of two computers over a serial synchronous linek. Both computers can include the DMC11 and DECnet software, or both computers can use the DMC11 and implement their own communications software. For remote operations, a DMC11 can also communicate with a different type of synchronous interface, provided that the remote system has implemented the DDCMP protocol.

The DMC11 supports full- or half-duplex operation. Full-duplex operation offers the highest throughput and is used when the communications facilities permit two-way operation. The DDCMP protocol permits continuous simultaneous transmission of data messages in both directions when buffers are available and there are no errors on the channels.

When both computers are located in the same facility, the DMC11 permits transmission at speeds up to 1,000,000 bps over coaxial cable up to 6,000 feet long, or speeds

SOFTWARE STORAGE REQUIREMENTS

	Main Memory		Disk Storage	
	Minimum	Typical	Minimum	Typical
FORTRAN IV-PLUS	50KB	75KB	350KB	550KB
COBOL-74/VAX	65KB	65KB	630KB	860KB
PDP-11 BASIC-PLUS-2/VAX	65KB	65KB	365KB	510KB

up to 56,000 bps over coaxial cable up to 18,000 feet long. The necessary modems for local interconnection are built in. Where the computers are located remotely and connected using common-carrier facilities, the DMC11 permits transmission at up to 19,200 bps using an EIA interface. A DMC11 can interface with synchronous modems such as the Bell 208 and 209, or with other synchronous modems conforming to the RS-232C standard.

COMMUNICATIONS CONTROL

DECnet is a family of network products that add networking capability to all of DEC's computer families, including the VAX-11/780. Using DECnet, various kinds of computer system networks can be constructed to facilitate remote communications, resource sharing, and distributed computation. DECnet is highly modular and flexible, and enables the user to select the appropriate hardware and software to build a network that satisfies a particular application's requirements.

DEC's Digital Network Architecture (DNA) provides the common network structure upon which all DECnet products are built. DNA is designed to handle a broad range of application requirements. All the functions of the network—from the user interface to physical link control—are completely modular. DNA allows nodes to operate as switches, front ends, terminal concentrators, or hosts.

Using DECnet, each node of the network has both common DECnet attributes and system-specific attributes. The attributes provided by DECnet/VAX include:

- Interprocess (task-to-task) communication. Programs executing on one system can converse with programs executing on other systems.
- Inter-system file transfer. A program or a command language user can transfer an entire data file from one system to another.
- Inter-system resource sharing. Programs executing on one system can access files physically located at other systems in the network. Access to devices in other systems is provided only through the file system of the target node and is subject to that system's Data Access Protocol (DAP) implementation or file system restrictions.
- Down-line system loading. Initial load images for RSX-11S systems in the network can be stored on the host VAX-11/780 system and loaded on request into PDP-11 systems configured for the RSX-11S operating system.
- Down-line command file loading. Programs or command language users can send command files to a remote node to be executed there. However, no status information or error messages are returned.
- High-level language interface. This facility allows programs written in any VAX-11/780 native programming language to access some of the network facilities.

Some of the DECnet protocols and their functions are as follows:

- Digital Data Communications Message Protocol (DDCMP) handles the physical link control and error recovery within DECnet. DDCMP has been designed to operate with existing hardware interfaces over full- and half-duplex facilities.
- Network Services Protocol (NSP) handles network management functions within DECnet. This is the protocol that allows interprocess communication within DECnet.
- Data Access Protocol (DAP) enables programs on one node of the network to use the file system services available on other network nodes. Each operating system in DECnet provides facilities for translating its own unique I/O structure into the DAP standard, and vice versa.
- Network Information Control Exchange Protocol (NICE) enables the user to control the status of a network node, and enables the nodes to exchange status information.
- Maintenance Operation Protocol (MDP) permits down-line loading and some loop-back tests.

Interprocess communication is the one feature common to all DECnet implementations. Interprocess communication allows programs (tasks, processes, etc.) to create one or more logical links, which are full-duplex virtual data paths. Programs have the capability to create these connections, transmit and receive data over them, and destroy them. Data transmission can be done on a normal or priority (i.e., interrupt) basis.

In DECnet/VAX, the interface itself can be transparent (each program looks like a sequential device to the other), or non-transparent (each program knows that it is using DECnet and has the opportunity to acquire information about the network). Each access method has its advantages. Transparent access is easier to learn, and it allows great flexibility in that the location of files, devices, and the program itself need not be determined until run time. Under non-transparent access, the programmer can take advantage of known properties of the protocols, providing the ability to transmit and receive interrupt messages, connect initiates, and disconnect notification.

Task-to-task communication and file access between systems is transparent. These inter-system facilities appear to be no different from the intra-system interprocess communication and file access facilities. In transparent access, the program opens the network interchange as if it were preparing device access, and then performs a series of reads and writes, just as it would to a pair of serial devices, one for input (reception) and the other for output (transmission). By its very nature, transparent access has no calls specifically associated with DECnet. The calls used for interprocess communication are the same as the calls used for accessing a sequential file in a high-level language—OPEN, CLOSE, READ, WRITE, etc. The programmer can choose to include the target node name in the OPEN statement, or he can defer assignment using logical names.

