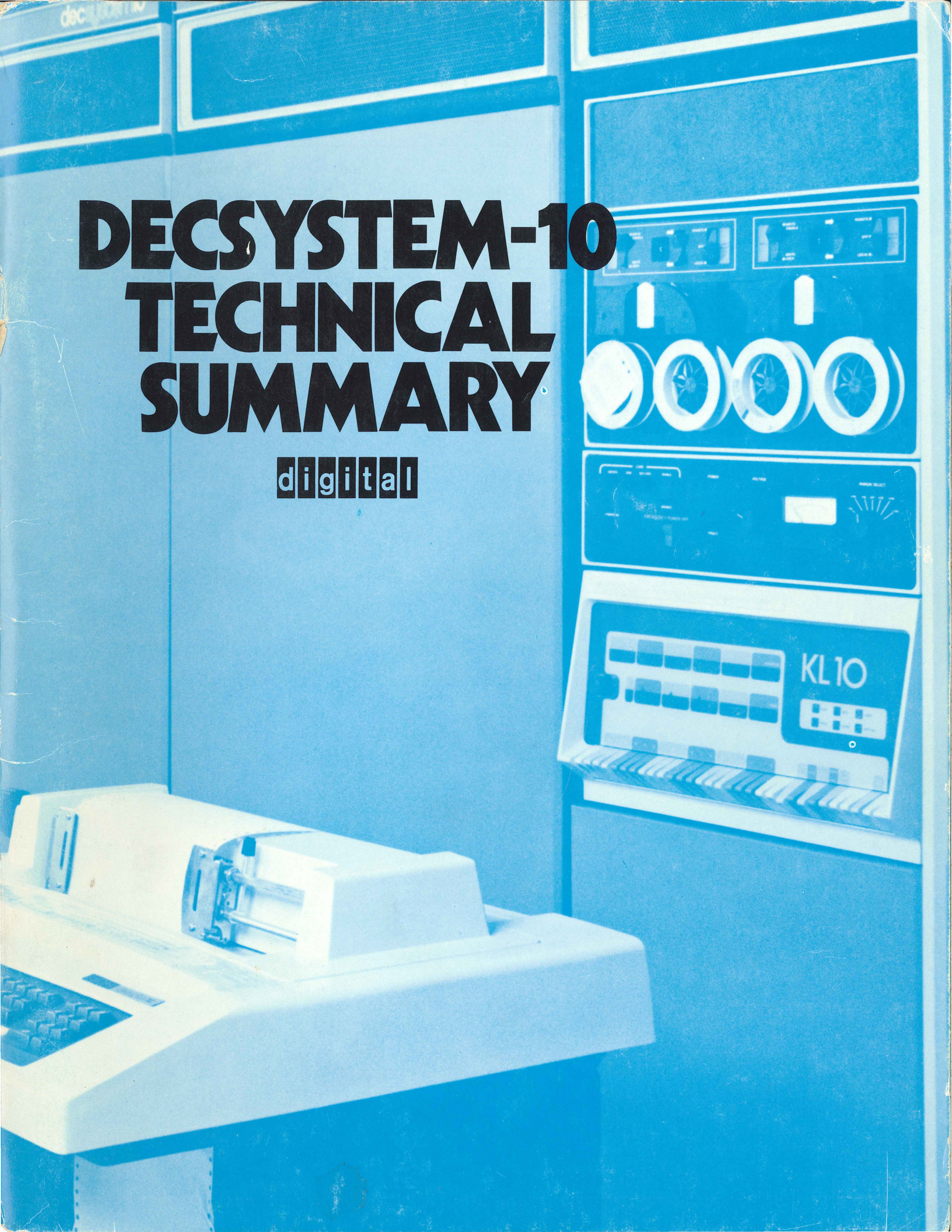
DECSYSTEM-10 TECHNICAL SUMMARY

digital



COPYRIGHT © 1976, 1977 DIGITAL EQUIPMENT CORPORATION

SYSTEM ARCHITECTURE

The DECsystem-1090 2

Central Processors 4

Instruction Set 6
- Instruction Groups 6
- Instruction Format 6
- Trap Handling on the KL10 and KI10 7

Processor Modes 8

Fast Register Blocks 9

Real-Time Clocks 9

Memory 10
- Cache Memory 12
- Memory Systems 12
- Interleaving 12

Input/Output 13
- Priority Interrupt System 13
- Multiplexed I/O Bus 13
- Massbus 14
- I/O Channel Capacity 14

Console/Diagnostic Computer 15

System Integrity Features 16
- Power Fail 16
- Error Handling 16

Multiprocessor Systems 17
- Dual CPU System 17

PERIPHERALS

PERIPHERALS 19

Disk Systems 20

DAS33 Buffered Data Channel 22

DECtape Transport 23

Magnetic Tape Systems 24

Hard Copy Equipment 26

Card Readers 26

Line Printers 26

Terminals 28

DATA COMMUNICATIONS

DATA COMMUNICATIONS 33

System Concepts 34
- Asynchronous Communications 34
- Synchronous Communications 34
- Synchronous vs. Asynchronous Transmission 35
- Remote Communications 36

Network Concepts 37
- Simple Topologies 37
- Complex Topologies 37
- Task-to-Task Communication 38
- IBM Device Emulation and Termination 38

Communications Products 39
- DN87/87S Universal Synchronous/Asynchronous Front End Subsystem 39
- Complex Topologies 42
- DECnet-10 42
- DAS80 Series Remote Station 43
- DAS92 Remote Station 45
- DAS 61 IBM 2780/3780 Support 46
- DAS 62 IBM HASP Multi-leaving Support 46

CONFIGURATOR

CONFIGURATOR 47

SOFTWARE

SOFTWARE 59

Languages 61
- AID 60
- ALGOL 60
- APL 60
- BASIC 62
- COBOL 61
- ISAM 62
- CPL 62
- FORTRAN 63
- FORTRAN Version 4 63
- MACRO 64

System Utilities 65
- SORT 65
- Data Base Management System 65
- ITPS-10 66
- LINK-10 66
- MACY11-LNKX11 67
- DDT 67
- FAILSAFE/BACKUP 67
- Message Control System 68
- PIP 68
- RUNOFF 68
- TECO 68

Operating System 69
- TOPS-10 Operating System 69
- Command Control Language 72
- File System 73
- Input/Output 76
- Memory Management 76
- Multi-Processor Systems 77
- Inter-job Communication 78
- Communications Software 78

Multimode Computing 79
- Timesharing 79
- GALAXY-10 Batch 79
- Real-Time Computing 81

Diagnostic Software 83
- High Availability 83
- Maintenance Features 83
- On-Line Diagnosis 83

SERVICES

SERVICES 85
- People 87
- Parts 87
- Support 87

Service Agreements 88

Education 88

Logistics 89

Technical Documentation 89

Software Services 90

Advanced Systems Group 91

THE DECsystem-1090

The DECsystem-1090 is an extension of the DECsystem-1080 which was announced in September of 1974 and first delivered in June of 1975. The DECsystem-1090 offers all the outstanding features of the 1080 plus many new ones. The key features of the DECsystem-1090 are:

The new KL10B Processor.

It features 1.4 mips, integrated channels, cache memory and a PDP-11 Console front end computer which permits on-line, remote diagnosis and maintenance. The new CPU permits more compact configuration and lower power consumption through the use of integrated channels; improved performance; and substantially lower system purchase and maintenance costs. Single and dual CPU systems, featuring high availability, are offered.

The new MH10 Memory System.

It offers 256K 36-bit words in one 31" system cabinet. Internal 2-way interleaving and eight memory ports accommodate up to 2 CPU's and all of DIGITAL's standard memory channel peripheral subsystems. The DECsystem-10 memory can be expanded with up to 16 MH10 units to over 4 million words or 20 million ASCII characters.

The new DN87S Communications front end.

It provides additional features and lower costs than previous communication front end offerings. Cost is reduced through the use of the integrated DTE channels which utilize high-speed microcode for data transfer. Important new features include DECnet compatibility and simultaneous support of synchronous and asynchronous communications on a single front end. All the standard ANF features are also supported. Up to three DN87S units per 1090 System can accommodate up to 336 full duplex asynchronous lines or 36 synchronous lines. The number of lines can be extended by adding a DN87A and up to three DN87D s, each of which can support 112 asynchronous or 12 synchronous lines.

The recently-announced RP06 200 megabyte Disks.

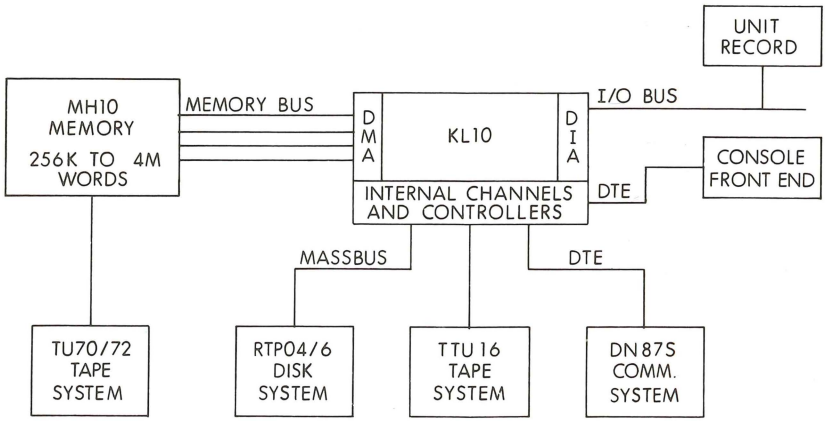
They are now available using the integrated MASSBUS® controllers. Not only is the cost of Disk Systems substantially reduced, but the use of integrated controllers and channels provide automatic update of the cache memory –thus reducing overhead. The on-line disk data storage can now be extended to over 6.2 billion ASCII characters or about 5 billion 8-bit bytes.

The new TTU16 Magnetic Tape Subsystem.

It now uses the integrated MASSBUS® Controllers to extend the DECsystem-1090 tape capability to three subsystems. Transfer rates of 72K (TU16), 320K (TU70) and 780K (TU72) characters per second allow DECsystem-1090 users to select the tape system which meets their needs. With the new TU72 system, 6250 bpi capability has been added to our product offerings at the high end and the TTU16 72KHz system greatly reduces the cost of 1600 bpi capability, for low performance requirements.

A complete range of Line Printers and Card Readers.

The new LP100 offers high quality "band" printing at 1200 to 1500 lines per minute. The LP10 and LU10 offer high quality mid-range and low-speed printers at lower cost to meet users' individual needs. The CR10E and CR10F 1200 and 300 card per minute readers offer a choice of two excellent card readers.



A DECsystem-1090 Block Diagram

SYSTEM ARCHITECTURE

Central Processors 4
Instruction Set 6
Instruction Groups 6
Instruction Format 6
Trap Handling on the KL10 and KI10 7
Processor Modes 8
Fast Register Blocks 9
Real-Time Clocks 9
Memory 10
Cache Memory 12
Memory Systems 12
Interleaving 12
Input/Output 13
Priority Interrupt System 13
Multiplexed I/O Bus 13
Massbus 14
I/O Channel Capacity 14
Console/Diagnostic Computer 15
System Integrity Features 16
Power Fail 16
Error Handling 16
Multiprocessor Systems 17
Dual CPU System 17

Central Processors

Within the family of six DECsystem-10 configurations, there are three central processor units. All operate under the same DECsystem-10 operating system (TOPS-10), execute the same software, and share most models of DECsystem-10 peripheral equipment. The three processors differ in their speed, method of address mapping, memory capacity, program size, auto-diagnostic and restart features, and priority interrupt system.

The KA10 central processor is used in the DECsystem-1050 and 1055. The more powerful KI10 central processor is used in the DECsystem-1070 and 1077 configurations. The top-of-the-line KL10 central processor is used in the DECsystem-1080, 1090 and 1099. The high speed 2K word cache memory (K=1,024) permits rapid instruction execution while minimizing accesses to main memory. A double precision floating multiply, for instance, executes in 4.8 μsec.

Because all processors operate with the same operating system, it is easy to field upgrade a system with a KA10 and replace it with a KI10 processor; or take a system with a KI10 and replace it with a KL10 processsor, providing a significant increase in the computing power with no necessary changes in user programs.

The 1090 configuration differs from the 1050 or 1070 in that the high speed data channels and communications processor channels are integral to the central processor. Up to eight high speed buffered data channels can be utilized which support DIGITAL's Massbus® peripherals. Because the data channels access four 36-bit words of memory in one channel request cycle the total throughput is in excess of one million words per second or 4 megabytes per second. For non-Massbus® peripherals, external channels which transfer directly to memory can be utilized. An important feature of the integrated high speed data channels is that cache memory is automatically updated during I/O transfers. The integrated front end processor channels will accommodate up to three PDP-11 communications front ends in addition to the KL10 console front end. These channels utilize high speed microcode to transfer and process data coming from the PDP-11 communications front ends. All channel data transfers are parity checked. A final benefit of the 1090 internal channels is configuration simplicity, improved maintainability and reduced power consumption.

DECsystem-10 at a Glance.

	1040/1050	1055*	1060/1070	1077*	1080/1090	1088*/1099*
Relative Performance	1.0	1.8	2.2	3.5	4.8	6.7
Avg. Number of Users	10-40	20-70	30-80	40-100	50-127	75-150
No. of CPUs	1	2	1	2	1	2
Memory Size in K words (min.-max.) (K=1024)	64-256	128-256	128-4096	128-4096	128-4096	128-4096
No. of Instructions:	366	366	378	378	398	398
Instruction Look-ahead	No	No	Yes	Yes	Yes	Yes
Virtual Memory	No	No	Yes	Yes	Yes	Yes
Memory Interleaving	2 or 4 way	2 or 4 way	2 or 4 way	2 or 4 way	2 or 4 way	2 or 4 way
Index Registers	15	15 each CPU	4 x 15	4 x 15 each CPU	8 x 15	8 x 15 each CPU
Accumulators	16	16 each CPU	4 x 16	4 x 16 each CPU	8 x 16	8 x 16 each CPU
Instruction Times (microseconds)**						
Fixed Point Add	2.8	2.8	1.6	1.6	.52	.52
Fixed Point Multiply	9.8	9.8	4.6	4.6	2.4	2.4
Unconditional Jump	1.5	1.5	1.1	1.1	.36	.36
Single Precision Floating Point Add	9.8	9.8	2.7	2.7	1.8	1.8
Double Precision Floating Point Add	59.4	59.4	3.9	3.9	2.2	2.2
Double Precision Floating Multiply	N.A.	N.A.	8.1	8.1	4.8	4.8
Double Precision Floating Divide	N.A.	N.A.	16.0	16.0	10.2	10.2
Increment and Move Byte	N.A.	N.A.	5.6	5.6	1.4	1.4
Move from Memory	N.A.	N.A.	1.6	1.6	.48	.48
I/O Bus Band width (words/second)	200K	200K	370K	370K	370K	370K
Memory Bus Band width (words/second)	4000K	4000K	4000K	4000K	4000K	4000K

* Dual-processor systems execute two instructions simultaneously.

**For 1080/1088 and 1090/1099 Operands in Cache Memory.

Instruction Set

Instruction Groups

The KA10 has 366 instructions, the KI10 378 instructions, and the KL10 398 instructions, an extremely large repertoire which provides the flexibility required for specialized computing problems. The instruction sets of the KA10 and KI10 are hardwired. By contrast, the instruction set of the KL10 is microprogrammed; that is, each instruction is actually a series of micro-instructions that perform various logical functions such as processor state control, data path control and actual implementation of each instruction in the KL10's set. The micro-code is loaded into a 1K 80-bit word RAM (random access memory) through the PDP-11 console/diagnostic computer. Since the set provides so many instructions to choose from, fewer instructions are required to perform a given function. Assembly language programs are therefore shorter than with other computers, and the instruction set simplifies the Monitor, language processors, and utility programs. For example, compiled programs on a DECsystem-10 are often 30 to 50 percent shorter, require less memory and execute faster than those of comparable computers.

In addition to these instructions, the DECsystem-10 provides 64 programmable operators, 33 of which "trap" to the Monitor (Monitor calls) and 31 of which trap to the user's call area. The remaining instruction codes are unimplemented and reserved for future expansion. An attempt to execute one of these unimplemented instructions results in a trap to the Monitor.

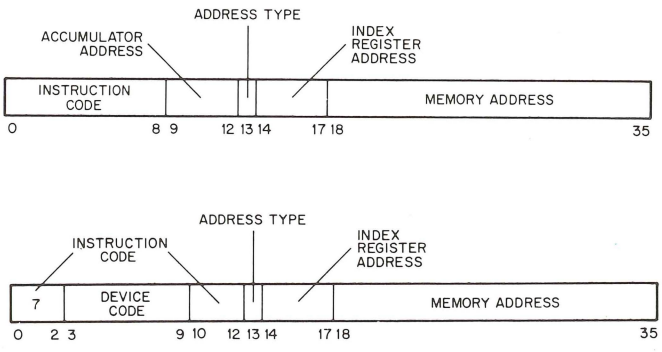
The instruction set, despite its size, is easy to learn. It is logically grouped into families of instructions and the mnemonic code is constructed modularly. All instructions are capable of directly addressing a full 256K (36 bit) words of memory without resorting to base registers, displacement addressing, or indirect addressing. Instructions may, however, use indirect addressing with indexing to any level. Most instruction classes, including floating point, allow immediate mode addressing, where the result of the effective address calculation is used directly as an operand in order to save storage and speed execution.

Instruction Format

In all the non-input/output instructions, the nine high-order bits (0-8) specify the operation, and bits 9-12 usually address an accumulator but are sometimes used for special control purposes, such as addressing flags. The rest of the instruction word always supplies information for calculating the effective address, which is used for immediate mode data or is the actual address used to fetch the operand or alter program flow. Bit 13 specifies the type of addressing (direct or indirect), bits 14-17 specify an index register for use in address modification (zero indicates no indexing), and the remaining eighteen bits (18-35) contain a memory address.

The instruction codes that are not assigned as specific instructions are performed by the processor as so-called "unimplemented operations", as are the codes for floating point and byte manipulation in any KA10 that does not have the hardware for these instructions.

An input/output instruction is designated by three 1s in bits 0-2. Bits 3-9 address the input/output device to be used in executing the instruction, and bits 10-12 specify the operation. The rest of the word is the same as in other instructions.



INSTRUCTION FORMAT

HALF-WORD DATA TRANSMISSION

The half-word data transmission instructions move a half-word and may modify the contents of the other half of the destination location. There are 16 instructions which differ in the direction that they move the chosen half-word and in the manner in which they modify the other half of the destination location.

FULL-WORD DATA TRANSMISSION

The full-word data transmission instructions move one or more full words of data from one place to another. The instructions may perform minor arithmetic operations such as forming the negative or the magnitude of the word being processed.

BYTE MANIPULATION

The five byte manipulation instructions pack or unpack bytes of any length anywhere within a word.

LOGIC INSTRUCTIONS

The logic instructions provide the capabilities of shifting and rotating, as well as performing the complete set of 16 Boolean functions on two variables.

FIXED-POINT ARITHMETIC

Two common conventions are to regard a number as an integer (binary point at the right) or as a proper fraction (binary point at the left); in these two cases, the range of numbers represented by a single word is -2^{35} to $2^{35}-1$ or -1 to $1-2^{-35}$. Since multiplication and division make use of double-precision numbers, there are special instructions for performing these operations to yield results that can be represented by a single word.

The format for double-length fixed-point numbers is just an extension of the single-precision format. The magnitude (or its twos complement) is the 70-bit string in bits 1-35 of the high- and low-order words. Bit 0 of the high-order word is the sign, 0 for positive, 1 for negative. Bit 0 of the low-order word is ignored. The range for double-length integer and proper fractions is thus -2^{70} to $2^{70}-1$ or -1 to $1-2^{-70}$. Zero is represented by a word containing all 0s.

FLOATING-POINT ARITHMETIC

All DECsystem-10 processors have instructions to perform scaling, negating, addition, subtraction, multiplication and division upon numbers in single-precision, floating point format. In the single-precision floating point format, one bit is reserved for the sign, eight bits are used for the exponent and 27 bits are used for the fraction.

As a superset of the KA10, the KI10 and KL10 have instructions to perform all of the above functions in double-precision floating-point format as well as single-precision floating-point. In double-precision floating-point format, one bit is used for the sign, eight bits are used for the exponent and 62 bits are used for the fraction.

FIXED/FLOATING CONVERSIONS

Special KI10 and KL10 instructions provide the capability of converting fixed-point formats to or from floating-point formats. Two sets of instructions are provided to perform this function: one set optimized for FORTRAN and a second set optimized for ALGOL.

ARITHMETIC TESTING

The arithmetic testing instructions may jump or skip, depending on the result of an arithmetic test and may first perform an arithmetic operation on the test word.

LOGICAL TESTING, MODIFICATION, AND SKIP

These instructions modify and/or test using a mask and/or skip on selected bits in an accumulator.

PROGRAM CONTROL

Program control instructions include several types of jump instructions and the subroutine control PUSHJ and POPJ instructions.

INPUT/OUTPUT OPERATIONS

Input/output instructions govern all direct transfers of data to and from the peripheral equipment and also perform many operations within the processor. Block transfer instructions handle bulk data transfers to and from I/O devices.

UNIMPLEMENTED USER OPERATIONS (UUOs)

Many of the codes not assigned as specific instructions are executed as unimplemented user operations wherein the word given as an instruction is trapped and must be interpreted by a routine included for this purpose either by the programmer or by the Monitor. Those UUOs reserved for use by the Monitor are called Monitor UUOs (MUUOs), while user UUOs are called Local UUOs (LUUOs). Instructions that are illegal in user mode also trap in the same manner as MUUOs.

BUSINESS INSTRUCTION SET

Five instructions comprise the Business Instruction Set in the KL10 central processor. Four of these are new arithmetic instructions to add, subtract, multiply, and divide using double precision, fixed-point operands. The new STRING instruction is capable of performing nine separate functions.

These functions include an EDIT capability; decimal to binary and binary to decimal conversion in both offset and translated mode, (offset mode is byte modification by addition of the effective address of the string instruction. This also occurs in EDIT. In addition to providing the translation function, those instructions which use translation can control the flags in ACs and can detect special characters in the source string.); Move String in both offset and translated mode; and Compare String in both offset and translated mode.

This Business Instruction Set provides faster processing since there are specific instructions for doing more comprehensive string operations. These instructions can be used on a variety of code types such as ASCII and EBCDIC.

Trap Handling on the KI10 and KL10

The KI10 and KL10 provide facilities for handling arithmetic overflow and underflow conditions, pushdown list overflow conditions, and page failures directly by the execution of programmed trap instructions. This trap capability avoids recourse to the program interrupt system.

Processor Modes

Instructions on the KA10 are executed in one of two modes depending upon whether a mode bit has been set. Programs operate in either User Mode or Executive Mode. In Executive Mode operations, all implemented instructions are legal, addresses are not relocated, and all core locations are accessible. The Monitor operates in Executive Mode and is able to control all system resources and the state of the processor. In User Mode operations, addresses are relocated, certain instructions are illegal, causing monitor traps when executed, and address references are confined within two user program segments.

The KI10 and KL10 further divide Executive and User Mode operation into two submodes each. User Mode is subdivided into public and concealed submodes and Executive Mode into supervisor and kernel submodes. For each 512-word page in the system, information is stored in a table maintained by the operating system which specifies whether or not a page can be accessed or altered, and if it is defined to be public or concealed.

If a program is running in a public submode, pages within the user's addressing space are accessible only if they are listed in the user's page map and are defined to be accessible from public mode. Pages designated public are, by definition, accessible. Pages designated concealed may be accessed only at defined entry points, i.e. portals which permit entry from public submode programs. In concealed submode operations, programs can access all of the virtual addressing space. However, if a program running in concealed submode executes an instruction from an area designated to be public, the state of the processor transfers over into public submode. Concealed areas can be used for proprietary coding that can be executed but not altered or examined by users in the public mode.

The supervisor and kernal submodes are similar but not identical to the public and concealed submodes. Supervisor submode programs can access but cannot alter areas designated as concealed. Also, any instruction executed out of a public area from either supervisor or kernel submode returns the processor to supervisor submode. In kernel submode operations all of memory is accessible and can be altered. Programs operating in kernel submode can address portions of memory directly, without paging, and it is through the kernel submode program that page restrictions are established. Functions delegated to supervisor submode generally include those affecting individual users as opposed to overall system management of input/output, priority interrupts, page map accounting, etc., which are handled by kernel submode programs. The ability of kernel submode programs to supply information which supervisor submode programs can read but not alter allows portions of the operating system to be hardware-protected from other portions undergoing modifications or design changes.

KI10 AND KL10 PROCESSOR MODES

USER MODE

Public Submode
- User programs
- 256K word address
- All instructions permitted unless they compromise integrity of system or other users
- Can transfer to concealed submode only at entry points

Concealed Submode
- Proprietary programs
- Can READ, WRITE, EXECUTE or TRANSFER to any location labeled Public

EXECUTIVE MODE (Monitor)

Supervisor Submode
- Performs general management of system
- Performs those functions that affect only one user at a time
- Executes in virtual address space labeled public

Kernel Submode
- Performs I/O for system
- Performs those functions that affect all users

Fast Register Blocks

General-purpose registers are a DECsystem-10 feature that help improve program execution. These sets of fast integrated circuit registers can be used as accumulators, index registers, and as memory locations. Since they may be addressed as memory locations, they do not require special instructions.

One set of 16 registers is included in the KA10, four sets of 16 fast registers are included in the KI10, and eight sets of 16 fast registers are included in the KL10. Context switching on the KA10 is performed by storing the register information into core locations. Program switching time for the KL10 between register stacks is 500 nanoseconds.

On the KI10 and KL10, different register blocks can be used for the operating system and individual users. This eliminates the need for storing register contents when switching from User Mode to Executive Mode. Also, a critical real-time program is able to maintain its own register block for handling data and interrupt sequences at maximum speed.

Real-time Clocks

The DK10 Real-time Clock is available with each KA10- or KI10-based DECsystem-10 and provides high-resolution timekeeping for time accounting, time-base maintenance, periodic high-frequency interrupts, and interval timing. Meeting the most demanding real-time requirements, the clock provides 10 μsec resolution and a choice of up to 2^{18} possible timing intervals, so that interrupts can be programmed at intervals from 10 μsec up to 2.6 seconds.

In addition to an interval register, the DK10 has a frequency counter which counts the pulses of an internal 100 kHz $\pm$0.01 clock, or an external clock having a maximum frequency of 400 kHz. The clock includes a comparator network which provides a running comparison between the frequency counter and the internal register. When the frequency counter reading equals the total on the internal register, a program interrupt is generated and the frequency counter is automatically reset so that it can time the next interval.

The clock, which is assignable to any interrupt channel, can be used to pace real-time, monitor, or other functions performed in either Executive or User Modes. In fact, a KA10 or KI10 system can have two clocks—one for each mode—since two device codes are available for clock use. Clock updating is interlocked with the DATAI instructions so that it can be read correctly—at any time—by the KA10 or KI10 without losing a clock pulse.

The four clocks built into the KL10 provide a number of timing and counting functions including an interval timer, a time base, an accounting meter and a performance analysis counter.

All of the operations on these clocks are accomplished by means of I/O instructions to internal devices. Many of these functions use a microsecond source of pulses which is counted down from the basic machine clock. The 0.005% tolerance will give less than five seconds drift over 24 hours.

The Interval Timer provides a programmable source of interrupts with a 1 microsecond resolution and is similar to the DK10 real-time clock. It is used for real-time applications and for page management by the Monitor. It is designed so that a real-time deadline schedule with varying deadlines can be implemented.

The Time Base provides a 60-bit microsecond resolution source of elapsed time. This gives over a 9,140 year maximum time before wrap-around.

The Accounting Meters provide an accurate and reproducible measure of the amount of processor resource used by a job, interrupts, page failures, and cache activity.

The Performance Analysis Counter provides a tool for studying hardware and software performance of the system. It may be used to point to hardware and/or software bottlenecks.

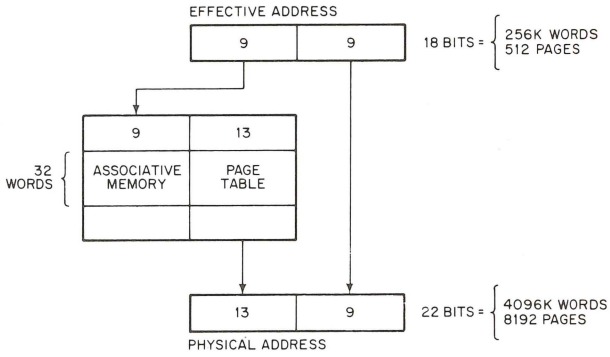
Memory

On the KA10, dual memory protection and relocation registers protect other users and the monitor from errors in the active user program. If a user attempts to reference memory outside his area, his program is stopped from operating and control is transmitted to the operating system. The feature also permits dynamic memory allocation, allowing each user program to be assigned either one or two segments of memory. One of these segments can be write-protected so as to protect critical programs or data from errors committed by the user. Segments—which are multiples or 1024 words—may be located anywhere in memory and may contract and expand as needs dictate. Memory management techniques, employed by the Monitor, insure efficient memory usage.

The division of programs into two segments also makes possible the reentrant or recursive use of language processors and systems programs. With the DECsystem-10, all DIGITAL-supported language processors and most utility programs are reentrant. In other words, each program is written in two parts. One part contains pure or reentrant code that is not modified during execution so the same copy can be used to simultaneously service any number of users. A separate second part of the program belongs strictly to each user and consists of non-reentrant code and data. This section is stored in a separate area of core. This feature allows more users to utilize a given amount of physical core simultaneously. Alternatively, less core is required to service a given number of users. The result is much more efficient use of computing resources.

