Commercial Computer Systems

General Information

TEXAS INSTRUMENTS.

DS990 Commercial Computer Systems

General Information

Published by

TEXAS INSTRUMENTS
INCORPORATED

Texas Instruments reserves the right to make changes at any time to improve design and supply the best product possible.

The information in this document is subject to change without notice and should not be construed as a commitment by Texas Instruments Incorporated. Texas Instruments assumes no responsibility for any errors that may appear in this document.

1st Edition, July 1978

Texas Instruments Incorporated
Digital Systems Division
P.O. Box 2909
Austin, Texas 78767

Table of Contents

DS990 Commercial Systems Overview

DS990 Software

DS990 Hardware

Customer-Support Services

Appendixes

List of Illustrations

List of Tables

DS990 Commercial Systems Overview

The Model 4, Model 6, and Model 8 Systems, shown in figure 1, combine the performance of the Model 990/10 Minicomputer with the power of the DX10 disk-based commercial operating system. The systems are specifically designed for commercial, interactive, multiuser, multitasking, multilanguage, and communications applications.

The models differ in disk-file capacity to provide an orderly growth path. The Model 4 System is a 10M-byte disk-based system suitable for a small software-development system or medium-scale application system. The Model 6 System is a dual 25M-byte disk-based system suitable for medium-scale software-development and application systems. The Model 8 System is a dual 50M-byte disk-based system intended for medium- to large-scale software-development and application systems. Physical packaging differences between the three models are based on the space requirements of the disk-storage units.

The base systems are offered in the minimum configurations that support the full functions of the operating system. This allows maximum flexibility in adding optional software and hardware features to customize a system. The various models are available in either equipment-only versions or in versions that include system software and installation. Software-included versions provide a licensed copy of the DX10 operating system and software-development facilities, documentation, and a one-year software subscription service for the DX10 system. A factory-provided sysgen is also installed on systems that include software. Systems are configured and tested before they are shipped from the factory.

Figure 1. DS990 Model 4 System (foreground) and Model 6 or Model 8 System (background)

DS990 Software

DS990 software is both versatile and efficient. It comprehends user requirements in a way not found in most computers of this class.

DX10 Operating System

Operators interface with the DX10 operating system through Model 911 Video Display Terminals (VDTs) that provide hierarchical menus with supporting fill-in-the-blank, prompting, and predefined functions.

More than 170 Texas Instruments-supplied system commands provide powerful and comprehensive system control. Users can also provide custom commands that reflect their application terminology. A broad range of utilities and supporting routines are built into the system-command structure.

Program memory is dynamically allocated. Resource management is enhanced by variable-location roll in/roll out and task-level priority assignment.

The DX10 operating system has flexible file management. Types of files supported include sequential, relative-record, and multikey-indexed files. The key-indexed method supports a unique real-time, self-maintenance capability.

Program and Memory Management. The DX10 operating system is a multiprogramming system. User programs that operate under control of the DX10 system include a composite of tasks, procedures, and overlays. Programs are installed and stored in program files. When a program is activated, its images are loaded into any available memory area.

An active program can be rolled in and out of several different locations in memory by the DX10 operating system several times during its execution to efficiently share memory and processor resources. When in memory and active, a program competes with other programs for execution time on a user-defined priority basis. When a program terminates, the operating system releases all program-owned resources including files, devices, and memory. This unique DX10 program structure is made possible by a memory-mapping technique. These advanced memory- and program-management techniques provide high processor utilization, resulting in high levels of throughput.

System Command Interpreter. Operators interface with the DX10 operating system through VDTs via the system command interpreter (SCI). The SCI is a collection of more than 170 procedures that provide system functions ranging from setting the time of day or initiating compiles to backing up disks. Commands are at the operator's fingertips via the SCI. This can save 10 to 30 percent of development effort on major programs. The completeness and flexibility of the functions performed by the SCI make it without parallel in the minicomputer market. Many of the functions performed by the SCI are found only on main-frame machines.

Activation of SCI commands is via a hierarchy of command menus made available to all types of system terminals. The command menus provide each terminal on-line command prompts by logical grouping.

Custom commands can be integrated into the framework of the DX10 operating system. Users can combine SCI primitives with their own application language to provide a user interface that is unique to the terminology and customary procedures of the application.

Interactive Operation. The DX10 operating system features an excellent interactive user interface for control of the system through SCI. All entries keyed by an operator are meaningfully prompted. Fields are easily edited by the operator and are verified by the system. The number of prompts, and therefore time, can be conserved since all arguments for a command can be entered before the system requests them by prompt. When a partial list of arguments is entered, any arguments not already supplied by the operator or default specified are then prompted.

Batch Operation. The background program at the terminal may be a copy of the SCI. In this case, the SCI is interpreting commands in the background (batch processing). Batch input is from any sequentially oriented file device but not from the terminal itself. An operator can initiate batch processing, query its status, and receive information concerning its normal or abnormal completion. Certain interactive commands are inappropriate for batch operation, but all other SCI commands are available.

File Management

The DX10 operating system provides a file-management package that includes a complete range of file structures and features. The DX10 system can accommodate many uniquely named data files on a disk cartridge and provides the necessary management for allocation of disk space to the files. The user can specify the amount of space to be allocated to a file or, more frequently, can specify that space is to be automatically allocated to the file as it is needed.

File Types. Three major file types are supported by the DX10 operating system: sequential, relative-record, and multikey-indexed files. Sequential files allow records with concurrent reads. Relative-record files provide rapid access to fixed-length records in either random or sequential mode. In multikey-indexed files, variable-length records are accessed by providing the DX10 operating system any one of up to fourteen keys by which the data is known. The keys are in sorted order and allow rapid access to data addressed by the keys.

Multikey indexing provides a unique self-maintenance capability. Deleted or added keys are automatically removed or inserted in the sorted key lists. The DX10 system automatically expands or contracts the key lists and eliminates much of the necessity for periodically rebuilding and reorganizing files.

File Features. Various file features and file types are available to the assembly-language user. High-level languages may or may not allow access to any given feature, depending on the syntax of the language. Some of the supported features include:

- Record locking
- Temporary files
- Blocked files
- Deferred or immediate write operation
- Delete and write protection
- Access privileges
- Blank compression and adjustment
- Expandable files
- Blank compression.

Error Control and System Log

The DX10 operating system incorporates several error-control features and supports an optional system log. The 990 mapping feature protects the DX10 system from destruction by errant application programs. An optional end-action routine analyzes abnormal termination and takes appropriate recovery steps.

High-Level Languages and Utilities

COBOL, RPG II, DBMS 990, BASIC, Business BASIC, FORTRAN, Pascal, and Sort/Merge packages are available as options on all DS990 systems.

The COBOL compiler conforms to the American National Standards Institute (ANSI) COBOL subset (ED 1X3.23-1974) and incorporates extensions to this subset to provide added capabilities.

The RPG II compiler conforms to the IBM System/3* RPG II specifications with certain equipment and teleprocessing exceptions. Texas Instruments version of RPG II is video-display-oriented and provides one-line-at-a-time forms or multiline listings.

Texas Instruments DBMS 990 is a modular data-base management system specifically designed for minicomputer applications. DBMS 990 includes a data-definition language (DDL) for defining the logical structure of data and a data-manipulation language (DML) that interfaces through COBOL for storing and retrieving data. A number of utilities and security features can optionally be used within the modular structure of DBMS 990.

The BASIC** language is a version of Dartmouth BASIC as described in *BASIC Programming,* by Kemmeny and Kurtz, with certain extensions to enhance its use. The extensions are integer arithmetic type, expanded string handling, CALL, and subprograms.

BASIC is aimed at the scientific user. Business BASIC, a variation of BASIC, is an easily understood, business-oriented, application-solving language. Single-key-indexed file input/output (I/O) and limited-output editing capabilities are included to provide a check-printing capability.

The FORTRAN compiler conforms to the ANSI standard FORTRAN, or FORTRAN IV. The compiler also incorporates the extensions recommended by the Instrument Society of America (ISA-S61.1, 1975 and ISA-61.2, 1976).

Pascal is a general-purpose language well suited for a variety of applications. Originally designed as a language for teaching a systematic concept of programming, Pascal is straightforward to learn and to use. Its readability makes the language especially useful when programs must be maintained by users other than the original author.

The DX10 operating system supports a comprehensive Sort/Merge package that can be accessed in several ways. SCI provides commands to access Sort/Merge in batch or interactive mode. COBOL, RPG II, FORTRAN, and BASIC programs can interface with Sort/Merge by using the CALL statement. Both sort and merge processes support record selection, reformatting on input, and summarizing on output. Ascending key order, descending key order, or an alternate collating sequence can be specified.

Program-Development Tools

In addition to a comprehensive set of utilities that operate in conjunction with the DX10 operating

*IBM System/3 is a registered trademark of IBM.
**BASIC is a registered trademark of Dartmouth College.

system, Texas Instruments provides four major program-development tools: interactive text editor, macro assembler, link editor, and debug package. Each of these operates under SCI and is easily invoked by operator commands.

Communications Software

The DX10 3780 Emulator provides the DS990 family of commercial systems with a means of remote-job-entry (RJE) communications with an IBM 360/370 host computer or another 3780 emulator-equipped 990 computer. Operation of the IBM 3780 Data Communications Terminal is emulated. DS990 systems so equipped can operate in unattended mode as central or satellite stations in a distributed network. Optional auto-call capability is also provided. Data files are transmitted over leased point-to-point or switched telephone lines at speeds up to 9600 baud. Any file or system device can be specified to transmit or receive data. Hardware support of 3780 emulation is provided with the 990 communications interface module to a customer-supplied modem or Texas-Instruments-supplied modem kit and optional auto-call unit.

DS990 Hardware

The DS990 disk-based systems require a fast, flexible computer architecture to meet the processing demands of multiple interactive operations. The Model 4, Model 6, and Model 8 Systems are based on a minimum 128K-byte 990 processor with 16K random-access-memory (RAM) technology. The processor features the TILINE* asynchronous, high-speed data bus and a memory-mapping technique that allows addressing of up to 2048K bytes of main memory.

All models employ moving-head disk drives with at least one removable disk pack and one additional fixed or removable disk pack. This allows copy, backup, and transportability of media that are so important in interactive systems.

Each system includes one 1920-character video display terminal and keyboard. A dual-terminal controller is included and allows the easy addition of a second VDT. Additional VDTs are available as options.

990 Processor

The 128K-byte 990 processor features an advanced memory-to-memory architecture and TILINE asynchronous, high-speed data bus. Current models offer main-memory capacities up to 384K bytes of metal-oxide-semiconductor (MOS), 16K dynamic-RAM, error-checking-and-correcting (ECC) memory. The additional ECC memory is available in various module sizes. TILINE expansion to a second chassis is recommended for these memory modules. Memory can be further expanded by adding standard 990 options.

The 990 processor includes a dual input/output system, a high-speed, asynchronous parallel TILINE bus, and a low- to medium-speed command-driven communications register unit (CRU).

TILINE Peripherals

TILINE peripherals are high-speed I/O systems which transfer data to and from 990 memory at rates that approach the instruction execution rate of the 990 computer.