In non-transparent access, a program can obtain information about the network status to control the nature of its communication with other processes or tasks. Non-transparent access is available only through calls to operating system service procedures. A program can issue the following requests:

- CONNECT—Establish a logical link (the analog of OPEN).
- CONNECT REJECT—Reject a connect initiate.
- RECEIVE—Receive a message (the analog of GET).
- SEND—Transmit a message (the analog of PUT).
- SEND INTERRUPT MESSAGE—Transmit a high-priority message.
- DISCONNECT—Terminate a conversation (the analog of CLOSE).

The process can send optional data along with the connect request, such as the size or number of messages that it wants to send. The receiving process or task can accept or reject the connect initiate. A process can accept multiple connect requests.

A process can send or receive unsolicited messages to or from another process or task. Unsolicited message traffic is essentially no different from solicited message traffic except that it uses a system's software interrupt mechanism to transmit a message. A logical link, therefore, has two subchannels over which messages can be transmitted, one for normal messages and another for high-priority messages. In DECnet/VAX, an interrupt message is written to a mailbox that a process supplies for that purpose. The process can request that an asynchronous system trap routine be executed when a message is queued to that mailbox.

A program can issue a synchronous disconnect, which guarantees the receiver that it got every every message that was sent, or it can issue a disconnect abort, which terminates the logical link immediately.

In DECnet/VAX, a program using non-transparent access normally opens a control path directly to the Network's Ancillary Control Process (NETACP) and designates a mailbox for receiving information from the NETACP about the logical or physical links over which the process is communicating. The NETACP can notify a process when: a connect is attempted, a partner requests a synchronous disconnect, a partner requests a disconnect abort, a partner exists, a partner has sent an interrupt message, a physical link goes down, or an NSP protocol error is detected. If the process has the diagnostic privilege, it can also stop and start DDCMP protocol over a physical link.

Using the VAX/VMS command language, operators and terminal users can copy files from a VAX-11/780 system to another system, delete files in other systems, and transfer command files for executing on other systems. Programs have access to all of the above file operations, and they can also read and write records sequentially and randomly.

The down-line loading and network status interface for the operator is a function of the Network Control Program (NCP) utility that accepts parameters such as the location and name of the RSX-11S system image and the name of the node to be sent the load image.

Booting the RSX-11S operating system requires the presence of a read-only memory bootstrap program and, for a cold start, an operator to power-on the system and switch the DMC11 to "remote load detect." For a warm start, when an operating system is already running, no operator intervention is required, as the system automatically jumps to the bootstrap when requested to do so via MDP. For a cold start, the operator of the RSX-11S system starts the ROM bootstrap. It sends a special BOOT-ME message to the host, which automatically sends the proper system over the link.

A watchdog timer is available that will automatically put the RSX-11S system into BOOT mode whenever the operating system fails to reset a bit within a specified period. In this case, no operator intervention is required at either end. In all cases, the satellite system can start automatically after a boot operation.

SOFTWARE

OPERATING SYSTEM: The operating system used on the VAX-11/780 is *VAX/VMS,* a general-purpose operating system that provides the environment for the concurrent execution of multi-user timesharing, batch, and time-critical applications. Applications can be divided into several independent subsystems whose data and code are protected from one another but which have general communication and data sharing facilities. Jobs can communicate using general, group, or local communication facilities.

Jobs can be scheduled as time-critical jobs that have strict priorities of execution. When a time-critical job is ready to execute, it executes until it becomes blocked or until another time-critical job of higher priority needs the resources of the processor. Normal jobs can be scheduled using a modified preemptive algorithm that ensures that they receive processor and peripheral resources at regular intervals commensurate with their processing needs.

If insufficient memory is available for keeping concurrently executing jobs resident, the operating system will swap jobs in and out of memory to allocate each its share of processor time. Time-critical jobs can be locked in memory to ensure that they can be started up rapidly when they need to execute.

The operating system provides a dynamic virtual-memory programming environment. Large programs can be executed in a portion of physical memory that is considerably smaller than the program's memory requirements, without requiring the programmer to define overlays. The operating system optimizes its virtual-memory system for program locality and provides tools that support optimization. It makes program performance predictable and controllable by restricting paging to the process program, and by allowing the user to cause large amounts of a program to be brought in at one time.

The operating system provides sophisticated peripheral device management for sharing, protection, and throughput. Devices can be shared among all jobs or reserved for exclusive use by particular jobs. Input and output for low-speed devices is spooled to high-speed devices to increase throughput. Files on mass storage devices can be protected from unauthorized access on an individual, group, or volume basis.