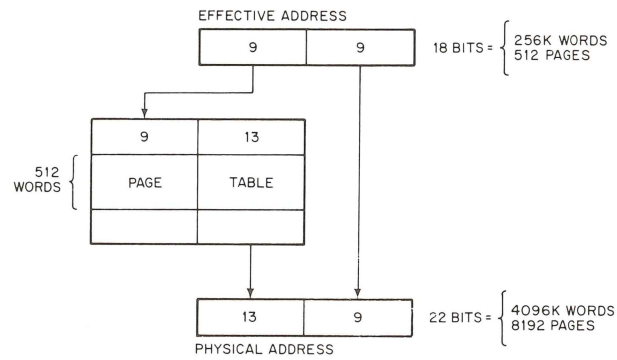
The ability to divide programs into two segments is not limited to DIGITAL-supplied language processors and utilities. User-written programs, including those written in higher-level languages such as FORTRAN and COBOL, can also take advantage of this feature. Thus, user developed applications can benefit from the same efficient treatment given DIGITAL-supplied software.

KI10 Processor Associative Memory



KI10 ASSOCIATIVE MEMORY

KL10 Processor Page Addressing



KL10 PAGE ADDRESSING

The KI10 and KL10 provide memory address mapping from the program's memory address space (referred to as the effective address) to the physical memory address space by substitution of the most significant bits of the memory address. This mapping provides access to the entire physical memory space which can be up to 16 times larger than the maximum user address space. The user's effective address space is 256K words addressed with 18-bit addresses; the physical address space is 4,096K words addressed with 22-bit addresses (where 4,096K is equivalent to 4,194,304 decimal).

The memory mapping process utilizes the most significant 9 bits of the effective address as an index into the appropriate page map (User or Executive). The data located by the index provides 13 bits which are appended to the least-significant 9 bits of the effective address in order to form the 22-bit physical address. Also provided are 3 bits which indicate what type of memory requests are allowed to the page in question (none, read-only, proprietary, etc.).

If this scheme were implemented exactly as outlined above, every user memory reference would require two actual memory references: one to obtain the memory mapping data and one to obtain the user's mapped memory reference. In order to reduce the number of actual memory references to nearly the same number as required by the program, an associative memory mapping unit is used in the KI10, while a copy of the entire page table is kept in the KL10 hardware.

If the address is in the range 0-17 (octal) inclusive, the hardware fast register blocks are referenced instead of the memory system. On the KI10, the User Mode bit and the high-order 9-bits of the virtual address are compared against the contents of the associative memory registers which are part of the memory mapping hardware. These 10-bits will either exactly match one of the associative registers or a no-match occurs. On the KL10, the User Mode bit and the high-order 9-bits of the virtual address are used to do a "table look-up" in the hardware page table.

On the KI10, if a match exists, the contents of the related register supply the 13-bit most-significant portion of the physical memory address and also supply 3 bits which indicate what type of memory references are allowed to this page, i.e., if the memory request is not consistent with the request type allowed bits, a page failure occurs. On the KL10, if a page table entry exists, the contents of the entry supply the 13-bit most-significant portion of the physical memory address and also supply 3 bits which indicate what types of memory references are allowed to this page.

For both the KI10 and KL10, if the memory request is consistent with the request type allowed bits, the physical address used consists of the 13 bits from the related register as the most significant bits of the physical address and the 9 least significant bits of the effective address as the least significant bits of the physical address.

When the relocation data for a referenced page does not exist in the associative memory of the KI10, or the hardware page table of the KL10 (i.e., a no-match), the hardware reads the relocation data from the page table in memory and stores it into the associative memory of the KI10 or the hardware page table of the KL10.

The associative memory in the KI10 has associated with it a reload counter that points to one of the 32 entries in the memory.

Any time a word of the associative memory of the KI10 is referenced and the reload counter is pointing at it, the reload counter is incremented to point at the next word in the KI10 associative memory. Whenever it is necessary to load a new entry into the associative memory, the location replaced is the one to which the reload counter points. Thus, at worst, a page which was just referenced would not have its word in the KI10 associative memory replaced by the next memory reference. If a particular word in the KI10 associative memory has not been referenced in some time, the reload counter would be left pointing to this word, having been pushed away from all words in associative memory which have been used. Thus, a 1-bit approximation to "least-recently-used" page table operation is obtained.

The monitor assigns the core area for each user by loading the various page tables, setting up the trap locations in the user page map, and responding appropriately when a trap occurs. The Monitor provides memory protection for itself and each user by filling the page tables only with those entries which are allowed to be accessed. A zero access bit in the page table will cause a reference to the associated page to initiate a page failure trap to the Monitor.

The TOPS-10 Operating System utilizes the KI10 and KL10 in-core page maps to create one or two segment programs in roughly the same fashion as it uses the protection and relocation registers of the KA10. The major benefits of the paging capability are a smaller unit of core allocation (512 words instead or 1024), the freedom to scatter the pages of a segment randomly through physical core (avoiding core fragmentation and the overhead of repacking core), and the opportunity to execute a program when all of its pages are not in physical core (i.e., a virtual memory capability).

Memory Systems

To meet the requirements of large systems, DECsystem-10 core memory can be modularly expanded to 256K words for KA10-based systems or 4,096K words for KI10- and KL10-based systems, all directly addressable. Memory can consist of combinations in 64K, 128K and 256K word modules. The structure of the memory bus gives the central processor and high-speed data channels simultaneous access to separate memory modules and allows each to operate at its own speed.

The MF10 provides up to 4 ports and the MG10 or MH10 up to 8 ports. Each port can be further expanded through the use of the MX10 memory multiplexer. The multiplexer handles up to eight channels, inter-leaving data from the channels on a word-by-word priority basis. Such parallel operation yields many improvements over systems which provide only a single path to memory. The memory bus station allows each data channel to transmit full 36-bit words in parallel at a speed of one million words per second. In total, the memory structure operates at rates of up to 10 million characters per second when I/O devices and processors are concurrently simultaneously transferring data. In addition, the memory bus is capable of handling a maximum of 16 memory modules. Each memory module provides switches which allow that particular module to represent any module of its size in the addressable memory space. Thus, one model can replace another without rewiring. The Switches also provide for memory interleaving.

The core memory systems available for the DECsystem-10 are described in the memory table below.

Interleaving

The concept of interleaving has long been used to increase the effective bandwidth of DECsystem-10 systems. On KA- and KI-based systems, successive memory locations were located, via memory addressing hardware, in alternating memory modules. Thus, the processor, when requesting successive logical locations, would not have to wait for the complete memory read/write cycle to be complete before affecting a fetch. The first memory module could be completing the write cycle, while the second memory module could be accomplishing a read cycle.

The memory implementation techniques for the KL dramatically extend this concept. The memory bus architecture of the KL10 differs from both the KA10 and the KI10 in that its width is four 36-bit words. That is, in an adequately configured memory system, the KL10 actually reads or writes four words concurrently from memory. The four logically successive memory locations (obtained from four physically separate memory modules) are placed in the KL's cache memory, and in typical programs, another core memory reference need not be initiated until the four instructions or data have been processed. Combined with the typical effects of the cache memory, therefore the KL has an effective memory word access time of 334 nanoseconds, using current DIGITAL memory.

Cache Memory

The KL10 features a cache or buffer memory with 160 nanosecond access time. Data being written to or read from memory is typically found in the cache 85 to 90 percent of the time, thus giving the KL10 an effective memory access time of 334 nanoseconds with current DECsystem-10 memories. Another feature of the "state-of-the-art" design cache memory on the KL10 is that it does not require write-through to memory. This eliminates, for example, the necessity of writing back into main memory each value of an index in a loop comprised of only a few instructions.

A cache sweep feature allows all pages or one selected physical page in main memory to be updated from the cache.

Core Memories

	MH 10	MG10	MF10
Word Size	36 bits plus parity	36 bits plus parity	36 bits plus parity
Minimum Memory Size	64K	32K	32K
Maximum Memory Size			
KA10	256K	256K	256K
KI10	4096K	2048K	1024K
KL10	4096K	2048K	1024K
Module Sizes			
32K words	No	Yes	Yes
64K words	Yes	Yes	Yes
128K words	Yes	Yes	No
256K words	Yes	No	No
Read Access Time			
Typical	735	620 ns	550 ns
Maximum	745	660 ns	610 ns
Cycle Time	1.2 μsec	1.0 μs	950 ns
Interleaving	2 or 4 way	2 or 4 way	2 or 4 way
Minimum memory that can be disabled	64K	32K	External Only 32K

Input/Output

Priority Interrupt System

The DECsystem-10 Priority Interrupt System is one of the most flexible available today. Devices are assigned under program control to any one of seven priority levels through the dynamic loading of a 3-bit register within the device. Each interrupt level has any number of high-speed programmable sublevels. Thus, a program can change the priority level of any device or disconnect the device from the system and later reinstate it at any other level. In the same manner, a program can set, enable, or disable, all or any combination of levels with a single instruction. In addition, the program can assign some or all devices to the same level.

A set of instructions (block-in and block-out) allow blocks of information to be transferred between a device and memory with a single instruction. These instructions identify the source of the interrupt, update a word count and data address, transmit or receive the block or information, and dismiss the interrupt.

The system can also generate interrupts through software. Real-time hardware can thus operate on a high-priority level while related computations, particularly if they are lengthy, can be performed on a lower level.

The DECsystem-10 program-assignable priority interrupt system provides much greater flexibility than permanently hard-wired systems. Hard-wired systems require a large number of levels, often operate with an extremely high overhead, and cannot change device priorities without system shutdown and rewiring.

An interrupt on the KI10 or KL10 can cause the processor and the interrupting device to immediately initiate one of several possible actions. In response to the "interrupt grant" signal from the processor, the device may supply a 33-bit word which is decoded as 18 bits address, 12 bits data, 3 bits function. The processor then does one of the following:

- Executes the instruction found at the supplied 18-bit address (KI10 or KL10)
- Transfers a word into or out of the addressed location (KI10 or KL10)
- Adds a signed 11-bit (12 bits+ + 11 bits) value to the addressed location (KL10 only)
- Traps to a predefined memory location (KA10)

Peripheral devices which are not equipped with the decoding logic perform an interrupt and transfer of control, as on the KA10, to one of the standard interrupt trap locations.

Multiplexed I/O Bus

The DECsystem-10 Multiplexed I/O Bus provides a 36-bit full word parallel path between memory and an I/O device for purposes of control or low-speed data transfer. To initiate high speed data transmission directly between memory and a device connected to the memory bus, a control word is first transferred over the I/O bus to the buffer of the high-speed device controller. Then, on command, entire data blocks are moved directly to or from memory with a single instruction. (For a description of the memory bus, see the discussion of Memory Systems.)

The I/O bus may also be used as a control and data path to/from a large number of low-speed I/O devices. Transfer is performed in 36-bit words in parallel at speeds of 200K words/sec on the KA10, and 370K words/sec on the KI10 and KL10. Thus each data transmission instruction moves one word of data between memory and the buffer of the device controller. When block input or output instructions are used, entire blocks of data are moved to or from the device with a single instruction.

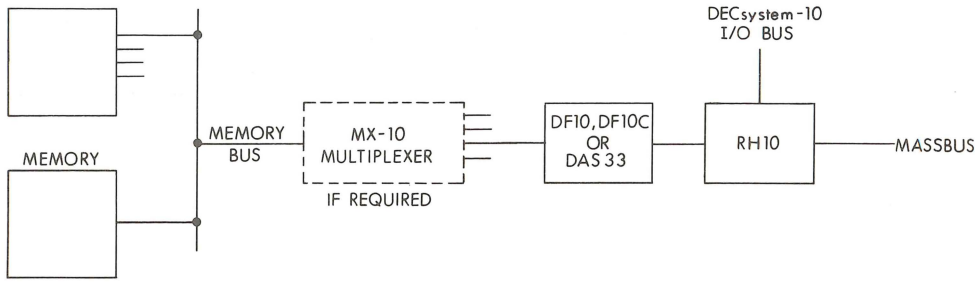
Massbus

The KL10 architecture and implementation now allow use of DIGITAL's latest Massbus controllers, channels, and peripheral devices. Although, via memory bus and I/O bus logic, the KL10 supports traditional peripheral control units, the efficient memory control unit provides paths and logic for transfers into and out of DECsystem-1090 memory. Virtual-to-physical address translation is performed by a 64 x 154-bit RAM. The memory control also includes a series of multiple word channel buffers for high-speed direct I/O over the channel bus to the individual Massbus devices. For each device, there is one RH20 channel containing a 16-word data buffer, a channel command word register, and a control word location pointer—effectively a program counter. The channels perform data transfer by executing channel programs which are set up in memory by device service routines.

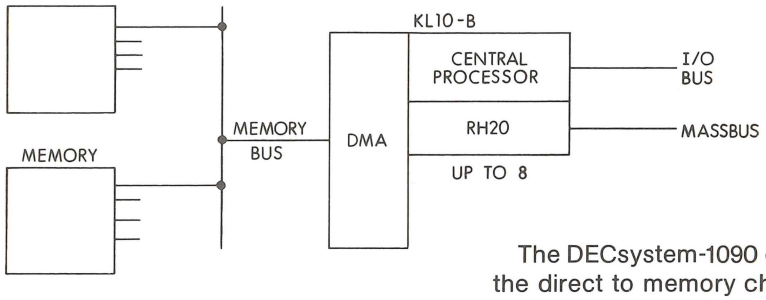
A storage bus provides a 1.6 million words/second transfer bandwidth between the memory control unit and memory. In diagnostic mode, the storage bus is used to access the internal memory controllers. The channel bus is a physically short, high-speed data transfer path providing a peak I/O bandwidth of six million words/second between the memory control unit and the Massbus controllers. The channel bus operates in a synchronous time division multiplexed mode, allowing the connection of multiple controllers to memory.

A high-speed data path (Massbus) connects the integrated channel controllers and mass storage devices, such as disk or tape. Operating either asynchronously or synchronously, the Massbus transfers control and status information or blocks of data between devices and controllers. The UNIBUS, an integral part of the PDP-11 architecture, provides a data and status control path between the Front End Processor and the devices making up the PDP-11 system, including the communications interface and unit record I/O devices.

1040, 1070, 1080 MASSBUS CHANNEL AND CONTROL BLOCK DIAGRAM



1090 MASSBUS CHANNEL



I/O CHANNEL CAPACITY

DECsystem-1050 and 1070 generation equipment implement high-speed data transfers from mass storage devices via a direct to memory channel using the DF10, DAS33 or DX10 channel and an appropriate controller. If the number of channels exceeds the available memory ports then an MX10 multiplexer can be used. In general, capacity is determined by the memory system word transfer rate capacity which will vary based on the type of memory used and the method of interleave. For current memories, four words can be transferred every microsecond in a four-way interleaved configuration. Conflict in memory port contention will reduce the effective word transfer rate somewhat and can result in "overruns" which occur when a rotating mass storage device cannot execute a memory transfer in the required time. TOPS-10 recovers from this situation and simply retries the transfer until it is completed. The DAS33 buffered channel will greatly reduce or eliminate "overrun" capacity loss.

The DECsystem-1090 can use either the direct to memory channels or the internal channels in the KL10B Central Processor. Since the processor is configured on a lower memory priority port than direct to memory channels there is no conflict between these devices. The KL10 cache and internal buffering further reduce any contention. The number of internal high-speed channels which can be operated simultaneously depends on the data transfer capacity between the CPU and Memory. This, in turn, is a function of the memory interleave method and corresponding number of data busses implemented. The following table summarizes this relationship based on RP04 or RP06 device transfer rates.

Memory Interleave and KL10 Bus Mode	1	2	4
Number of Channels Transferring Simultaneously	2	5	6

This table presumes no "overruns". Should these occur they would be handled by TOPS-10 in the same manner as direct to memory channels. It should be noted that for typical data transfer rates of under 1000 blocks per second there is no measurable conflict between instruction execution and I/O transfer.

Console/Diagnostic Computer

One of the more significant features of the KL10 is the PDP-11-based Console/Diagnostic Computer.

The PDP-11 communicates with the KL10 through a DECsystem-10/PDP-11 interface. The interface allows data transfers between the PDP-11 and the KL10 to take place simultaneously. On the PDP-11 side of the interface, data has direct memory access over the PDP-11 UNIBUS in 8- or 16-bit bytes. On the KL10 side of the interface, data may exist in any size byte up to 36 bits in length.

The PDP-11 serves as the console device for the KL10 processor and includes a TU56 DECtape for loading and diagnosing the PDP-11 and back-up for the RP04 which normally loads the KL10's micro-code memory. An LA36 DECwriter-11 provides operator/monitor communication.

As a diagnostic computer, the PDP-11 can examine the data paths and the control logic of the KL10 via a separate diagnostic bus, even if the KL10 is completely inoperative. Other features of the diagnostic computer allow all data busses to be checked. Remote diagnostic checking over telephone connection can also be performed via the PDP-11 computer.

The console/diagnostic computer is a state-of-the-art concept to provide the DECsystem-1090 and DECsystem-1099 with greatly improved mean-time-between-failure and reduced mean-time-to-repair. The KLINIK diagnostic system permits remote diagnosis of system failures via the console/diagnostic computer.

System Integrity Features

Power Fail

If system power fails, a power failure detection circuit senses the condition and causes an interrupt. The interrupt can trigger the operation of a program which saves all valuable registers so that the system can be restarted in a minimum amount of time.

On the KI10 and KL10 (through the PDP-11 console computer) an automatic restart capability has been added to resume normal operations in the event of a power outage. All three phases of AC power are monitored. Low voltage on any phase will initiate a sequence of power-down operations. A program-selectable automatic restart capability is provided to allow resumption of operations when power returns. Alternatively, a manual restart may be used.

Temperature sensors strategically placed within the equipment detect high temperature and low air flow conditions and cause power shutdown. This, in turn, initiates the power failure interrupt.

Error Handling

The TOPS-10 Monitor records errors that occur in the slave CPU (on a dual CPU system) or peripheral subsystems and notifies the operator. If the error can be corrected then users affected simply continue running. Alternatively, the operator may request that the faulty device be detached in which case the user job will be notified and continued or restarted as required. If the error detected is in memory then TOPS-10 will attempt to move its own data, if any, out of the failing memory. In the case of disks, the system will attempt to migrate any swapping space to another unit to prevent loss of jobs. TOPS-10 also provides diagnostics to facilitate on-line maintenance of a particular device concurrent with user mode operation.

Multiprocessor Systems

Dual CPU System

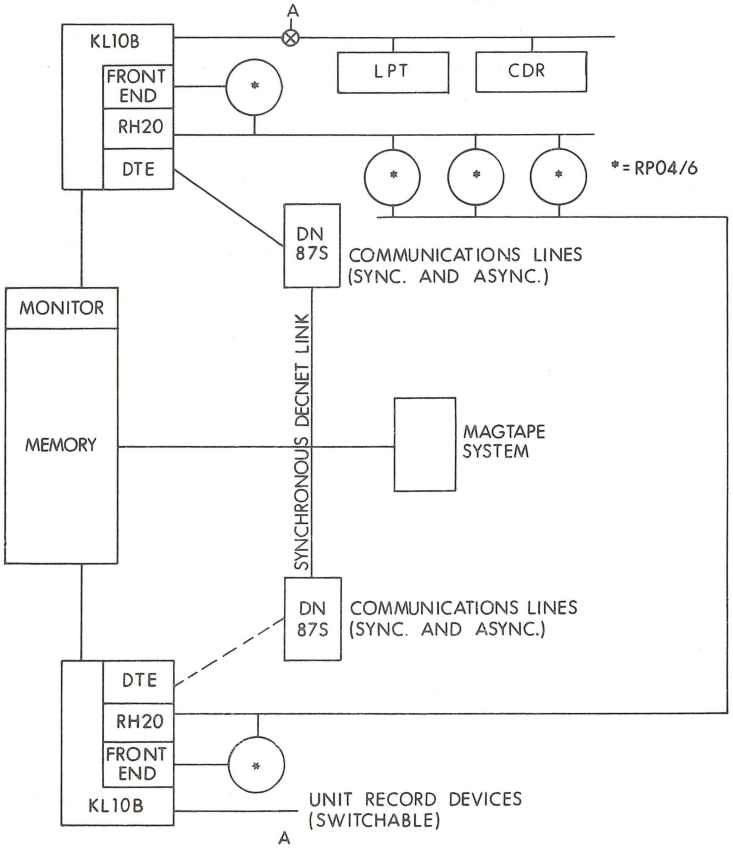
Multiprocessor systems, built around all of DIGITAL's current CPU's, offer attractive, economical solutions needed for capacity and availability.

In these configurations, two processors divide responsibilities as master and slave. The master runs the TOPS-10 operating system and handles all scheduling and I/O. One processor can take over the other's role through the use of strategically-placed bus switching hardware, which allows peripheral devices to be under the control of either processor. If the master malfunctions, the system stops and an operator can have it back into action in under five minutes by reconfiguring all I/O to the slave, which then becomes the master. Users just temporarily have somewhat less computing power to work with, but can still get their jobs done. Minutes after a CPU malfunction that could keep a single system down for hours, the dual processor configuration can be back in action. Redundant disks and communications subsystems can provide increased capacity while all components are fully operational and back-up in case of an isolated controller or subsystem malfunction.

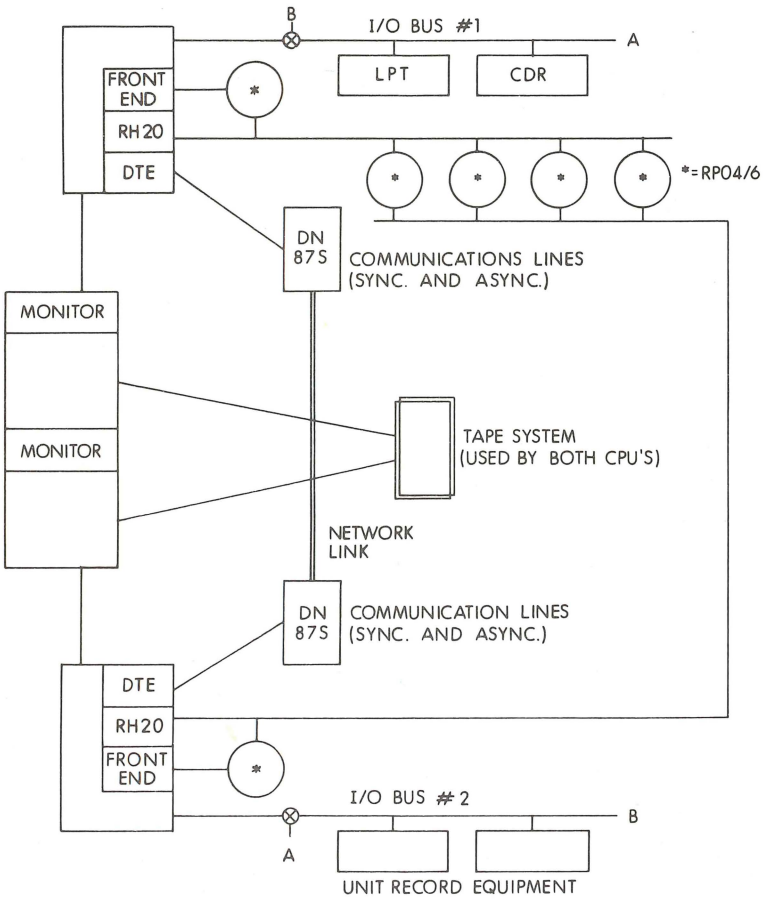
An extremely important and valuable side benefit of the redundant, switched configuration is system maintainability. Malfunctioning subsystems, (such as disk controllers, etc.) which require a CPU to fully diagnose, repair, and certify as operational, may be included in a "maintenance configuration" while the remaining components of the system continue with the production tasks. Users need not suffer from any lack of capability until scheduled maintenance activity or "non-prime time" allows the testing of the complete system. When the subsystem or component is released, fully tested, from Field Service, a scheduled reconfiguration (providing minimum impact on users) can be accomplished, providing full capability again.

Dual CPU/peripheral configurations can be initially installed or, through the addition of appropriate redundant system components and bus switching hardware, can provide attractive upgrade/migration paths for existing systems.

DECSYSTEM 1099 DUAL PROCESSOR SYSTEM



DUAL DECSYSTEM 1090 SYSTEM



Peripherals

Disk Systems	20
DAS33 Buffered Data Channel	22
DECtape Transport	23
Magnetic Tape Systems	24
Hard Copy Equipment	26
Card Readers	26
Line Printers	26
Terminals	28

Disk Systems

There are four disk systems available on the DECsystem-10. The RHS04 system is designed to provide the DECsystem-10 with a fast-access, high transfer rate swapping system which greatly enhances system performance and load handling capability. The DECsystem-10's three disk pack systems offer rapid access, on-line storage, and large capacities at low cost. Modularly expandable, a DECsystem-10 allows up to four controllers each with eight drives giving a total capacity in excess of 6.3 billion ASCII characters. In addition to storage, a disk system may be used as either the primary or secondary swapping system.

The RTP06 is DIGITAL's newest disk system. It operates through a controller which transfers data directly to and from memory via the KL10 internal channels. Since the controller provides overlapped positioner operation, the TOPS-10 operating system will simultaneously position two or more disk drives, shortening the effective access time and increasing throughput.

Each RP04 or RP06 drive configuration can have an optional dual Mass Bus® channel connection which allows access and data transfer via two paths for each individual drive.

The RH04 and RP06 offer a high level of data reliability. A phase locked loop clock system and MFM (modified frequency modulation) recording provide the latest in reliable reading and recording techniques. In addition, error detection and correction hardware provides information as to the position and pattern of any error burst up to 11 bits within the data field. Correction of any data field errors is then accomplished under TOPS-10 software control.

Disks

	RP03	RP04	RP06	RS04
Disk Drive Capacity (m words)	10.24	20.48	40.96	.256
Transfer Rate (μs/word)	15	5.6	5.6	4.0
Access Time: (msec)				
Track-to-track	7.5	7	10	0
Average	30	28	28	8.5
Maximum	55	50	55	8.5
Organization:				
words/sector	128	128	128	128
sectors/track	10	20	20	64
tracks/cylinder	20	19	19	64
cylinders/pack	400	411	815	
Number of Heads	20	19	19	64
Number of Recording Surfaces	20	19	19	1
Number of Disks	11	12	12	1
Number of Drives/Controller	8	8	8	8
Number of Drives/System	32	32	32	16
Maximum Storage/System (b ASCII Chrs)	1.96	3.19	6.30	.004

DAS33 Buffered Data Channel

The DAS33 is intended to replace DF10 data channels in large DECsystem-10 configurations containing many high-speed peripherals where memory transfer throughout is a critical consideration in a peak memory contention environment.

- Hardware and software compatible with the standard Data Channels.
- 256-word data channel buffer reduces memory contention for high-speed, direct-to-memory peripherals, increasing memory transfer throughput.
- Control word and data word "look-ahead" logic
- 22-bit memory addressing (four million words of 36-bit memory)

The DAS33 has been developed by the Advanced Systems Group to reduce peak memory contention and to increase memory transfer throughput for the new generation of high-speed peripherals such as swapping disk pack systems. It is fully hardware and software compatible with the existing data channels. By incorporating a 256-word buffer (equivalent to two disk sectors), block transfers are averaged over available memory bandwidth. This results in a minimization of memory contention during peak memory transfer and an increase in usable memory transfer throughput for high-speed peripherals. Throughput is further enhanced by the "look-ahead" feature for control and data words.

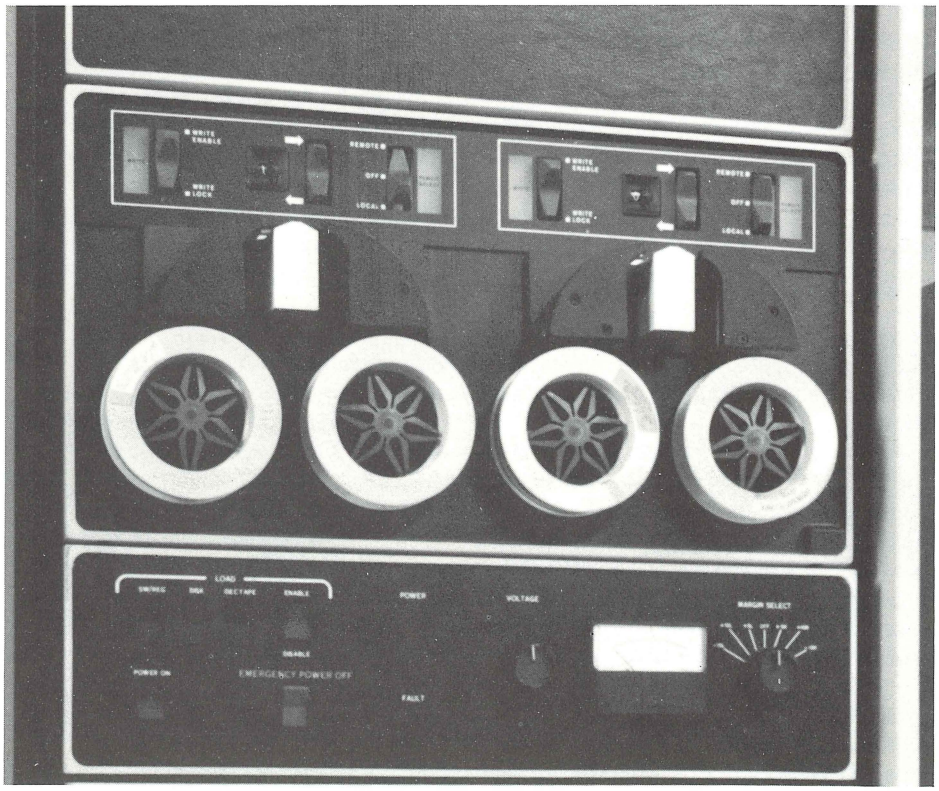
The DAS33 is recommended for large DECsystem-10 configurations having swapping and disk pack subsystems. By minimizing memory contention (with the two-disk sector buffer) and by providing control and data work "look-ahead", the DAS33 will increase the system memory throughput to/from these high-speed devices. Its 22-bit-wide memory addressing but enables I/O controllers to address up to four million (36-bit) words of memory.