The DS990 base system includes one of the following disk systems, depending on the model selected. The DS31 Disk System has a 2315-type disk cartridge with 2.8M bytes of formatted capacity. This disk drive is an option on the Model 6 and Model 8 Systems.

The DS10 Disk System has a single-spindle, fixed- and removable-platter disk drive employing the 5440-type disk cartridge. Each platter has a 4.7M-byte capacity for a combined disk-drive capacity of 9.4M bytes. This is the disk drive supplied with the Model 4 System and is offered as an option on the Model 6 and Model 8 Systems. One DS10 disk controller can accommodate up to two DS10 disk drives.

The DS25 Disk System has a multiplatter disk pack with 22.3M bytes of formatted capacity. Dual DS25 disk drives are employed on the Model 6 System. One DS25 disk controller can accommodate up to four DS25 disk drives.

The DS50 Disk System has a multiplatter disk pack with 44.6M bytes of formatted capacity. Dual DS50 disk drives are employed on the Model 8 System. One DS50 disk controller can accommodate up to four DS50 disk drives.

The Model 979A Magnetic-Tape Transport is an optional TILINE peripheral. Two versions are available: an 800-bits-per-inch, nonreturn-to-zero format and a 1600-bits-per-inch, phase-encoded format. Both versions use industry-compatible, nine-track tape formats. The transport operates at 953 millimetres per second (37.5 inches per second).

CRU Peripherals

The command-driven communications-register-unit (CRU) peripherals are low- to medium-speed I/O systems that transfer data to and/or from the 990 processor. These peripherals are optional features that further enhance the DS990 system.

*TILINE is a registered trademark of Texas Instruments.

The Model 810 Printer is an impact printer with 9 x 7 dot-matrix character structure and a ninety-six-character, full ASCII, compressed print set. The 810 printer prints 132-column lines at 150 characters per second with eight-channel vertical-format control.

The Model 2230 and Model 2260 Line Printers print 136-column lines at 300 lines per minute and 600 lines per minute, respectively, with vertical-format control, internal self-test, static eliminator, and standard ASCII, sixty-four-character set.

Additional Model 911 Video Display Terminals are available as single- or dual-terminal displays with keyboards and interfaces. A maximum of twelve VDTs can be installed in their predefined slots in a two-chassis configuration. Additional VDTs can be installed in other vacant slots by redefining the interrupt assignments. VDTs can also be installed in add-on chassis.

The Model 804 Card Reader is a 400-card-per-minute reader that takes standard-sized, eighty-column punched or marked cards.

The *Silent 700** Model 733 ASR and Model 743 KSR Data Terminals use Texas Instruments unique solid-state, thermal printheads for virtually silent printing of eighty-column lines at thirty characters per second. A typewriter-style, limited ASCII keyboard allows operator entries.

The Model FD800 Floppy-Disk System provides transportable diskette media to smaller members of the 990 family of systems.

Communications Equipment and Special Interface Devices

Optional communications equipment includes an RS-232-C communications interface module for asynchronous and synchronous transmission at selectable baud rates from 75 to 9600. Other supporting options include asynchronous and synchronous modems and auto-call. A variety of interface modules for custom-device interface or process monitoring and control are available including analog-to-digital converters and digital input and output interfaces. A variety of Teletypewriter/Electronics Industries Association (TTY/EIA) interfaces are also available.

990 Chassis Considerations

The 990 computer is packaged in a thirteen-slot chassis with a programmer panel. The chassis includes a 40-ampere power supply and a disk-loader read-only memory (ROM). One or more chassis can be added to the system to provide additional mounting space or dc power.

Customer-Support Services

Texas Instruments customer services encompass the following areas: hardware installation, software installation, hardware maintenance, software update, education classes, and telephone hot line. Individual services can be selected to best suit application and customer requirements.

**Silent 700* is a registered trademark of Texas Instruments.

DS990 Software

DS990 Software

DS990 software features a versatile and powerful operating system, the DX10 operating system. File-management and error-control features make this operating system an extremely efficient one. A wide variety of high-level languages are available with the DS990 system. Among these are COBOL, FORTRAN, RPG II, BASIC, Pascal, and DBMS 990. Software-development tools include an interactive text editor, macro assembler, link editor, and debug package. These features enable DS990 software to comprehend user requirements in a way not found in most computers of this class.

DX10 Operating System

The operating-system software selected for a computer installation has a major effect on the throughput, reliability, and usability of the system. The disks, processors, memories, terminals, and other physical resources of a computer system represent a potential for performance. The operating system controls the total system resources and allocates them to the various tasks required in the user's application programs. An operating system also provides other necessary system services and utilities, such as interrupt handling and input/output (I/O) device service routines. The operating system relieves the individual application program of these overhead responsibilities, which greatly reduces the opportunity for error and simplifies application programming.

The DX10 operating system is Texas Instruments powerful disk-based operating system; it is one of the most sophisticated, flexible, and versatile computer operating systems available. First released in 1976, the DX10 system is a proven product that has been refined by operation within Texas Instruments and at customer sites around the world.

The DX10 system is a general-purpose, multitasking operating system, which is optimized for multiterminal interactive operations. The operating system makes each terminal appear to have exclusive control of the system; so the existence of other terminals is transparent to any individual user. Batch-processing tasks can be initiated from any terminal without interfering with interactive terminal operations.

The DX10 operating system has extensive file-management capabilities. The DX10 system creates files, allocates disk and memory space for files, transfers files between disk and memory as required, and controls read access, write access, deletion, and file sharing by multiple users. The operating system supports sequential files, relative-record files, and multikey-indexed files with up to fourteen keys.

The DX10 operating system incorporates a number of program-development utilities, which greatly reduces development time and improves effectiveness. Programs can be developed under the DX10 operating system to execute on the DS990 system or on the smaller floppy-disk-based or memory-resident systems in the 990 family. Program-development packages are sold as extra-cost options to many operating systems; however, they are included as part of the DX10 system because program development is a major application of the DS990 systems. The program-development utilities include an interactive text editor, relocatable macro assembler, link editor, and interactive debugger.

Many users prefer to write programs in one or more of the major high-level programming languages. Compilers/interpreters for FORTRAN (with Instrument Society of America (ISA) process-control extensions), COBOL, BASIC, Business BASIC, RPG II, DBMS 990, and Pascal are available as optional features of the DX10 operating system.

The DX10 operating system includes the device service routines necessary to communicate with the standard and optional DS990 I/O devices, such as the Model 911 Video Display Terminals (VDTs), disk systems, Model 810 Printers, Model 2230 and Model 2260 Line Printers, Model 979A Magnetic-Tape Transport, and *Silent 700* terminals. Special device service routines can be linked to the DX10 system. The DX10 system uses logical I/O, which further isolates an individual user program from the characteristics of the I/O device. Logical I/O allows an I/O device to be treated as a file that is referenced by a logical unit number (LUNO).

This simplification allows the user to concentrate on the application problem rather than on device details. Logical I/O is particularly useful for program development in which a file or another I/O device can be substituted for the device that will ultimately be used.

A versatile, interactive, fully prompted system command interpreter (SCI) allows direct communication between the operating system and the user. This powerful command interpreter is a major convenience of the DX10 system.

The DX10 operating system is a modular operating system that can be customized for each customer's site. Customizing the operating system improves the efficiency of resource utilization. An interactive system-generation (sysgen) program is supplied with the DX10 system to provide a fast and convenient means of implementing the custom system.

To protect Texas Instruments investment and its customers' interests, the DX10 operating system is only available on a licensed basis. The license includes software updates for one year following purchase with annual renewal subscriptions available. (The software subscription service is also available to users who allow their subscriptions to lapse and then decide to update their systems.) The DX10 software license also includes installation support (basic system generation and verification) by a Texas Instruments customer engineer. For additional information, refer to the *990 Computer Family Price List*.

System Generation

The DX10 operating system is a modular operating system that can be customized for each customer's site. Customizing the operating system has the following advantages:

- Reduces disk- and memory-space requirements (Eliminates unnecessary modules, such as device service routines for equipment not included in the customer's system)
- Increases operating speed (Eliminates code sequences associated with unnecessary modules)
- Eliminates replication of device service routines for multiple installations of a device type
- Adds device service routines for nonstandard devices
- Adds user-defined operating-system service calls
- Adds user-defined extended operation (XOP) processors
- Adjusts operating-system parameters for best efficiency in a given installation.

The DX10 operating system is supplied to a DS990 purchaser on a media that is compatible with the system. Unless otherwise specified, the DX10 system is supplied on a DS10 (5440-type) cartridge for DS990 Model 4 Systems, on a DS25 disk pack for DS990 Model 6 Systems, and on a DS50 disk pack for DS990 Model 8 Systems.

The DX10 disk includes a standard DX10 operating system that is compatible with the basic DS990 configuration (including standard options). The disk also includes a complete set of individual DX10 component modules, which can be used in the development of the custom DX10 system. An interactive sysgen program builds the custom system to user specifications. The sysgen program uses prompting and tutorial displays to lead the user through system generation. When system generation is complete, a complete set of specifications is displayed for user approval. At this point, the user may back out, revise specifications, store the custom system as an alternate system, back it up with an additional copy, or activate it as the primary operating system.

Program Management

A multitasking operating system allocates the resources of a single computer system to a number of individual user programs in such a way that each user appears to have exclusive control of the computer system. Multitasking maximizes the amount of useful work obtained from a given amount of computer hardware.

User programs that operate under control of the DX10 operating system include a composite of tasks, procedures, and overlays. Programs are installed and stored in disk program files in memory-image form. When a program is activated, its memory images are rolled in from the disk and loaded into any available memory areas. This specific activation of a program is called a task. Multiple tasks can be in memory at any time. A scheduler in the DX10 operating system allocates central-processing-unit (CPU) execution time to the various tasks. Conceptually, every task is in some partial state of completion with one task actually executing while the others are either active (in queue, ready to execute) or suspended (not ready to execute).

The process of removing one task from execution and putting another into execution is called scheduling. Tasks scheduled by expiration of a time limit are called time-shared tasks. Time-shared tasks are queued at different priority levels. A sophisticated scheduling algorithm allocates time slices at the various priority levels; so no task is locked out.

The architecture of the 990 computer family supports multitasking software. The workspace architecture allows rapid context switches with a minimum of overhead time. The general registers are not saved or restored each time a different task is scheduled. Memory mapping allows each task to execute in an independent, fully protected address space. It permits the logical address space of tasks to be segmented into up to three physical memory areas. Mapping also protects the operating system from destruction by errant application programs.

Priority Scheduling. The DX10 operating system requires that each task have a defined priority level. Four priority levels are available: level 0 (highest) through level 3 (lowest) and level 4 (floating). Level 0 is reserved for DX10 internal use. Higher priority tasks are granted execution preference over lower priority tasks. However, even lower priority tasks are guaranteed some execution time. Tasks designated with equal priority execute in round-robin fashion. Each priority level is allocated a number of execution units of time. When all the allocated units have been used, one unit of time is given to the next lower priority task.