The I/O request processing system is optimized for throughput and interrupt response. The operating system provides the user with several data accessing methods, from logical record accessing for device-independent programming to direct I/O accessing for rapid data processing. Files can be stored in any of several ways to optimize subsequent processing.

VAX/VMS provides the programming tools, scheduling services, and protection mechanisms for multi-user program development. Programmers can write, execute, and debug programs interactively, and can also create batch command files that perform repetitive program development operations without requiring their attention.

The VAX/VMS operating system's own jobs run as independent activities. They include the Job Controller, which initiates and terminates user processes and manages spooling; the Operator Communications Manager, which handles messages queued to the system operators; and the Error Logger, which collects all hardware and software errors detected by the processor and the operating system.

A command interpreter executes as a service for interactive and batch jobs. It enables the general user to request the basic functions that the operating system provides, such as program development, file management, and system information services.

Both hardware-detected and software-detected exception conditions are tracked through the exception dispatcher. The exception dispatcher passes control to user-programmed condition handlers or, in the case of system-wide exception conditions or the absence of user routines, to operating system condition handlers.

The operating system's memory management routines include the virtual activator, which controls the mapping of virtual memory to system and user jobs, and the pager, which moves portions of a process in and out of memory as required. They respond to a program's dynamic memory requirements and enable programs to control their allocated memory, share data and code, and protect themselves from one another.

The scheduler controls the allocation of processor time to system and user jobs. The scheduler always ensures that the ready-to-execute time-critical job of highest priority receives control of the processor until it relinquishes it. When no time-critical jobs are ready to execute, the scheduler dynamically allocates processor time to all other jobs according to their resource requirements. The swapper works in conjunction with the scheduler to move entire jobs into and out of memory when memory requirements exceed memory resources. The swapper ensures that the jobs most likely to execute are kept in memory.

The operating system's I/O processing software includes interrupt service routines, device-dependent I/O drivers, device-independent control routines, and user-programmed record processing services. The I/O system ensures rapid interrupt response and processing throughput, and provides programming interfaces for both special-purpose and general-purpose I/O processing.

LANGUAGES: VAX/VMS provides both a native programming environment and a compatibility-mode programming environment. The native programming environment consists of the language processors that produce native object code and the program development tools that support native program development. The VAX-11 BLISS-32 and VAX-11 FORTRAN IV-PLUS compilers produce VAX-11 native-mode code.

The compatibility-mode programming environment consists of the language processors that produce PDP-11 compatibility-mode object code and the program development tools that support compatibility-mode program development. The PDP-11 COBOL-74/VAX compiler and the PDP-11 BASIC-PLUS-2/VAX compiler product compatibility-mode code.

The differences between the native programming environment and the compatibility-mode programming environment are that:

- Native programs can be built from procedures written in any native language, fully exploit the new features of the VAX/11-780, and share a common run-time procedure library.

- Compatibility-mode programs can be built from procedures written in a given compatibility-mode language, and each compatibility-mode program has its individual object-time library.

Native and compatibility-mode programs can communicate with each other, share global data areas of memory, and read and write the same files, except that native programs can read and write sequential and relative files only, while compatibility-mode programs can read and write sequential, relative, and indexed files.

The compatibility-mode programming environment can be extended to provide an RSX-11M program development environment. The option includes RSX-11M language processors and tools that can be used to create programs to be executed in PDP-11 compatibility mode on the VAX-11/780 system, or to be executed on PDP-11 systems running the RSX-11M or RSX-11S operating systems and, in some cases, the IAS operating system.

VAX-11 FORTRAN IV-PLUS is based on the American National Standard FORTRAN X3.9-1966 language and includes features of the proposed ANS FORTRAN-77. The FORTRAN IV-PLUS compiler produces optimized VAX-11 native object code, makes use of the VAX-11 floating-point and character string instructions, and produces shareable code.

The VAX-11 FORTRAN IV-PLUS language is upward-compatible with the PDP-11 FORTRAN IV and FORTRAN IV-PLUS languages. The compiler supports the same enhancements to the language standard as PDP-11 FORTRAN IV and FORTRAN IV-PLUS, as well as providing additional enhancements.

The OPEN and CLOSE statements extend the file manipulating characteristics of the FORTRAN language. The OPEN statement can contain specifications for file attributes that direct file creation or subsequent processing. Attributes include: file organization (sequential, relative), method of access (sequential, direct), protection (read-only, read/write), record type (formatted, unformatted), record size, and file allocation or extension. List-directed input and output statements provide a method for obtaining simple sequential formatted input or output without the need for FORMAT statements. On input, values are read, converted to internal format, and assigned to the elements of the I/O list. On output, values in the I/O list are converted to characters and written in a fixed format according to the data type of the value.