The DAS33 includes its own 19-inch cabinet and memory bus cables. One DAS33 is needed for each controller.

DECtape Transport

Digital Equipment Corporation's popular computer data storage medium, DECtape, is available in a dual-transport version. This fixed-address, bi-directional magnetic tape storage system provides random access for high-speed reading or writing of files on 260 foot reels of ¾-inch-wide magnetic tape contained on a reel less than four inches in diameter. Redundant recording (each bit of data is recorded on two separate tracks) assures high reliability and eliminates the need for character parity checking. In addition, block check summing insures data integrity.

DECtape does not require the rewind and count record operations or the writing of separate header records typical of ordinary magnetic tapes. The high reliability and convenient size make DECtape the ideal medium for program storing and transportation.



Magnetic Tape Systems

DIGITAL provides a complete range of magnetic tape systems from the low-cost THU16 to the mid-range TU70/TU71, to the high-performance TU72.

The THU16 subsystem provides 200, 556, 800 bpi (NRZI) and 1600 (Phase Encoded–PE) capability at low cost. With a tape speed of 45 ips, these transports exhibit a 72K character per second maximum transfer speed. For systems with moderate duty requirements, the THU16 is an excellent price/performance choice.

The TU70 series of high-speed magnetic tape drives represents the latest state-of-the-art design in tape transport technology.

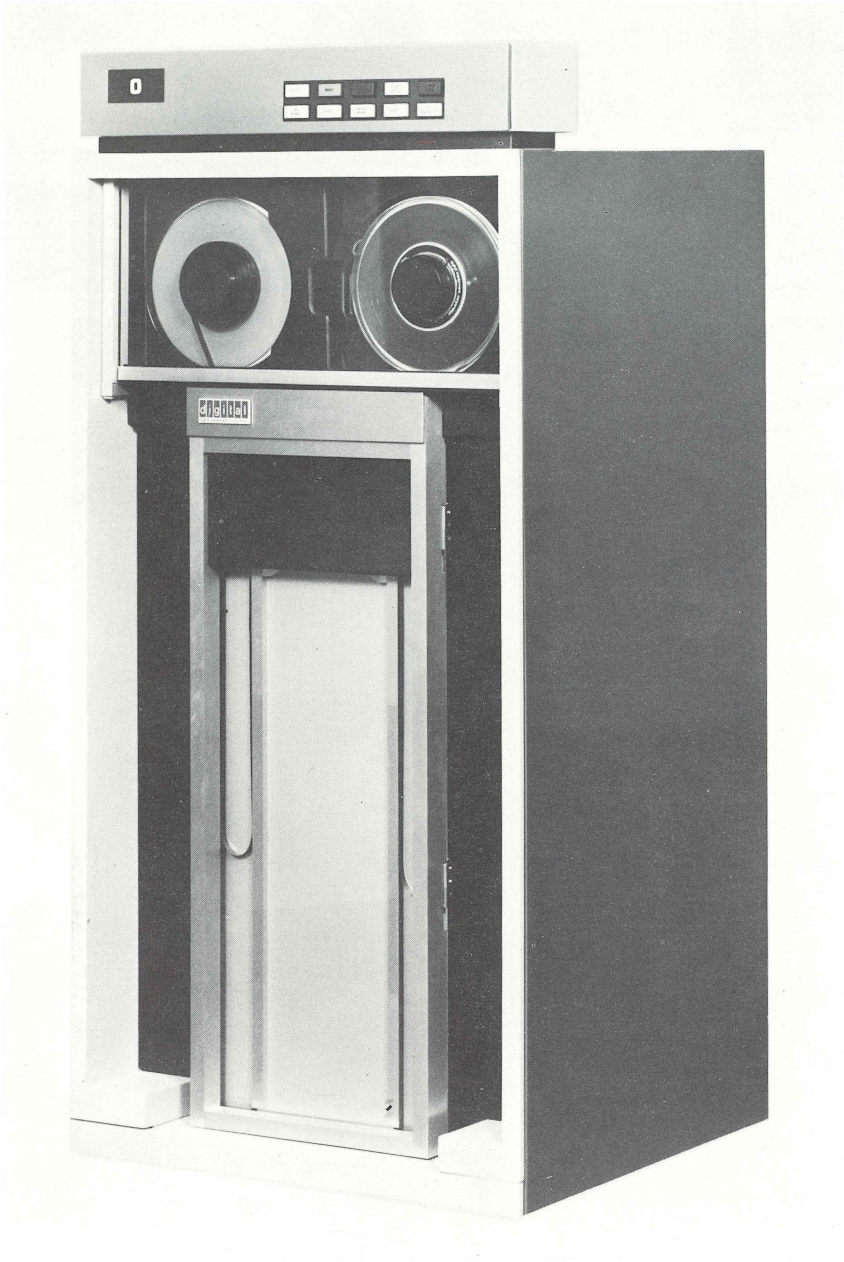
The TU70 provides 800 bpi (NRZI) and 1600 bpi (PE) capability at 200 ips, resulting in a maximum transfer rate of 320K characters per second. The TU71 provides 7-track capability at 220/556/800 bpi (NRZI) at 200 ips for a maximum transfer rate of 160K characters per second.

For very high performance systems DIGITAL offers the TU72, providing faster processing, fewer tapes, faster dumps, and improved error handling for DEC-10 sites. This 6250 bpi drive provides 3 to 1 recording density compression of an equivalent 1600 bpi drive. Using the Group Coded Recording (GCR) technique the higher density provides more capacity per tape, higher data rates and better error correction.

Magnetic Tape Systems

The TU70-series drives incorporate such features as automatic threading and loading of 5-, 8½- and 10½-inch reels as well as industry-standard cartridges. Dynamic amplitude control during read operations improves performance during 1600 bpi operations by reducing read/write errors, avoids the need for preamplifier adjustments, and allows optimum performance with different tape brands on the same drive. All TU70 tape drives are equipped with an automatic reel hub to speed and ease tape loading and alleviate reel slippage. An analog capstan control provides constant motor drive control during read/write and rewind for smoother operation and greater drive-to-drive compatibility, while a velocity feedback reel control provides for protection and velocity sensing of the tape loop in the vacuum columns to minimize reel overspeed; this prevents tape matching, results in longer tape file, and greater machine reliability. A linear high-speed rewind mechanism reduces rewind time and lessens stress on the reel drive system resulting in smoother operation. A tape storage pocket is provided for convenience.

The controller handles up to 8 tape drives and provides direct access to the DECsystem-10 memory via an integrated data channel. The controller includes such features as improved phase-encoded and group coded recording error correction techniques for fewer permanent and temporary read errors. A read-only memory control for incorporation in microprogram diagnostics, monolithic MSI design, for greater reliability, loadable in-line microprogram diagnostics, programmable maintenance memory, adjustment-free read detection, and a radial interface which allows each individual drive to be maintained or physically removed without effecting the operation of other units.

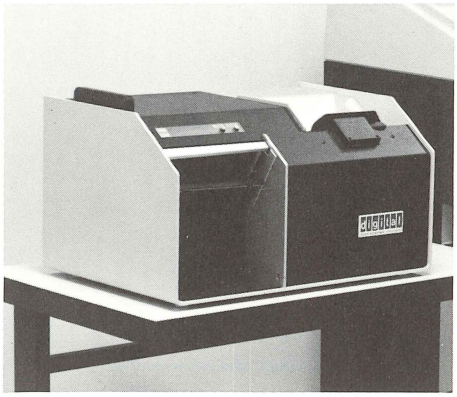


TU70/TU72

Magnetic Tape

	TU10W	TU16	TU70	TU71	TU72
Tape Speed	45 ips	45 ips	200 ips	200 ips	125 ips
Transfer Rate at:					
200 bpi	9 K char/sec	9 K char/sec	–	40 K char/sec	–
556 bpi	25 K char/sec	25 K char/sec	–	111.2 K char/sec	–
800 bpi	36 K char/sec	36 K char/sec	160 K char/sec	160 K char/sec	–
1600 bpi	–	72 K char/sec	320 K char/sec	–	200 K char/sec
6250 bpi	–	–	–	–	780 K char/sec
Recording Technique	NRZI	PE/NRZI	PE/NRZI	NRZI	PE/GCR
Nominal Inter-record Gap:					
9-track	0.6 inches	0.5 inches	0.6 inches	–	0.3 inches
7-track	0.75 inches	–	–	0.75 inches	–
Rewind Time (2400 ft)	195 seconds	3 minutes	45 seconds	45 seconds	55 seconds

Hardcopy Equipment



Card Readers

Designed to meet varying throughput requirements and provide quiet, reliable trouble-free operation, the CR10 card readers accept 80-column EIA-ANSI standard cards. For fast throughput, the user can choose the console model CR10-E which processes 1200 cards per minute or the table model CR10-D which processes 1000 cards per minute. The CR10-F card reader is a lower-cost table model which operates at 300 cards per minute, but utilizes the same excellent features as the high-throughput model.

Line Printers

The LP10-F and LP10-H are drum-type line printers designed with emphasis on output quality and operating reliability. The two-speed capability of the 64-character LP10-F and the 96-character LP10-H increases the printing quality for both low-speed operation and multi-part forms handling.

The LP10-F operates at 1250 or 925 lines per minute and the LP10-H at 925 or 675 lines per minute. Both line printers feature optical vertical format unit using 12-channel tape, and a quick-change drum which allows the character set to be changed in just 5 minutes. Both printers are available with either an EDP or Scientific character set for the quick-change drum.

For user sites requiring high-grade print quality, flexibility of printing fonts, heavy print volumes and high reliability from a horizontal font-motion line printer, DIGITAL offers the LP100. The LP100, an impact printer using a Charaband*® available with a 64-character EDP character set font on one side of the band and a 96 character set on the other. The Charaband* can be reversed by the operator in about one minute. The LP100 stores a stream of up to 132 characters in a print line buffer and upon command prints the contents of the buffer and advances the forms as specified.

* ®Trademark of Dataproducts Corporation

Line Printers

PRINTERS	TYPE	PRINT SET # CHARACTERS	MAXIMUM SPEED LINES/MINUTE	MAXIMUM COLUMN WIDTH	PAPER TYPE	# PARTS OF PAPER FORMS
LP10-F	impact	64	1250/925	132	std.	6
LP10-H	impact	96	925/675	132	std.	6
LP100	impact	64/96	1200/900	132	std.	6

Terminals

Each DECsystem-10 is capable of supporting up to 512 interactive terminals. A wide variety of EIA and 20 mA Current Loop compatible terminals is supported by the DECsystem-10.

The LA36 DECwriter II is designed to be industry's lowest-priced, best performing, most reliable 30-cps impact teleprinter. DECwriter II is packed with capability: it's fast—a true 30-cps machine; it's quiet—you'll almost forget it's there; it has many people-oriented design considerations you wouldn't expect in a cost-conscious product; it's versatile—you can use 6-part forms and even standard 132-column line printer paper; and it's built for a long, reliable lifetime of heavy use.

The LA35 Receive Only terminal is an LA36 without the input key board.

The LA37 APL terminal is really three terminals in one. It provides the user with the choice of either full APL, full upper/lower case ASCII, or upper case TTY in one reliable, easy-to-use hard-copy terminal.

The LA180 DECprinter I is an inexpensive 7x7 dot matrix 132 column printer that prints at 180 characters per second. It features the 128 character ASCII upper-lower case print set, an extensive array of human engineering features, and may be used either locally or as a remote printer hard copy unit.

The DECprinter I extends the field-proven technology of the LA36 DECwriter II into applications demanding higher speed capabilities.

Hardcopy Terminals

CHARACTERISTICS	LA35 RECEIVE ONLY	LA36	LA37APL	LA180 RECEIVE ONLY
Print speed (chars./sec.)	10/15/30	10/15/30	10/15/30	up to 180
Print positions per line	132	132	132	132 (with 133 bit buffer)
Character set	96 ASCII upper/ lower case	96 ASCII upper/ lower case	Full APL; Full ASCII upper case TTY	128 ASCII upper/ lower case
Character formation	7x7 dot matrix	7x7 dot matrix	7x7 dot matrix	7x7 dot matrix
Interfaces	20mA standard; EIA optional	20mA standard; EIA optional	20mA standard; EIA optional	Parallel
# of copies	Original +6	Original +6	Original +6	Original +5
Purpose of Unit	A hardcopy output device (e.g. CRT display attachment)	Interactive hard copy terminal	APL Terminal	A medium-speed hard copy unit
Comments	Basically an LA36 without keyboard			Prices include controller/interface to attach to PDP-8 or PDP-11

DECscopes—DIGITAL's VT50 series of character oriented video terminals—eliminate the need for electromechanical devices. They are all simple for operators to master, quieter than mechanical terminals, and inexpensive. The three models—VT50, VT50H, and VT52—differ only in advanced application features.

The VT61 is the first DECscope that can transmit in either block or character mode. It offers over 80 commands, including character and line insertion and deletion, text justification, reverse video and protected forms, and a number of other editing features not found on the VT50 and VT52 DECscopes.

The VT61 is an upper and lower case ASCII video terminal with a 1920 character display and 19-key function pad. However, the VT61 uses a faster microprocessor with larger memory than do the other DECscope models, so it can offer more stand-alone capabilities.

CRT Display Terminals

CHARACTERISTICS	VT50	VT50H	VT52	VT61
Screen format (lines and chars./line)	12x80 (960 chars.)	12x80 (960 chars.)	80x24 (1920 chars.)	24x80 (1920 chars.)
Character matrix	5x7	5x7	7x7	7x8
Character set	64 ASCII upper case	64 ASCII upper case	96 ASCII upper/ lower case	128 char. upper/lower
Keyboard	Standard typewriter	Standard typewriter plus 19-key pad (10 numeric, enter, period & 3 unlabeled function keys)	Standard typewriter plus 19-key pad (10 numeric, enter, 4 censor, period & 3 unlabeled function keys)	DEC standard +19-key pad + 12 user defined keys
Communications interface	20 mA std., EIA optional	20 mA std., EIA optional	20 mA or EIA standard	20 mA std., EIA optional
Data rates (bps)	75-9600	75-9600	75-9600	75-9600
Editing capabilities				
–Erase to end of line	yes	yes	yes	yes
–Erase to end of screen	yes	yes	yes	yes
–Downward scroll	no	no	yes	yes
–Upward scroll	yes	yes	yes	yes
–Insert/delete character	no	no	no	yes
–Insert/delete line	no	no	no	yes
Cursor control	Cursor movement via ESCAPE sequences	Direct cursor addressing	Direct cursor addressing plus cursor keys	Direct cursor addressing
Optional features	No optional attachments	No optional attachments	Electrostatic copier, LA35RO	LA35 LA180 202T modem interface

DATA COMMUNICATIONS SYSTEMS

The DECsystem-10 Data Communications System unifies software and hardware into a capability designed and implemented for flexibility and ease of use. The system provides a wide spectrum of capability to distribute computing power to a continually-growing user community. In concept, the DECsystem-10 Communications System provides a transparent link between remote users and the central site. Because of this transparency, users, at any location, may work with the same facility as those physically located at the central site, with the same set of commands and machine resources used at the central site.

DECsystem-10 Data Communications also links networks of computers together from both local and remote sites. Data collection stations, remote control stations, remote concentrators, and remote job entry stations are easily tied into a single data network.

An important features of these capabilities, which translates into a direct benefit to users in ease-of-use, is that communications is handled as an integral subsystem of the DECsystem-10 computer system. Communications protocol is handled by the communications front end system, fully in league with the TOPS-10 Operating System. Users need not concern themselves with a "tacked on" communications handler; TOPS-10 interfaces directly with the user environment in a transparent fashion. The powerful and flexible communications capability of the DECsystem-10 is just one of the features of TOPS-10. Users may design and implement their systems by concentrating on the application; the DECsystem-10 Communications products take care of the communications problems.

System Concepts **34**
Asynchronous Communications 34
Synchronous Communications 34
Synchronous vs. Asynchronous Transmission 35
Remote Communications 36
Network Concepts **37**
Simple Topologies 37
Complex Topologies 37
Task-to-Task Communication 38
IBM Device Emulation and Termination 38
Communications Products **39**
DN87/87S Universal Synchronous/Asynchronous Front End Subsystem 39
Complex Topologies 42
DECnet-10 42
DAS80 Series Remote Station 43
DAS92 Remote Station 45
DAS 61 IBM 2780/3780 Support 46
DAS 62 IBM HASP Multi-leaving Support 46

System Concepts

Asynchronous Communications

Asynchronous communications equipment, with associated terminals, serve a broad spectrum of user needs within the DECsystem-10 interactive environment. These needs include interactive program development, operator control of the system, program production and control, data entry, program and data preparation, interactive problem solving, student instruction, and information storage and retrieval, to name a few. Asynchronous communications is used to link the DECsystem-10 with terminal equipment of three types: hard copy, such as the LA37 or LA36 DECwriters, CRT terminals such as the VT52 DECscope, and buffered terminals, such as the new VT61.

Local terminals within 1500 feet of the computer site can be connected over dedicated hard-wired lines. Remote terminals located beyond this distance are connected over dedicated or dial-up telephone lines.

Speeds of asynchronous hard-copy terminal equipment generally range from 10 to 30 characters per second. CRT terminals range in speed from 10 to 960 characters per second, depending upon their communications interface and type of line used for transmission.

The goal of an interactive terminal system is to match all users to the system in such a way that they feel the terminal is identified closely with the computer. In this way, the machine becomes an easy and natural extension of each application. To do this, the computer must be a willing partner in the interchange so that the user feels that he is in control of the terminal, and not the reverse. Typically, a user types input sporadically, occasionally makes mistakes, and reads fast on output. The Data Communications system for the DECsystem-10 has been engineered to deal with these user characteristics.

The system, which includes hardware and software elements, accepts characters at uneven rates, allows for single character or line correction of mistakes, and allows the user to type at his own speed—even typing ahead of the computer's response to his characters. When users are entering large volumes of data and need no prompting, the system accumulates data and anticipates ongoing entries so that the user can continue to type despite fluctuations in the system's response.

Features of DECsystem-10 asynchronous communications include the ability to delete one or more characters, delete entire lines, retype lines and characters, interact on a character or complete line basis (depending upon the application), suppress unwanted output, and make use of prompting messages to call for input. Terminals may communicate with the system administrator or operator at the central or remote sites as well as with other terminals.

Synchronous Communications

DECsystem-10 synchronous communications provides error-free, high-speed paths between the central computer and remote stations or other computer systems. DECsystem-10 synchronous equipment controls transmission over high-speed synchronous lines. This transmission is on a message basis in contrast with the character-by-character basis of lower-speed, asynchronous transmission.

Full duplex protocol software supplied with the DECsystem-10 makes efficient use of high-speed transmission in both directions simultaneously on a full-duplex line. Front end hardware and software handles hardware control, message formatting, and message acknowledgements. The data transmission is "pipelined", a technique which increases line efficiency by fully overlapping the acknowledge-continue signals.

Transmission errors are detected using block-oriented techniques including longitudinal data checks. Data errors are corrected through retransmission of the erroneous block. Hardware is available to interface to a broad range of communications modems.

Synchronous vs. Asynchronous Transmission

Asynchronous transmission is employed to transmit data on a character-by-character basis; synchronous transmission is preferred for moving large blocks of data. Asynchronous transmission, at 2400 baud, is limited to conveying no more than 240 eight-bit characters per second; synchronous transmission, at 2400 bits/second, can convey 300 eight-bit characters per second. This difference comes about because synchronous transmission uses independent, block-oriented clock synchronization, while asynchronous transmission re-establishes its character-oriented clock synchronization with each character. Thus, in the asynchronous case, each eight-bit character requires, minimally, one start bit and one stop bit.

The most important distinction between asynchronous and synchronous methods of transmitting data is the means of error detection and recovery. Error detection for asynchronous devices is normally accomplished by insertion of a parity bit into the data field of each character transmitted. Half-duplex (local character echo) receivers will typically check this parity bit and notify the sender of the existence of an error. In full-duplex systems, such as the DECsystem-10, the character echo is generally provided by the communications computer (DN87, DAS80, etc.) to which the user is immediately attached. In this case, transmission error detection and corrective action are thus the responsibility of the sender, who simply compares what was sent with what the computer says it received. The communications computer which echoes the character to the user then guarantees that it will deliver that character, without error, through the network, to the appropriate DECsystem-10 host.

This mode of error detection is highly efficient and fully acceptable for interactive, terminal-oriented systems, but is less efficient and less than acceptable for message-oriented communications with other computers. Synchronous communication normally uses a block-oriented error detection technique, such as the Cyclic Redundancy Check (CRC-16) polynomial. This technique is most efficient for block-oriented data transmission and for this reason, communications with remote computers is accomplished using synchronous data transmission.

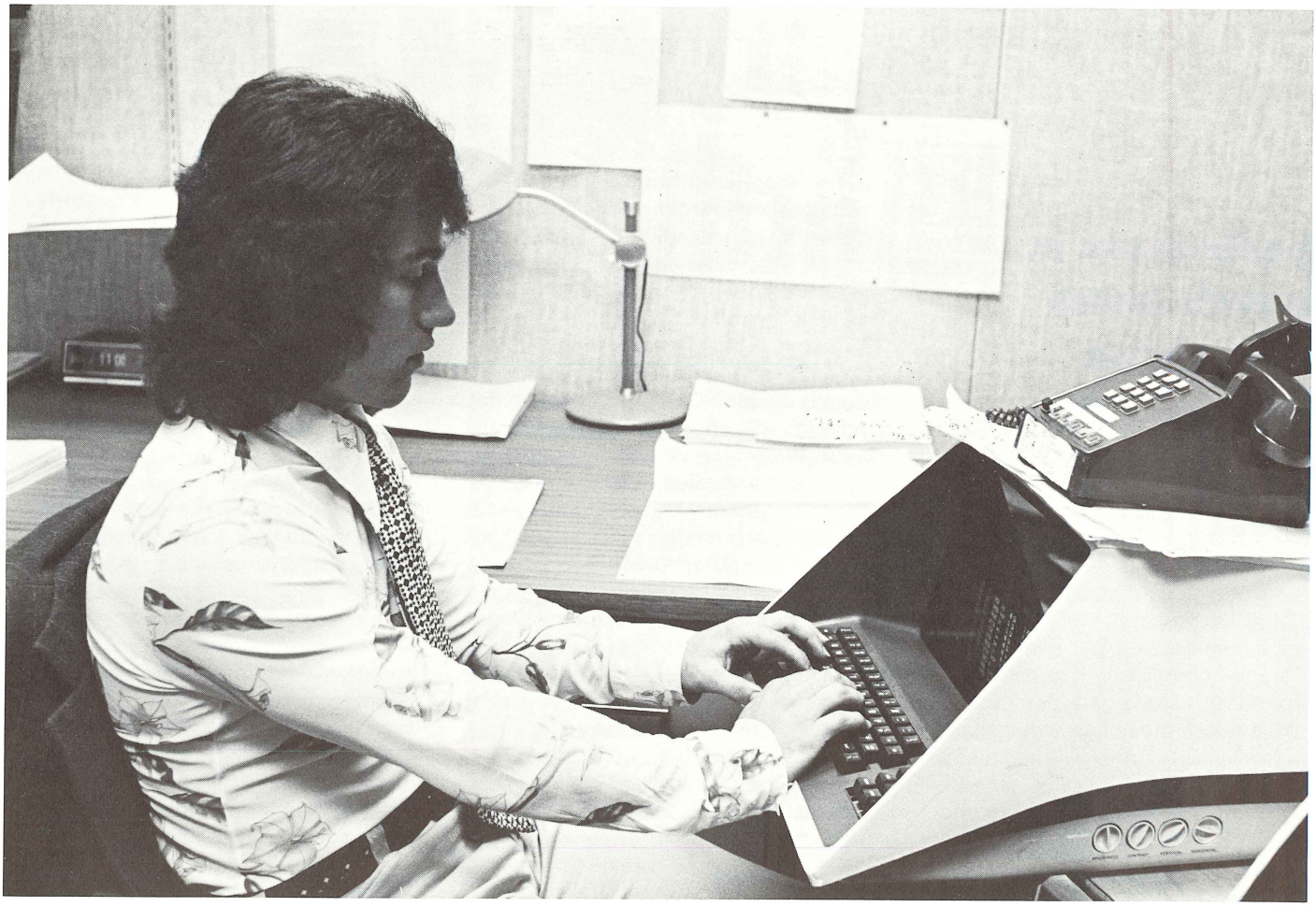
Remote Communications

Prior to the current line of DECsystem-10 remote communications products, each remote user had been individually linked to the computer center by separate long-distance telephone lines.

The only device that the remote user had available at his location was the terminal; he was able to utilize available devices at the central station, but he could not obtain output other than his terminal output at the remote site. Either the user had to travel to the central station to obtain a listing, or he had to have the listings delivered to him. However, with remote communications hardware and software, listing files and data can be sent via a single synchronous full-duplex line to a small remote computer. That remote computer in turn services many remote peripherals, including terminals, card readers and line printers. This eliminates the need for the user to travel to the central site to obtain his output. The remote computer and its associated peripherals constitute a remote station.

Remote station use of the central computer is essentially the same as local use. All sharable programs and peripherals available to local users at the central computer station are also available to remote users. The remote user specifies the resources he wants to use and, if available, these resources are then allocated in the same manner as to a local user. In addition to utilizing the peripherals at the central station, the remote user can access devices located at his station or at another remote station. Local users at the central station can also make use of the peripherals at remote stations. Therefore, by specifying a station number in appropriate commands to the operating system, each user of the DECsystem-10 is given considerable flexibility in allocating system facilities and in directing input and output to the station or DECsystem-10 of his choice.

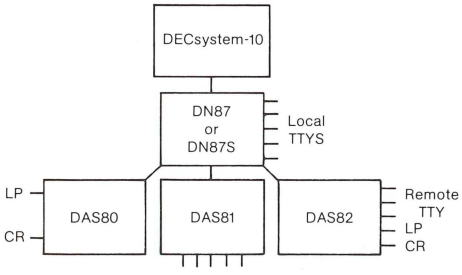
The DECsystem-10 allows for simultaneous operation of multiple remote stations. Software provisions are incorporated in the operating system to differentiate one remote station from another. By utilizing peripheral devices at various stations, the user is provided with increased capabilities. For example, data can be collected from various remote stations, compiled and processed at the central station, and then the results of the processing can be sent to all contributors of the data.



Network Concepts

Simple Topologies

The most basic network topology is a point-to-point, or star topology. It is best illustrated by the following diagram.

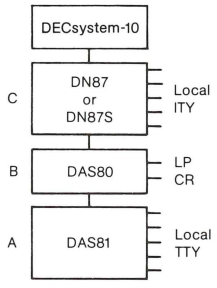


Complex Topologies

This is the name given to a group of general network building capabilities which, when put together, allow the user to make many different cost, performance and high availability trade-offs so that his network may be configured in a way which is very much optimized for him and therefore very favorable to his environment. These building block capabilities are:

Route-thru

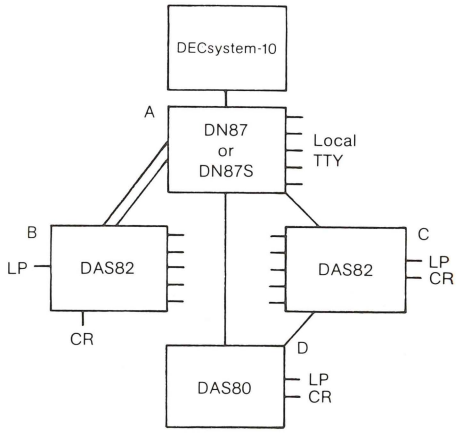
This is the ability for a Remote Station to send information to a DECsystem-10, via the Universal Front End, to which it is only indirectly attached—i.e., indirectly through another Remote Station or Universal Front End. This is most easily illustrated in the following picture:



Information flows from A to C only by being routed through B. This is a very useful price performance trade-off since the cost of a communications line from A to B and from B to C may be substantially cheaper than the cost of lines from A to C and from B to C.

Multi-pathing

This is the ability to have more than one path between nodes in the network. In general, these links are automatically load leveled depending upon the relative speed and error rate of the lines. This is best illustrated by the following diagram:



The two links between A and B will be automatically load leveled with data flowing through each link. This may be a desirable configuration for high availability systems with, for example, one link via microwave and one by ground link. This would allow degraded operation if either link went out for any reason. There are also two routes from D to A. The direct link could be a high speed link, while the link routed through C could be lower speed and used as hot backup to the direct link. Again, both of these configurations have desirable benefits that the user, in building his network, can pick and choose from depending upon his specific needs and desires. The ability exists to pick and choose rather than having a very rigid set of network configuration rules.