Tasks with Variable Priorities. When a task is installed, it may be designated with a floating priority level (level 4). In this event, the task is loaded at priority level 1 when execution is first desired. After initial execution, its priority is dynamically set to level 1 for terminals and level 2 for other devices on each I/O request. Task priority is lowered to the next lower level (never below level 2) after a specified number of time slices when it is executing. A floating priority (managed by the DX10 operating system) permits rapid response to I/O events and deemphasizes the task during periods when task processing could compute bind the system. For example, application programs that function interactively normally are installed with a floating priority level.

Shared Procedures and Replicated Tasks. Having several concurrent executions of the same program is desirable in many multiterminal environments or in industrial applications where symmetrically alike devices are controlled. An example might be a program serving bank tellers that interacts concurrently with several tellers.

In many cases, the procedural part of the program is common to each of the concurrent executions; however, the data is unique to each separate execution. Under the DX10 operating system, the shared procedural part is called a procedure while the unique part is called a task. The three allowed program segments are therefore allocated to two procedures and one task.

A program operating under the DX10 operating system can consist of a task and none, one, or two procedures. The procedures can be shared with other executing tasks. Sharing procedures conserves memory usage because replicating the procedural part of a program is unnecessary. Conversely, the task part is indeed unique to each separate execution. The DX10 operating system provides a convenient mechanism to replicate tasks for multiple executions. In cases where each concurrent activation of a program has the same initial data, only one program image need be stored on disk. For each terminal, a task can be replicated from a single image installed in a program file on disk. Replicating tasks conserves disk space and time because installing a copy of the same initial task for each possible concurrent activation of a program is unnecessary.

Sharing Data among Tasks. Under the DX10 operating system, a block of data can be shared among two or more tasks through the use of a shared-procedure segment. A procedure can contain data that is shared among several tasks.

Overlays. As programs become large, they can be partitioned to allow only a portion of the program to be resident in memory at a given time. The overlay support provided by the DX10 operating system provides the mechanism to accomplish the disk-resident part. The remainder is divided into disk-resident overlays.

When an overlay module is required, the program initiates a supervisor call that loads it into memory. Overlay modules can be further segmented into a lower level consisting of a "root" and overlays. Multilevel overlay structures are supported by the link editor.

Task Activation by a Program. Any task can request that another task be activated. Both tasks then become concurrently active. The DX10 operating system supports the identification of a station with which the new task is to be associated. In this manner, all of the station-local logical-I/O-unit assignments are available to the next task. Furthermore, the requesting task can specify that it be suspended until termination of the activated task. This provides a convenient mechanism for a master application program serving a station to activate subprocesses either in parallel with the master program or instead of the master program. In the latter case, the master program resumes execution when the subprocess completes.

Program Files. All tasks, procedures, and overlays are installed in structures referred to as program files. These files are based on the expandable relative-record file type and contain program images in blocks corresponding to file records. An internal directory is maintained within the program file. This internal directory contains pointers to each image on the file as well as relevant information about the images. Typically, the DX10 operating system requires two disk accesses to load a disk-resident task, procedure, or overlay: one for the directory entry and one for the image.

One program file is designated as the system program file. The system program file initially contains only programs that constitute parts of the DX10 operating system. Other program files can be created to hold application programs.

Program Identification. Program parts stored in program files can be retrieved by task, procedure, or overlay numbers specified at installation time. A program can be installed with or without the replicative attribute.

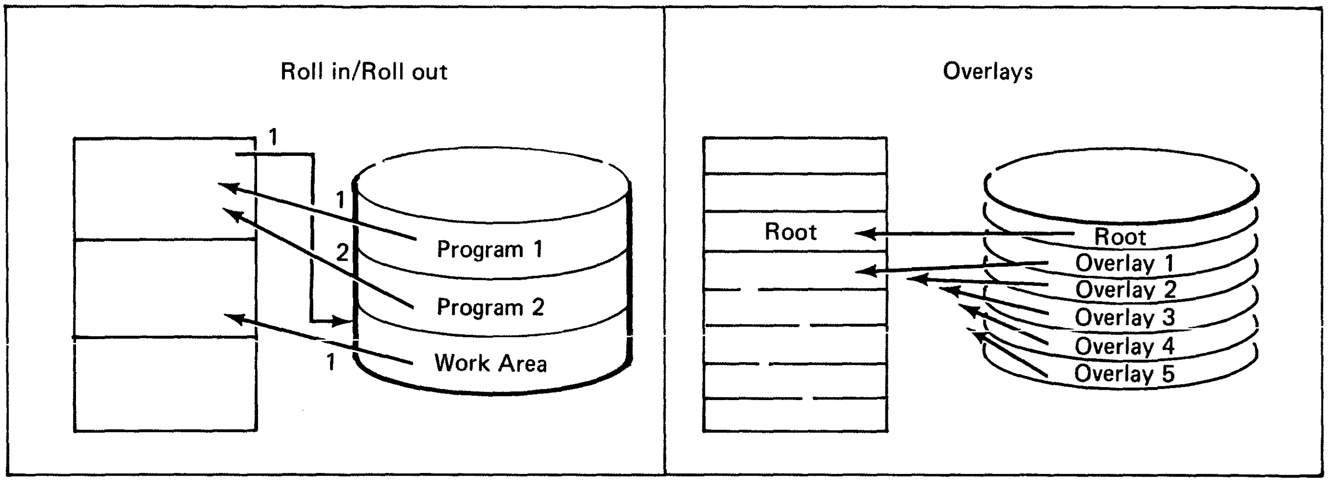


Figure 2. Memory Conservation through Disk Management

Memory Management

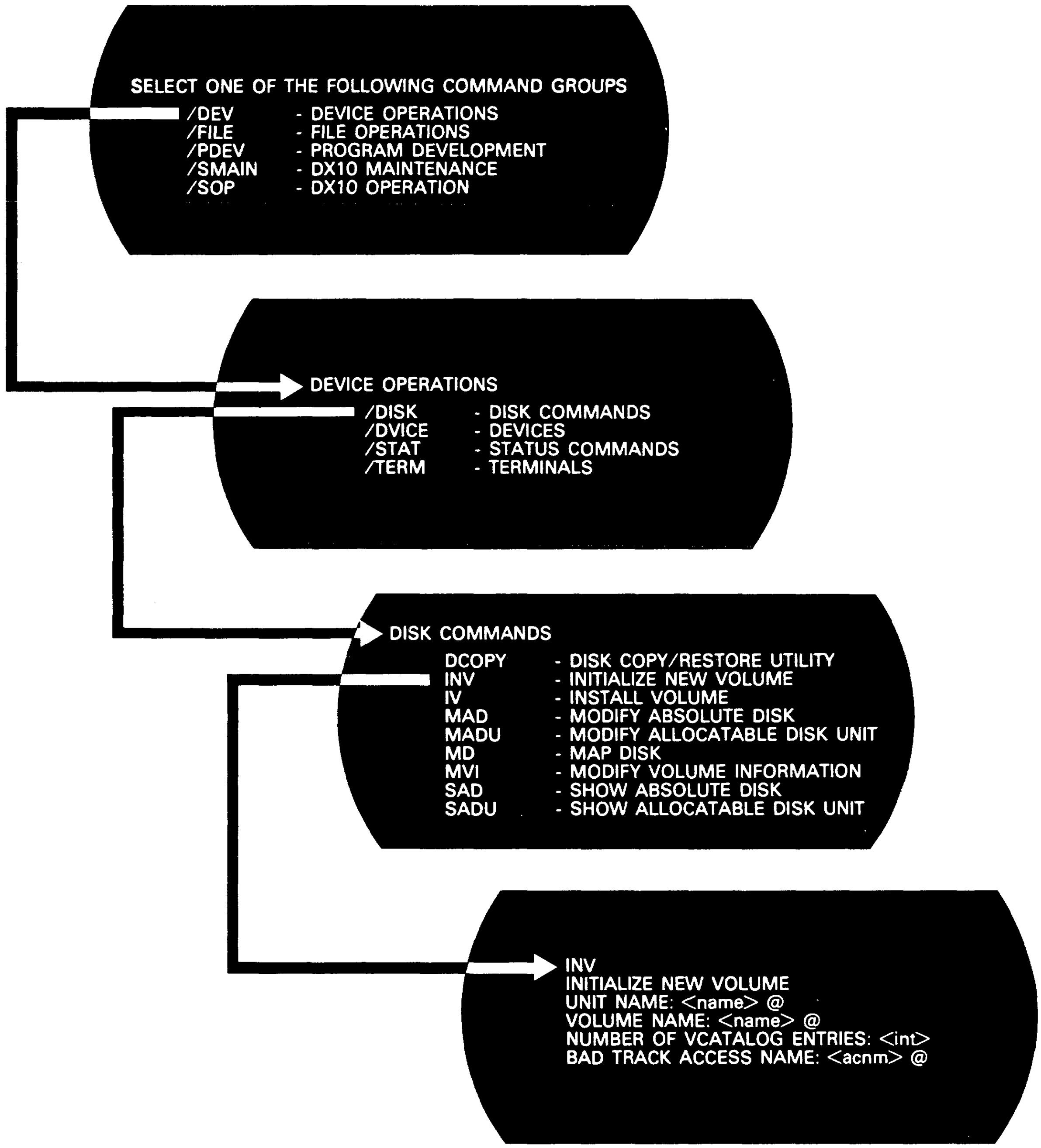
The DX10 operating system uses the 990 mapping option to dynamically allocate memory to the disk-resident task segment, procedure segment, and file-blocking buffers. The allocated blocks can be released from memory and rolled to disk as needed. The roll-in/roll-out mechanism, shown in figure 2, ensures efficient use of main memory and CPU time.

The DX10 operating system incorporates an algorithm that permits programs of high priority to preempt memory space from those of lesser or equal priority. Any program can preempt space from a suspended program. Whenever insufficient memory space is available to permit the operating system to execute a program, the DX10 system seeks lower-priority or suspended-task segments and dispatches these programs to disk. This process is called roll out. Similarly, when the task and priority mix indicates, the rolled-out program is rolled in from the disk and execution resumes. The memory-mapping feature permits a program segment to be restored to a different physical memory space than it occupied at the time it was rolled out. The priority-oriented roll-in/roll-out mechanism guarantees high-priority tasks immediate access to memory to respond to users or other external stimuli.

File blocks are allocated in the dynamic memory area. Any blocks of data retained by the DX10 operating system from recent disk transfers can be preempted by the operating system if the memory space is needed. These blocks are written to their appropriate file location on the disk (if they have been updated) to acquire memory space.

System Command Interpreter

The DX10 system command interpreter (SCI) provides an interactive, conversational user interface at a terminal and also provides a background batch-processing mode. The SCI is a collection of more than 170 procedures that provide system functions ranging from setting the time of day or initiating compiles to backing up disks. Commands are at the operator's fingertips via the SCI. This can save 10 to 30 percent of development effort on major programs. The completeness and flexibility of the functions performed by the SCI make it without parallel in the minicomputer market. Many of the functions



Any portion of the sequence can be skipped by going directly to the command desired.