A program can create fixed-length CHARACTER variables and arrays to store ASCII character strings. The VAX-11 FORTRAN IV-PLUS language provides a concatenation operator, substring notation, CHARACTER relational expressions, and CHARACTER-valued functions. CHARACTER constants, consisting of a string of printable ASCII characters enclosed in string quotes, can be assigned symbolic names using the PARAMETER statement. The CHARACTER data type implemented in VAX-11 FORTRAN IV-PLUS is a superset of the FORTRAN-77 data type.

The INCLUDE statement provides a mechanism for writing modular, reliable, and maintainable programs by eliminating duplication of source code. A section of program text that is used by several program units, such as a COMMON block specification, can be created and maintained as a separate source file. All program units which reference the COMMON block then simply INCLUDE this common file. Any changes to the COMMON block will be reflected automatically in all program units after compilation.

FORTRAN programs can call MACRO assembly language subroutines, and can call the system services and record management services using the VAX-11 procedure calling standard. Special operators exist for passing argument values directly, by reference, or by descriptor. A special operator also exists for obtaining the location of argument values used by the record management services procedures.

The FORTRAN IV-PLUS language can be used to create shared programs. FORTRAN IV-PLUS subprograms can also be used to create shareable image libraries, which can be made available to any program written in a native programming language.

FORTRAN IV-PLUS programs can be linked with the symbolic debugger to give programmers a way of examining and depositing locations dynamically.

VAX-11 BLISS-32 supports the development of modular software and provides access to most of the VAX-11/780 hardware features to facilitate time-critical and hardware dependent applications. It is intended for the development of operating systems, compilers, run-time system components, data base file systems, communications software, and utilities.

BLISS features include:

- Separately compiled modules for modularity and convenient development.
- BLISS-32 provides expressions for describing the actions to be performed and declarations for allocating, describing, and initializing data, for defining macros and literals, etc.
- All data is manipulated as longwords or 32 bits. Interpretation of data is provided by operators.
- The operators provide a set of operations for integer arithmetic, for comparison, maximization, and minimization of signed integer, unsigned integer, and address values, and for Boolean operations.
- Field references allow values to be fetched from or assigned to any contiguous field of from 1 to 32 bits located anywhere in the VAX-11 virtual address space.
- Character sequence functions provide for run-time manipulation of character data. Operations include moving, concatenating, comparing, and translating character sequences, as well as searching for particular characters or substrings of characters.
- The choice of an expression or group of expressions to be executed based on programmed tests.
- General loops that cycle as long as a programmed test is satisfied. The test may be made at either the beginning or the end of the loop.
- Counted loops that execute a computed number of times under control of a loop variable.
- Static storage allocation; names are made available to the Linker to resolve external data declarations in other modules.
- Stack-like allocation using either the execution stack or the general registers.
- Programmed definition of arbitrary data structures in terms of na accessing algorithm for locating elements of a structure. Built-in declarations for vectors and blocks provide for commonly needed structures.
- Routine declarations provide subroutines or procedures in BLISS-32. Routines are recursive and reentrant, and can be linked in sharable images for use by multiple processes.
- Sources from other files can be automatically included in the module being compiled.
- Library declarations allow special compiler-produced binary declaration files to be included in the module being compiled. The effect is substantially the same as require, but is more efficient because a restricted set of declarations can be precompiled into internal form.
- Macro declarations allow compile-time definition of both positional and keyword oriented macros. Macro definition and replacement are in terms of source lexical units called lexemes (atoms, tokens) rather than character text. Macro calls and macro declarations may be nested and recursive.
- Conditional compilation of BLISS source depending on programmed compile-time tests. These can be used to control expansion of macros or in their own right.
- Lexical functions allow a variety of compile time operations such as concatenation of strings, construction of names, testing properties of macro parameters, testing compiler qualifiers, generating compiler diagnostic messages, and controlling macro expansion.
- Linkages declarations allow definition of specialized calling sequences for time critical or unusual applications. Options allow for use of call and return instructions, for passing parameters in VAX-11 general registers or in parameter blocks, for controlling the preservation and use of registers by a routine, and for the sharing of registers across a set of routines as high speed common storage. Built-in linkage declarations for BLISS and FORTRAN fully support the VAX-11 calling sequence conventions.
- Use of link-time program sections for efficient use of the virtual address space. By default, generated code sections are position independent.
- Use of VAX-11 machine-specific functions for access to VAX-11 features not otherwise provided by the BLISS-32 language. Machine specific functions generally correspond to VAX-11 instructions. Over 50 such functions are provided.
- Enable declarations, together with signal, signal-stop, and setunwind functions, allow use of the VAX/VMS condition handling and error message reporting mechanisms.

The BLISS-32 compiler performs both global and local optimization. The following optimizations are employed; common subexpression elimination, removing loop invariants, constant folding, block register allocation, peephole replacement, test instruction elimination, jump vs. branch instruction resolution, branch chaining, and cross-jumping.