Multiple Host Support

If there is more than one DECsystem-10 in a network, the Universal Front Ends on the DECsystem-10s may be hooked up to each other via a synchronous link, or the Remote Stations may be hooked up to each Universal Front End. This allows Teletype users anywhere in the network to select on an individual basis, before they LOGIN, which DECsystem-10 they wish to LOGIN to. This is accomplished by means of the SET HOST command.

Dynamic Topology

This feature allows individual nodes in a network to go down or come up without affecting the remainder of the network. The exact implications of an individual node going down depend on the type of node and its network functions, but any new node may be added to the network dynamically.

Task-to-Task Communication

DECnet-10 provides the capability to have tasks or jobs in different and geographically removed processors communicate with each other in a straightforward manner. These jobs appear to each other as mere I/O devices much like a magtape. Jobs in different DECsystem-10s may communicate with each other, or DECsystem-10 jobs may communicate with other DECnet systems, such as RT-11, RSTS, RSX-11 M, D, & S, IAS, and RTS-8 when they are equipped with DECnet software.

This interprocessor communication allows only for jobs to talk to each other, not for terminal concentration. It does not directly allow for remote file access, remote device access or program sharing. It does, though, present a mechanism or a building block which could be used to provide those features.

IBM Device Emulation and Termination

Through the use of IBM device *emulation* the DECsystem-10 can be attached to an IBM 360/370 computer via a synchronous link. The DECsystem-10 will be treated by the IBM system as if it were an IBM 2780 or 3780 or HASP multi-leaving remote station. This facilitates job submissions and file transfers back and forth between the two computer systems.

Through IBM device *termination* IBM 2780's, and HASP multi-leaving work stations can be used as Remote Job Entry stations to enter batch jobs into the DECsystem-10; batch jobs may be placed in the card reader of the 2780, 3780 or HASP work station, read into the DECsystem-10, executed and have their output returned to the 2780, 3780 or HASP work station line printer.

Communications Products

DN87/DN87S Universal Synchronous/ Asynchronous Front End Subsystem

The DECsystem-10 data communications products are based on the DN87 Universal Synchronous/Asynchronous Communications Front End Subsystem. The DN87 is configurable in asynchronous only mode (up to 112 asynchronous lines), synchronous only mode (up to 12 synchronous lines) or with a mixture of synchronous and asynchronous lines in the same DN87. These three modes allow the DN87 to be configured very cost effectively in a wide variety of customer specific configurations. This great flexibility is the essence of the DN87 Communications Subsystem.

The DN87 is capable of synchronous communication with the DAS80 series Remote Stations, the DAS92 Remote Station and the DC72NP Remote Station. These allow Remote Job Entry, Remote Concentration of interactive Teletype lines or a combination of the two. The DN87 with these Remote Stations supports complex topologies such as multi-pathing, route-thru, and multiple host support. With DECnet-10, it is possible to have a DECsystem-10 communicate with any DECnet system—for example RSX-11M, RSX-11D, RSX-11S, IAS, RT-11, RSTS/E, RTS-8, DECsystem-10 or DECSYSTEM-20 if equipped with DECnet software.

The DN87S has the same functionality as a DN87 except that it is attached to the DECsystem-10 via the DTE10 interface rather than the DL10. Up to three DN87S's may be attached to the DTE10 interface. The DN87 and the DN87S require TOPS-10 version 6.03 or later monitor release.

Asynchronous Functionality

On an eight-line group basis, the DN87/87S is capable of terminating 20 mA current loop, local EIA or EIA with full modem control type of asynchronous line/terminal interfaces. On an individual line basis the DN87/87S asynchronous lines can be:

- ASCII Teletype-compatible code or 2741 EBCD or Correspondence Code.
- Full duplex with echoplex (i.e. echo generated by computer) or full-duplex with local copy (simulated half-duplex).
- Program selectable line speeds of from 50 through 9600 baud.
- Split transmit/receive speeds.
- Automatically baud rate detected for 110, 134.5, 150 and 300 baud lines.

Some off-loading of the DECsystem-10 host is accomplished, in that the DN87/87S does the majority of the echoing for asynchronous lines. It does not echo special characters, nor does it echo when the user is in character-at-a-time mode (e.g., DDT—when an individual character can be a command). The DN87/87S also does fill character generation.

Synchronous Functionality

The DN87/87S is capable of terminating EIA and/or current-loop type synchronous links. The line speeds may be 2400, 4800, 7200, 9600, 19.2K, 38.4K or 40.8K baud on an individual line basis. These links operate only in full-duplex with simultaneous bidirectional transmission. The synchronous links use DDCMP protocol for error checking and correction and for point-to-point link control.

These synchronous links communicate only with the DC72NP Remote Station (DC72NP is a software only upgrade of the DC71 or DC72), the DAS80 Series Remote Stations, the DAS92 Remote Station or another DN87/87S. On an individual synchronous line basis the DN87/87S can also communicate on a task to task basis with systems running DECnet software.

Options List

The DN87 is the Universal Synchronous/Asynchronous Front End Communications Subsystem. Software to provide complex topology support is included with the DN87. The DN87 requires the addition of DN81-xx synchronous and/or asynchronous line options.

DN87-DA, DB—DN87 including DL10-C port only. Requires DL10-A with available port. Note that this is what would be ordered with a 1080 or 1060 System Package, which includes a DL10.

DN87-AA, AB—DN87 including DL10-A Communications Interface and one DL10-C port.

DN87S-AA, AB—DN87 with DTE10 interface.

ASYNCHRONOUS LINE OPTIONS

DN81-EA, EB—Asynchronous Expansion Cabinet including one DN81-EC 16-Line Asynchronous Expansion Group. Requires two DN81-Fx 8-Line Terminator Groups to activate the lines.

DN81-EC, ED—Asynchronous 16-Line Expansion Group. This requires two DN81-Fx 8-Line Terminators to activate the lines.

DN81-FA—8-Line Terminators each with 20 mA Current Loop Local Interfaces.

DN81-FB—8-Line Terminators each with EIA Local Interfaces.

DN81-FC—8-Line Terminators each with EIA Full Modem Control Interfaces.

DN81-FD—8-Line Terminators with Integral Auto Answer Modems. (These modems require customer supplied DAA's).

SYNCHRONOUS LINE OPTIONS

DN81-EE, EF—Synchronous Expansion Cabinet. Includes one DN81-H.

DN81-H—Synchronous Line Controller Expansion Line. For data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.

DN81-J—Synchronous Line Controller. For data transmission speeds of up to 40.8K baud and for attachment to 303-type current mode modems.

NOTE: Option designations are such that if there are two designations the first is 115V/60Hz and the second is 230V/50Hz. If there is only a one option designation that option is NOT power dependent.

Hardware Configuration Guidelines

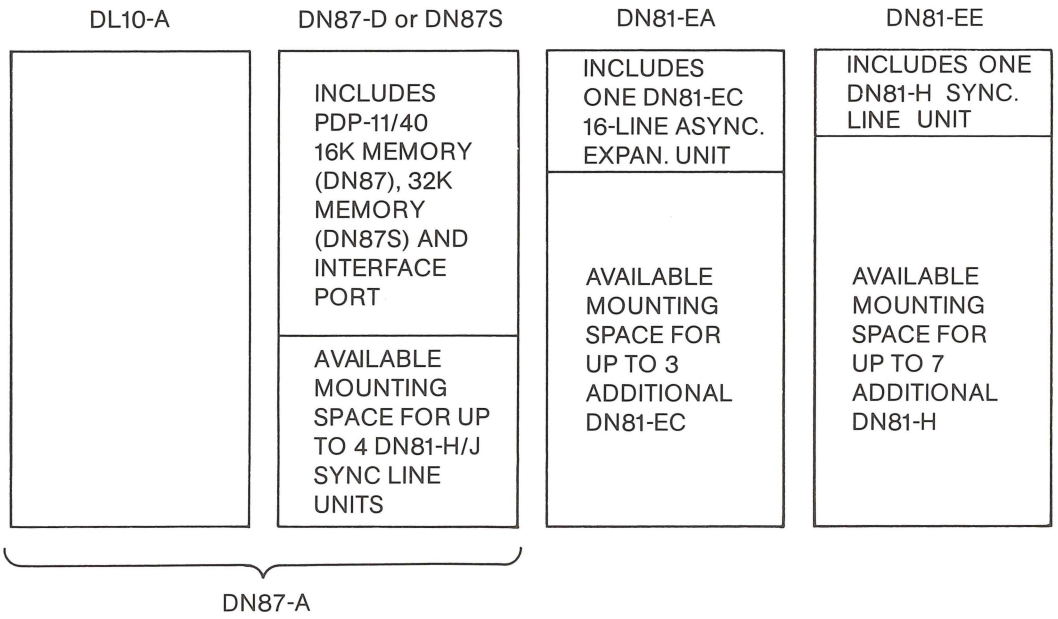
A pictorial representation of the DN87 cabinet arrangements should help to simplify the explanation of the configuration rules.

Asynchronous Configuration Rules

- The DN87 cabinet (DN87-A or DN87-D) has no available mounting space for asynchronous lines.
- If any asynchronous lines are required a DN81-EA Asynchronous Expansion Cabinet must be ordered. This cabinet includes one DN81-EC Asynchronous 16-Line Expansion Group and has available mounting space for up to three additional DN81-EC units. Therefore, the DN81-EA is capable of containing up to 64 asynchronous lines.
- A maximum of two DN81-EA Asynchronous Expansion Cabinets are permitted per DN87.
- Each DN81-EC Asynchronous 16-Line unit requires two DN81-Fx (FA, FB, FC or FD) 8-Line Terminators to activate the lines.

Synchronous Configurations Rules

- The DN87 cabinet (DN87-A or DN87-D) has available mounting space for up to four DN81-H/J synchronous line units.
- If more than four Synchronous Line Units are required a DN81-EE Synchronous Expansion Cabinet must be ordered. This cabinet includes one DN81-H Synchronous Line Unit, and available mounting space for up to seven additional DN81-H units.



LINE MAXIMUMS PER DN87/DN87S

Max. No of Sync. Lines	Max. No. of Async. Lines
0	112
4	64
8	32
12	0

- Proper number of asynchronous lines are made up using the Asynchronous Configuration Rules from above.
- Proper number of synchronous lines are made up using the Synchronous Configuration Rules from above.

Complex Topologies

Complex topologies are available to all DECsystem-10s using the DAS80 Series Remote Stations and the DN87 Front End (includes DC75NP and DAS85) with the 6.03 version of TOPS-10. This means that route-through, multpathing, multiple host support and dynamic topologies (SET HOST monitor command) are all a standard part of the 6.03 monitor and available to those who have the appropriate communications hardware and software configurations.

DECnet-10

DECnet is the set of software products which extend various DIGITAL operating systems so they may be interconnected with each other to form computer networks. The DECnet user can configure a variety of networks, satisfying a variety of constraints, by choosing the appropriate CPU's, line interfaces (and speeds) and operating systems software.

Digital Network Architecture, implemented across a wide range of operating systems and hardware architectures, enables users to build a variety of networks.

DECnet-10 provides interprocessor task-to-task communications for the DECsystem-10. Thus users may communicate on a program to program basis with any other DECnet system through the DN87 front end. This will include RSX-11S, RSX-11M, RSX-11D, IAS, RT-11, RSTS, RTS-8, DECSYSTEM-20 and other DECsystem-10s.

DAS 80 Series Remote Station

The DAS80 Series Remote Stations are a family of remote stations with a wide range of functionality. The DAS80 is a Remote Job Entry Station with a 300 cpm Card Reader, a 300 lpm Line Printer and an LA36 Console. The DAS81 is a Remote Concentrator capable of concentrating up to 32 asynchronous lines. The DAS82 is a combination of the DAS80 and DAS81, facilitating both RJE and Remote Concentration.

When linked to a DN87/DN87S Universal Front End the DAS80 Series Remote Stations will support complex topologies such as route-thru, multi-pathing, and multiple host support.

Asynchronous Functionality

The asynchronous terminal lines attached to the DAS80 Series Remotes have the same functionality as the asynchronous lines attached directly to the DN87/DN87S.

Synchronous Functionality

The synchronous line is used as a full duplex dedicated line to any one of a DC75NP, DAS85 or DN87/DN87S DECsystem-10 Front Ends. The DAS80 Series Remote Stations are all downline loadable from the DECsystem-10 through the synchronous link.

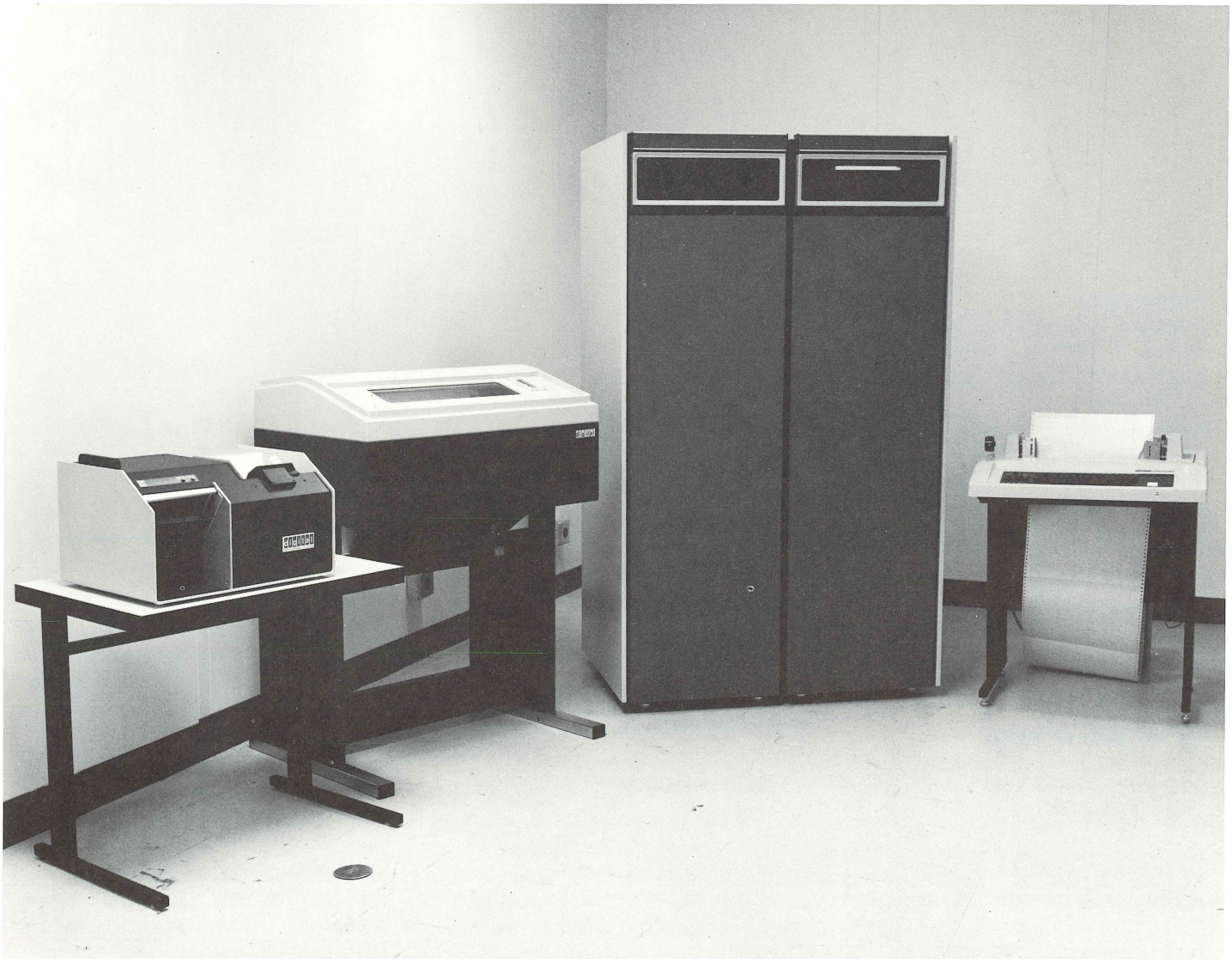
Unit Record Devices

The Line Printer (LP) and the Card Reader (CR) on a DAS80 or DAS82 Remote Station are available from the DECsystem-10 host in a transparent manner. This means that any user at any terminal can use this remote LP or CR very similarly to how he would use a LP or CR attached directly to the host. This is accomplished by use of the DECsystem-10 Monitor Command LOCATE which sets the default LP and CR for a specific user.

The LA36 DECwriter, which is included with the DAS80 Series Remotes, can be used as the Operator Controlling Console for the spoolers which are running for the LP and CR at that station. This means that any operator-type requests (e.g., LP out of paper) appear at the remote location with the LP and CR rather than at the host.

Host Interfaces

The DAS80 Series Remote Stations can only communicate with the DECsystem-10 via the DC75NP, the DAS85 or the DN87/DN87S Front Ends. The DN87/DN87S requires the 6.03 or later version of the TOPS-10 monitor, while the DC75NP and the DAS85 require the 6.02A or later monitor. The DC75 Front End will not run under 6.02 or later monitors unless it is equipped with the "NP" upgrade (additional 4K memory, KG11 CRC unit and software).



OPTIONS LIST

DAS80-AA, AB—Remote Batch Station Includes 1 DN81-H, LA36 Console, DAS80-CA/CB, DAS80-LA/LB, PDP-11 and software.

DAS81-AA, AB—Remote Concentrator Includes PDP-11, 1 DN81-H, DN81-EA and software. One more DN81-EC can be added for a total of 32 asynchronous lines.

DAS82-AA, AB—Remote Batch Station and Concentrator. Includes PDP-11, LA36 Console, DAS80-CA/CB, DAS80-LA/LB, DN81-H, DN81-EA and software. DN81-Fx asynchronous interfaces are also required and must be ordered separately.

UNIT RECORD OPTIONS

DAS80-CA, CB—Card Reader, 300 cards per minute.

DAS80-LA, LB—Printer, 300 lines per minute, 132 printing positions, 64 printing characters, EDP character set.

DAS80-LC, LD—Printer, 230 lines per minute, 132 printing positions, 96 printing characters, EDP character set.

ASYNCHRONOUS LINE OPTIONS

DN81-EA, EB—Asynchronous Expansion Cabinet including one DN81-EC 16 Line Asynchronous Expansion Group. Requires two DN81-Fx 8-Line Terminator Groups to activate the lines.

DN81-EC, ED—Asynchronous 16-Line Expansion Group. This requires two DN81-Fx 8-Line Terminators to activate the lines.

DN81-FA—8-Line Terminator each with 20mA Current Loop Local Interfaces.

DN81-FB—8-Line Terminators each with EIA Local Interfaces.

DN81-FC—8-Line Terminators each with EIA Full Modem Control Interfaces.

DN81-FD—8-Line Terminators with Integral Auto Answer Modems. (These modems require customer supplied DAA's).

SYNCHRONOUS LINE OPTIONS

DN81-H—Synchronous Line Controller Expansion Line. For data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.

DN81-J—Synchronous Line Controller. For data transmission speeds of up to 40.8K baud and for attachment to 303-type current mode modems.

Note: These same line options (DN81-xx) are used on the DAS80 series remote stations, as described here, on the DN87 and for DC76 add-ons.

DAS 92
Remote Station

The DAS92 Remote Station is a low cost remote job entry concentrator that is attached to the DECsystem-10 via one synchronous link to a DN87/DN87S front end. The DAS92 may be configured in one of the following manners:

1. up to 16 asynchronous terminal lines
2. up to 12 asynchronous terminal lines plus a card reader *or* line printer
3. up to 8 asynchronous terminal lines plus a card reader *and* line printer

Only one synchronous interface is permitted on the DAS92, therefore the DAS92 must be a sequential node. This means it may not do multi-pathing or route-thru. Data to the DAS92 may be routed from some other node, but no data may be routed through the DAS92 since it is only capable of having one synchronous link.

Options

1. DAS92-AA, AB Basic Unit, including 16K, PDP-8/A processor, VT52 console terminal, desk, ROM for down-line loading, one (1) synchronous line interface and software. Requires at least one (1), but a maximum of five (5) optional units listed below:

(a) DAS92-EA Asynchronous 4-line multiplexer, including line drivers. The DAS92-EA Asynchronous 4-line Multiplexer will accommodate either 20 ma. or EIA lines in any mixture. One out of every four can be full modem control. The baud rate is switch selectable anywhere from 50 to 4800 baud. Maximum of four (4) DAS92-EA's per DAS92-A.

(b) DAS92-CA, CB Card Reader. The DAS92-CA, CB Card Reader is a Digital CR8-F for 80-column (only) cards, and operates at 285 CPM. Maximum of one (1) Card Reader per DAS92-A.

(c) DAS92-PA LA180 Printer. The LA180 operates at 180 characters per second and uses a 96-character set (upper and lower case). The carriage is 132 columns wide. Maximum of one (1) LA180 per DAS92-A.

(d) DAS92-VA, VD LP05 Line Printer, (64-characters). This is the LP05 (LE8-V) and operates at 300 lines per minute using a 64-character set. The carriage is 132 columns wide. Maximum of one (1) LP05 per DAS92-A.

(e) DAS92-WA, WD LP05 Line Printer, (96 characters). This is the same as the DAS92-VA except that it uses the 96-character set (upper and lower case). Maximum of one (1) LP05 per DAS92-A.

DAS61
IBM 2780/3780
FRONT END

The DAS61 Front End allows IBM 2780 and 3780 emulation and termination. This means that IBM 2780's and/or 3780's can be used as RJE stations into the DECsystem-10 or that the DECsystem-10 can emulate (or simulate) an IBM 2780 or 3780 to an IBM 360/370 computer. The DAS61 can handle a maximum of 12 synchronous lines each operating independently with any combination of 2780 or 3780 emulation or termination. The maximum aggregate throughput of a DAS61 is 100,000 bits per second (100K baud). The DAS61-D is interfaced to the DECsystem-10 via the DL10 and the DAS61-S is interfaced via the DTE10. Multiple DAS61's are supported per DECsystem-10.

OPTIONS

DAS61-DA, DB–IBM 2780 and/or 3780 Emulation and Termination Front End. Including DL10-C port and requiring DL10-A with available port. Requires addition of DAS61-H and/or DAS61-J synchronous line units.

DAS61-SA, SB–IBM 2780 and/or 3780 Emulation and Termination Front End including DTE10 interface. Requires addition of DAS61-H and/or DAS61-J synchronous line units.

DAS61-EA, EB–Synchronous Expansion Cabinet for expansion past 4 synchronous lines. Includes no synchronous lines.

DAS61-H–Synchronous Single Line Controller for data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.

DAS61-J–Synchronous Single Line Controller for data transmission speeds of up to 50K baud and for attachment to 303-type current modems.

DAS61-QC–DAS61 Software Only.

DAS61-QX–DAS78 to DAS61 Software Only Upgrade (prerequisite GALAXY).

DAS62
IBM HASP
MULTI-LEAVING
FRONT END

The DAS62 Front End combines all of the functionality of the DAS61 with support of IBM HASP multi-leaving work stations as DECsystem-10 RJE stations and also allows for the DECsystem-10 to simulate a HASP multi-leaving work station to an IBM host computer. The DAS62 can handle a maximum of 12 synchronous lines each operating independently with any combination of 2780 or 3780 or HASP multi-leaving emulation or termination. The maximum aggregate throughput of the DAS62 is 100,000 bits per second (100K baud). The DAS62-D is interfaced to the DECsystem-10 via the DL10 and the DAS62-S is interfaced via the DTE10. Multiple DAS62's are supported per DECsystem-10.

OPTIONS

DAS62-DA, DB–IBM 2780 and/or 3780 and/or HASP multi-leaving work station Front End. Including DL10-C port and requires DL10-A with available port. Requires addition of DAS62-H and/or DAS62-J synchronous line units.

DAS62-SA, SB–IBM 2780 and/or 3780 and/or HASP multi-leaving work station Front End including DTE10 interface. Requires addition of DAS62-H and/or DAS62-J synchronous line units.

DAS62-EA, EB–Synchronous Expansion Cabinet for expansion past 4 synchronous lines. Includes no synchronous lines.

DAS62-H–Synchronous Single Line Controller for data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.

DAS62-J–Synchronous Single Line Controller for data transmission speeds of up to 50K baud and for attachment to 303-type current modems.

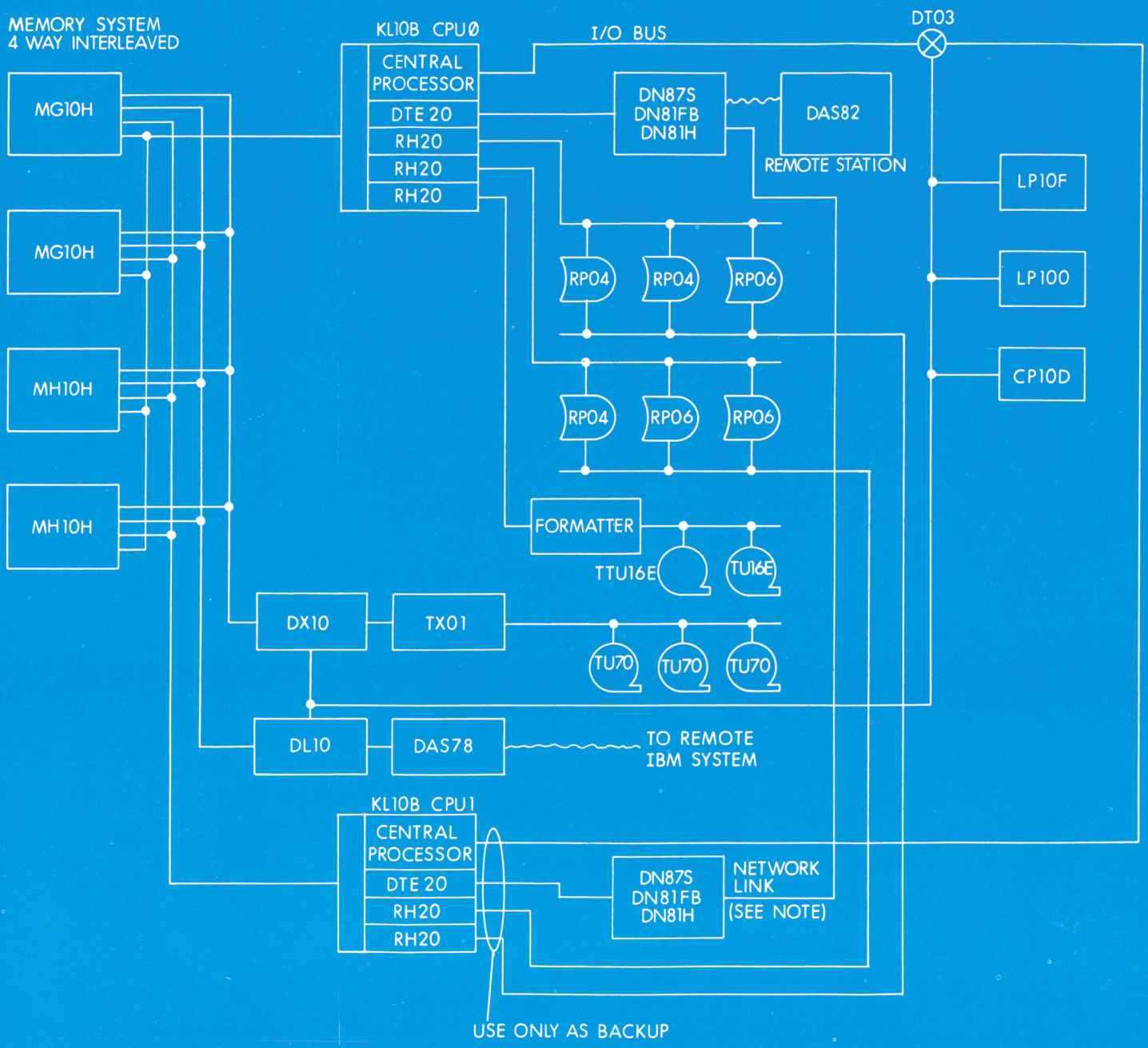
DAS62-QC–DAS62 Software Only

DAS62-QX–DAS78 to DAS62 Software Only Upgrade (prerequisite GALAXY)