Figure 3. Example of the Menus for Initialization of a Disk Volume

performed by the SCI are found only on main-frame machines. Table 1 lists the general categories of functions provided by the DX10 operating system and initiated by the SCI.

Appendix A lists all standard SCI commands available with the DX10 operating system. Activation of SCI commands is via a hierarchy of command menus made available to all types of system terminals. The command menus provide each terminal on-line command prompts by logical grouping. The example in figure 3 illustrates the menus for initialization of a disk volume.

Custom commands can be integrated into the framework of the DX10 operating system. Users can combine SCI primitives with their own application language to provide a user interface that is unique to the terminology and customary procedures of the application. Figure 4 is an example of a custom procedure.

Table 1. Areas Served by DX10 SCI Commands

Log in and out
Time and date setup and inquiry
Disk volume initialization, installation, and unloading
Disk directory backup, restore, and copy
Directory and file creation and deletion
Synonym support
File alias name
File name changing and protecting
Directory and file viewing and listing
Directory and file copying
Logical unit assignment, positioning, and release
System I/O status display
System task status display
Program activation and control
Batch command input, activation, and status
Station control (user ID, terminal status, etc.)
Program installation and deletion
System log activation
Program debugging including:
Breakpoints
Memory/disk dump or display
Decimal/hexadecimal arithmetic aid
Interactively controlled program trace
Text Edit control
COBOL, RPG II, DBMS 990, Business BASIC, BASIC, FORTRAN, and Pascal compilers and assembly-language assemblies
Link Edit activation
Sort/Merge activation

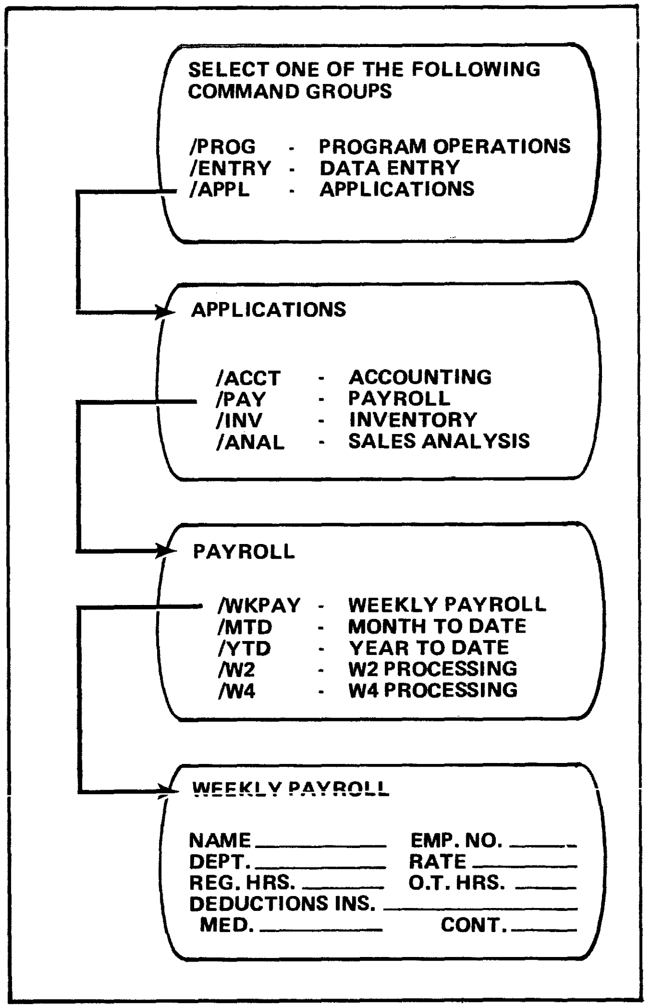


Figure 4. Custom Procedure Example

Interactive Operation. The DX10 operating system features an excellent interactive user interface for control of the system through SCI. All entries keyed by an operator are meaningfully prompted. Fields are easily edited by the operator and are verified by the system. The number of prompts, and therefore time, can be conserved since all arguments for a command can be entered before the system requests them by prompt. When a partial list of arguments is entered, any arguments not already supplied by the operator or default specified are then prompted.

Batch Operation. The background program at the terminal may be a copy of the SCI. In this case, the SCI is interpreting commands in the background (batch processing). Batch input is from any sequentially oriented file device but not from the terminal itself. An operator can initiate batch processing, query its status, and receive information concerning its normal or abnormal completion. Certain interactive commands are inappropriate for batch operation, but all other SCI commands are available.

File Management

The DX10 operating system is based on a file-management package that includes a complete range of file structures and features. The DX10 system can accommodate many uniquely named data files on a disk pack and can allocate disk space to the files. The amount of space allocated to a file can be as small as two disk sectors or as large as all of the available space on the disk cartridge. The user can specify the amount of space to be allocated to a file or, more frequently, can specify that space is to be automatically allocated to the file as it is needed.

File Types

Three major file types are supported by the DX10 operating system: sequential, relative-record, and multikey-indexed files.

Sequential Files. Sequential files are useful for recording data records in the order received. Similarly, data is returned in the same order. A pointer to the current file position is kept by the DX10 operating system for each active assignment to the file. As each record is read or written, the pointer is advanced. In sequential files, no valid data can exist beyond the most recently written record. The only exception to this rule is that a limited rewrite of a record is supported. A sequential file can be segmented, and support for multiple end-of-files within a sequential file is provided. Several programs can concurrently read a sequential disk file at different positions in the file. A sequential file cannot be shared if it is being modified.

Relative-Record Files. Relative-record files are optimized for rapid random access. Fixed-length records are accessed by supplying the DX10 operating system the record number within the file. Such files are useful when the data lends itself to computation of a record number. The DX10 system increments the caller's record number after each read or write; so a sequential access is permitted. One end-of-file record is maintained.

Multikey-Indexed Files. In multikey-indexed files, variable-length records are accessed by providing the DX10 operating system any one of up to fourteen keys by which the data is known. (A key is a string of up to 100 characters at a fixed position within the record.) One of the fourteen possible keys must be selected as the primary key. All other keys are known as secondary keys. Primary keys must be present in all records, but secondary keys can be optionally absent in any given record within the file.

For example, an employee file can be constructed so the data record for any given employee can be accessed by supplying the employee's name, employee number, social security number, or any other designated key.

Key values for both primary and secondary keys are kept in indexes within the files. These indexes are hierarchically structured. This allows both rapid random access to any record within the file or sequential access to all records in the file in the stored order of any selected key.

The user can perform the following types of functions with key-indexed files:

- Randomly read any record.
- Insert and delete records. (Records are automatically blank suppressed on insertion, and duplicate key values are allowed.)
- Establish a generic file position as in a COBOL start command.
- Read files sequentially in the sort order of a given key starting at an established file position. (Files can be read in ascending or descending order.)
- Add, delete, or change a secondary-key value when records are updated.

Multikey indexing provides a unique self-maintenance capability. Deleted or added keys are automatically removed or inserted in the sorted key lists. The DX10 operating system automatically expands or contracts the key lists and eliminates much of the necessity for periodically rebuilding and reorganizing files.

File Features

Various file features and file types are available to the assembly-language user. High-level languages may or may not allow access to any given feature, depending on the syntax of the language. Assembly-language programs can be written and called from high-level-language programs to allow indirect access to a feature.

Record Locking. The DX10 operating system supports locking individual records within a file. This feature allows a program exclusive access to the locked record until that record is unlocked. An example might be locking an inventory record while updating the quantity in stock. The lockout prevents programs responding to other terminals from updating the same quantity before the first update is complete.

Temporary Files. The DX10 operating system allows the use of temporary files. These files are subject to subsequent deletion by the DX10 system. This feature allows a trial preparation of a file. If the prepared file is satisfactory, it can be renamed and designated as permanent.

Blocked Files. Multiple logical records can be automatically combined by the DX10 operating system into larger physical records. These larger records are called file blocks or physical records. Blocking conserves disk space and reduces the number of physical transfers of data between memory and disk.

Deferred or Immediate Write. The physical transfer of logical-record blocks to disk is normally deferred by the DX10 operating system until the memory space held by the blocking buffer is required for some other purpose. This reduces the number of physical disk accesses since data may be recalled. The DX10 system updates the image of the disk before the file is closed. In some cases, it may be desirable (e.g., for security) for all writes to a file to occur immediately upon request. The DX10 system supports this immediate write option.

Delete and Write Protection. After each file is created, it can be protected from accidental deletion from the volume. In some cases, it may be desirable to further protect a file. The DX10 operating system permits file write protection, which only allows the data to be read. Files that are write-protected are automatically delete-protected.

Access Privileges. For any use of a file, a DX10 program can request specific access privileges. A use is defined as the entire file transaction from open through close. These privileges include:

Privilege	Function
Exclusive Access	Only the calling program can access the file.
Exclusive Write Access	Only the calling program can write to the file.
Shared Access	The calling program shares the file for read and write operations.
Read Only	The calling program is prohibited from writing to the file.

Blank Compression and Adjustment. For file types that support variable-length records (i.e., all files except relative-record files), blank characters can be optionally removed from each record. Blank compression encodes strings of consecutive blanks within the record in a shortened form. Blank adjustment removes trailing blanks on a write operation and replaces them on a subsequent read operation. Blank adjustment is available to devices as well as files.

Expandable Files. The DX10 operating system permits declaration of the file size at file creation. Unless otherwise specified, additional space is allocated when the file exceeds this initial allocation. In this way, files continue to grow beyond their initial bounds. These secondary allocations to the file become increasingly larger as the file expands beyond its current extent.

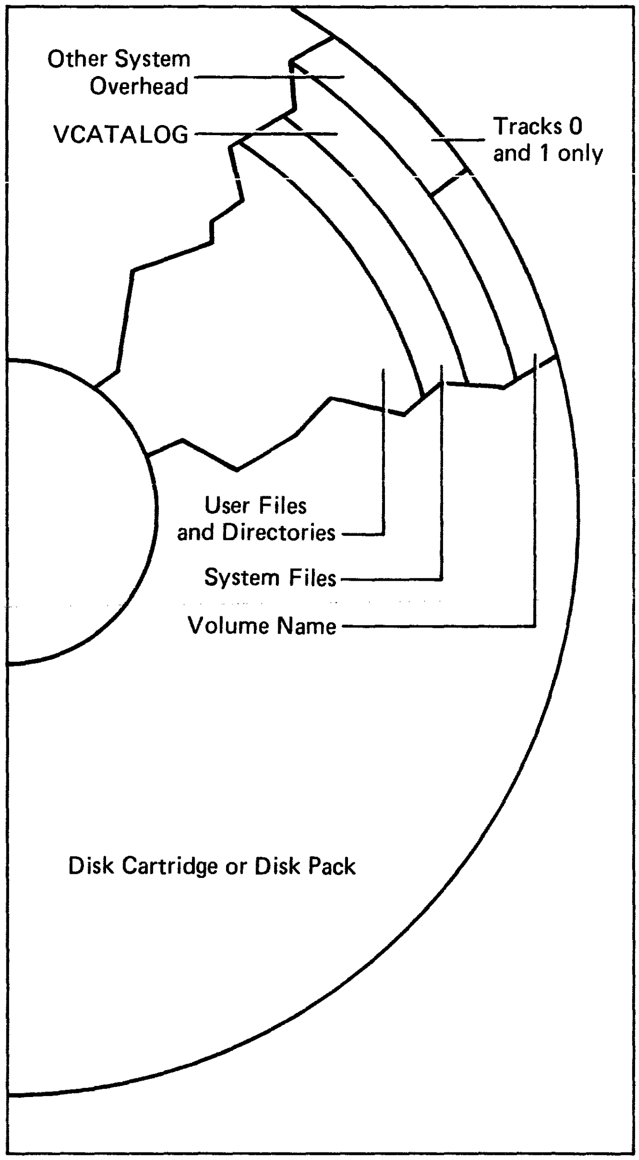


Figure 5. Disk-Volume Layout

Physical Disk Characteristics

Figure 5 shows the disk-volume layout. Under the DX10 operating system, each disk volume contains overhead space reserved for user files. The DX10 system uses one disk drive from which the operating system is loaded. That disk is designated as the system disk and is used by the DX10 system to perform its internal disk-based functions. The DX10 system disk or secondary disks are described in the following paragraphs.