The BLISS-32 compiler optionally produces a listing file that shows the source text compiled and the generated code in a format that closely resembles a VAX-11 MACRO assembly listing. Multiple listing options allow selective inclusion or exclusion or source generated code, use of source names in the listing in place of machine register names (where feasible), macro expansion and tracing information, and use of names from library files.

PDP-11 COBOL-74/VAX is an optional language processing system that provides data processing for commercial applications. The PDP-11 COBOL-74/VAX compiler produces code that executes in PDP-11 compatibility mode, and is the same compiler that is available as an optional language processor for the RSTS/E, RSX-11M, and IAS operating systems.

The PDP-11 COBOL-74/VAX language conforms in language elements, representation, symbology, and coding format to ANS-74 COBOL, Specification X3.23-1974. The language elements include the Level 2 Nucleus module, Level 2 Table Handling module, Level 2 Sequential I/O module, Level 2 Relative I/O module, Level 2 Indexed I/O module, Level 2 Segmentation module, Level 1 Library module (with partial Level 2 REPLACING facility), Level 1 Interprogram Communication module, cross reference compilation listing, DISPLAY verb WITH NO ADVANCING clause, and nested conditionals.

The COBOL language processing system includes three utility programs: a source program reformatter (RFRMT), a report program generator (COBRG), and a program section description merge utility (MERGE). In addition, the SORT utility procedures can be called from within COBOL programs using the SORT verb.

PDP-11 BASIC PLUS-2/VAX is an optional language processing system that includes a compiler and an object-time system. It is the same BASIC-PLUS-2 system that is available as an optional language processor for the RSTS/E, RSX-11M, and IAS operating systems, and it produces code that executes in PDP-11 compatibility mode.

BASIC-PLUS-2 is an extended BASIC language that features programming facilities not found in most BASIC languages, including program formatting and commenting facilities, long variable names, virtual arrays, PRINT USING statement, a COMMON statement, a subprogram CALL statement, and extended debugging facilities.

The programmer can use the BASIC-PLUS-2 compiler in several ways. He can request the compiler to load in a source program for editing; he can compile a source program, produce an executable load module, and execute it (load and go); or he can compile a source program and produce an object module which can be linked with previously compiled object modules.

The object-time system (OTS) is a collection of library modules used during BASIC-PLUS-2 program execution. The library routines include math and floating-point functions, input/output operations, error handling, and dynamic storage functions. Since the OTS is a library, the linker can select only those functions needed at run time to be included in a program. Unnecessary routines are omitted from the program, and memory usage is thereby reduced.

UTILITIES: The VAX utility programs (or, as DEC categorizes them, program development tools) include two text editors, a linker, a librarian, a common run-time procedure library, and a debugger. These tools are available to the programmer through the VAX/VMS command language.

The text editors can be used to create memos, documentation, and data files, as well as source program modules for any language processor. The linker, librarian, debugger, and run-time procedure library described below are used only in conjunction with the language processors that produce native code. The language processors that produce compatibility-mode code offer their own task building, library, and debugging facilities and include their own object-time system libraries.

SOS is an interactive text editor that enables the programmer to create and modify text files using commands entered from either a hard-copy or video terminal. The user can insert, delete, and replace lines, find and substitute strings, or modify the text a character at a time. Lines can be identified by line number, by relative position, or by content. An adjacent group of lines can be copied or transferred from one place to another. Editing can be done in any order in the file. Editor parameters can be set to user-specified values, and the current values can be shown. User-specific parameters can be set automatically at editor start-up.

SLP is a programmed text editor that enables the .user to modify an existing file by supplying a command file containing a list of the modifications to be made. The command file provides a reliable way to duplicate the changes made to a file at a later time or on another system. SLP provides a formal record of changes made to files, both in the source file and in an audit-trail listing.

The *VAX/VMS Linker* accepts one or more native object modules produced by an assembler or compiler, resolves the symbols and procedure references between them, and produces an executable program image. The linker also enables a programmer to create shareable images that can be linked subsequently with other modules to produce an executable image. Furthermore, the linker not only accepts object modules to produce executable or shareable images, but can also accept object module libraries, shareable images, and shareable image libraries.

The *Librarian* enables a programmer to create, update, modify, list, and maintain library files. A library file can be a collection of object modules or shareable images. A programmer can request the linker to use one or more library files from which the linker can obtain modules to resolve references during linking.

The *Run-Time Procedure Library* is a collection of general-purpose and language-specific libraries available to any native program, regardless of the source language in which the program was written. The run-time library is a shareable program that allows the choice of either incorporating procedures from the library into an executable image or mapping the global sections into a process virtual address space at run time. A single copy of the library can be shared by all processes, and a new library can be installed without the need to relink existing programs. The run-time library includes a mathematical library, a general utility library, a condition-handling facilities library, a language-independent support library, and a FORTRAN IV-PLUS language-specific support library.