DAS62-QY–DAS61 to DAS62 Software Only Upgrade

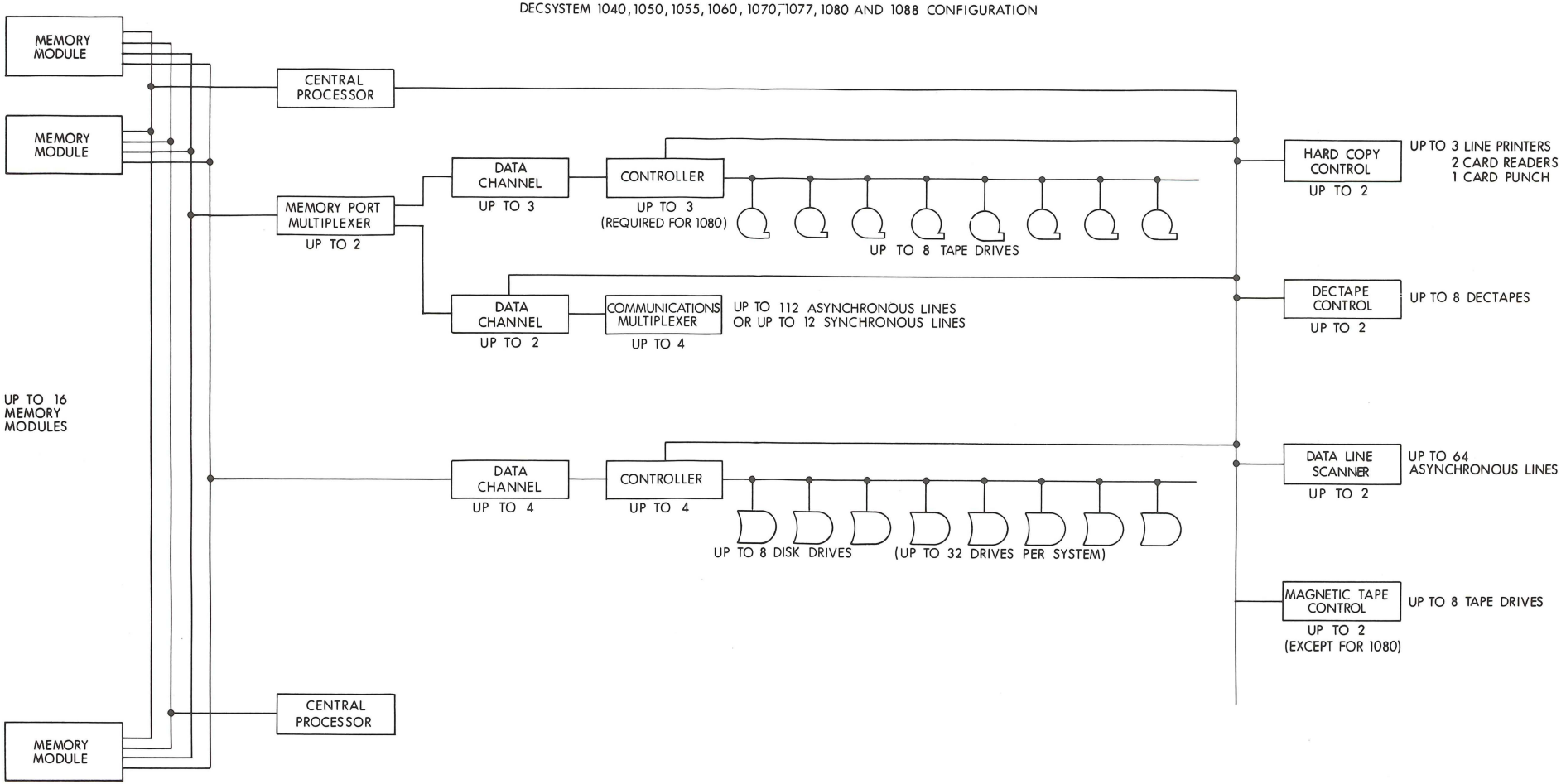
CONFIGURATOR

SAMPLE COMPLEX CONFIGURATION
1090 WITH HIGH AVAILABILITY SWITCHOVER

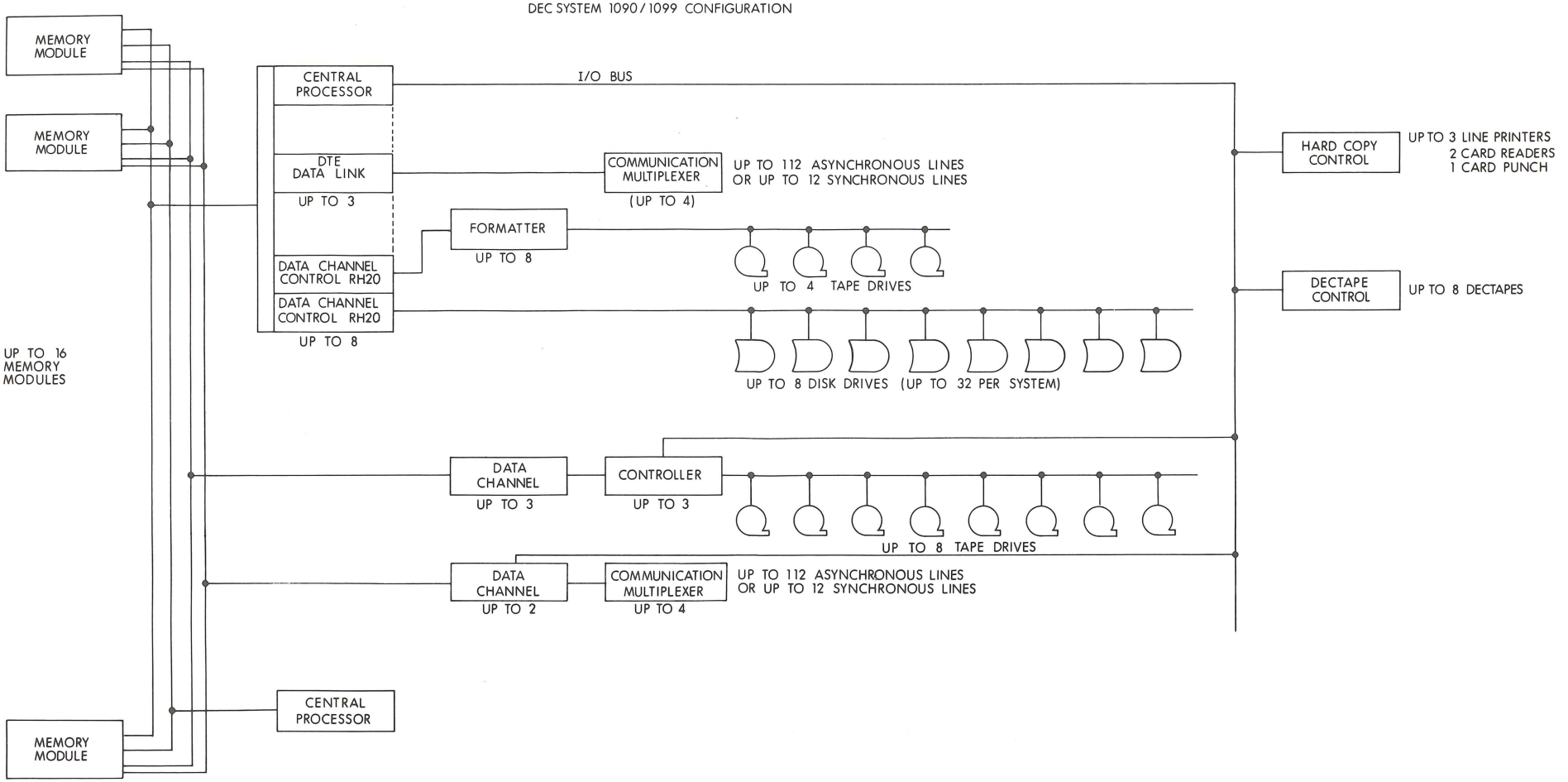


NOTE:
This system would normally run as a 1099 with CPU 0 the master and CPU 1 the slave. The DN97S Subsystem connected to CPU 1 would function as a DAS 81 remote concentrator to the DN87S on CPU 0. Should the master CPU 0 fail, the system would be reconfigured as a 1090 and TOPS-10 reloaded to run on CPU 1. Route-through would be used with the DN87S on CPU 0 now operating as a DAS 81 to the DN87S or CPU 1.

DECSYSTEM 1040, 1050, 1055, 1060, 1070, 1077, 1080 AND 1088 CONFIGURATION



DEC SYSTEM 1090/1099 CONFIGURATION



OPTION DESIGNATOR 115V/60Hz 230V/50Hz	DESCRIPTION	PREREQUISITE	MAX. NO. SUP-PORTED (NOTE 1)
Central Processors			
KA10-A, KA10-C	KA10 Central Processor; 366 hardware instructions, 10-character-per-second console terminal.	None	2
KI10, KI10	K110 Central Processor; 378 hardware instructions, memory paging, instruction look-ahead, virtual memory, 10-character-per-second console terminal.	None	2
KL10-BA, KL10-BB	KL10 Central Processor; 395 microprogrammed instructions, business instruction set, high-speed cache memory, virtual memory, instruction look-ahead, 30-character-per-second console terminal.	None	2
Memories and Channels			
MF10-A, MF10-A	MF10 Core Memory; 32K words of 1.0-microsecond memory.	KA10, KI10 or KL10	Note 2
MF10-E, MF10-E	MF10 Core Memory expansion module; 32K words of 1.0-microsecond memory to expand MF10-A.	MF10-A	Note 2
MF10-G, MF10-G	MF10 Core Memory; 64K words of 1.0-microsecond memory.	KA10, KI10, or KL10	Note 2
MG10-HA	MG10 Core Memory; 128K words of 1.0 microsecond memory	KA-10, KI10, or KL10	Note 2
MH10-H	MH-10 Core Memory 256K words of 1.2 microsecond memory	KA-10, KI10 or KL10	Note 2
MC10-D, MC10-D	Memory Access Port; for MD10 Memories	MD10	64
MC10-F, MC10-F	Memory Access Port; for MF10 Memories.	MF10	64
MX10, MX10	Memory Port Multiplexer; provides direct memory access for up to eight FD10 Data Channels using 18-bit address logic.	MC10	3
MX10-C, MX10-C	Memory Port Multiplexer; provides direct memory access for up to eight DF10-C Data Channels using 22-bit address logic.	MC10	3
DF10, DF10	Data Channel; permits data transfer between high-speed devices and core memory; uses 18-bit address logic	MC10 or MX10	8
DF10-CA, DF10-CB	Data Channel; permits data transfer between high-speed devices and core memory; uses 22-bit address logic.	MC10 or MX10-C	8
DAS33	Buffered Data Channel; permits block data transfers between I/O controllers and core memory; contains 256-word data buffer; uses 22-bit address logic.	Avail. memory port MX10 or MX10-C	8
Disk Systems			
RP02-A, RP02-B	Disk Drive; 5.12-million words capacity; average access time 47.5 milliseconds; transfer rate 15 microseconds per word.	RP10 or RP10-C	32
RP03-AS, RP03-BS	Disk Drive; 10.24-million words capacity; average access time 47.5 milliseconds; transfer rate 15 microseconds per word.	RP10-C	32
RP02-P, RP02-P	Disk Pack; additional disk pack for RP02 or RP03 disk drive.	RP02 or RP03	n/a
RP10-CA, RP10-CB	Disk Controller; controller for up to eight RP02 and/or RP03 Disk Drives.	DF10 or DF10-C	3
RP02-CA, RP02-CB	Disk System; includes DF10 Data Channel, RP10-C Controller and one RP02 Disk Drive.	MC10, MX10	3
RP03-CA, RP03-CB	Disk System: includes DF10 Data Channel, RP10-C Controller and one RP03 Disk Drive.	MC10, MX10	3

OPTION DESIGNATOR 115V/60Hz 230V/50Hz	DESCRIPTION	PREREQUISITE	MAX. NO. SUP-PORTED (NOTE 1)
RP04-AA, RP04-AB	Disk Drive; 20.48-million words capacity; average access time 27 milliseconds; transfer rate 5.6 microseconds per word.	RHP04-A, RHP04-B, or RHP04-D	32
RP04-BA, RP04-BB	Disk Drive; 20.48-million words capacity; average access time 27 milliseconds; transfer rate 5.6 microseconds per word; controller select option for attachment to two controllers.	RHP04-E, RHP04-F, or RHP04-H	Note 3
RP04-C, RP04-C	Controller Select Option; converts RP04-A to RP04-B.	RP04-A	n/a
RP04-P, RP04-P	Disk Pack; additional disk pack for RP04 Disk Drive.	RP04-A or RP04-B	n/a
RHP04-AA, RHP04-AB	Disk System; includes controller and one RP04-A Disk Drive.	DF10 or DF10-C	4
RHP04-BA, RHP04-BB	Disk System; includes controller and one RP04-B Disk Drive.	DF10 or DF10-C	4
RHP04-DA, RHP04-DB	Disk System; includes DF10 Data Channel, controller and RP04-A Disk Drive.	MC10, MX10	4
RHP04-EA, RHP04-EB	Disk System; includes DF10 Data Channel, controller and one RP04-B Disk Drive.	MC10, MX10	4
RHP04-FA, RHP04-FB	Disk System; includes DF10-C Data Channel, controller and one RP04-A Disk Drive.	MC10, or MX10-C	4
RHP04-HA, RHP04-HB	Disk System; includes DF10-C Data Channel, controller and one RP04-B Disk Drive.	MC10, MX10-C	4
RTP04-AA, RTP04-AB	Disk System including Controller and RP04-A (single access) Disk Drive	KL10-B	4
RTP04-BA, RTP04-BB	Disk System including Controller and RP04-B (dual access) Disk Drive	KL10-B	4
RTP04-C	Dual Channel Access Kit	KL10-B, RTP04-A	N/A
RHP06-AA, RHP06-AB	Disk System; including Controller and RP06-A (single access) Disk Drive.	DF10 or DF10-C	4
RHP06-BA, RHP06-BB	Disk System; including Controller and RP06-B (dual access) Disk Drive.	DF10 or DF10-C	4
RP06-AA, RP06-AB	Add-on Disk Drive (single access); includes one RP06-P Disk Pack. 39, 40 million word capacity average access time. 27 milliseconds, transfer rate 5.6 microseconds per word.	RHP06-A or RHP06-B	32
RP06-BA, RP06-BB	Add-on Disk Drive (dual access); includes one RP06-P Disk Pack. 39, 40 million word capacity average access time, 27 milliseconds, transfer rate 5.6 microseconds per words.	RHP06-A or RHP06-B	32
RP06-C	RP06 Dual Access Kit	RP06-A	
RTP06-AA, RTP06-BB	Disk System including Controller and RP06-A (single access) Disk Drive	KL10-B	4
RTP06-BA, RTP06-BB	Disk System including Controller and RP06-B (dual access) Disk Drive	KL10-B	4
RTP06-C	Dual Channel Access Kit	KL10-B, RTP06-A	7
RHS04-DA, RHS04-DD	Swapping Disk; 256K words of swapping storage; transfer rate of four microseconds per word; room for one additional RHS04-C.	RHS04-G, RHS04-H or RHS04-J	16
RHS04-CA, RHS04-CD	Swapping Disk; 256K words of swapping storage; transfer rate four microseconds per word; for expansion of RHS04-D.	RHS04-D, RHS04-G RHS04-H, RHS04-J	16
RHS04-GA, RHS04-GD	Swapping System; includes controller and one RHS04-D Swapping Disk.	DF10 or DF10-C	2
RHS04-HA, RHS04-HD	Swapping System; includes DF10 Data Channel, controller, and one RHS04-D Swapping Disk.	MC10, MX10	2
RHS04-JA, RHS04-JD	Swapping System; includes DF10-C Data Channel, controller, and one RHS04-D Swapping Disk.	MC10 or MX10-C	2

OPTION DESIGNATOR 115V/60Hz 230V/50Hz	DESCRIPTION	PREREQUISITE	MAX. NO. SUP-PORTED (NOTE 1)
Magnetic Tape Systems			
TU10A-EE, TU10A-EJ	Magnetic Tape Drive; 9-channel drive, 45-inches-per-second; recording densities of 200, 556 and 88 bpi.	TM10-A or TM10-B	8
TU10A-FE, TU10A-FJ	Magnetic Tape Drive; 7-channel drive, 45-inches-per-second; recording densities of 200, 556 and 800 bpi.	TM10-A or TM10-B	8
TM10-AA, TM10-AB	Magnetic Tape Controller; Interfaces to I/O bus of KA10 or KI10 Central Processors for control of up to eight TU10A Magnetic Tape Drives.	KA10 or KI10	1
TU10-CA, TU10-CB	Magnetic Tape System; includes TM10-A Controller and one TU10A Magnetic Tape Drive.	KA10 or KI10	1
THU16-EA, THU16-ED	Magnetic Tape System; including RH10 and TM02 Controllers and TU16-E 9-track, 45 ips Tape Drive. Recording densities of 800/1600 bpi	DF10 or DAS33	8
TTU16-EA, TTU16-ED	Magnetic Tape System including Massbus Controller, Tape Controller and TU16-E Tape Drive	KL10-B	4
TU16B-E, TU16B-ED	Tape Controller and TU16-E Tape Drive (for expansion beyond 4 drives)	THU16-E or TTU16-E	8
TU16-EE, TU16-EJ	Add-on Magnetic Tape Drive, 9-track, 45 ips. Recording densities of 800/1600 bpi	THU16-E	32
TU40-A, TU40-B	Magnetic Tape Drive; 9-channel drive, 150-inches-per-second; recording densities of 200, 556 and 800 bpi.	TU40-C, TU41-C	16
TU41-A, TU41-B	Magnetic Tape Drive; 7-channel drive, 150-inches-per-second; recording densities 200, 556, and 800 bpi.	TU40-C, TU41-C	16
TM10-BA, TM10-BB	Magnetic Tape Control; attaches to DF10 or DF10-C Data Channel to control up to eight TU40 and/or TU41 Magnetic Tape Drives.	DF10 or DF10-C	2
TU40-CA, TU40-CB	Magnetic Tape System; includes DF10 Data Channel, TM10-B Controller and one TU40 Magnetic Tape Drive.	MC10, MX10	2
TU41-CA, TU41-CB	Magnetic Tape System; includes DF10 Data Channel TM10-B Controller and one TU41 Magnetic Tape Drive.	MC10, MX10	2
TU70-AA, TU70-AB	Magnetic Tape Drive; 9-channel drive, 200-inches-per-second; recording densities 800 and 1600 bpi.	TU70-C	16
TU71-AA, TU71-AB	Magnetic Tape Drive; 7-channel drive, 200-inches-per-second; recording densities 200, 556, and 800 bpi.	TU70-C	16
TU70-CA, TU70-CB	Magnetic Tape System; includes Data Channel, controller and one TU70-A Magnetic Tape Drive.	MC10, MX10 or MX10-C	2
TU56-A, TU56-B	DECTAPE Unit; dual-drive DECTAPE.	TD10 or TD10-G	8
TD10-A, TD10-B	DECTAPE Control; provides control for up to four TU56 DECTAPE units.	KA10	2
TD10-CA, TD10-CB	DECTAPE Control; provides control for up to four TU56 DECTAPE units.	KI10, KL10	2
TD10-GA, TD10-GB	DECTAPE System; includes TD10 Controller and one TU56 DECTAPE unit.	KI10 or KL10	2
TU72-C	Magnetic Tape System; includes DX10 data channel, TX02 tape control unit and TU72-E 9-track 125 ips Tape Drive. Recording densities of 1600 and 6250 bpi.	KI10 or KL10	2

OPTION DESIGNATOR 115V/60Hz 230V/50Hz	DESCRIPTION	PREREQUISITE	MAX. NO. SUP-PORTED (NOTE 1)
TU72-E	Magnetic Tape Drive; 9-track, 125-inches-per-second; recording densities 1600 and 6250 bpi.	TU72-C	8
Hard-copy Equipment			
CR10-DA, CR10-DB	Card Reader; table-top card reader, 1000-cards-per minute.	KA10, KI10 or KL10	2
CR10-EA, CR10-EB	Card Reader; console card reader, 1200-cards-per-minute.	KA10, KI10 or KL10	2
CR10-FA, CR10-FB	Card Reader; table-top card reader, 300-cards-per-minute.	KA10, KI10 or KL10	2
LSP10-LA, LSP10-LB	Line Printer; 245-lines-per-minute, 64-character print set.	KA10, KI10 or KL10	1, Note 5
LP10-FE, LP10-FE	Print Drum; 64-character EDP print drum for LP10-F Line Printer.	LP10-F	N/A
LP10-FF, LP10-FF	Print Drum; 64-character Scientific Print drum for LP10-F Line Printer.	LP10-F	N/A
LP10-HE, LP10-HE	Print Drum, 96-character EDP Print drum for LP10-H Line Printer.	LP10-H	N/A
LP10-HF, LP10-HF	Print Drum, 96-character Scientific Print drum for LP10-H Line Printer.	LP10-H	N/A
LP10-FA, LP10-FB	Line Printer; 1,250-lines-per-minute, 64-character EDP print set.	KA10, KI10, KL10	3
LP10-FC, LP10-FD	Line Printer; 1,250-lines-per-minute, 64-character Scientific print set.	KA10, KI10, KL10	3
LP10-HA, LP10-HB	Line Printer, 925-lines-per-minute, 96-character EDP print set.	KA10, KI10, KL10	3
LP10-HC, LP10-HD	Line Printer, 925-lines-per-minute, 96-character Scientific print set.	KA10, KI10, KL10	3
LP100-BA, LP100-BB	Line Printer; up to 1500LPM, EDP print set—see LP07-Y.	KA10, KI10, KL10	3
LP07-YA	Charaband for LP100 with 2 sets of 64-char. fonts.	LP100-B	N/A
LP07-YB	Charaband for LP100 with 2 sets of 96-char. fonts.	LP100-B	N/A
LP07-YC	Charaband for LP100 with 1 set of 64-char. fonts and 1 set of 96-char. fonts	LP100-B	N/A
XY10, XY10	Plotter Control; for CALCOMP Model, 563 or 565 plotter or equivalent.	KA10, KI10, KL10	2
XY10-A, XY10-A	Plotter; CALCOMP Model 565 plotter with control; uses 12-inch paper; specify step size: Step size — Steps/minute 0.01 inches — 18,000 0.005 inches — 18,000 0.1 inches — 18,000	KA10, KI10, KL10	2
XY10-B, XY10-B	Plotter; CALCOMP Model 563 plotter with control; uses 31-inch paper; specify step size: Step size — Steps/minute 0.01 inches — 12,000 0.005 inches — 18,000 0.1 inches — 18,000	KA10, KI10, KL10	2

Option Designator 115V/60Hz 230V/50Hz	DESCRIPTION	Prerequisite	Maximum No. Supported (Note 1)
Asynchronous Data Communication Systems			
DC10-AA, DC10-AB	Data Line Scanner; provides controller plus four units of cabinet space; DC10 System handles up to 64 lines per DC10-A.	KA10, KI10	2
DC10-B, DC10-B	Eight-line Group; Interface for eight local or dial-up lines; requires one unit of cabinet space; dial-up lines require DC10-E for full modem control.	DC10-A	16
DC10-C, DC10-C	Telegraph Relay Assembly; provides conversion from local to long lines using full- or half-duplex facilities; requires two units of cabinet space.	DC10-B and DC10-D	8
DC10-D, DC10-D	Telegraph Power supply; provides a standard line-voltage power supply used with DC10-C (120 volt, 2 amps); no cabinet space required.	DC10-C	8
DC10-E, DC10-E	Expanded Data Set Control; provides expanded control of eight data sets in the DC10; recommended for maximum system security; requires two units of cabinet space.	DC10-B	8
DC10-F, DC10-F	Expander Cabinet; provides eight units of cabinet space and power supplies for expansion beyond DC10-A.	DC10-A	2
DC10-H, DC10-H	Eight-line Group; interface for eight local or dial-up 2741-type lines; requires one unit of cabinet space; dial-up lines require DC10-E for full modem control.	DC10-A	8 Note 6
Remote Stations			
DAS80-AA, DAS80-AB	Remote Batch Station. Includes 1 DN81-H, LA36 Console, DAS80-CA/CB, DAS80-LA/LB, PDP-11 and software.	DAS 85, DN87, or DC75NP	N/A
DAS81-AA, DAS81-AB	Remote Concentrator. Includes PDP-11, 1 DN81-H, DN81-EA and software. Can add one more DN81-EC, ED for a total of 32 asynchronous lines. DN81-Fx asynchronous interfaces are also required and must be ordered separately.	DAS85, DN87, or DC75NP	N/A
DAS82-AA, DAS82-AB	Remote Batch Station and Concentrator. Includes PDP-11, DAS80-CA/CB card reader, DAS80-LA/LB line printer, 1 DN81-H, DN81-EA and software. DN81-Fx asynchronous interfaces are also required and must be ordered separately.	DAS85, DN87, or DC75NP	N/A
DAS92-AA, DAS92-AB	Includes 16K PDP-8/A processor, VT52, ROM for down-line loading, one synchronous line interface.	DECsystem-10 DC75-NP, DAS85 or DN87	N/A
DAS92-EA	Asynchronous 4 line multiplexer including line drivers. Will accommodate either 20 mA or EIA lines in any mixture.	See DAS92 configurator	4
DAS92-CA, DAS92-CB	Card Reader for 80-column (only) cards; operates at 285 cpm.	See DAS92 configurator	1
DAS92-PA, DAS92-PD	LA 180 Printer operates at 180 cps and uses a 96-character set.	See DAS92 configurator	1
DAS92-VA, DAS92-VD	LP05 Line Printer (64 characters) operates at 300 lpm; 132 columns wide carriage	See DAS92 configurator	1
DAS92-WA, DAS92-WD	LP05 Line Printer (96 characters).	See DAS92 configurator	1

Option Designator	DESCRIPTION	Prerequisite	Maximum No. Supported (Note 1)
Unit Record Options			
DAS80-CA, DAS80-CB	Card Reader, 300 cards per minute.	DAS80 Series Remote	1
DAS80-LA, DAS80-LB	Printer, 300 lines per minute, 132 printing positions, 64 printing characters, EDP character set.	DAS80 Series Remote	1
DAS80-LC, DAS80-LD	Printer, 230 lines per minute, 132 printing positions, 96 printing characters, EDP character set.	DAS80 Series Remote	1
Front Ends			
DN87-DA, DN87-DB	DN87 including DL10-C port only. Requires DL10-A with available port. Requires addition of DN81-xx synchronous and/or asynchronous line options.	KA10, KI10, KL10 DL10	6
DN87-AA, DN87-AB	DN87 including DL10-A Communications Interface and one DL10-C port. Requires addition of DN81-xx synchronous and/or asynchronous line options.	KA10, KI10, KL10	2
DN87S-AA, DN87S-AB	DN87 with DTE10 interface. Requires addition of DN81-xx synchronous and/or asynchronous line options.	1090	3
DN87-U	DC76 to DN87 software only upgrade (including KG11-A).	DC76	N/A
Asynchronous Line Options			
DN81-EA, DN81-EB	Asynchronous Expansion Cabinet including one DN81-EC 16-Line Asynchronous Expansion Group. Requires two DN81-Fx 8-Line Terminator Groups to activate the lines.	DN87, 87S*	2
DN81-EC, DN81-ED	Asynchronous 16-Line Expansion Group. This requires two DN81-Fx 8-Line Terminators to activate the lines.	DN87, 87S*	5
DN81-FA	8-Line Terminators each with 20 mA Current Loop Local Interfaces.	DN87, 87S	N/A
DN81-FB	8-Line Terminators each with EIA Local Interfaces.	DN87, 87S	N/A
DN81-FC	8-Line Terminators each with EIA Full Modem Control Interfaces.	DN87, 87S	N/A
DN81-FD	8-Line Terminators with Integral Auto Answer Modems. (These modems require customer supplied DAA's).	DN87, 87S	N/A
Synchronous Line Options			
DN81-EE, DN81-EF	Synchronous Expansion Cabinet. Includes one DN81-H.	DN87, 87S	1
DN81-H	Synchronous Single Line Controller Expansion Line. For data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.	DN87, 87S DAS 80, 81, 82	12 4
DN81-J	Synchronous Single Line Controller. For data transmission speeds of up to 40.8K baud and for attachment to 303-type current mode modems.	DN87, 87S DAS 80, 81, 82	12 4

*Used on DN87, DN87S Front Ends and on DAS80 Series Remote Stations.