System Overhead Area. Tracks 0 and 1 are always allocated to system overhead on the system-disk and secondary-disk volumes. Each disk under the DX10 operating system is identified by a user-assigned name. User-assigned names are a part of specific volumes and are recorded on its associated disk. The assigned name is called the volume ID. The DX10 operating system maintains a map of used/unused disk space for each volume on the disk. A system loader is stored on disk to initialize memory with a DX10 image at initial-program-load (IPL) time. The DX10 system maintains a map of unusable surface on the disk. The bad-disk-surface map is initialized to reflect disk condition at disk initialization.

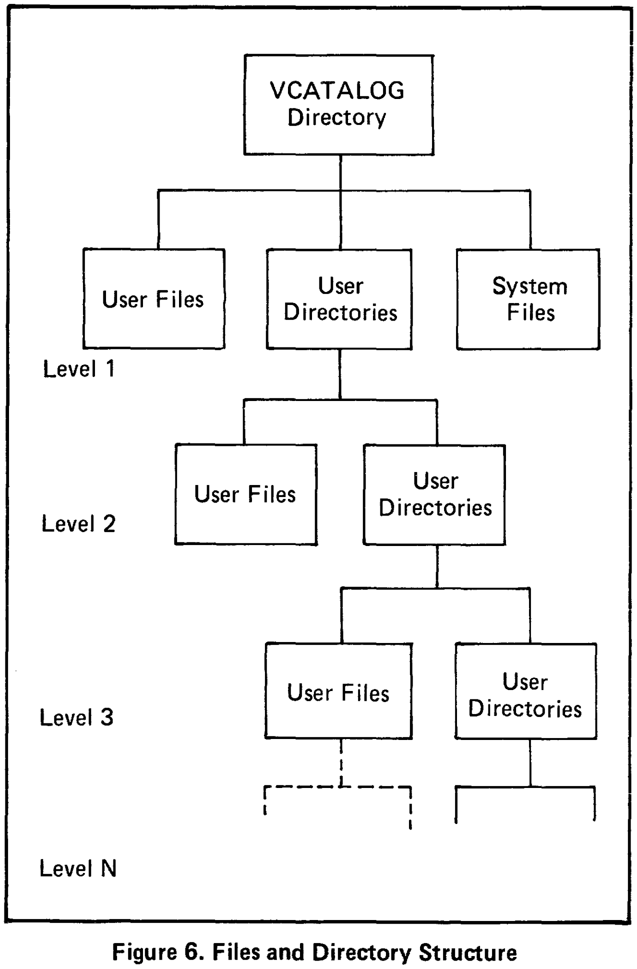


Figure 6. Files and Directory Structure

Dynamic File Area. Area on a disk volume other than tracks 0 and 1 is dynamically allocated to files. Each disk volume has a specific file directory named VCATALOG where the DX10 operating system maintains a volume table of contents. As shown in figure 6, the files described in VCATALOG can be data files or directory files.

At IPL time, any disk drive can be selected to contain the system volume. The DX10 operating system maintains certain files on system-disk volumes to support DX10 internal disk-based functions. System files also can be maintained on volumes installed in secondary drives. Such disks can be used to back up system volumes or can be selected as an alternate system at IPL time.

The DX10 operating system maintains user files on disk volumes. User files can be data files or directory files. User-directory files contain file names that are unique to the user directory and can be duplicated in the system directory or other user directories without conflict.

Any given file on a disk volume is referenced by its pathname. The pathname for a file is a concatenation of the volume name, the directory levels (excluding VCATALOG) leading to the file, and the file name itself. Components of the pathname are separated by periods. The following is an example pathname:

VOLONE.AGENCY.RECORDS

If temporary files are used by a task, they are automatically deleted from the disk volume when the task terminates. Otherwise, operator commands and supervisor calls are available to delete a file (or a directory).

Logical Input/Output

The DX10 operating system supports the assignment of four-character names to each peripheral device at system-generation time. Peripherals are identified by these names when operating within the DX10 system. Files are identified by pathnames. However, a volume in which a file resides may be alternately referenced by the physical device name of the disk drive rather than volume ID. File identifiers follow the device name, volume name, or system-disk designation. Generically, the names of devices and files are called access names. Access names can be either device names or file pathnames.

Logical Unit Numbers. The DX10 operating system performs input/output to logical units instead of physical units making programs more flexible in the disposition of input and output. For example, a program may be written to accept input from logical unit number (LUNO) 82. Before the program operation, an assignment is made associating LUNO 82 with the desired access name (which may be either a device or file name). A LUNO assignment can be made by either an interactive user or a program via a supervisor call.

A LUNO assignment can apply only to the program that made the assignment, to all programs running for a given terminal, or to all programs. Task and terminal local LUNO assignments allow different programs and different users of the same program to use the same LUNO but to assign it different access names.

File-Oriented Devices. Certain devices can be declared as file-oriented at system-generation time. When a device is file-oriented, it becomes the exclusive property of the program that successfully performs an open operation to a LUNO assigned to the device. The device becomes available to other programs only after it is closed.

Record-Oriented Devices. The alternative to the file-oriented device is the record-oriented device. In this case, records can be freely interspersed among programs.

I/O Supervisor Calls. The I/O supervisor call supports all record-transfer and file-positioning operations to devices and files from a program. Utility operations, such as file creation and LUNO assignment, are also available through the I/O call. A complete list of these operations is provided in table 2.

Table 2. Device and File Operations Available through I/O Supervisor Calls

Assign LUNO
Release LUNO
Fetch Characteristics of Device or File
Verify Legality of Access Name
Open LUNO
Close LUNO
Close LUNO and Write End-of-File
Open LUNO and Rewind
Forward Space Record
Backward Space Record
Read Record
Write Record
Read Direct (used to acquire data with special formats)
Write Direct (used for special data formats)
Write End-of-File
Rewind
Unload
Rewrite the Previous Record
Create a File
Delete a File
Establish Immediate/Deferred Write Mode
Change a File Name
Write Protect/Delete Protect/Unprotect a File
Add an Alias Name for a File
Delete an Alias Name for a File
Unlock a Record
Key-Indexed File Operations
Open Extend
Modify Access Privileges

Error Control and System Log

The DX10 operating system incorporates several error-control features. When failures are detected by the DX10 system during I/O data transfers, the transfer is retried. After several attempts, a code indicating the device failure is returned to the originating program. The failure can be optionally recorded in a system log.

Error codes are returned to programs that issue illegal supervisor calls. In some cases, additional codes are returned that convey information concerning potential errors.

The 990 mapping feature protects the DX10 operating system from destruction by errant application programs. Application programs are protected from errant interaction in a similar fashion, except where they are overtly sharing a procedure. When application programs specify an illegal function (such as referencing memory outside the legal range or issuing an illegal instruction), they are abnormally terminated by the DX10 system. In these cases, an advisory message is directed to the system log. The message displays a code indicating the cause of abnormal termination.

Any task can optionally include a sequence of instructions designated as an end-action routine. If this option is selected and an abnormal termination occurs, the DX10 operating system returns to the beginning of the end-action routine in that task. The user must provide code in the end-action routine to analyze the termination code provided by the DX10 system and to take appropriate recovery steps.

The DX10 operating system also supports an optional system log. Information logged by the DX10 system includes I/O errors and task errors. I/O errors are logged for device retries as well as failures. Tasks that are abnormally terminated provide task error messages to the system log. Application programs can specify additional messages to be logged by issuing the appropriate supervisor call. A system command is provided to start the system log. At this time, output to disk files and/or output to a hard-copy device is specified. A pair of files is supported such that when one file fills (is saved to) an external archive, the other is written to. All logged messages are time stamped.

High-Level Languages and Utilities

Texas Instruments provides a choice of high-level programming languages to users of the disk-system software. The following languages run under the DX10 operating system. A comprehensive Sort/Merge utility also is available.

- COBOL for business environments
- RPG II for business environments
- DBMS 990 for efficient data management
- BASIC for simplified interactive scientific programming
- Business BASIC for simplified business-application programming
- FORTRAN for mathematical and scientific application
- Pascal for scientific and engineering programming.

COBOL

COBOL (Common Business-Oriented Language) is a high-level programming language consisting of English words and symbols. Because it is easy to read and understand, COBOL programs are largely self-documenting and generally require little explanation. For standardization, COBOL is divided into a nucleus and eleven functional processing modules. Each of these twelve modules is subdivided into several levels according to sophistication and completeness. For example, a standard version of COBOL can include level 0 indexed I/O (none), level 1 indexed I/O, or level 2 indexed I/O. Level 2 indexed I/O is the most powerful and includes level 1. Full COBOL implements the highest level of each of the twelve modules. Although COBOL language is quite extensive, it is rarely implemented in full, even on large-scale computers. Table 3 indicates the relative power of Texas Instruments version of COBOL, which significantly extends the minimum-standard COBOL. Minimum-standard COBOL as specified by the National Standard Institute (X3.23-1974) consists of:

- Nucleus level 1
- Table handling level 1
- Sequential I/O level 1.

Texas Instruments version of COBOL includes the minimum and also adds:

- Relative I/O level 1+
- Indexed I/O level 1+
- Segmentation level 1
- Library level 1
- Interprogram communication level 1
- Nonstandard support for debugging and communications.