The *Symbolic Debugger* can be linked with a native program image to control program execution during development. The debugger can be used interactively or controlled from a command procedure file. The debugging language is similar to the VAX/VMS command language. Expressions and data references are similar to those of the source language used to create the image being debugged. Debugging commands include the ability to start and interrupt program execution, to step through instruction sequences, to call routines, to set break or trace points, to set default modes, to define symbols, and to deposit, examine, or evaluate virtual memory locations.

An an option, VAX/VMS supports an *RSX-11M program development package* that enables users to write, assemble or compile, and link RSX-11M task images. The task images can be written to execute on a PDP-11 under RSX-11M or RSX-11S; or, if properly coded, they can be written to execute in the VAX-11/780 compatibility-mode environment.

The RSX-11M development package includes both the PDP-11 MACRO assembler and the PDP-11 FORTRAN IV compiler, plus RSX-11M program development utilities. Programmers can use the VAX/VMS command language to

➤ edit, copy, or back up their files. They can also use the RSX-11M MCR command interface on the VAX-11/780 system. MCR enables the programmer to run the standard RSX-11M utilities such as the EDI and SLP editors, the PIP and FLX file transfer utilities, and the Task Builder, Librarian, and patch utilities.

PRICING

POLICY: DEC provides the VAX-11/780 system on a purchase basis, with separately priced maintenance agreements. Leasing arrangements are available through DEC's joint venture with U.S. Leasing Corporation at a monthly charge of 2.4% of the purchase price. DEC software is licensed rather than sold. Users purchase licenses and distribution rights separately.

On-site installation and basic education are included in the system prices. One-time installation charges are generally made to install add-on equipment. Separately priced training is available.

EQUIPMENT: Three of the standard VAX-11/780 systems for end users follow. All VAX/11/780 systems have provisions for additional memory and peripherals. System components are available to OEM's.

DUAL RK07 DISK-BASED SYSTEM: Includes a VAX-11/780 CPU with 256K bytes of ECC MOS memory, an LA36 DECwriter II console terminal, two RK07 28-megabyte disk drives, a multiplexer that provides eight EIA terminal connections, and the VAX/VMS operating system. Purchase price is $134,600.

RM03 DISK/TE16 MAGNETIC TAPE-BASED SYSTEM: Includes a VAX-11/780 CPU with 256K bytes of memory, one RM03 67-megabyte high-performance disk, one TE16 800/1600-bpi magnetic tape unit, an 8-line multiplexer, and the VAX/VMS operating system. Purchase price is $167,000.

RP06 DISK/TE16 MAGNETIC TAPE-BASED SYSTEM: Includes a VAX-11/780 CPU with 512K bytes of memory, one RP06 176-megabyte high-performance disk drive, one TE16 800/1600-bpi magnetic tape unit, an 8-line multiplexer, and the VAX/VMS operating system. Purchase price is $207,000.■

EQUIPMENT PRICES

		Purchase Price	Monthly Maint.
STANDARD SYSTEMS			
SV-AXHH	CPU with 512 bytes of memory, console terminal, two 28-megabyte disk drives, multiplexer with eight EIA terminal connections, console, operating system	141,330	753
SV-AXTV	CPU with 512K bytes of memory, one 67-megabyte disk drive, one 800/1600-bpi 45 ips magnetic tape unit, 8-line multiplexer, console, operating system	175,350	795
SV-AXTB	Same as SV-AXTV except 125 ips magnetic tape unit	185,850	898
SV-AXCV	CPU with 1024K bytes of memory, one 176-megabyte disk drive, one 800/1600-bpi 45 ips tape drive, 8-line multiplexer, console, operating system	217,350	955
SV-AXCB	Same as SV-AXCV except 125 ips magnetic tape unit	227,850	1,058
PROCESSOR AND MEMORY OPTIONS			
MA780-AA	256K-byte ECC MOS Multiport Memory subsystem for up to 4 VAX-11/780's	37,343	170
MA780-BA	Additional MA780-AA subsystem (max 3)	32,742	170
MA780-C	Multiport memory interface; required for 3rd or 4th CPU connection	6,955	20
MA780-D	Multiport Memory Selective Cache Invalidate Option; recommended for three- and four-processor systems	8,988	55
FP780-AA	High-performance floating-point accelerator	10,593	45
KU 780	12K writable control store	10,700	50
MS780-AA	128K bytes ECC MOS memory with controller	24,075	70
MS780-BA	128K bytes ECC MOS expansion memory for MS780-AA	8,560	30
MS780-BB	256K bytes ECC MOS expansion memory for MS780-AA	13,910	60
MS780-BC	512K bytes ECC MOS expansion memory for MS780-AA	23,540	120
MS780-CC	512K bytes ECC MOS memory with controller	26,643	150
MS780-DA	256K bytes ECC MOS expansion memory for MS780-CC	9,095	55
MS780-DB	512K bytes ECC MOS expansion memory for MS780-CC	13,910	110
MS780-DC	1024K bytes ECC MOS expansion memory for MS780-CC	19,795	220
MS780-DD	2048K bytes ECC MOS expansion memory for MS780-CC	31,886	440
H7112-A	MOS memory battery backup	1,225	10
H9602-HA (HB)	Memory expansion cabinet	4,227	NC
H9602-DF (DH)	Unibus expansion cabinet	3,199	NC
DR11-B	Direct memory access interface including control registers	1,733	13
PCL11-B	Multiple computer link for local network	7,330	60
BA11-KE	Extension mounting box; provides mounting space for 5 system units	3,210	16
BB11	Blank mounting panel for custom interface design and mounting system units	235	NC
DD11-C	Backpanel mounting unit; provides space for 2 hex and 2 quad slot modules	428	NC
DD11-D	Backpanel mounting unit; provides space for 7 hex and 2 quad slot modules	856	NC
MASS STORAGE			
REM03-A	Single-access 67M-byte removable disk pack drive and MASSBUS adapter; expandable to 8 RM03 disk drives	26,750	170
REM03-B	Dual-access 67M-byte removable disk pack drive and two MASSBUS adapters; expandable to 8 dual-access RM03 disk drives	35,310	215
REM03-D	RM03 dual-access kit containing drive logic, cables, and second MASSBUS adapter to convert REM03-A to REM03-B	8,560	45