Option Designator 115V/60Hz	230V/50Hz	DESCRIPTION	Prerequisite	Maximum No. Supported (Note 1)
IBM DEVICE EMULATION AND TERMINATION FRONT ENDS				
DAS61-DA, DB		IBM 2780 and/or 3780 Emulation and Termination Front End. Including DL10-C port and requiring DL10-A with available port. Requires addition of DAS61-H and/or DAS61-J synchronous line units.	KI10, KL10, DL10 TOPS10 6.03	8
DAS61-SA, SB		IBM 2780 and/or 3780 Emulation and Termination Front End including DTE10 interface. Requires addition of DAS61-H and/or DAS61-J synchronous line units.	KL10B	3
DAS61-EA, EB		Synchronous Expansion Cabinet for expansion past 4 synchronous lines. Includes no synchronous lines.	DAS61-D, DAS61-S	1
DAS61-H		Synchronous Single Line Controller for data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.	DAS61-D, DAS61-S	12
DAS61-J		Synchronous Single Line Controller for data transmission speeds of up to 50K baud and for attachment to 303-type current modems.	DAS61-D, DAS61S	12
DAS61-QC		DAS61 Software Only	DAS61-D, DAS61-S	Note 6
DAS61-QX		DAS78 to DAS61 Software Only Upgrade (prerequisite GALAXY).	DAS61-D, DAS61-S	Note 6
DAS62-DA, DB		IBM 2780 and/or 3780 and/or HASP multi-leaving work station Front End. Including DL10-C port and requires DL10-A with available port. Requires addition of DAS62-H and/or DAS62-J synchronous line units.	KI10, KL10, DL10 TOPS10 6.03	4
DAS62-SA, SB		IBM 2780 and/or 3780 and/or HASP multi-leaving work station Front End including DTE10 interface. Requires addition of DAS62-H and/or DAS62-J synchronous line units.	KL10B	8
DAS62-EA, EB		Synchronous Expansion Cabinet for expansion past 4 synchronous lines. Includes no synchronous lines.	DAS62-D, DAS62-S	3
DAS62-H		Synchronous Single Line Controller for data transmission speeds of up to 10K baud and for attachment to EIA RS232C compatible modems.	DAS62-D, DAS62-S	12
DAS62-J		Synchronous Single Line Controller for data transmission speeds of up to 50K baud and for attachment to 303-type current modems.	DAS62-D, DAS62-S	12
DAS62-QC		DAS62 Software Only.	DAS62-D, DAS62-S	Note 6
DAS62-QX		DAS78 to DAS62 Software Only Upgrade (prerequisite GALAXY).	DAS78	Note 6
DAS62-QY		DAS61 to DAS62 Software Only Upgrade.	DAS61	Note 6
Terminals				
LA36-CC, LA36-CD		DECwriter-II; 30-character-per-second teleprinter with 20-mA current loop interface; handles up to 6-part forms, 132-column printing, variable-width forms, 96 upper- and lower-case characters.	Asynchronous Line	Note 4
LA37		APL-ASCII Terminal, 30-character-per-second, features full APL, full upper/lower case, or upper case TTY character sets.	"	Note 4
VT05B-DA, VT05-DB		Alphanumeric Terminal; 110 to 2400-baud CRT display terminal; features 128-character keyboard, 1440-character capacity screen, direct cursor addressing, 20-mA current loop or EIA interface.	"	Note 4

Option Designator 115V/60Hz 230V/50Hz	DESCRIPTION	Prerequisite	Maximum No. Supported (Note 1)
GT40-AA, GT40-AB	Graphic System; computer-level graphic system consisting of a graphical display system and general-purpose minicomputer; features light pen, 128-character keyboard and EIA serial interface.	"	Note 4
VT50-CA, VT50-CD	DECscope; 75 to 9600-baud CRT terminal, 20-mA current loop interface, 960-character capacity screen, 64-character keyboard.	"	Note 4
VT52	DECscope; 75 to 9600-baud CRT terminal, EIA interface, 1920-character capacity screen, 96-character keyboard.	"	Note 4
VT50-EA, VT50-ED	DECscope; 75 to 9600-baud CRT terminal, EIA interface, 960-character screen, 64-character keyboard.	"	Note 4
VT61	DECscope Video Display Block-mode Terminal; 1920-character display and full ASCII keyboard; 128-character 7x8 dot matrix character set.	"	Note 4 Note 6
Miscellaneous			
DA10, DA10	Interface between KA10 or KI10 CPU and PDP-8 minicomputer.	KA10, KI10, KL10	n/a
DK10, DK10	Real-time Clock; provides resolution to 10 micro-seconds via crystal oscillator.	KA10, or KI10	2
PC10-C	High speed paper tape reader/punch	KL10	1
GP10, GP10	General Purpose Interface; provides interface to KA10 or KI10 I/O bus; includes cabinet, two model 728 power supplies, one model 844 power control indicator in panels, convenience outlet and BS10-A-15 cablet set; logic provides a status register, device decoding, read-in gating and line buffering.	KA10, KI10, KL10	N/A
GP10-L, GP10-L	GP10 Interface Logic; interface logic only; does not include power supply or indicators.	KA10, KI10, KL10	N/A
GP10-M, GP10-M	GP10 less logic; includes power supplies and indicators; does not include cables.	KA10, KI10, KL10	N/A
DF01-AA, DF01-AB	Acoustic Telephone Coupler.	DECsystem-10 Terminal	Note 4
CAB-9B, CAB-9B	Cabinet; includes full-length single doors, front and back with indicator panel, but without end panels (old style).	DECsystem-10	N/A
H-956, H-956	Cabinet; includes full-length single doors, front and back with indicator panel, but without end panels (new style).	DECsystem-10	N/A

Note: Specifications subject to change without notice.

Note 1: Maximum number supported by standard software. Configurations with large numbers of disk drives, high-speed magnetic tape drives and/or data communication lines must be reviewed by DECsystem-10 Marketing.

Note 2: Maximum memory capacity of KA10 is 256K words in 16 memory modules. Maximum memory capacity of KI10 or KL10 is 4,096K words in 16 memory modules.

Note 3: Currently under software development. Contact DECsystem-10 Marketing before configuring.

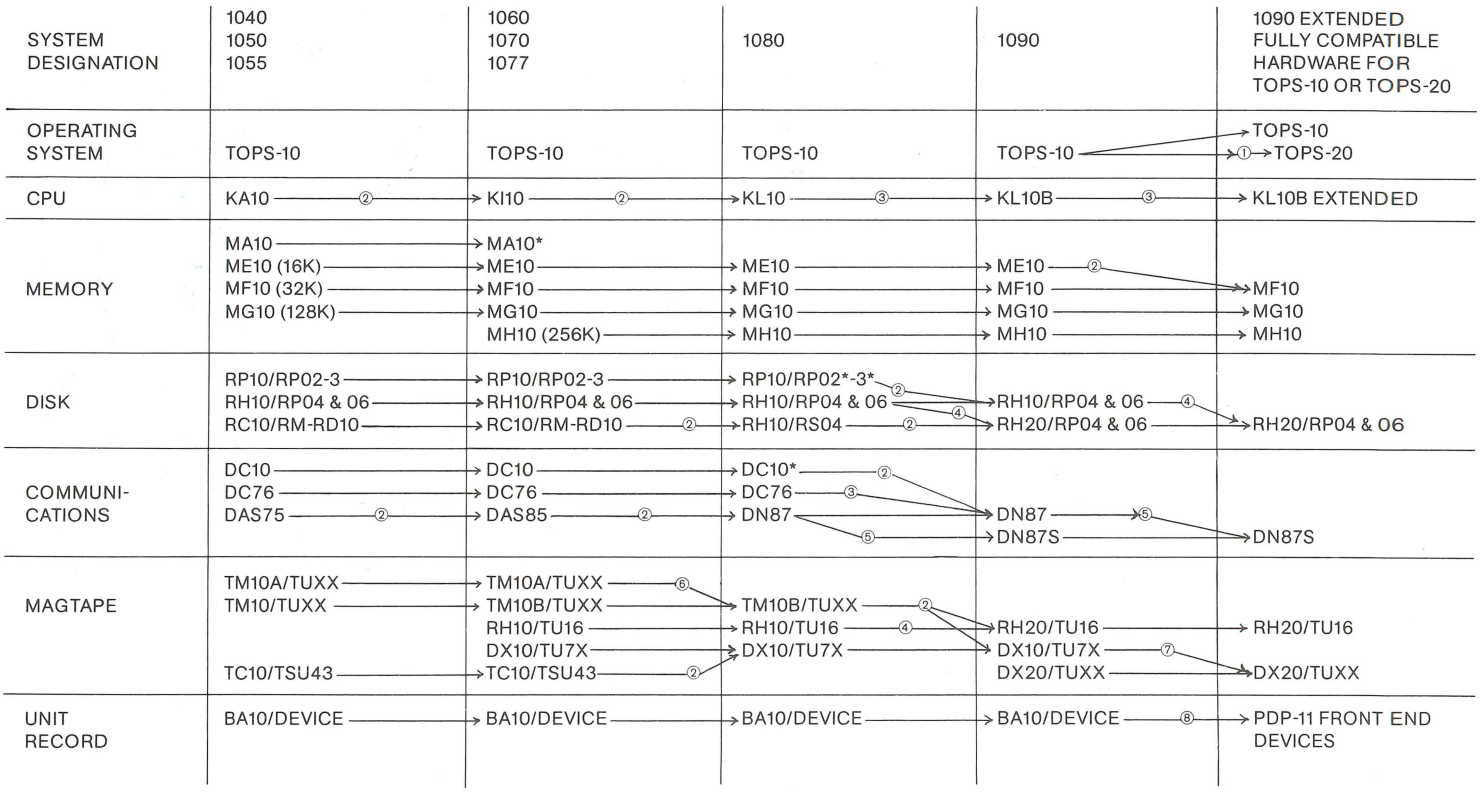
Note 4: DECsystem-10's support a maximum of 512 simultaneously-active terminals through 127 simultaneously-active jobs.

Note 5: Software Supported by Computer Special Systems.

Note 6: Software Supported by Advanced Systems Group.

DECsystem-10 UPGRADE/MIGRATION PATHS

This chart illustrates the migration of different DECsystem-10 subsystems toward the common TOPS-10/TOPS-20 hardware system. The chart is useful for determining what equipment is supported within each DECsystem-10 family member. It also provides insight into what equipment needs to be upgraded in moving from one CPU-based system to another.



NOTES:
1 New microcode; TOPS-20 Release,
2 Drop-in upgrade replacement (old component typically returned/retired)
3 Field retrofit (in-place)
4 Substitution of RH20 for DF10/RH10
5 Field Upgrade: replace DL10 with DTE connection, S/W Release
6 Field Upgrade/Addition: replace/upgrade TM10A with TM10B and add DF10/DF10-C
7 Tape channel replaced with DX20 (internal channel)
8 BA10 Hard Copy Control Unit replaced with PDP-11-based controller(s)
* Starred items are those hardware components which are not manufactured, difficult/impossible to support, or whose successor provides cost/performance or other features which eventually make their support uneconomical.

SOFTWARE

SOFTWARE 59
Languages 61
AID 60
ALGOL 60
APL 60
BASIC 61
COBOL 62
ISAM 62
CPL 62
FORTRAN 63
FORTRAN Version 4 63
MACRO 64
System Utilities 65
SORT 65
Data Base Management System 65
ITPS-10 66
LINK-10 66
MACY11-LNKX11 67
DDT 67
FAILSAFE/BACKUP 67
Message Control System 68
PIP 68
RUNOFF 68
TECO 68
Operating System 69
TOPS-10 Operating System 69
Command Control Language 72
File System 73
Input/Output 76
Memory Management 76
Multi-Processor Systems 77
Inter-job Communication 78
Communications Software 78
Multimode Computing 79
Timesharing 79
GALAXY-10 Batch 79
Real-Time Computing 81
Diagnostic Software 83
High Availability 83
Maintenance Features 83
On-Line Diagnosis 83

Languages

AID

The Algebraic Interpretive Dialogue, AID, is the DECsystem-10 adaptation of the language elements of JOSS, a program developed by the RAND Corporation. To write a program in the AID language requires no previous programming experience. Commands to AID are typed in via the user's terminal as imperative English sentences. Each command occupies one line and can be executed immediately or stored as part of a routine for later execution. The beginning of each command is a verb taken from the set of AID verbs. These verbs allow the user to read, store and delete items in storage; halt the current processing and either resume or cancel execution; type information on his terminal; and define arithmetic formulas and functions for repetitive use that are not provided for in the language. However, many common algebraic and geometric functions are provided for the user's convenience.

The AID program is device-independent. The user can create external files for storage of subroutines and data for subsequent recall and use. These files may be stored on any retrievable storage media, but for accessibility and speed, most files are stored on disk.

ALGOL

- Programs may comprise separately-compiled procedures
- Assignments are permitted within expressions
- Remainder operator
- Unique implementation of dynamic own arrays
- Octal boolean constants and integer/boolean and boolean/integer transfer functions
- Alternative reserved delimiter word or "quoted delimiter word" representations

The DECsystem-10 ALGOL system is based on the ALGOL-60 specifications. It is composed of the ALGOL compiler, and the ALGOL object time system. The compiler is responsible for reading programs written in the ALGOL language and converting these programs into machine language. The user may specify a parameter at compile time which produces a sequence numbered listing with cross-referenced symbol tables.

The ALGOL object time system provides special services, including the input/output service for the compiled ALGOL program. Part of the object time system ALGOL library is a set of routines that the user's program can call in order to perform various functions including mathematical functions and string and data transmission routines. These routines are loaded with the user's program when required; the user need only make a call to them. The remainder of the object time system is responsible for the running of the program and providing services for system resources, such as core allocation and management and assignment of peripheral devices.

Source level, interactive debugging is provided through ALGDDT. The user may stop his program at any point, may examine and change the contents of data locations, may insert connected code and then continue execution to test his changes.

APL

APL (A Programming Language) is a concise programming language specially suited for dealing with numeric and character data which can be collected into lists and arrays. The conciseness with which APL expressions can be written greatly enhances programmer productivity and allows for very compact and readable code which can be executed in a highly efficient manner. APL finds application not only in mathematics and engineering, but also in financial modeling and text handling situations.

Some of APL's features include:

- The user's active workspace size is dynamically variable
- The workspace symbol-table is dynamically variable
- Immediate-mode line editing of complex APL expressions
- Statement branching may occur anywhere in a statement line
- Multiple statements may appear in a single line
- User-controlled tab positioning for I/O operations
- All floating point operations are performed to an accuracy of eighteen decimal digits

APL is a fully interactive system with both immediate (desk calculator) and function (program) modes of operation. It includes its own editor as well as debugging tools, tracing of function execution, typeout of intermediate values, setting break points, etc. APL for the

DECsystem-10 includes many extensions not normally found in other APL implementations. Some of these features include a flexible file system which supports both ASCII sequential and binary direct access files, the dyadic format operator, expanded command formations which allow the user to take advantage of the DECsystem -10 file system, the execute operator and tools for error analysis and recovery. In addition, workspaces are dynamically variable in sizes up to 512K bytes and the system supports double precision calculations which allow up to 18 decimal digits of precision.

Finally, APL on the DECsystem -10 is available to all users at all times without the need to run a separate subsystem which provides extra terminal handling, workspace swapping, etc. These tasks are all accomplished by the TOPS-10 operating system.

BASIC

- Multiple users share the BASIC compiler's pure code.
- A variety of program manipulation commands which include functions for savings, running and retrieving BASIC programs
- Immediate mode statements to facilitate debugging and "desk calculator" mode
- Sequential data storage
- String capability; including string arrays and functions
- Chaining with COMMON to accommodate large programs
- Formatted output using the PRINT USING statement

BASIC is a conversational problem solving language that is well suited for timesharing and easy to learn. It has wide application in the scientific, business, and educational communities and can be used to solve both simple and complex mathematical problems from the user's terminal.

The BASIC user types in computational procedures as a series of numbered statements that are composed of common English terms and standard mathematical notation. After the statements are entered, a run-type command initiates the execution of the program and returns the results.

If there are errors during execution, the user, from his terminal, simply changes the line or lines in error by deleting, modifying, or inserting lines.

The beginning user has many facilities at his disposal to aid in program creation:

- **Program Editing Facilities**—An existing program or data file can be edited by adding or deleting lines, by renaming it, or by resequencing the line numbers. The user can combine two programs or data files into one and request either a listing of all or part of it on the terminal or a listing of all of it on the high-speed line printer.
- **Documentation Aids**—Documenting programs by the insertion of remarks within procedures enables recall of needed information at some later date and is invaluable in situations in which the program is shared by other users.
- As a BASIC user, you can type in a computational prodedure as a series of numbered statements by using simple common English syntax and familiar mathematical notation. You can solve almost any problem by spending an hour or so learning the necessary elementary commands.
- **Functions**—Occasionally, you may want to calculate a function, for example, the square of a number. Instead of writing a program to calculate this function, BASIC provides functions as part of the language.

More advanced users will want to take advantage of some of the more sophisticated computation and data handling tools which BASIC provides. For example:

File Input—The user may create a name date file using the BASIC editing facilities. This file may now be referenced. within a program.

Output Formatting—The user can control the appearance of his output to the terminal or printer.

Data Access—Data files may be read facilities. This file may now be referenced and written either sequentially or randomly.

String Manipulation—Alphanumeric strings may be read, printed, concatenated and searched.

COBOL

The COmmon Business Oriented Language, COBOL, is an industry-wide data processing language that is de-designed for business applications, such as payroll, inventory control, and accounts receivable.

Because COBOL programs are written in terms that are familiar to the business user, he can easily describe the formats of his data and the processing to be performed in simple English-like statements. Therefore, programmer training is minimal, COBOL programs are self-documenting, and programming of desired applications is accomplished quickly and easily.

Major features include:

- **On-line editing and debugging**—The programmer may create and modify the source program from a terminal using an easily-learned editing facility. After the program has been submitted to the compiler—again from the terminal—any syntax errors may be immediately corrected using the editor, and the program resubmitted. The program listing is sequence numbered with symbols cross-referenced to show when they are defined and where used. COBDDT, the on-line, interactive COBOL debugging package, allows the programmer to check out the program:
- By selectively displaying a paragraph name
- By causing the program to pause at any desired step during execution
- By allowing the program to examine and modify data at will before continuing execution

Easy program development and debugging increases programmer productivity by eliminating tedious waiting periods.

- **Batch**
  Using the same commands for time-sharing, the programmer may submit the program via a control file to the Batch system. This further increases efficiency because other terminal work may be done concurrent with the Batch processing
- **Access Methods**
  The programmer has a choice of three access methods; sequential, indexed sequential, and random.

Indexed Sequential Access Mode (ISAM)

ISAM requires a minimum amount of programming while providing a large data file handling capacity. It is supported by the COBOL Object Time System which automatically handles all of the searching and movement of data.

All reading and writing of an index file (ten levels of indexing) are performed by the run-time operating system (LIBOL). This does not involve the user. When using the indexed sequential files, the programmer need only specify which record is to be read, written, or deleted.

Whenever records are added to the file, the index is automatically updated, additions to the file will not degenerate the file as with other computers. The common technique for using overflow areas for added records has been avoided. Whenever records have been deleted from the file, the empty space is used again for later additions. The net effect of the addition and deletion techniques significantly increases the time between major "overhauls" of the data files, because the time required to access a file is independent of the number of changes made to it.

- **Sorting**
  The SORT package permits a user to rearrange records or data according to a set of user-specified keys. The keys may be ascending or descending, alpha or numeric, and any size or location within the record. Multi-reel file devices may be specified.
- **Source Library Maintenance System**
  This system lists file entries or adds, replaces, and/or deletes source language data on a file. It can also add, replace, or extract an entire file from the library.
- **Device Independence**
  The operating system allows the programmer to reference a device with a user-assigned logical name as well as its physical name, thus allowing dynamic assignment of peripheral devices at run time.

CPL

CPL is an interpreter supporting a subset of the draft ANSI PL/I language. The following is a summary of the key features of CPL:

- Data types include FIXED, FLOAT, CHARACTER, CHARACTER VARYING, BIT VARYING, POINTER, and arrays of these data types.
- Storage classes include AUTOMATIC, STATIC, CONTROLLED and BASED
- Almost all PL/I statement types are supported including READ, WRITE, DECLARE, DEFAULT, GET, PUT, ON and FORMAT
- Subroutines and function procedures are recursive
- The ON statement provides the capability of recovering from program error under program control.
- Complete string manipulation package
- CPL supports the following classes of PL/I built-in functions: arithmetic, mathematical, string array, storage control and pseudo-variables.

CPL is intended to be easy-to-use for the beginning programmer or non-programmer. At the user's option, statements will be executed immediately or saved for deferred execution. A beginning programmer can start by executing simple computational statements and can proceed to building programs. Since CPL is an interpreter, a user can track a program very closely. Debugging features, such as source level breakpoints and program modification are available.

- **Data Base Management System Interface**
  The programmer may call upon the data base management system facilities from within his program. This system, which allows the CODASYL specifications, allows data files to be consolidated into one or more data bases. Application programs are then permitted to access the data in the way best suited to their needs.

FORTRAN

The FORmula TRANslator language, FORTRAN, is a widely-used and procedure-oriented programming language. It is designed for solving scientific problems and is thus composed of mathematical-like statements constructed in accordance with precisely formulated rules. Therefore, programs written in FORTRAN consist of meaningful sequences of these statements that are intended to direct the computer to perform the specified computations.

FORTRAN has a wide use in every segment of the computer market. Universities find that FORTRAN is a good language with which to teach students how to solve problems via the computer. The scientific community relies on FORTRAN because of the ease with which scientific problems can be expressed. In addition, FORTRAN is used as the primary data processing language by many timesharing utilities.

FORTRAN Version 4

FORTRAN-10 is a superset of the American National Standard FORTRAN. Both the compiler and object-time system are reentrant (shareable). The compiler produces optimized object code. The following is a summary of key features and extensions of FORTRAN-10:

- PARAMETER statement–Allows symbolic specification of compile-time constants
- INCLUDE statement–Allows user to include in the compilation of a given program unit source code which resides on a file other than primary source file
- GLOBAL OPTIMIZATION-FORTRAN-10 performs extensive local and optional global optimizations
- FORDDT–the FORTRAN-10 debugger
  FORDDT in conjunction with the FORTRAN-10/"DEBUG" switch will allow:
  - Display and modification of program data
  - Tracing of the program statement by statement
  - Setting of pauses on any statement or routine
- OPEN/CLOSE file-specification statements
- Array bounds checking can be generated optionally
- N-dimensional arrays
- ENCODE/DECODE statements
- Boolean operations equivalence (EQV) and exclusive or (XOR), in addition to OR, AND, NOT.
- The NAMELIST and list-directed I/O features provide format-free input and output operations
- Random-access I/O capabilities
- Compatibility with IBM type declaration statements
- Implied DO loops in I/O statements and data statements
- Full mixed-mode arithmetic in expressions
- Octal Constants
- Logical Operations–full-word masking operations for all logical functions (rather than a result of just true or false)
- END= and ERR= in I/O statements
- Device independence

The FORTRAN-10 object-time system, FOROTS, controls the input/output, format interpretation and numerical conversion for compiled programs. The FORTRAN user may reference any I/O device. All special editing, conversion, and file structuring tasks are handled by the object-time system. Devices are normally specified by logical assignment so that physical device selection need not be made until run-time. The devices corresponding to the special I/O statements READ, PRINT, ACCEPT, and TYPE are also assignable at run-time. The object-time system is shareable.

FOROTS implements all program data file functions and provides the user with an extensive run-time error reporting system. Device independence is provided to allow the programmer or operator to determine the physical device at run-time.

FORTRAN-10 is easy to use in both timesharing and batch processing environments. Under timesharing, the user operates in an interactive editing and debugging environment. FORDDT, an interactive program that is used as an aid in debugging.
FORTRAN programs uses the language constructs and variable names of the program itself–thereby eliminating the need to learn another language in order to accomplish on-line debugging. Under batch processing, the user submits his program through the batch software in order to have the compiling, loading, and executing phases performed without his intervention.

FORTRAN programs can be entered into the FORTRAN system from a number of devices: disk, magnetic tape, user terminal and card reader. In addition to data files created by FORTRAN, the user can submit data files or FORTRAN source files created by the system editor. The data files contain the data needed by the user's object program during execution. The source files contain the FORTRAN source text to be compiled by the FORTRAN compiler. Output may be received on the user's terminal, disk or magnetic tape. The source listing cross references all symbols to show where they are defined and where used.

MACRO

MACRO is the symbolic assembly program on the DECsystem-10. It makes machine language programming easier and faster for the user by

- Translating symbolic operation codes in the source program into the binary codes needed in machine language instructions.
- Relating symbols specified by the user to numeric values.
- Assigning absolute core addresses to the symbolic addresses of program instructions and data.
- Preparing an output listing of the program which includes any errors detected during the assembly process.

MACRO programs consist of a series of statements that are usually prepared on the user's terminal with a system editing program. The elements in each statement do not have to be placed in certain columns nor must they be separated in a rigid fashion. The assembler interprets and processes these statements, generates binary instructions or data words, and performs the assembly.

MACRO is a two-pass assembler. This means that the assembler reads the source program twice. Basically, on the first pass, all symbols are defined and placed in the symbol table with their numeric values, and on the second pass, the binary (machine) code is generated. Although not as fast as a one-pass assembler, MACRO is more efficient in that less core is used in generating the machine language code and the output to the user is not as long.

MACRO is a device-independent program; it allows the user to select, at runtime, standard peripheral devices for input and output files. For example, input of the source program can come from the user's terminal, output of the assembled binary program can go to a magnetic tape, and output of the program listing can go to the line printer. More commonly, the source program input and the binary output are disk files.

The MACRO assembler contains powerful macro capabilities that allow the user to create new language elements. This capability is useful when a sequence of code is used several times with only certain arguments changed. The code sequence is defined with dummy arguments as a macro instruction. Thus, a single statement in the source program referring to the macro by name, along with a list of the real arguments, generates the entire correct sequence. This capability allows for the expansion and adaptation of the assembler in order to perform specialized functions for each programming job.

System Utilities

SORT

The TOPS-10 SORT arranges the records of one or more files according to a user-specified sequence. The user designates the keys on which the records are sorted from one or more fields within a record. The keys can be in either ascending or descending order. SORT compares the key fields values of all records. Then it arranges the records in the specified sequence and merges them into a single output file. Sort may be used under timesharing and batch, and may be called from within a COBOL program.

Data Base Management System

Certain data—such as that within commercial, accounting, inventory control and administrative systems—are used in computer applications which have common relationships and processing requirements with data in other applications. This can prove to be a problem because as file organizations are defined in one application or program, restructuring, redundant appearance, and even repetitive processing of the same data are often required in another application.

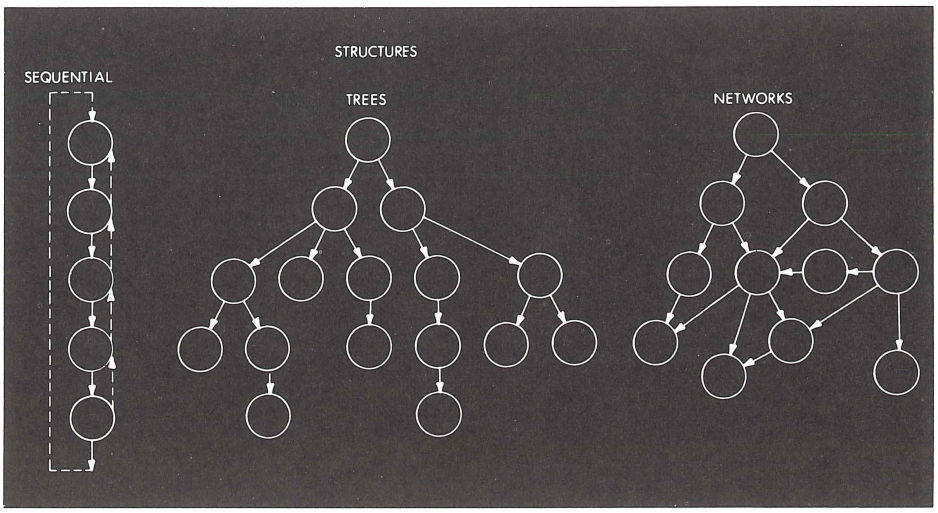
To further complicate the problem, on-line processes (i.e., customer order entry, shipment planning, and student information retrieval systems) tend to different data structures than the processes used to create and maintain primary data files. This means that as new applications requirements for existing data are determined and additional data are defined, program development personnel must either alter existing data file forms and programs or create and maintain redundant copies.

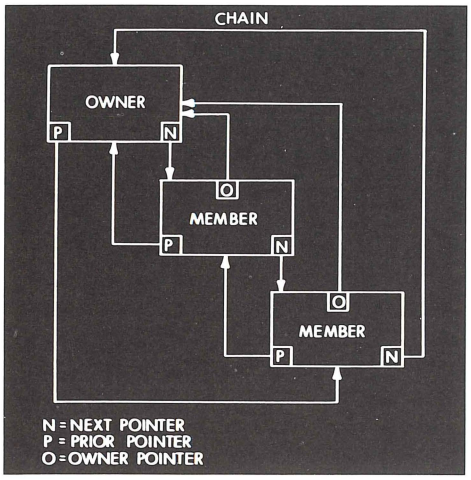
DIGITAL offers a solution to the problem in the Data Base Management System—DBMS. It enables DEC system-10 users to organize and maintain data in forms more suitable to the integration of a number of related but separate processes and applications. DBMS is ideal in situations where data processing control and program development functions require structures and techniques not satisfied by traditional data management facilities.

Features

The DECsystem-10 Data Base Management System is predicated upon the proposals of the CODASYL Data Base Task Group (DBTG) which appear in their report of April, 1971. DIGITAL's goal for DBMS is to provide features which assist users in obtaining the most significant objectives stated in the CODASYL DBTG report. These features include:

- **Hierarchical Data Structures**
  —In addition to sequential structures, simple tree structures and more complex network structures can be created and maintained. Data items can be related within and between various levels of the structure established.





- **Non-redundant Data Occurrence**–Data items may appear in a number of different structural relationships without requiring multiple copies of the data. Data structures may be established and modified in a manner most suitable to a given application without altering the occurrence of the data in structures maintained for other applications.
- **Variety of Access and Search Strategies**–Data records can be maintained in chains, as illustrated below. Access may be through DIRECT, CALCULATED, or VIA set location modes. Sort keys, in addition to the normal storage key, may be defined.
- **Concurrent Access**–Multiple run units (or programs) that use the same re-entrant code module further supplement the DECsystem-10 monitor's extensive centralized file handling capabilities. Any number of concurrent retrievals to the same data areas can be handled. Concurrent updates to the same area can be handled by a multiple update queuing mechanism.
- **Protection and Centralized Control**–Usage of a common data definition language establishes data base structures maintained on direct access storage devices that are selectively referenced by individual application programs through privacy lock and key mechanisms. Physical placement of data is specified by a centrally-controlled data definition processor.
- **Device Independence**–The common input/output and control conventions of the DECsystem-10 monitor provide basic device independence. Applications programs deal with logical areas rather than physical devices. Data base areas may reside on the same or different direct access storage devices as non-date base files.
- **Program Independence of Data**–Significant steps toward program independence are achieved by using the SCHEMA and SUB-SCHEMA concepts of data definition. Individual programs reference only user-selected data elements rather than entire record formats. Program alterations due to adding new data and relationships are minimized. Alterations of the original form (i.e., binary, display) and element sizes require program recompilations.
- **User Interaction Without Structural Maintenance Responsibilities**–Individual users, applications, and programs may access data structures without the responsibility of maintaining detailed linkage mechanisms that are internal to the data base software. Common and centralized authorization, error recovery, and building techniques reduce individual activity to maintain data structures.
- **Multiple Language Usage**–The DBMS data manipulation language is available to both COBOL and FORTRAN as host languages
- **DBMS** Software Modules Consistent with the CODASYL
Data Base Task Group report, the body of the DECSYSTEM-20 data base management software includes:
- **DDL**–data description language and its processor
- **DML**–data manipulation language for COBOL and FORTRAN programs
- **DCBS**–data base manager module of re-entrant run-time routines
- **DBU**–group of data base system support utilities

ITPS-10

ITPS-10, In-House Text Preparation System allows DECsystem-10 users to prepare office-quality or camera-ready (typeset) copy using the interactive capability of the DECsystem-10. High-quality, camera-ready copy for pamphlets, advertising brochures, product flyers, user manuals, and many other types of documentation can be produced and updated rapidly and inexpensively using ITPS-10. The system allows use of many different input and output devices, including interactive CRT or hard copy terminals, line printers, typewriter like devices, and photocomposition equipment. Documents are stored as standard DECsystem-10 files and can be easily updated, edited and re-generated.