Table 3. Relative Power of COBOL

NUCLEUS	FUNCTIONAL PROCESSING MODULES										
	TABLE HANDLING	SEQ. I/O	RELATIVE I/O		SORT/ MERGE	REPORT WRITER	SEGMEN-TATION	LIBRARY	DEBUG	INTER-PRO. COMM.	COMMUNI-CATION
2 NUC 1, 2	2 TBL 1, 2	2 SEQ 1, 2	2 REL 0, 2	2 INX 0, 2	2 SRT 0, 2	1 RPW 0, 1	2 SEG 0, 2	2 LIB 0, 2	2 DEB 0, 2	2 IPC 0, 2	2 COM 0, 2
					! SRT 0, 2				1 DEB 0, 2 *		

Minimum COBOL: 1 NUC, 1 TBL, 1 SEQ *Nonstandard support provided.
Texas Instruments version of COBOL: 1 NUC, 1 TBL, 1 SEQ, 1 REL, 1 SEG, 1 LIB, 1 IPC (shown in gray)

The COBOL compiler/interpreter runs under the DX10 disk operating system on a 990 computer. The compiler is controlled by the DX10 disk-system software. The COBOL translator is a compiler/interpreter; a COBOL program is compiled to produce an intermediate language and at execution time an interpreter executes the intermediate language. This feature provides a powerful debugging tool; COBOL can be executed in single-statement mode, allowing the user to easily detect an erroneous statement.

Features of COBOL. COBOL is a powerful minicomputer programming language with the following features.

Data Types. COBOL can process decimal numbers with up to eighteen digits (with or without a sign). Numbers can be scaled with automatic decimal-point alignment. The USAGE IS COMPUTATIONAL-1 statement allows use of 16-bit binary numbers. Character strings of fixed length are allowed, and the COBOL data structures allow a series of contiguous data fields to be treated as a single character string. The data structures support arrays of up to three dimensions.

Arithmetic Operations. The basic arithmetic operators of COBOL are ADD, SUBTRACT, MULTIPLY, and DIVIDE. Rounding of results can be specified on any arithmetic operation, and the user can specify the action to be taken if a SIZE ERROR (field overflow) should occur.

Character-String Handling. Full COBOL numeric and alphanumeric editing capabilities give great flexibility in formatting numbers for output. Character strings can be right- or left-justified, and the comparison of unequal-length strings assumes that the shorter string is extended by blanks. The INSPECT statement provides the capability for counting and/or replacing occurrences of a specified character in a character string.

Program Control. The IF statement controls execution based on the logical truth or falsity of a condition. The condition may be an expression, named condition, or logical combination of conditions linked by AND and OR. IF statements can have ELSE clauses and can be nested.

The GOTO statement transfers control to a specified location in the program or to one of several possible locations, depending on the value of a variable.

The PERFORM statement allows a portion of the program to be executed as a parameterless subroutine. The code segment specified in the PERFORM statement can be executed once, some number of times determined by a variable, or repetitively until a condition is satisfied. The VARYING option can be used to increment a variable for each repetition.

The CALL statement allows the program to call routines that have been separately compiled and linked together; this simplifies program development.

Input/Output. COBOL supports sequential files and devices in addition to random-access files accessed by record numbers or character-string keys. On random-access files, COBOL can READ records, WRITE new records, REWRITE (replace) records, and DELETE records. Random and sequential accesses can be mixed on random-access files. For example, the user might use a key to locate and read a specific record (in a key-indexed file) and then sequentially read the following records.

Sequential files can be opened with or without rewinding. A sequential file can be opened at the end of the file so subsequent WRITE operations will extend the file.

READ WITH LOCK can be used to prevent other users of a shared file from accessing the file while it is being updated.

Two files can share the same buffer area under COBOL. The user can specify the FILE STATUS variable to receive error codes from I/O operations and can specify code to be executed in the event of I/O errors.

Other Features. The COPY clause can be used to incorporate source text from a library file into the program. Commas and semicolons can be used to separate clauses and statements. Data names and paragraph names may be qualified by the structure or section they are in. Programs can be separated into code and data portions for partitioning into read-only memory (ROM) and random-access memory (RAM). Current time and date (maintained by the operating system) can be accessed via COBOL.

I/O with Video Display Terminal. COBOL includes significant extension to the ACCEPT and DISPLAY statements to facilitate interactive I/O with a video display terminal. The ACCEPT and DISPLAY statements can specify the position of the field on the screen to be referenced as a line number, column number, and field length. More than one screen field can be referenced in a single statement. In addition, the DISPLAY statement has an option to clear the entire screen before output. The ACCEPT statement has options for converting numeric inputs, initializing the screen field with asterisks, and echoing the received data in field-justified form.

Debugging Facilities. A COBOL program can include debug lines, which are indicated by a D in column 7. These lines will either be compiled as part of the program or treated as comments, depending on a compiler option.

The COBOL compiler has an option of printing a cross-reference listing that lists all of the identifiers (data-item names, index names, condition names, file names, section names, and paragraph names) used in the program, along with the line number of each appearance. Where a variable appears, the line numbers are flagged as a declaration, reference, or modification of the variable.

COBOL programs are executed under control of the COBOL debug monitor. Besides normal execution, the debug monitor allows the user to execute the program up to a specified point or to execute a single statement. The contents of data locations can be displayed between executions of portions of a program.

Elements of COBOL. The following paragraphs discuss Texas Instruments version of COBOL. This discussion is divided into parts corresponding to the modules in COBOL.

Nucleus. COBOL implements all of the level 1 nucleus with limited capabilities for ACCEPT, ADD, ALTER, DIVIDE, DISPLAY, IF, INSPECT, MOVE, MULTIPLY, PERFORM, and SUBTRACT and with full capabilities for EXIT, GO, and STOP. COBOL also implements the following features from level 2:

- Full capabilities for ALTER and IF
- Full capabilities for PERFORM
- Separators comma and semicolon
- Level 2 data names (Names need not begin with an alphabetic character.)
- Full level 2 qualifications (Names need not be unique if they can be made unique through qualification.)
- Figurative constants including ZERO, ZEROS, ZEROES, SPACE, SPACES, HIGH-VALUE, HIGH-VALUES, LOW-VALUE, LOW-VALUES, QUOTE, and QUOTES
- Full level 2 continuation capabilities (A word or literal can be broken in such a way that part of it appears on a continuation line.)
- Full level 2 condition-name conditions
- Data levels 11-49
- THRU and THROUGH as equivalent
- Level 2 relational characters >, <, and =
- Comparisons of nonnumeric literals of unequal length
- AND, OR, and NOT connectives.

In addition, COBOL has powerful extensions to the ACCEPT and DISPLAY statements to allow the user exact control of the video display terminal.

Table Handling. COBOL provides the capability to use tables of up to three dimensions and to vary the access index by an increment or decrement. The OCCURS, USAGE, and SET clauses are supported.

Sequential I/O. COBOL provides limited capabilities for OPEN, CLOSE, USE, and WRITE statements and full capabilities for the READ, REWIND, and REWRITE statements. COBOL extends level 1 CLOSE to include multiple files and WITH LOCK and also extends level 1 OPEN to include EXTEND or NO REWIND.

Relative I/O. COBOL provides level 1+ capabilities for relative I/O. Included are level 1 capabilities for the FILE-CONTROL, I-O-CONTROL, and FD entries and full level 2 capabilities for CLOSE, DELETE, READ, REWIND, REWRITE, START, USE, WRITE, DYNAMIC, READ NEXT, and START. The only level 2 features not present are the RESERVE and SAME RECORD AREA clauses.

Indexed I/O. COBOL provides level 1+ capabilities for FILE-CONTROL, I-O-CONTROL, and FD entries and full capabilities for CLOSE, DELETE, READ, REWRITE, START, USE, WRITE, DYNAMIC, READ NEXT, and START. The dynamic access feature of level 2 is supported; alternate and duplicate keys are not supported.

Segmentation. COBOL provides level 1 segmentation capabilities to allow efficient generation of overlay capabilities. Actually, since COBOL is a compiler/interpreter, the size of a program does not provide as significant a limitation as it would with a compiler language. The COBOL intermediate-language interpreter need only keep the relevant data base and the single instruction being executed in memory at a given instant. The remainder of the program can reside on disk.

Library. COBOL implements level 1 library functions including the COPY statement. The COPY statement extends the standard in that the text produced by a COPY statement can include another COPY statement. Up to five levels of nesting are allowed. Since some elements of COBOL syntax are global, the syntactic correctness of a COBOL source program cannot be determined until all COPY statements (including those in COPYed text) have been fully expanded.

Interprogram Communication. COBOL implements level 1 CALL statement and USING clause for passing parameters. CALL and EXIT PROGRAM allow control to pass back and forth between programs in a run unit.

Debugging. The COBOL compiler/interpreter supplies powerful nonstandard debugging tools. In addition to optionally compiled lines (specified by a D in column 7), the interpreter supplies two options. The user can specify COBOL execution for any block of code including statement-at-a-time execution. Or, the user can obtain a dump of any data item (including type specification).

Comparison with Level 1 and Level 2 COBOL. Texas Instruments version of COBOL differs from standard (level 1) COBOL in several ways. All signed, numeric data fields except COMP-1 have the attribute SIGN IS TRAILING SEPARATE. Thus, the S in the picture is counted in the size of the field, contrary to the usual COBOL convention. In addition, the RERUN feature and the CODE-SET feature are not provided.

Some of the more significant (level 2) COBOL features found on larger machines that are not provided on the 990 computer are discussed in the following paragraphs.

Data. Data levels do not include level 66. CORRESPONDING cannot be used, and the DEPENDING option cannot be used on an OCCURS clause. There is no figurative constant ALL literal. SIGN IS LEADING cannot be specified.

Arithmetic. No REMAINDER option is available on the DIVIDE statement. Only one destination is provided on arithmetic statements. There is no COMPUTE statement. Expressions are not allowed in IF statements. There are no sign condition tests (POSITIVE, NEGATIVE, or ZERO) and no SEARCH statement.

Character Strings. There are no STRING and UNSTRING statements.

Input and Output. The Sort/Merge feature and the report writer features are not provided. There is no label processing. Cards with over-punched signs on numeric fields cannot be read. No more than one record on an indexed file can have the same key.

RPG II

RPG II (Report Program Generator, version II) is an easy-to-use, high-level language for business data processing. Based upon a predetermined sequence that reads a record, processes the data, and outputs the results, RPG II is especially suited for applications requiring file maintenance or report generation. A series of six basic specification formats are used to input the specific actions to be taken within the RPG II sequence of execution.

Texas Instruments version of the RPG II language is closely compatible with the widely used IBM System/3 RPG II. Extensions of many of the System/3 features have been included in RPG II to provide more flexible programming. A utility program is provided with RPG II to copy System/3 or System/32 source programs or files from diskette to DS990 disk files.

The RPG II package also includes an RPG II-oriented VDT text editor and a trace feature that prints each major step occurring during execution of an RPG II program. A System/3-compatible sort/merge capability is provided by the optional Sort/Merge package, and communication of RPG II files is available through the optional DX10 3780 Emulator package.

RPG II Hardware. Texas Instruments version of RPG II is especially suited for users with rapidly growing applications. The minimum system for RPG II is the Model 4 System. The Model 4 System includes a 990 processor, 128K-byte memory, 10M-byte disk drive, 911 VDT terminal, single-bay desk enclosure, and DX10 disk-system software license. Expansion capabilities can provide large amounts of memory, multiple 50M-byte disk drives, and a variety of additional standard 990 peripherals.