DEC VAX-11/780

EQUIPMENT PRICES

		Purchase Price	Monthly Maint.
MASS STORAGE (Continued)			
RM03-A	Single-access 67M-byte removable disk pack drive	20,330	140
RM03-B	Dual-access 67M-byte removable disk pack drive	22,470	155
RM03-C	RM03 dual-access kit containing drive logic and cables to convert RM03-A to RM03-B	2,140	15
RM03-P	67M-byte removable disk pack for RM03	637	NC
REP06-A	Single-access 176M-byte removable disk pack drive and MASSBUS adapter; expandable to a total of 8 single-access RP drives (RP05, RP06)	47,080	220
REP06-B	Dual-access 176M-byte removable disk pack drive and MASSBUS adapter; expandable to a total of 8 dual-access RP drives (RP05, RP06)	60,562	270
REP06-D	RP06 dual-access kit containing drive logic, cables, and second MASSBUS adapter to convert REP06-A to REP06-B	15,729	50
RP06-A	Single-access 176M-byte removable disk pack drive	36,380	190
RP06-B	Dual-access 176M-byte removable disk pack drive	41,880	210
RP06-C	RP06 dual-access kit containing drive logic and cables to convert RP06-A to RP06-B	5,511	20
RP06-P	176M-byte removable disk pack for RP06	803	NA
RP06-U	RP05 to RP06 upgrade kit	11,235	NA
RK611-E	Single-access 14M-byte cartridge disk drive and control unit; expandable to a total of 8 single-access RK06 drives	15,515	129
RK611-F	Dual-access 14M-byte cartridge disk drive and control unit; expandable to a total of 8 dual-access RK06 units	23,540	169
RK611-C	Dual-access kit containing drive logic and hardware, one controller, and cables to convert RK611-E to RK611-F	11,182	40
RK06-E	Single-access 14M-byte disk drive	11,235	99
RK06-F	Dual-access 14M-byte disk drive	14,980	99
RK06-C	Dual-access kit containing drive logic, hardware, and cables to convert RK06-E to RK06-F	4,120	10
RK06 K-D	14M-byte data cartridge for RK06 subsystems	288	NA
RK711-E	Single-access 28M-byte cartridge disk drive and control unit; expandable to a total of 8 single-access RK06 or RK07 units	15,515	145
RK711-F	Dual-access 28M-byte cartridge disk drive and two control units; expandable to a total of 8 dual-access RK06 or RK07 units	23,450	190
RK711-C	Dual-access kit containing drive logic and hardware, one controller, and cables to convert RK711-E to RK711-F	11,182	45
RK07-E	Single-access 28M-byte disk drive	11,235	115
RK07-F	Dual-access 28M-byte disk drive	14,980	130
RK07-C	Dual-access kit containing drive logic, hardware, and cables to convert RK07-E to RK07-F	4,120	15
MAGNETIC TAPE EQUIPMENT			
TEE16-A	Program-selectable 800- or 1600-bpi, 9-track, 45-ips magnetic tape transport and MASSBUS adapter; industry-compatible; expandable to a total of eight TE16 transports	20,170	132
TRU45-K	Same as TEE 16-A except 75 ips, eight transports max.	24,610	235
TEU77-A	Same as TEE 16-A except 125 ips, four transports max.	29,960	235
TE16-A	Add-on transport for TEE 16-A	12,840	72
TU45-K	Add-on transport for TEU45-K	14,980	175
TU77-A	Add-on transport for TEU77-A	20,865	175
PUNCHED CARD EQUIPMENT			
CR11	80-column card reader and control; 300 cpm	7,490	61
CR11-B	Same as CR11 except 600 cpm	10,700	63
PRINTERS			
LA11-PA	180-cps, 132-column, 128-character printer and control unit	4,034	55
LP11-CA	900-lpm, 132-column, 64-character printer and control unit	25,680	185
LP11-DA	660-lpm, 132-column, 96-character printer and control unit	27,499	185
LP11-VA	300-lpm, 132-column, 64-character printer and control unit	12,626	127
LPR11-VA	Same as LR11-VA except remote, 250 lpm	19,046	207
LP11-WA	230-lpm, 132-column, 96-character printer and control unit	15,034	127
LPR11-WA	Same as LR11-WA except remote	21,454	207
LP11-YA	600-lpm, 132-column, 64-character printer and control unit	20,223	150
LPR11-YA	Same as LR11-YA except remote, 525 lpm	26,643	230
LP11-ZA	436-lpm, 132-column, 96-character printer and control unit	21,935	150
LPR11-ZA	Same as LP11-ZA except remote, 420 lpm	28,355	230
TERMINALS			
LA36-CE	DECwriter II 96-character set; upper/lower case keyboard; requires asynchronous line unit; 20 mA interface; 10, 15, or 30 cps	2,247	19
LA36-HE	Same as LA36-CE except with EIA interface	2,354	19
LA38-GA	DECwriter IV Printer 96-character set; upper/lower case keyboard; up to 30 cps	1,712	16
LA120-BA	EIA interactive hardcopy terminal; 7 x 7 dot matrix; up to 180 cps	2,782	25
LAXX-KG	EIA/CCITT adapter; allows an LA36 to connect to an EIA/CCITT interface	70	NC
VT52-AA/AE	Alphanumeric CRT; 80 columns by 24 lines, 96-character keyboard, EIA or 20-ma interface	2,247	20
VT100-AA	DECscope; 80 character x 24 lines or 132 character by 14 lines, 96-character ASCII keyboard plus 19-key keypad; up to 19,200 bps	2,033	17