The system is easy-to-use and easy to learn. Current successful applications range from preparing form letters to setting daily newspapers.

LINK-10

LINK-10, the DECsystem-10 linking loader, merges independently-translated modules of the user's program into a single module and links this module with system modules into a form that can be executed by the operating system. It provides automatic relocation and loading of the binary modules producing an executable version of the user's program. When the loading process has been completed, the user can request LINK-10 either to transfer control to his program for immediate execution or to output the program to a device for storage in order to avoid the loading procedure in the future.

While the primary output of LINK–10 is the executable version of the user's program, the user can request auxiliary

output in the form of map, log, save, symbol, overlay plot, and expanded core image files. This additional output is not automatically generated: the user must include appropriate switches in his command strings to LINK-10 in order to obtain this type of output. The user can also gain precise control over the loading process by setting various loading parameters and by controlling the loading of symbols and modules. Furthermore, by setting switches in his command strings to LINK-10, the user can specify the core sizes and starting addresses of modules, the size of the symbol table, the segment into which the symbol table is placed, the messages he will see on his terminal or in his log file, and the severity and verbosity levels of the messages. Finally, he can accept the LINK-10 defaults for items in a file specification or he can set his own defaults that will be used automatically when he omits an item from his command string.

LINK-10 has an overlay facility to be used when the total core required by a user's program is more than the core available to the user. The user organizes his program so that only some portions of the program are required in core at any one time. The remaining portions reside in a disk file and are transferred in and out of core during execution. Because the portion brought into core may overlay a portion already core-resident, the amount of core required by the entire program is reduced. Overlays can be invoked either by runtime routines called from the user's program or by automatic calls to subroutines outside the current overlay link.

MACY11 – LNKX11

MACY11 is a cross-assembler which operates on the DECsystem-10 and assembles MACRO-11 source code for the PDP-11 family of computers.

MACY11 will produce either an absolute binary or a relocatable object module, depending upon the assembly mode. The resulting absolute binary files are transferrable to the PDP-11 for execution or, the relocatable object modules may be used as input to the LNKX11 linker on the DECsystem-10 and then transferred to the PDP-11 operating environment. MACY11 will append a symbol table and (optionally) a cross-reference table to the listing file. MACY11 produces a "side-by-side" assembly listing of symbolic source statements, their octal equivalents, assigned addresses and error codes.

DDT

The Dynamic Debugging Technique, DDT, is used for on-line program composition of object programs and for on-line checkout and testing of these programs. For example, the user can perform rapid checkout of a new program by making a change resulting from an error detected by DDT and then immediately executing that section of the program for testing.

After the source program has been compiled or assembled, the binary object program with its table of defined symbols is loaded with DDT. In command strings to DDT, the user can specify locations in his program, or breakpoints, where DDT is to suspend execution in order to accept further commands. In this way, the user can check out his program section-by-section and, if an error occurs, insert the corrected code immediately. Either before DDT begins execution or at breakpoints, the user can examine and modify the contents of any location. Insertions and deletions can be in source language code or in various numeric and text modes. DDT also performs searches, gives conditional dumps, and calls user-coded debugging subroutines at breakpoint locations.

FAILSAFE/BACKUP

The FAILSAFE/BACKUP programs are used to selectively or universally save disk files on magnetic tape. Files may be copied to magnetic tape, in a special format, selectively by file name or groups of file names, by project/programmer number, by disk structure (logical or physical), or universally. In addition to providing backup of files, these programs allow a method of extending disk space by allowing infrequently-used files to be stored on tape. Restoring of files from FAILSAFE/BACKUP format tapes can likewise be handled on an individual file, groups of files, project/programmer, logical or physical disk basis. An added benefit is achieved when restoring files in that disk fragmentation is often minimized by the failsafe/restore operation.

Message Control System

The Message Control System (MCS-10) is to provide the DECsystem-10 with an efficient, transaction processing-oriented system that facilitates control of communications between a network of terminals and applications processes. The intent is to provide generalized routines that are readily used from COBOL programs and network definitions. Message control in its fullest sense includes a complex of hardware and software systems:

- Terminals and transmission devices.
- Remote and central site concentrators.
- Central site communications multiplexers.
- DECsystem-10 Monitor and its Message Service module.
- Message Control System (MCS-10) Routines.
- Extended Communications Language module and interfaces (COBOL/LIBOL).
- Network Definition, Activation and Operations Control Routines.
- Message Oriented Applications Programs.
- Data Base Management capability via DBMS-10.

The activities normally associated with message control are cooperative processes distributed among communications concentrators, multiplexers, and a central system. Actual I/O transmission, line control, error handling, device selection, code translation, and message buffering are handled by the remote and local concentrators and front-end processors. The DECsystem-10 central processor, Monitor, and Message Control modules accept, route, queue and log terminal and application program-generated messages. Upon the occurrence of specific user-specified events, application programs are activated to perform message processing. Applications programs receive messages from queues through interface modules, perform processes related to data within messages and send messages back through queues to terminals and/or other application processes.

PIP

The Peripheral Interchange Program, PIP, is used to transfer data files from one I/O device to another. Commands to PIP are formatted to accept any number of input (source) devices and one output (destination) device. Files can be transferred from one or more source devices to the destination device as either one combined file or individual files. Switches contained in the command string to PIP provides the user with the following capabilities:

- Naming the files to be transferred.
- Editing data in any of the input files.
- Defining the mode of transfer.
- Manipulating the directory of a device if it has a directory.
- Controlling magnetic tape and card punch functions.
- Recovering from errors during processing.

RUNOFF

RUNOFF facilitates the preparation of typed or printed manuscripts by performing line justification, page numbering, titling, indexing, formatting, and case shifting as directed by the user. The user creates a file with an editor and enters his material through his terminal. In addition to entering the text, the user includes information for formatting and case shifting. RUNOFF processes the file and produces the final formatted file to be output to the terminal, the line printer, or to another file.

With RUNOFF, large amounts of material can be inserted into or deleted from the file without retyping the text that will remain unchanged. After the group of modifications have been added to the file, RUNOFF produces a new copy of the file which is properly paged and formatted.

TECO

The Text Editor and COrrector program, TECO, is a powerful editor used to edit any ASCII text file with a minimum of effort. TECO commands can be separated into two groups: one group of elementary commands that can be applied to most editing tasks, and the larger set of sophisticated commands for character string searching, text block movement, conditional commands, programmed editing, and command repetition.

TECO is a character-oriented editor. This means that one or more characters in a line can be changed without retyping the remainder of the line. TECO has the capability to edit any source document: programs written in MACRO, FORTRAN, COBOL, ALGOL, or any other source language; specifications; memoranda, and other types of arbitrarily-formatted text. The TECO program does not require that line numbers or other special formatting be associated with the text.

Editing is performed by TECO via an editing buffer, which is a section within TECO's core area. Editing is accomplished by reading text from any device into the editing buffer (inputting), by modifying the text in the buffer with data received from either the user's terminal or some other device (inserting), and by writing the modified text in the buffer to an output file (outputting).

A position indicator, or buffer pointer, is used to locate characters within the buffer and its position determines the effect of many of TECO's commands. It is always positioned before the first character, between two characters, or after the last character in the buffer. Various commands, such as insertion commands, always take place at the current position of the buffer pointer.

There are TECO commands to manipulate data within the editing buffer. Input and output commands read data from the input file into the buffer and output data from the buffer to the output file. There are other commands to have one or more characters inserted into the editing buffer, deleted from the buffer, searched for, and/or typed out. In addition, the user can employ iteration commands to execute a sequence of commands repeatedly and conditional execution commands to create conditional branches and skips.

Operating System

TOPS-10 Operating System

The resident operating system is made up of a number of separate and somewhat independent parts, or routines. Some of these routines are cyclic in nature and are repeated at every system clock interrupt (tick) to ensure that every user of the computing system is receiving the requested services. These cyclic routines are

- The command processor, or decoder
- The scheduler
- The swapper

The command decoder is responsible for interpreting commands typed by the user on his terminal and passing them to the appropriate system program or routine. The scheduler decides which user is to run in the interval between the clock interrupts, allocates sharable system resources, and saves and restores conditions needed to start a program interrupted by the clock. The swapper rotates user jobs between secondary memory (usually disk or drum) and core memory after deciding which jobs should be in core but are not. These routines constitute the part of the operating system that allows many jobs to be operating simultaneously.

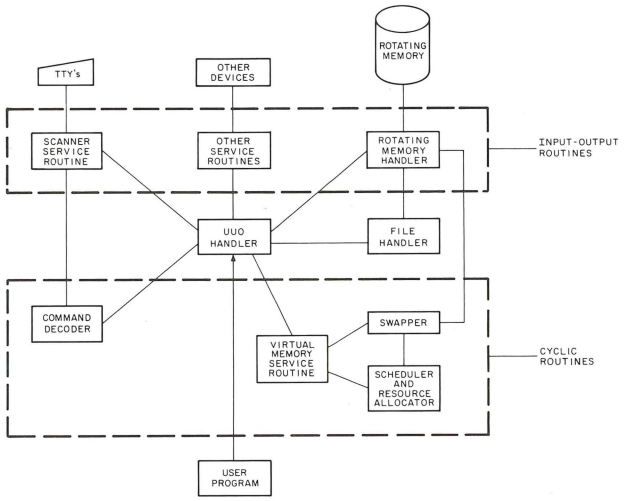
The non-cyclic routines of the operating system are invoked only by user programs and are responsible for providing these programs with the services available through the operating system. These routines are

- The UUO handler
- The input/output routines
- The file handler

The UUO handler is the means by which the user program communicates with the operating system in order to have a service performed. Communication is by way of programmed operators (also known as UUO's) contained in the user program which, when executed, go to the operating system for processing. The input/output routines are the routines responsible for directing data transfers between peripheral devices and user programs in core memory. These routines are invoked through the UUO handler, thus saving the user the detailed programming needed to control peripheral devices. The file handler adds permanent user storage to the computing system by allowing users to store named programs and data as files.

Scheduler

The DECsystem-10 is a multiprogramming system; i.e., it allows several user jobs to reside in core simultaneously and to operate sequentially. It is then the job of the scheduler to decide which jobs should run at any given time. In addition to the multiprogramming feature, the DECsystem-10 employs a swapping technique whereby jobs can exist on an external storage device (e.g., disk or drum) as well as in core. Therefore, the scheduler decides not only what job is to be run next but also when a job is to be swapped out onto disk or drum and later brought back into core.



THE RESIDENT OPERATING SYSTEM.

All jobs in the system are retained in ordered groupings called queues. These queues have various priorities that reflect the status of each job at any given moment. The queue in which a job is placed depends on the system resource for which it is waiting and, because a job can wait for only one resource at a time, it can be in only one queue at a time. Several of the possible queues in the system are

- Run queues for jobs waiting for, or jobs in, execution.
- I/O wait queues for jobs waiting for data transfers to be completed.
- Resource wait queues for jobs waiting for some system resource.
- Null queue for all job numbers that are not currently being used.

The job's position within certain queues determines the priority of the job with respect to other jobs in the same queue. For example, if a job is first in the queue for a sharable device, it has the highest priority for the device when it becomes available. However, if a job is in an I/O wait queue, it remains in the queue until the I/O is completed. Therefore, in an I/O wait queue, the job's position has no significance. The status of a job is changed each time it is placed into a different queue.

The scheduling of jobs into different queues is governed by the system clock. This clock divides the time for the central processor into sixtieths of a second. When the clock ticks, the scheduler decides which job will run during the next cycle. Each job, when it is assigned to run, is given a time slice. When the time slice expires for the job, the clock notifies the central processor and scheduling is performed. The job whose time slice just expired is moved into another, perhaps lower, priority run queue, and the scheduler selects another job to run in the next time slice. If the currently running job is the null job and a higher priority job (any job) becomes ready to run before the clock ticks it will immediately run. If the currently running job is not the null job and a high priority (HPQ real-time) job becomes runnable it will run at the next clock tick. Finally if a job just becoming runnable is not an HPQ job, but is of higher priority than the current job, it will only preempt the current job when the applicable time slice has expired.

Each job, when it is assigned to run, is given a time slice. When the time slice expires for the job, the clock notifies the central processor and scheduling is performed. The job whose time slice just expired is moved into another, perhaps lower, priority run queue, and the scheduler selects another job to run in the next time slice.

Scheduling may be forced before the clock ticks if the currently-running job reaches a point at which it cannot immediately continue. Whenever an operating system routine discovers that it cannot complete a function requested by the job (e.g., it is waiting for I/O to complete or the job needs a device which it currently does not have), it calls the scheduler so that another job can be selected to run. The job that was stopped is then requeued and is scheduled to be run when the function it requested can be completed. For example: when the currently running job begins input from a DECtape, it is placed into the I/O wait queue, and the input is begun. A second job is scheduled to run while the input of the first job proceeds. If the second job then decides to access a DECtape, it is stopped because the DECtape control is busy, and it is placed in the queue for jobs waiting to access the DECtape control. A third job is set to run. The input operation of the first job finishes, freeing the DECtape control for the second job. The I/O operation of the second job is initiated, and the job is transferred from the device wait queue to the I/O wait queue. The first job is transferred from the I/O wait queue to the highest-priority run queue. This permits the first job to preempt the running of the third job. When the time slice of the first job becomes zero, it is moved into the second run queue, and the third job runs again until the second job completes its I/O operation.

In addition, data transfers allow the scheduler to permit the user to overlap computation with data transmission. In unbuffered data modes, the user supplies an address of a command list containing pointers to locations in his area to and from which data is to be transferred. When the transfer is initiated, the job is scheduled into an I/O wait queue where it remains until the device signals the scheduler that the entire transfer has been completed.

In buffered modes, each buffer contains information to prevent the user and the device from using the same buffer at the same time. If the user requires the buffer currently being used by the device as his next buffer, the user's job is scheduled into an I/O wait queue. When the device finishes using the buffer, the device calls the scheduler to reactivate the job.

TOPS-10 also provides the ability to tune the system scheduler to specific loads. The system administrator can set aside percentages of the CPU to given classes of timesharing and batch users. This facility can be dynamically changed while the system is running and need not be defined at system generator time, thus facilitating optimal system usage over a wide variety of system loads.

Swapper

The swapper is responsible for keeping in core the jobs most likely to be run. It determines if a job should be in core by scanning the various queues in which a job may be. If the swapper decides that a job should be brought into core, it may have to take another job already in core and transfer it to secondary storage. Therefore, the swapper is not only responsible for bringing a job into core but is also responsible for selecting the job to be swapped out.

A job is swapped to secondary storage for one of two reasons:

- A job that is more eligible to run needs to be swapped in and there is not enough room in core for both jobs.
- The job needs to expand its core size and there is not enough core space to do so.

If the latter case is true, the job must be swapped out and then swapped in later with new allocation of core.

The swapper checks periodically to see if a job should be swapped in. If there is no such job, then it checks to see if a job is requesting more core. If there is no job wishing to expand its size, then the swapper does nothing further and relinquishes control of the processor until the next clock tick.

UUO Handler

The UUO handler is responsible for accepting requests for services available through the operating system. These requests are made by the user program via software-implemented instructions known as programmed operators, or UUO's. The various services obtainable by the user program include:

- Communicating with the I/O devices on the computing system, including connecting and responding to any special devices that may be desired on the system for real-time programming.
- Receiving or changing information concerning either the computing system as a whole or the individual program.
- Altering the operation of the computing system as it concerns the user job, such as controlling execution by trapping or suspending, or controlling core memory by locking.
- Communicating and transferring control between user programs.

The UUO handler is the only means by which a user program can give control to the operating system in order to have a service performed. Contained in the user program are operation codes which, when executed, cause the hardware to transfer control to the UUO handler for processing. The routine obtains its arguments from the user program. The core location at which the UUO operation was executed is then remembered. After the UUO request has been processed, control is returned to the user program at the first or second instruction following the UUO. In this way, the software supplements the hardware by providing services that are invoked through the execution of a single core location just as the hardware services are invoked.

Device Service Routines

I/O programming in the DECsystem-10 is highly convenient for the user because all of the burdensome details of programming are performed by the operating system. The user informs the operating system of his requirements for I/O by means of UUO's contained in his program. The actual input/output routines needed are then called by the UUO handler.

Since the operating system channels communication between the user program and the device, the user does not need to know all the peculiarities of each device on the system. In fact, the user program can be written in a similar manner for all devices. The operating system will ignore, without returning an error message, operations that are not pertinent to the device being used. Thus, a terminal and a disk file can be processed identically by the user program. In addition, user programs can be written to be independent of any particular device. The operating system allows the user program to specify a logical device name, which can be associated with any physical device at the time when the program is to be executed. Because of this feature, a program that is coded to use a specific device does not need to be rewritten if the device is unavailable. The device can be designated as a logical device name and assigned to an available physical device with one command to the operating system.

Data is transmitted between the device and the user program in one of two methods: unbuffered mode or buffered mode. With unbuffered data modes, the user in his program supplies the device with an address, which is the beginning of a command list. Essentially, this command list contains pointers specifying areas in the user's allocated core to or from which data is to be transferred. The user program then waits until the operating system signals that the entire command list has been processed. Therefore, during the data transfer, the user program is idly waiting for the transfer to be completed.

Data transfers in buffered mode utilize a ring of buffers set up in the user's core area. Buffered transfers allow the user program and the operating system's I/O routines to operate asynchronously. As the user program uses one buffer, the operating system processes another one by filling or emptying it as interrupts occur from the device. To prevent the user program and the operating system from using the same buffer at the same time, each buffer has a use bit that designates who is using the buffer. Buffered data transfers are more efficient than unbuffered transfers because the user program and the operating system can be working together in processing the data.

Several steps must be followed by the user program in order for the operating system to have the information it needs to control the data transfers. Each step is indicated to the operating system with one programmed operator. In the first step, the specific device to be used in the data transfer must be selected and linked to the user program with one of the software I/O channels available to the user's job (OPEN or INIT programmed operators). This device remains associated with the software I/O channel until it is disassociated from it (via a programmed operator) or a second device is associated with the same channel. In addition to specifying the I/O channel and the device name, the user program can supply an initial file status, which includes the type of data transfer to be used with the device (e.g., ASCII, binary), and the location of the headers to be used in buffered data transfers. The operating system stores information in these headers when the user program executes programmed operators, and the user program obtains from these headers all the information needed to fill or empty buffers.

Another set of programmed operators (INBUF and OUTBUF) establishes the actual buffers to be used for input and output. This procedure is not necessary if the user is satisfied to accept the two buffers automatically set up for him by the operating system.

The next step is to select the file that the user program will be using when reading or writing data. This group of operators (LOOKUP and ENTER) is not required for devices that are not file-structured (e.g., card reader, magnetic tape, paper-tape punch). However, if used, they will be ignored, thus allowing file-structured devices to be substituted for non-file-structured devices without the user rewriting the program.

The third step is to perform the data transmission between the user program and the file (IN, INPUT, OUT, and OUTPUT). When the data has been transmitted to either the user program on input or the file on output, the file must be closed (CLOSE, fourth step) and the device released from the channel (RELEASE, fifth step). This same sequence of programmed operators is performed for all devices; therefore, the I/O system is truly device-independent because the user program does not have to be changed every time a different device is used.

In addition to reading or writing data to the standard I/O devices, provisions are included in the operating system for using the terminal for I/O during the execution of the user program. This capability is also obtained through programmed operators. As the user program is running, it can pause to accept input from or to type output to the terminal. The operating system does all buffering for the user, thus saving programming time. This method of terminal I/O provides the user with a convenient way of interacting with a running program.

Command Control Language

By allowing resources to be shared among users, the timesharing environment utilizes processor time and system resources that are wasted in single-user systems. Users are not restricted to a small set of system resources, but instead are provided with the full variety of facilities. By means of his terminal, the user has on-line access to most of the system's features. This on-line access is available through the operating system command control language, which is the means by which the timesharing user communicates with the operating system.

Through the command language, the user controls the running of a task, or job, to achieve the desired results: create, edit, and delete files; start, suspend, and terminate a job; compile, execute, and debug a program. In addition, since GALAXY batch software accepts the same command language as the timesharing software, any user can enter a program into the batch run queue. Thus, any timesharing terminal can act as a remote job entry terminal.

When the user types commands and/or requests on his terminal, the characters are stored in an input buffer in the operating system. The command decoder examines these characters in the buffer, checks them for correct syntax, and invokes the system program or user program as specified by the command.

On each clock interrupt, control is given to the command decoder to interpret and process one command in the input buffer. The command appearing in the input buffer is matched with the table of valid commands accepted by the operating system. A match occurs if the command typed in exactly matches a command stored in the system, or if the characters typed in match the beginning characters of only one command (i.e., constitute a unique abbreviation). When the match is successful, the legality information (or flags) associated with the command is checked to see if the command can be performed immediately. For instance, a command must be delayed if the job is swapped out to the disk and command requires that the job be resident in core; the command is executed on a later clock interrupt when the job is back in core. If all conditions as specified by the legality flags are met, control is passed to the appropriate program.

File System

Mass storage devices, such as disks and drums, cannot be requested for a user's exclusive use, but must be shared among all users. (An exception is the assignment of a disk structure to a single user; i.e., a "private" disk structure.) Because many users share these devices, the operating system must ensure independence among the users; one user's actions must not affect the activities of another unless the users desire to work together. To guarantee such independence, the operating system provides a file system for disks, disk packs, and drums. Each user's data is organized into groups of 128-word blocks called files. The user gives a name to each of his files, and the list of these names is kept by the operating system for each user. The operating system is then responsible for protecting each user's file storage from intrusion by unauthorized users.

In addition to allowing independent file storage for users the operating system permits sharing of files among individual users. For example, programmers working on the same project can share the same data in order to complete a project without duplication of effort. The operating system lets the user specify protection rights, or codes, for his files. These codes designate if other users may read the file, and after access, if the files can be modified in any way. A new facility called File DAEMON allows the user to specifically permit or deny access to any file, or set of files, for specific users. The user may also create a log of accesses to his files for later review, proprietary program billing, etc.

The user of the DECsystem-10 is not required to preallocate file storage; the operating system allocates and deallocates the file storage space dynamically on demand. Not only is this convenient for the user because he does not have to worry about allocation when creating files, but this feature also conserves storage by preventing large portions of storage from being unnecessarily tied up. However, a large batch job which needs to preallocate space may do so.

Files are assigned protection levels for each of three classes of users: self; users with a common project number; and all users. Each user class may be assigned a different access privilege; there are eight levels in each of the three user classes. The owner of a file may always change its protection.

In addition the File DAEMON is called on all access failures if the owner protection is 4,5,6,7 . . . Note that the protection used by the system to determine access right after a File DAEMON call is provided by the file DAEMON.

Table VIII. File Protection Scheme.

Protection Level	Access Code	Access Privileges
Greatest Protection	7	No access privileges (except by owner)
	6	EXECUTE ONLY
	5	READ, EXECUTE
	4	APPEND, READ, EXECUTE
	3	UPDATE, APPEND, READ, EXECUTE
	2	WRITE, UPDATE, APPEND, READ, EXECUTE
	1	RENAME, WRITE, UPDATE, APPEND, READ, EXECUTE
Least Protection	0	CHANGE PROTECTION, RENAME, WRITE, UPDATE, APPEND, READ, EXECUTE

File Handler

To reference a file, the user does not need to know where the file is physically located. A named file is uniquely identified in the system by a file name and extension, an ordered list of directory names (UFDs and SFDs) which identify the owner of the file, and a file structure name which identifies the group of disk units containing the file.

Usually a complete disk system is composed of many disk units of the same and/or different types. Therefore, the disk system consists of one or more file structures—a logical arrangement of files on one or more disk units of the same type. This method of file storage allows the user to designate which disk unit of the file structure he wishes to use when storing files. Each file structure is logically complete and is the smallest section of file memory that can be removed from the system without disturbing other units in other file structures. All pointers to areas in a file structure are by way of logical block numbers rather than physical disk addresses; there are no pointers to areas in other file structures, thereby allowing the file structure to be removed.

File Structures

A file structure contains two types of files; the data files that physically contain the stored data or programs, and the directory files that contain pointers to the data files. Included in these directory files are master file directories, user file directories, and sub-file directories. Each file structure has one master file directory (MFD). This directory file is the master list of all the users of the file structure. The entries contained in the MFD are the names of all the user file directories on the file structure. Each user with access to the file structure has a user file directory (UFD) that contains the names of all his files on that file structure; therefore, there are many UFDs on each file structure. As an entry in the user file directory, the user can include another type of directory file, a sub-file directory (SFD). The sub-file directory is similar to the other types of directory files in that it contains as entries the names of all files within that sub-directory. This third level of directory allows groups of files belonging to the same user to be separate from each other. This is useful when organizing a large number of files according to function. In addition, sub-file directories allow non-conflicting, simultaneous runs of the same program using the same file names.

As long as the files are in different sub-file directories, they are unique. Sub-file directories exist as files pointed to by the user file directory, and can be nested to the depth specified by the installation at system generation time.

File Protection

All disk files are composed of two parts: data and information used to retrieve data. The retrieval part of the file contains the pointers to the entire file, and is stored in two distinct locations on the device and accessed separately from the data. System reliability is increased with this method because the probability of destroying the retrieval information is reduced; system performance is improved because the number of positionings needed for random-access methods is reduced. The storing of retrieval information is the same for both sequential and random-access files. Thus a file can be created sequentially and later read randomly, or vice versa, without any data conversion.

One section of the retrieval information is used to specify the protection associated with the file. This protection is necessary because disk storage is shared among all users, each of whom may desire to share files with, or prevent files from being written, read, or deleted by other users. As discussed above, these protection codes are assigned by the user when the file is created and designate the users who have privileges to access the file.

Disk Quotas

Disk quotas are associated with each user (each project-programmer number) on each file structure in order to limit the amount of information that can be stored in the UFD of a particular file structure. When the user gains access to the computing system, he automatically begins using his logged-in quota. This quota is not a guaranteed amount of space, and the user must compete with other users for it. When the user leaves the computing system, he must be within his logged-out quota. This quota is the amount of disk storage space that the user is allowed to maintain when he is not using the system and is enforced by the system program that is used in logging off the system. Quotas are determined by the individual installation and are, therefore, used to ration disk resources in a predetermined manner.

To the user, a file structure is like a device; i.e., a file structure name or a set of file structure names can be used as the device name in command strings or UUO calls to the operating system. Although file structures or the units composing the file structures can be specified by their actual names, most users specify a general, or generic, name (DSK) which will cause the operating system to select the appropriate file structure. The appropriate file structure is determined by a job search list. Each job has its own job search list with the file structure names in the order in which they are to be accessed when the generic name is specified as the device. This search list is established by LOGIN and thus each user has a UFD for his project-programmer number in each file structure in which LOGIN allows him to have files.

File Operations

File writing on the disk can be defined by one of four methods: creating, superseding, updating and simultaneous updating. The user is creating a file if no other file of the same name exists in the user's directory on the indicated file structure. If another file with the same name already exists in the directory, the user is superseding, or replacing, the old file with the new file. Other users sharing the old file at the time it is being superseded continue using the old file and are not affected until they finish using the file and then try to access it again. At that time, they read the new file. When a user updates a file, he modifies selection parts of the file without creating an entirely new version. This method eliminates the need to recopy a file when making only a small number of changes. If other users try to access a file while it is being updated, they receive an error indication issued by the system.

Many users can update the same file through the use of the simultaneous update feature. This allows users to work with the same data base simultaneously. An ENQ/DEQ facility is also offered to allow record level lock out and thus eliminate race conditions where multiple users are attempting to read and/or write the same record.

Disk Storage Management

File storage is dynamically allocated by the file handler during program operations, so the user does not need to give initial estimates of file length or the number of files. Files can be any length, and each user may have as many files as he wishes, as long as disk space is available and the user has not exceeded his logged-in quota. This feature is extremely useful during program development or debugging when the final size of the file is still unknown. However, for efficient random access, a user can reserve a contiguous area on the disk if he desires. When he has completed processing, he can keep his preallocated file space for future use or return it so that other users can have access to it.

Input / Output

Peripheral Device Assignment

With the command language, the user can also request assignment of any peripheral device (magnetic tape, DECtape, and private disk pack) for exclusive use. When the request for assignment is received, the operating system verifies that the device is available to this user, and the user is granted its private use until he relinquishes it. In this way, the user can also have complete control of devices such as card readers and punches, paper tape readers and punches, and line printers.