Features of RPG II. The RPG II compiler has the following significant features:

- Efficient one-pass compiler
- Run-time trace that speeds the checking of the program
- Right- or left-hand sign handling
- ASCII or EBCDIC internal character set
- Capability to produce more than 500 unique diagnostic messages
- Alphabetic summary listing of all fields, labels, arrays, and tables
- Listing of all indicators specified in a program.

Comparison with System/3. RPG II is less restrictive than the System/3 in a number of areas.Table 4 lists these areas and the requirements of both the System/3 and RPG II. RPG II is compatible with the System/3 where the hardware and operating systems permit.

Several of the System/3 features are not provided on Texas Instruments version of RPG II. A Telecommunications Specifications sheet is not supported by RPG II. The P (Punch Card) function is not supported on the control card specifications.

Table 4. RPG II Enhancements

Feature	RPG II Implementation	System/3 Implementation
Number of files allowed in one source program	Unlimited*	Maximum of 20
Number of demand and/or chained files allowed in one source program	Unlimited*	Maximum of 15
Name specified for Label Exit	Any six-character name is allowed.	Name must be SUBRxx or SRyzzz.
Number of tables and arrays in one source program	Unlimited*	Maximum of 63. Only 60 may be at compilation time.
Number of spread-card specification lines	Unlimited*	Maximum of 128
Device used with spread cards	Any sequential input device is valid.	Device must be card reader.
Format of trailer fields on spread cards	Format can be unpacked decimal, packed decimal, or binary.	Format must be unpacked decimal.
Number of AND/OR lines allowed in input specifications	Unlimited*	Maximum of 20
Number of AN/OR lines in calculation specifications	Unlimited	Maximum of 7
Number of digits allowed with Y edit code	1 to 15	3 to 6
Number of characters in matching fields	Maximum of 256	Maximum of 144

*This field is limited by memory space only.

The Core Size to Execute function is not supported on the control card specifications. The Inquiry function is not supported on the control card specifications. The multifunction-card-unit (MFCU) device is not supported by the 990 system. The Nonprint Characters function is not supported on the control card specifications. The *PRINT function as a special word is not supported on the output specifications.

DBMS 990

The DBMS 990 (Data-Base Management System) is designed for minicomputer data-base applications. Specifically, this system handles applications with fast data-access requirements which need to be accessed in a logical format that can be easily equated with physical documents or records used in daily business transactions. The DBMS 990 allows the user to define and access a centralized, integrated data base using logical format without the physical data-access requirements imposed by conventional file-management software. Physical considerations such as access method, record size, blocking, and relative-field positions are resolved when the data base is initially defined. Thus, the user can concentrate fully on the logical data structures needed for interface.

Features of DBMS 990. The independence of the data definitions from the application software allows modification of the data base without impact to existing programs. It also provides a single, centralized copy of the data for all application subsystems. (Conventional file management provides fragmented and multiple copies of data held in a wide variety of files with each used by only one application.) This centralized copy results in more efficient data storage on disk, uniform processing of data requests, and centralized control of the Data-

Base Maintenance function. In addition, DBMS 990 provides optional password security for the most elementary data level; this provides control and protection of the data base from unauthorized access or tampering.

DBMS 990 User Interface. The primary user interfaces to DBMS 990 consist of the data-definition language (DDL) and data-manipulation language (DML).

DDL provides the means to completely describe the DBMS 990 data base and its associated data elements. The DDL logical data-base-definition source is compiled by the DDL compiler, and the output is stored with its associated data on disk.

DML provides the user the means to manipulate DBMS 990 data by supporting the reading and/or writing of DBMS 990 data. DBMS 990 data can be accessed by imbedding the appropriate DML syntax in a COBOL application program. COBOL is used to construct a call to DBMS 990 that specifies the function to be performed and the data element to be manipulated. DBMS 990 processes the request and returns the results to the COBOL program.

BASIC

BASIC is a powerful, easy-to-learn, high-level programming language used in a wide variety of scientific and business problem-solving applications. Texas Instruments version of BASIC is an interactive language that supports operation by multiple users. Interaction with BASIC involves operator-prompted user input of program statements and system commands. BASIC programs consisting of program statements can be easily created, entered, modified, debugged, and executed through the use of BASIC commands.

The BASIC system operates under DX10 disk-system software. The language implemented by this system is similar to Dartmouth BASIC with certain extensions to enhance its use in scientific and business applications.

The system includes the following components:

- Executive to process commands entered from a terminal
- Reentrant, multiterminal interface modules to coordinate the interaction of the executive, interpreter, and operating system
- Reentrant run-time interpreter to perform arithmetic and logical functions during the execution of a BASIC program.

BASIC and Business BASIC Systems. The 990 computer supports two BASIC systems: scientific-oriented BASIC and business-oriented Business BASIC.

The BASIC system is similar to the Dartmouth BASIC as described in *BASIC Programming* by Kemeny and Kurtz. Extensions to the Dartmouth BASIC include:

- Integer arithmetic
- Expanded string manipulation
- CALL and subprograms
- Formatted output and input.

The Business BASIC system differs from BASIC in that —

- A floating decimal replaces real arithmetic.
- Trigonometry and matrix arithmetic are not supported.
- An extended I/O package is provided including key-indexed files.

Components of BASIC. The BASIC system consists of five major components. Figure 7 shows the functional relationship of these components.

- Executive
- Compiler
- POPS interpreter
- Run-time support
- Operating-system interface.

The executive and compiler are distinguished from the rest by their storage management and their language. They are written primarily in a macrotype interpreter language called POPS, which is a single-address-formulated language. Each instruction is interpreted by a short machine-language program that branches to a routine to perform the function described by the POPS instruction. This action provides for faster program execution.

Executive. The executive controls the operations of the system. It processes the commands that are entered from the terminal. When required, it invokes the compiler and the run-time-support components. The executive performs all editing commands directly.

Compiler. The compiler is invoked by the executive, and diagnostics are listed on a line-by-line basis.

POPS. The POPS interpreter is written in assembly language. It executes the POPS and assembly subroutines required by the executive and compiler.

Run-Time Support. The run-time support performs arithmetic and logical functions that may be required

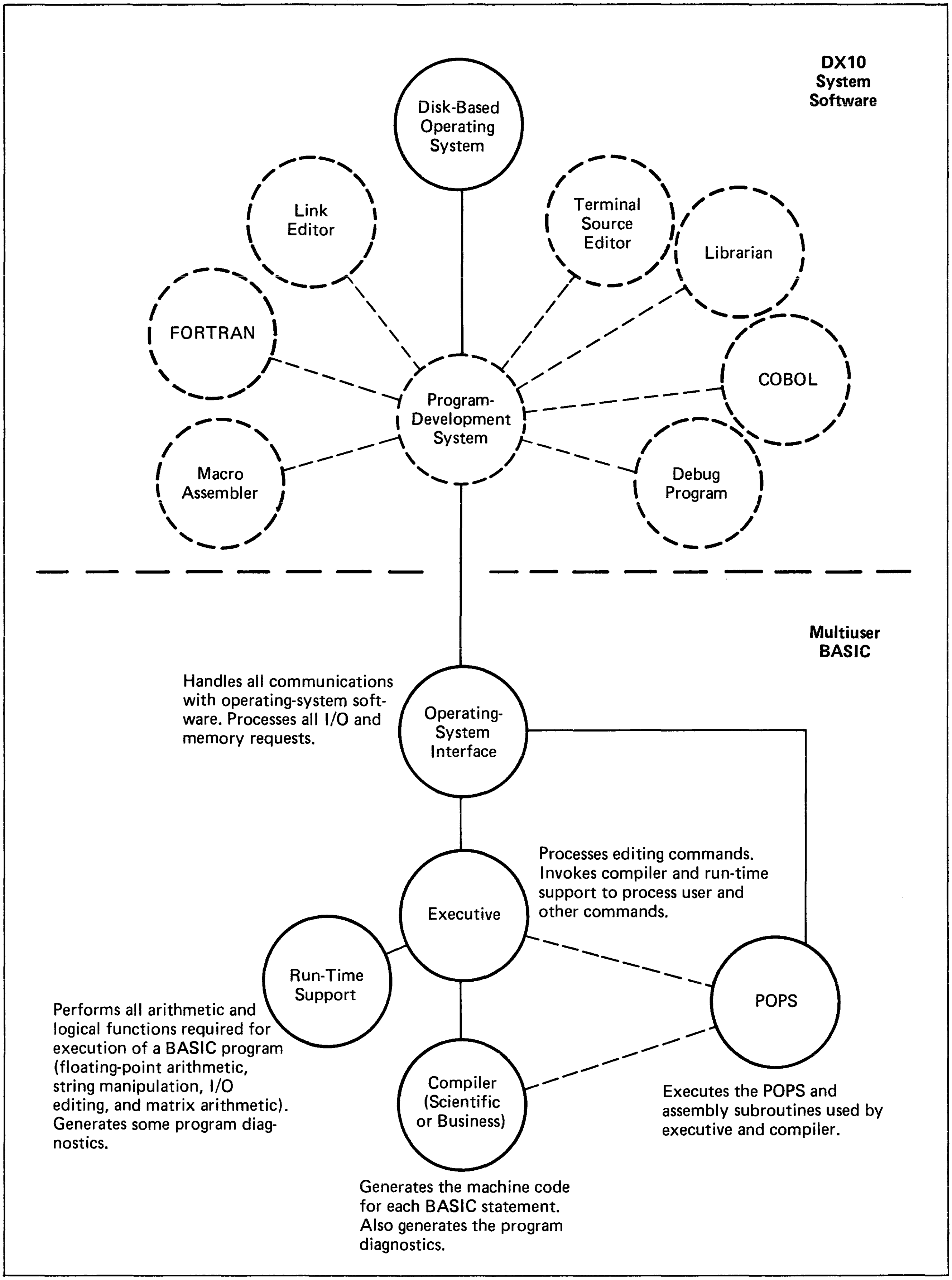


Figure 7. Multiuser BASIC Components

Table 5. BASIC and Business BASIC Statements

Statement	Purpose	Statement	Purpose
CALL	Transfer to external subroutine	ON	Computed GOTO or GOSUB
CLOSE	Close a file	OPEN	Opens a file
DATA	Defines internal data block	PRINT	Prints to output device
DEF	Defines statement	PRINTUSING	Prints to output device under control of image
DELETE	Deletes a record from a key-indexed file	READ	Reads from internal data block
DIM	Dimensions strings, vectors, and matrices	READ	READS record from file
END	Stops compilation	READUSING	Inputs values from a record on file using IMAGE
FIND	Sets record pointer	REAL	Specifies listed variables to be real type
FOR	Defines top of loop and loop parameters	REM	Comment line
GOTO	Transfers unconditionally	RESET	Sets file to first record
GOSUB	Transfers to internal subroutine	RESTORE	Reads internal READ to start of data block
IF . . . THEN	Transfers conditionally	STOP	Stops program
IMAGE	Picture and formatted print	SUB	Defines start and parameters of external subroutine
INPUT	Reads from terminal	SUBEND	Returns from external sub-routine
INTEGER	Specifies listed variable to be of integer type	WRITE	Writes record to file
LET	Evaluates expression and assigns value, all types	WRITEUSING	Writes record on file using IMAGE
NEXT	Closes loop		

Table 6. BASIC Matrix Statements

Statement	Purpose	Statement	Purpose
MAT INPUT	Matrix read from terminal	MAT INV	Matrix inverse
MAT READ	Matrix read from data block	MAT TRN	Matrix transpose
MAT PRINT	Matrix print on terminal	MAT CON	Matrix of all ones
MAT +	Matrix addition	MAT IDN	Identify matrix
MAT -	Matrix subroutine	MAT IDN	Identify matrix
MAT *	Matrix multiplication	MAT ZER	Matrix of all zeros
MAT ( )*	Scalar multiplication	MAT = (expression)	Matrix of elements all set to the expression value

during execution of a BASIC program. These functions include floating-point arithmetic, string manipulation, I/O editing, and matrix arithmetic. Referenced functions are read from a function library stored on a sequential disk file and read at run time.