DEC VAX-11/780

EQUIPMENT PRICES

COMMUNICATIONS EQUIPMENT

		Purchase Price	Monthly Maint.
DMC11-AL	Network Link Microprocessor Module for local applications; data rates to 1 million bps, full- or half-duplex; includes firmware for unattended operation; requires DMC11-MA or DMC11-MD line unit module; requires one hex SPC slot	1,712	13
DMC11-AR	Network Link Microprocessor Module for remote applications; data rates to 19,200 bps, full- or half-duplex; includes full data set controls and firmware for unattended operation; requires DMC11-AD line unit module; requires one hex SPC slot	1,712	13
DMC11-DA	Network link; remote line unit module	1,177	6
DMC11-MA	Network link; local line unit module; 1 million bps	1,177	6
DMC11-MD	Network link; local line unit module; 56,000 bps	1,177	6
BC03N-AO	100-ft. (30.5M) cable for DMC11 line units	130	NC
DZ11-A	EIA/CCITT asynchronous 8-line multiplexer; speeds and formats are programmable on a per-line basis; expandable to 16 lines	2,472	25
DZ11-B	EIA/CCITT 8-line multiplexer expansion unit for DZ11-A	1,926	21
DZ11-C	20-ma asynchronous 8-line multiplexer; speeds and formats are programmable on a per-line basis; expandable to 16 lines	2,525	25
DZ11-D	20-ma 8-line multiplexer expansion unit for DZ11-C	1,980	21
DZ11-E	EIA/CCITT asynchronous 16-line multiplexer; speeds and formats are programmable on a per-line basis	4,120	46
DZ11-F	20-ma asynchronous 16-line multiplexer; speeds and formats are programmable on a per-line basis	4,327	46
CS11-ML	Communications line switch for 8-channel RS 232-C lines	4,387	20

SOFTWARE PRICES

		Purchase Price
QE001-A	VAX/VMS operating system	13,800
QE100-AY	FORTRAN IV-PLUS compiler	3,795
QE101-AY	VAX-11 COBOL-74/VAX compiler and run-time system, report generator and reformat utility programs	8,855
QE102-AY	PDP-11 BASIC-PLUS-2/VAX compiler and run-time system	5,060
QE110-AY	VAX-11-DASCAL compiler	5,750
QE106-AY	BLISS compiler	13,800
QED01-AY	DECnet/VAX	3,105
QE105-AY	PDP-11 Datatrieve/VAX	2,588
QE111-AY	VAX-11 2780/3780 Protocol Emulator	5,060
MD-VAX	Maintenance documentation and diagnostic listings	3,450
MD-VAX-R	Yearly update service	2,070
MD-VAX-L/D	Diagnostic listings or maintenance diagnostics only	1,725
MD-VAX-LR/DR	Yearly update service	1,035