Spooling

When private assignment of a slow-speed device (card punch, line printer, paper tape punch, and plotter) is not required, the user can employ the spooling feature of the operating system. Spooling is a method by which output to a slow-speed device is placed on a high-speed disk or drum. This technique prevents the user from consuming unnecessary system resources while waiting for either a device to become available or output to be completed. In addition, the device is managed to a better degree because the users cannot tie it up indefinitely, and the demand fluctuations experienced by these devices are equalized.

Memory Management

The DECsystem-10 is a multiprogramming system; i.e., it allows multiple independent user programs to reside simultaneously in memory and to run concurrently. This technique of sharing memory and processor time enhances the efficient operation of the system by switching the processor from a program that is temporarily stopped because of I/O transmission to a program that is executable. When core and the processor are shared in this manner, each user's program has a memory area distinct from the area of other users. Any attempt to read or change information outside of the area a user can access immediately stops the program and notifies the user.

Because available memory can contain only a limited number of programs at any one time, the computing system employs a secondary memory, usually disk or drum, to increase the number of users serviced. User programs exist on the secondary memory and move into memory for execution. Programs in memory exchange places with the programs being transferred from secondary memory for maximum use of available main memory. Because the transferring, or swapping takes place directly between main memory and the secondary memory, the central processor can be operating on a user program in one part of memory while swapping is taking place in another. This independent, overlapped operation greatly improves system utilization by increasing the number of users that can be accommodated at the same time.

To further increase the utilization of memory, the operating system allows users to share the same copy of a program or data segment. This prevents the excessive memory usage that results when a program is duplicated for several users. A program that can be shared is called a reentrant program and is divided into two parts or segments. One segment contains the code that is not modified during execution (e.g., compilers and assemblers) and can be used by any number of users. The other segment contains non-reentrant code and data. The operating system provides protection for shared segments to guarantee that they are not accidentally modified.

Virtual Memory

The virtual memory option permits a user program to execute with an address space greater than the physical memory actually allocated to that program during execution. User jobs are swapped as described above. However, the entire program may not necessarily be in core during execution. Programs are divided into pages each of which is 512-words long. Some of these pages may remain on secondary storage while the program executes. When a virtual memory job attempts to access a page that is not in core, a page fault handler decides which page or pages to remove from core and which to bring in from secondary storage.

Unlike the virtual memory implementation on other systems, this DECsystem-10 feature is an option. Each site may determine its own need for virtual memory and install it at their convenience. The system administrator may grant the privilege for using virtual memory only to those users who truly need its capabilities. Those users who are granted the privilege of using virtual memory may elect to invoke the feature for only those programs that could not execute without the virtual memory capability.

The virtual memory users may elect to use the system page fault handler or they may use a handler that is more tailored to the particular application or program behavior. Finally, it is important to point out that only those users actually using the virtual memory feature are affected by any additional overhead associated with a demand paging system. Non-virtual users execute as they would in a non-virtual system with no discernible difference in performance.

Multi-Processor Systems

DECsystem-10 Dual Processor systems are composed of two CPU's, designated the primary processor (master) and the secondary processor (slave). The primary processor is connected to all of the memory in the system and has all of the system's peripheral I/O equipment connected to its I/O bus. The secondary processor also has access to all of memory, however, there are normally no I/O devices on this processor's I/O bus. The primary processor performs exactly the same operations as the processor in a single processor system. This includes all I/O operations, swapping, core allocation, resource allocation and command decoding. The secondary processor also performs scheduling and execution of user jobs according to the same algorithm used in a single processor system.

The secondary processor executes user jobs and scans the same job queues as the primary processor. However, since the slave cannot do any standard I/O, it looks for any compute bound jobs which are in core and runnable. A software interlock has been added to the scheduler to prevent the possibility of both processors trying to execute the same job at the same time. Whenever a job being executed by the secondary processor requests an I/O operation to be performed the job is stopped and marked for execution on the master only. Thus both processors run completely asynchronously, both executing the same scheduler, doing the same job accounting and using the same job queues.

The existence of dual processors gives DECsystem-10 users a large scale computing capability, especially in the areas of highly compute bound jobs and non-interactive batch jobs. The existence of a slave more or less dedicated to user computation allows these jobs to be carried on in the systems with little interference with time sharing users. The performance goal is to provide a system in which each customer can improve the service offered to his users over a single processor compute bound system. Either the customer can add more users with the same response time or he can keep the same number of users and reduce turn around time for compute bound jobs.

nter-job Communication

nared Data Areas

The DECsystem-10 operating system ables a user to communicate with ner jobs through the use of sharable ta areas. This also enables a data alysis program, for example to read write an area in the real-time job's re space. Since the real-time job as- ciated with the data acquisition would locked in core, the data analysis pro- am residing on secondary memory ould become core resident only when e real-time job had filled a core buf- with data. Operating system calls n be used to allow the data analysis ogram to remain dormant in secon- ry memory until a specified event curs in the real-time job: e.g., a buffer s been filled with data for the data alysis.

Inter-Process Communication Facility

The Inter-Process Communication Facility (IPCF) provides the capability for independent jobs to communicate with one another. For example, if several programs are involved in processing or maintaining a data base, it is possible that one program might want to inform the others of any modifications it made to the data. A job using IPCF cannot make any changes to another job so protection is in no way sacrificed when using the IPCF feature.

In order to use the IPCF, each participating job that wishes to receive communication from other jobs must request a unique process identifier (PID) from the system. The transmitting job then may send a "packet" of information to another job. (In addition to the information, the system automatically provides a "return address" so that the receiving program can respond to the sender.) The monitor maintains a linear queue (the "mailbox") for each job using IPCF. The packet (or packets) will be kept in the mailbox until the receiving job retrieves it. This queue is not created until a job sends an IPCF packet and it does not occupy any space until such time. The maximum number of packets allowed in a queue at any one time is determined by a "receive" quota that may be set at each installation for each user. (If no quota is set by the installation, the standard default is five.)

On systems with the virtual memory option, the packet could be an entire page. In this case, the monitor takes advantage of the page mapping hardware of the KI10 and KL10 to transmit the page without actually copying it.

Communications Software

DECsystem-10 users have a wide selection of communications products to enhance or facilitate their computing needs. Within the multi-task environment of TOPS-10, functionality exists for:

- Asynchronous communication—the typical vehicle for interactive time-sharing or transaction-oriented terminals
- Synchronous communications—for connection of remote batch, remote terminal concentration and computer-to-computer links.

The important feature of DECsystem-10 communications software is its implementation as an integral part of the TOPS-10 Operating System. The monitor, not the user, handles the communications housekeeping, and all communications products are fully supported within the TOPS-10 environment. Appropriate synchronous line protocols—DIGITAL's own DDCMP or BISYNC—are supported.

Users may configure networks with simple or complex topologies utilizing the DN87 family of Universal Front End equipment. The DAS-80 and DAS-90-series remote stations/concentrators can then be used to expand the network topologies. In addition, DECNET-10 permits the DECsystem-10 to utilize task-to-task communication with other DECsystem-10's or with any other DEC-supplied computer system which supports DECNET. The DAS-61 and DAS-62 can be used to do IBM 2780, 3780 or HASP multi-leaving emulation and terminination.

Additional details concerning the DECsystem-10 communications capability and products are included in the Data Communications Section.

Multimode Computing

The DECsystem-10 is designed for the concurrent operation of timesharing, multistream batch, real-time, and remote communications in either single- or dual-processor systems. In providing these multifunction capabilities, the DECsystem-10 services interactive users, operates local and remote batch stations, and performs data acquisition and control functions for on-line laboratories and other real-time projects. By dynamically adjusting system operation, the DECsystem-10 provides many features for each class of user and is therefore able to meet a wide variety of computational requirements.

Timesharing

Timesharing takes maximum advantage of the capabilities of the computing system by allowing many independent users to share the facilities of the DECsystem-10 simultaneously. Because of the interactive, conversational, rapid-response nature of TOPS-10 time-sharing, a wide range of tasks—from solving simple mathematical problems to implementing complete and complex information gathering and processing networks—can be performed by many users concurrently. The number of users on the system at any one time depends on the system configuration and the job mix on the system. DECsystem-10 timesharing is designed to support in excess of 200 active users. Interactive terminals can include CRTs, hard copy terminals and other devices which operate at speeds of 110 to 9600 baud. Terminal users can be located at the computer center or at remote locations connected to the computer center by communication lines.

Timesharing on the DECsystem-10 is general purpose; i.e., the system is designed in such a way that the command language, input/output processing, file structures, and job scheduling are independent of the programming language being used. In addition, standard software interfaces make it easy for the user to develop his own special language or systems. This general purpose approach is demonstrated by the many special-purpose programming languages implemented by DECsystem-10 users.

GALAXY-10 Batch

GALAXY-10 batch software enables the DECsystem-10 to execute in excess of 30 batch jobs concurrently with timesharing jobs. Just as the timesharing user communicates with the system by way of his terminal, the batch user normally communicates by way of the card reader. (However, he can also enter his job from an interactive terminal.) Unlike the timesharing user, the batch user can punch his job on cards, insert a few appropriate control cards, and leave the job for an operator to run. In addition, the user can debug the program in the time sharing environment and then run it in batch mode without any additional coding.

The GALAXY-10 system consists of a series of programs: QUASAR, the system queue manager and scheduler, SPRINT, the input spooler, BATCON, the batch controller, and the output spoolers, LPTSPL and SPROUT. The input spooler is responsible for reading the input from the input device and for entering the job into the batch controller's input queue. The input spooler sends a message to QUASAR to do the actual entering into the batch input queue. Although the input spooler is oriented toward card reader input, disk and magnetic tape also can be handled. The input information is then separated according to the control commands in the input deck and placed into disk files for subsequent processing. In addition, the input spooler creates the job's log file and enters a report of its processing of the job, along with a record of any operator intervention during its processing. The log file is part of the standard output that the user receives when his job terminates.

fter the input spooler reads the end-
ile and closes the disk files, it makes
ntry in the batch controller's input
ue. The batch controller processes
ch jobs by reading the entries in its
ue. The control file created by the
t spooler is read by the batch con-
ler, and data and user program com-
nds are passed directly to the user's
. Operating system commands are
ected by the batch controller and
sed to the operating system for ac-
. Most operating system and user
gram commands available to the
esharing user are also available to
batch user. Therefore, only one
trol language need be learned for
h timesharing and batch. During the
cessing of the job and the control
the batch controller adds informa-
to the log file for later analysis by
user.

UASAR is responsible for scheduling
s and maintaining both the batch
troller's input queue and the output
oling queues. A job is scheduled to
under the batch controller accord-
to external priorities, processing
e limits, and core requirements
ch are dynamically computed ac-
ding to parameters specified by the
r for his job, such as start and dead-
time limits for program execution.
ASAR makes an entry for the job in
batch input queue based upon the
ous priorities. After the job is com-
ed an entry is made in the output
ues for the job's spooled output and
job's LOG file.

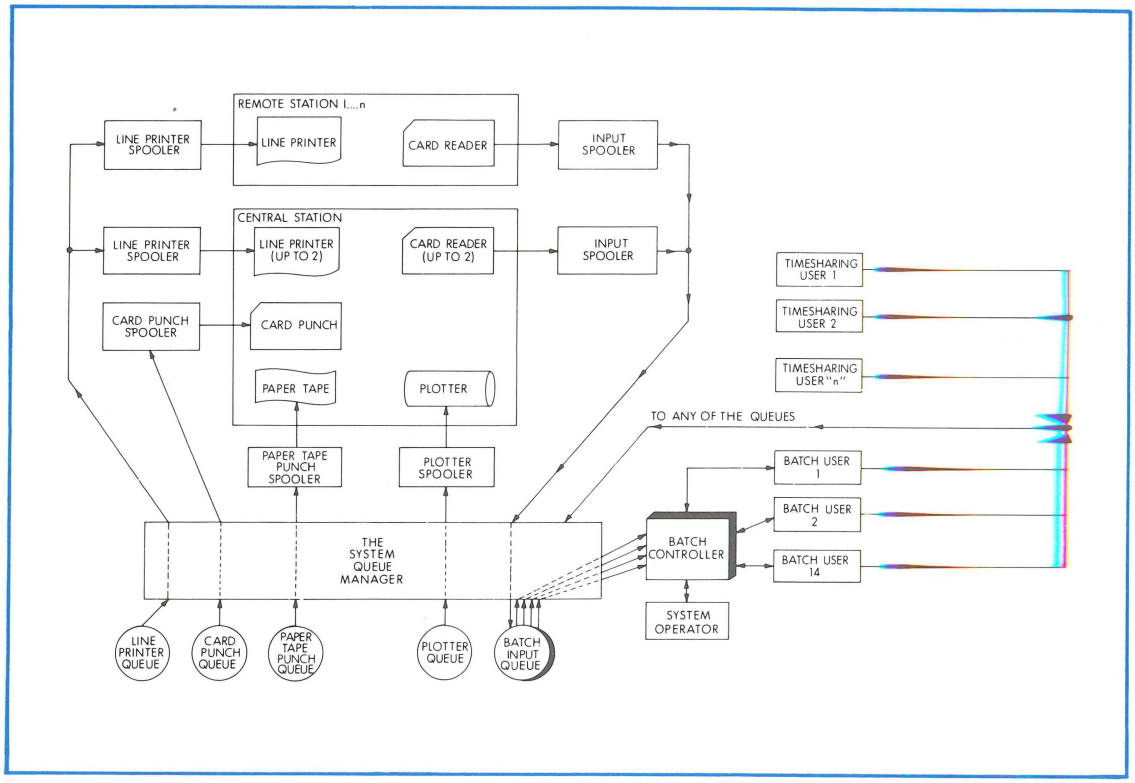
he output spooling programs im-
ve system throughput by allowing
output from a job to be written tem-
arily on the disk for later transfer
ead of being written immediately
a particular output device. The log
and all job output are placed into
or more output queues to await
cessing. When the specified device
vailable, the output is then pro-
sed by the appropriate spooling
gram. These spooling programs
be utilized by all users of the com-
ng system.

Use of System Features

The GALAXY-10 batch software employs many of the computing system's features in order to operate with maximum efficiency. Because core memory is not partitioned between batch and timesharing jobs, batch jobs can occupy any available area of core. Fast throughput for high-priority batch jobs is accomplished with the same swapping technique used for rapid response to interactive users. When available core is not large enough for a high-priority batch job, the operating system transfers programs of lower priority to secondary memory in order to provide space for the job. This transfer is done at the same time that the processor is operating on another job. Thus, processing can be overlapped with swapping (and other I/O) to utilize time that would otherwise be wasted. Batch jobs can also share programs with timesharing and other batch jobs. Only one copy of a sharable program need be in core to service any number of batch and timesharing jobs at the same time.

Flexibility

GALAXY-10 batch allows the user great flexibility. The input spooler normally reads from the card reader but can read from magnetic tape or disk in order to create a control file on disk and to enter the job into the batch controller's input queue. However, a job can be entered from an interactive terminal, in which case the user bypasses the input spooler and creates a control file on disk for the batch controller. The control file contains the operating system commands and user program commands necessary to run the job. The user then enters the job into the batch controller's input queue by way of an operating system command string. In the command string, the user can include switches to define the operation and set the priorities and limits on core memory and processor time.



Job Dependency

Although jobs are entered sequentially into the batch system, they are not necessarily run in the order that they are read because of priorities either set by the user in an input spooler control command or computed by the queue manager when determining the scheduling of jobs. Occasionally, the user may wish to submit jobs that must be executed in a particular order; in other words, the execution of one job is dependent on another. To ensure that jobs are executed in the proper order, the user must specify an initial dependency count in a control command of the dependent job. This dependency count is then part of the input queue entry. A control command in the job on which the dependent job depends decrements the count. When the count becomes zero, the dependent job is executed.

Error Recovery

The user can control system response to error conditions by including commands to the batch controller which will aid in error recovery. These commands are copied into the control file by the input spooler. With error recovery commands, the user specifies the action to be taken when his program contains a fatal error, as for example, to skip to the next program or to transfer to a special user-written error handling routine. If an error occurs and the user did not include error recovery conditior s in his job, the batch controller initiates a standard dump of the user's core area and terminates the job. This core dump provides the user with the means to debug his program.

Although the batch system allows a large number of parameters to be specified, it is capable of operating with very few user-specified values. If a parameter is missing, the batch system supplies a reasonable default value. These defaults can be modified by the individual installations.

Operator Intervention

Normal operating functions performed by the programs in the batch system require little or no operator intervention; however, the operator can exercise a great deal of control if necessary. He can specify the number of system resources to be dedicated to batch processing by limiting the number of programs and both the core and processor time for individual programs. He can stop a job at any point, requeue it, and then change its priorities. By examining the system queues, he can determine the status of all batch jobs. In addition, the programs in the batch system can communicate information to the operator and record a disk log of all messages printed at the operator's console. All operator intervention during the running of the input spooler and the batch controller causes messages to be written in the user's log file, as well as in the operator's log file, for later analysis.

Real-time Computing

For a system to be satisfactory for real-time operations, two important requirements must be met. The more important requirement is fast response time. Because real-time devices may not be able to store their information until the computing system is ready to accept it, the system would be useless for real-time if the response requirements of a real-time project could not be satisfied. The operating system allocates system resources dynamically in order to satisfy the response and computational requirements of real-time jobs.

The second requirement is protection. Each user of the computing system must be protected from other users, just as the system itself is protected from all user program errors. In addition, since real-time systems have special real-time devices associated with jobs, the computing system must be protected from hardware faults that could cause system breakdown. And, because protection is part of the function of the operating system, the real-time software employs this feature to protect users as well as itself against hardware and software failures. Inherent in the operating system is the capability of real-time, and it is by way of calls to the operating system that the user obtains real-time services. The services obtained by calls within the user's program include:

- Locking a job in core.
- Connecting a real-time device to the priority interrupt system.
- Placing a job in a high-priority run queue.
- Initiating the execution of FORTRAN or machine language code on receipt of an interrupt.
- Disconnecting a real-time device from the priority interrupt system.

ocking Jobs

Memory space is occupied by the esident operating system and by a mix f real-time and non-real-time jobs. The nly fixed partition is between the resi- ent operating system and the remain- er of memory so as not to lose informa- on when its associated real-time evice interrupts (since there may not e sufficient time to swap-in the job) the ob can request that it be locked into ore. This means that the job is not to be wapped to secondary memory and uarantees that the job is readily avail- ble when needed. Because memory is ot divided into fixed partitions, it can e utilized to a better degree by dyna- ically allocating more space to real- me jobs when real-time demands are igh. As real-time demands lessen, ore memory can be made available regular timesharing and batch usage.

Real-time Devices

The real-time user can connect real-time devices to the priority interrupt system, respond to these devices at interrupt level, remove the devices from the interrupt system, and/or change the priority interrupt level on which these devices are assigned. There is no requirement that these devices be connected at system generation time. The user specifies both the names of the devices generating the interrupts and the priority levels on which the devices function. The operating system then links the devices to the operating system.

The user can control the real-time device in one of two ways: single mode or block mode. In single mode, the user's interrupt program is run every time the real-time device interrupts. In block mode, the user's interrupt program is run after an entire block of data has been read from the real-time device. When the interrupt occurs from the device in single mode or at the end of a block of data in block mode, the operating system saves the current state of the machine and jumps to the user's interrupt routine. The user services his device and then returns control to the operating system to restore the previous state of the machine and to dismiss the interrupt. Any number of real-time devices may be placed on any available priority interrupt channel for the program to read. When the specified event occurs, the dormant program is then activated to process the data. The core space for the real-time job's buffer area or the space for the dormant job does not need to be reserved at system generation time. The hardware working in conjunction with the operating system's core management facilities provides optimum core usage.

High-Priority Run Queues

The real-time user can receive faster response by placing jobs in high-priority run queues. These queues are examined before all other run queues in the computing system, and any runnable job in a high-priority queue is executed before jobs in other queues. In addition, jobs in high-priority queues are not swapped to secondary memory until all other queues have been scanned. When jobs in a high-priority queue are to be swapped, the lowest priority job is swapped first and the highest priority job last. The highest priority job swapped to secondary memory is the first job to be brought into core for immediate execution. Therefore, in addition to being scanned before all other queues for job execution, the high-priority queues are examined after all other queues for swapping to secondary memory and before all other queues for swapping from secondary memory.

Diagnostic Software

"High Availability"

TOPS-10 provides a number of features which allow the system to continue operating although some devices may be inoperable.

1) When errors occur on disks, tapes, memory, or the slave CPU, TOPS-10 will record the error data and notify the operator. If the error can be fixed (e.g., put a disk back on-line), the user may continue running without further delay. Alternatively, the operator may instruct the system to remove (detach) the device, in which case the jobs in error will be notified and the system will continue operation without the specified device. If the error is in a bank of memory, TOPS-10 will even attempt to move its own code (if any) out of the failing memory. In the case of disks, the system will attempt to migrate any swapping space to another unit to prevent loss of jobs.
2) RP04, and RP06 type disks may be dual ported so that in case of controller failure, the packs may be accessed via an alternate path. In case of a unit failure, the pack may be moved to a working unit.
3) With use of appropriate I/O bus, memory bus, and device control switches, DECsystem-10's may be configured to provide redundancy of hardware. Such systems may allow any piece of hardware to be taken off-line and repaired while the system continues operation (in some cases, it may be necessary to reload the system after a significant reconfiguration).

Maintenance Features

The Diagnostic Strategy has produced a high quality of maintenance aids which enforce the increased performance and availability of a DECsystem-10. The design of the KL10 processor uses integrated diagnostic logic to enable maximum visibility and reduce "hard-core" requirements. Special software techniques are employed to maximize test coverage per diagnostic load, thereby decreasing the mean time to diagnose.

Board Level Diagnosis

The Writable Control Store feature of the KL10 has enabled the "Micro Diagnostic" alternate path methodology of testing. This technology has enabled generation of diagnostics which narrow malfunctions to the board level with a degree of confidence which exceeds industry standards.

Maintainability was attacked from a total system's concept, for not only are the CPU diagnostics capable of supporting the "Module Swap" philosophy of repair, the new generation of peripheral diagnostics gives the maintainer the same qualities. The programs utilize special diagnostic wrap around logic to isolate suspected malfunctions to subsystem level. Then, through comprehensive logic analysis, not just functionality, the repair person is directed to the failing module.

On-line Diagnosis

To further increase system availability, special features have been incorporated into the monitor to enable on-line preventive maintenance.

Emphasis has also been placed on testing the thoroughness of detectability and the correctness of isolation. Semiautomatic physical fault insertion is qualifying and providing maturity enhancements for this product.

KLINIK Remote Diagnosis

One final feature of the Maintenance and Diagnostic Philosophy is the support of remote diagnosis (KLINIK). This can provide distant maintenance personnel with hardware performance statistics to improve the efficiency of service on KL10 Based Systems.

Services

Quality in DIGITAL computer products is acknowledged throughout the world. Our engineers and programmers have provided reliable computer hardware and software to solve a host of problems in a variety of fields. To complement our product offering, we have developed a comprehensive customer service capability.

SERVICES	**85**
Hardware Maintenance	**87**
People	87
Parts	87
Support	87
Service Agreements	**88**
Education	**88**
Logistics	**89**
Technical Documentation	**89**
Software Services	**90**
Advanced Systems Group	**91**

Hardware Maintenance

Over the past few years, Datamation Magazine has run an annual survey of computer users, asking their opinions on the quality of service they receive from their vendors. One major area of concern is post-installation maintenance services. DIGITAL has continually ranked first in quality and availability of service.

Major factors in this capability are the number and placement of service personnel; parts inventories available to support local efforts; an organizational structure that provides increasing expertise at each level; a management reporting system that continuously monitors all areas of service; and a wide range of service agreements to meet every customer need.

People

DIGITAL's hardware maintenance organization includes 3000 hardware specialists supported by an administrative staff of over 1000. These specialists are located in more than 300 worldwide offices for the greatest decentralization of service availability.

Large Systems service engineers typically enter DIGITAL with more than four years of field experience. They then undergo four months of intensive training at DIGITAL, followed by yearly updates in new products and technology. They work with product support teams who assure smooth transitions of new products—from development to testing to installation—and provide consultation and assistance for all service locations.

Parts

The size and location of DIGITAL's parts inventories—at local, district, regional, and headquarters operations—are computer-optimized by our own DECsystem-10, which takes into account the needs of both existing and new installations. The DECsystem-10, in concert with PDP-11 computers at regional offices, assures that parts resources are available locally to solve more than 90% of user problems.

For those problems requiring parts not available immediately, DIGITAL has a priority parts system utilizing inventories at district, regional and corporate levels. All priority parts are shipped by specialized air-freight carriers.

Support

Field Service's hierarchical organization places the greatest concentration of resources at the local level, with supporting facilities at each ascending level in the group. Continuous monitoring of service activity at all levels of Field Service management focuses attention on our worldwide systems base so that resources can be mobilized as required.

Service Agreements

To give flexibility to this structure and provide services that meet users exact needs, DIGITAL offers a variety of service agreements. These agreements range from coverage 12 hours/day, five days/week to around-the-clock maintenance coverage. Occasional service, on a time and materials basis, is available to users who elect to perform self-maintenance, as well as off-site repair of electro-mechanical assemblies through 16 Product Repair Centers in the U.S. and in Europe.

Education

Education is an integral part of the total Digital customer service system. We believe in training, not only for ourselves, but also for our customers. Your employees will, through education, extract greater performance from your computer.

To provide this training, we have established completely equipped training centers in a dozen locations around the world. Our staff at these centers consists of full-time instructors dedicated to computer training.

The education DIGITAL offers includes standard and custom courses in both hardware and software. Our current schedule includes over 90 standard courses, ranging in duration from one to five weeks.

DIGITAL recognizes that you may have a unique requirement that can best be met with education at your location. Our on-site program meets this need by designing and conducting courses where and when you choose.

Logistics

Good service demands outstanding logistics. Spares availability minimizes computer repair time. Our materials inventory network recognizes this concept—and strengthens the total DIGITAL support system.

The first level of spares is often at your own computer site. From this point the inventory is in echelons at the Branch, District, Regional, and Headquarters locations. Our own DECsystem-10 keeps track of this support system, providing replacement parts when needed.

For DIGITAL computers located in areas on the fringes of our service capabilities, and for customers who use our maintenance on an "as available" basis, DIGITAL supplies spares planning and inventory procurement assistance.

Technical Documentation

DIGITAL is a major publisher of manuals, handbooks, and engineering literature. These documents support more than 65,000 installed DIGITAL computers. They meet user requirements for installation, maintenance, education and operation. They provide an authoritative guide to DIGITAL systems.

One example is DEC-O-LOG. Maintaining current systems and documents is a critical task. DEC-O-LOG satisfies this requirement by listing and explaining engineering improvements as they evolve. This information is available to keep you abreast of the latest technological developments.

Software Services

The DIGITAL software organization represents over 3,000 man-years of experience gained from the development and support of operating systems used in more than 65,000 computer installations. DIGITAL's software varies in complexity from real-time executives for minicomputers to the TOPS-10 Monitor of the DECsystem-10. Applications cover the spectrum from process control and monitoring scientific experiments to implementing reservation and inventory control systems.

DIGITAL offers a wide range of software services. These services *range* from the *personal attention* of a skilled software consultant to the *distribution of up-to-date software and software information.* In this way you can get the most out of your DECsystem-10 and keep pace with advancements in software.

You purchase only what you need. Software components, including manuals and updates, can be purchased as part of a Software Maintenance Service or ordered separately from the Software Distribution Center. Software maintenance plans for the DECsystem-10 include:

1. Software Notebook Updates–Each customer receives two sets of comprehensive software notebooks which can be kept current via periodic update packages. You are assured of always having the most current documentation published.
2. Program Updates–Subscribers receive the latest software versions along with all available "bug-fixes" on magtape.
3. Software DISPATCH–This service provides all available code corrections as well as a copy of all customer-generated Software Performance Reports published as an early warning service. This is a comprehensive publication aimed at the customer in a dynamic, growing software environment.
4. Expedited Software Performance Report Service–This service provides a very fast priority response system with personalized replies returned directly to the originator, thus bypassing normal lags associated with publication and program update cycles. This insures that any problems you encounter will be corrected.
5. Consulting Service–These are available on a short-term, per-call basis or for a longer term scheduled or resident period. Consultants can install software updates, train new employees, and recommend procedures that result in top performance by your software system.

Advanced Systems Group

The Advanced Systems Group, an integral part of the DECsystem-10 product group, meets special customer's needs by augmenting the standard products and services of the DECsystem-10. The Advanced Systems Group extends the range of the DECsystem-10 product and services by providing:

1. Hardware/Software Design and Development–Engineering for special products and modifications to standard products, with emphasis in the areas of communications, network systems, high availability and redundant systems, and special configurations.
2. Systems Project Management for large and complex configurations including such functions as systems evaluation and design, systems planning for future expansion, extended factory systems testing, special installation backup and support, and extensions to the DECsystem-10 acceptance test procedures.
3. Repeat Manufacture of previously designed and developed special products.

Advanced Systems Group products maintain DIGITAL's high standard of quality. All hardware products are supportable under DECsystem-10 field service support plans and are supplied with full documentation, prints and diagnostics.

Support of special software is provided through a centralized support group on an individual basis for both installation and Software Performance Reports (SPR's).

Because of the customized nature of these products, the Advanced Systems Group provides individual system configuration review, system integration and system installation assistances when needed.

digital

DIGITAL EQUIPMENT CORPORATION
Large Computer Group
Marlborough, Massachusetts 01752

In Europe: Digital Equipment Corporation International, Geneve 26, Switzerland

PRINTED IN U.S.A. EH 07127 63/77 010 03 30