Operating-System Interface. The operating-system-interface component handles all communications with the DX10 operating system. All I/O, memory requests, data handling, and other such tasks are performed by this component.

Table 8. Conditional Relations for BASIC

Operator	Meaning
EQ or =	Equal to
LT or <	Less than
LE or <= or =<	Less than or equal
GT or >	Greater than
GE or >= or =>	Greater than or equal
NE or <> or ><	Not equal

Table 7. BASIC and Business BASIC Intrinsic Functions

BASIC System* BASIC	Business BASIC	Function	Argument Type**	Function Type	Definition
		Numeric Functions			
A	A	SQR (X)	I, R	R	Positive square root
A	N	SIN (X)	I, R	R	Trigonometric sine
A	N	COS (X)	I, R	R	Trigonometric cosine
A	N	TAN (X)	I, R	R	Trigonometric tangent
A	N	ATN (X)	I, R	R	Trigonometric arctangent
A	A	LOG (X)	I, R	R	Natural logarithm
A	A	EXP (X)	I, R	R	Exponential
A	A	ABS (X)	I, R	R	Absolute value
A	A	SGN (X)	I, R	R	Algebraic sign (-1, 0, +1)
A	A	INT (X)	I, R	I	Largest (signed) integer less than X
A	A	RND (X)	I, R	R	Random value
A	N	DET		R	Determinant of last matrix inverted
A	A	CRI (X)	R	I	Convert real to integer
A	A	CIR (X)	I	R	Convert integer to real
		String Functions			
A	A	LEN (S)	S	I	Length of string
A	A	POS (S1, S2)	S	I	Position of string S2 in string S1
A	A	VAL (S)	S	I	Convert string to integer
A	A	CSR (S)	S	R	Convert string to real
A	A	ASC (S)	S	R	Value of binary representation of first character of S
A	A	SEG$ (S, I)	S	S	Substring of S from position I to end of S
A	A	SEG$ (S, 11, 12)	S, I, I	I	Substring of S from position 11 of length 12
A	A	STR$ (I)	I	S	Convert integer to string
A	A	CRS$ (R)	R	S	Convert real to string
A	A	CHR$ (I)	I	S	Character whose binary representation is I modulo 128
A	A	DAT$		S	Eight-character string giving current date: "MM/DD/YY"
A	A	CLK$		S	Eight-character string giving time of day: "HH:MM:SS"

*A = Available and N = Not available

**I = Integer, R = Real, and S = String

Features of BASIC. Interaction with BASIC involves program statements and system commands.

BASIC Statements. Tables 5 and 6 list the BASIC statements. Each statement of a BASIC program occupies one line, and each line has a distinct statement number. Statements can contain numeric or string expressions specifying constant or variable data. All variables can be dimensioned. Arithmetic operators on numeric expressions include addition (+), subtraction (-), multiplication (*), division (/), and exponentiation (**). Numeric expressions can be integer, real (BASIC), or decimal (Business BASIC). Integer values are represented internally as 16-bit two's complement numbers in the range -32767 to 32767. Real values are represented internally as two-word floating binary numbers in the range 10^{-77} to 10^{75} (16^{-63} to 16^{63}). Decimal values occupy four words in the range 10^{-99} to 10^{98}. String values are a sequence of characters stored one character per byte. Strings can be empty or can contain up to 255 characters. Numeric arrays are ordered sequences of numeric variables. They can have single or double subscripts.

Functions. Functions can be defined by the user; they may then be executed by reference to the function's name and parameters. A function's value can be numeric or string. A set of intrinsic, predefined functions (both mathematical and string) is given in table 7.

External Subroutines. Definition and reference of external subroutines that can be written in either BASIC or assembly language is provided through the use of (1) a CALL statement that transfers control to the subroutine or (2) a SUB statement that defines the name and parameters of the subroutine.

Control Statements. Definition and reference of internal subroutines by means of GOSUB and RETURN statements are provided. A computed transfer is provided by the ON statement. Conditional (IF . . . , THEN) and unconditional (GOTO) transfers are provided. The relational operators are given in table 8. A means is provided for looping through the use of the FOR and NEXT statements.

Input/Output. Formatted I/O is provided through the use of the PRINTUSING, READUSING, and WRITEUSING statements when referenced to an IMAGE statement. Format is specified through the use of the IMAGE statement. It is possible to specify I-format, F-format, or E-format. File reference I/O can be sequential or random.

Matrix Statements. Matrix statements can be created (in BASIC only) for the manipulation of matrices. Table 6 lists the BASIC matrix statements.

BASIC Commands. A set of sixteen operating commands is given in table 9. These commands provide a means for saving, modifying, deleting, and implementing programs written in BASIC.

Table 9. Operating Commands for BASIC

Command	Purpose	Abbreviation
APPEND	Appends a library program to the current program	AP
BREAK	Pause at specified line number	BR
CLEAR	Deletes all source and object programs from the current library	CL
DISPLAY	Display the current values of BASIC variables; values are printed	DI
DELETE	Deletes lines of the program	DE
GO	Resumes execution following BREAK	GO
LIST	Lists the program	LI
LOAD	Load the object program from the library	LO
NEW	Deletes all source programs from the current library	NE
OLD	Recalls a program	OL
QUIT	Terminates the BASIC session	QU
RENUMBER	Renumbers the program	RE
RUN	Runs the current program	RU
SAVE	Saves the current program in its library	SA
STEP	Forces next statement in the program to be executed	ST
COMPILE	Compiles the program from the library	CO

Commands are executed immediately upon entry. Error messages will be displayed when an improper command is attempted.

Diagnostic error messages include the line number of compiler errors. Programming errors can be easily modified by listing and correcting the incorrect statement and rerunning the program.

FORTRAN

FORTRAN IV is a standard, high-level programming language used to facilitate the solution of complex arithmetic and scientific computations. It is used in a wide variety of statistical, engineering, scientific, industrial, and business applications. Texas Instruments version of FORTRAN supports the development of programs under both the disk-based DX10 and the floppy-based TXDS systems software. FORTRAN run-time directories allow execution of compiled FORTRAN programs on 990/4 and 990/10 target hardware systems under control of their respective operating systems (or user-written operating systems) or stand-alone.

The DX10 disk-system software also allows concurrent execution of multiple FORTRAN programs. The number of concurrently executing programs is limited by system memory, CPU loading, and disk loading. Under the DX10 operating system, each FORTRAN program is limited to a 64K-byte address space; however, transparent overlay capabilities allow much larger programs to execute in a 64K-byte segment. Relative-record files are supported within FORTRAN syntax, while direct-disk I/O is supported through the use of supplied assembly-language subroutines.

TXDS FORTRAN allows the development of FORTRAN programs on 990-series floppy-disk software systems. Compiled FORTRAN programs can be linked to execute on 990 target systems under control of TXDS or TX990 system software or can operate stand-alone (with user-supplied I/O).

Features of FORTRAN. The FORTRAN compiler is an extended superset of the American National Standards Institute (ANSI) FORTRAN IV (X3.9-1966). It is implemented on the 990 computer with certain enhancements to the standard that provide increased flexibility. These enhancements include:

- Internal data-manipulation statements
- Variable names of any length
- General integer expressions in subscripts
- VDT data-handling statements
- Direct-access I/O
- Mixed-mode expressions
- Hollerith and hexadecimal constants and assignments
- Extended integers
- 16-bit fixed-point arithmetic
- Implicit variable typing
- Debug and trace options.

Additional extensions recommended by the ISA (S61.1-1975) are also included as follows:

- Program start control
- Delay of program-continuation capability
- Analog and digital I/O control
- Logical operations
- Bit-string shift capability
- Bit testing and setting capability
- Inclusion of time and date information.

DX10 FORTRAN also includes ISA procedures for file access and file contention control (S61.2-1976). These external procedures provide a means of accessing files and resolving file-access contention problems in a multiprogramming environment. These procedures include:

- Create a file
- Delete a file
- Open a file
- Close a file
- Modify access privileges
- Input/Output to unformatted direct access files.

The ISA-recommended extensions are implemented as library subroutines and do not impact the structure of FORTRAN, although some of the ISA extensions generate in-line code rather than function calls.

Compiler Options. Compiler options are specified by the letter C, D, F, M, O, R, S, or X. The letter M can be specified as a TXDS option only. Input format is free form with delimiters (commas, blanks, or no delimiters). The options can be entered in any order.

Conditional Compilation (C). The letter C entered as an option specifies that all program input records having a letter D in column 1 are to be compiled with the rest of the program statements. When the C option is not used, these records are treated as comments.

Debug-Trace Compilation (D). The letter D entered as an option specifies that a program trace listing is generated during execution. Trace messages are output on entry and exit of subroutines and under certain defined conditions.

Free Format (F). The letter F entered as an option specifies that the input program is not in the standard format of columns 1-5, column 6, and columns 7-80. The compiler scans the input program according to predefined rules.

Object-Code Listing (O). The letter O entered as an option specifies that the compiler print an object-code listing on the LIST FILE defined in the compiler prompting-message response. The O option should not be selected until the FORTRAN source is debugged.

Reentrant Object (R). The letter R entered as an option specifies that the compiler generate an object code for the entire program that uses a base-relative addressing method and can therefore be used in reentrant applications. Use of the R option should be restricted to programs that operate in unmapped processors with user-supplied operating systems. The R option is not required to have reentrant (shared) FORTRAN programs under the DX10 operating system.

Assembly Source Code (S). The letter S entered as an option specifies that the compiler generate 990 assembly-language source code instead of object code. The source can be used as input to the macro assembler to generate an object module.

Variable Cross-Reference List (X). The letter X entered as an option specifies that the compiler print a listing of the program variables on a LIST FILE defined in the compiler prompting-message response.

Memory Option (M). The letter M followed by a numeric field specifies the M option (for TXDS only). The M option is invoked to override the default memory size of 4K bytes. Available memory can be calculated for a FORTRAN compile by reducing the size of the dynamic task area by the size of the FORTRAN compiler.

Calling Assembly-Language Subroutines. FORTRAN programs employ standard calling sequences that allow the user to write FORTRAN programs that call or are called by assembly-language programs. A set of macroinstructions for the DX10 operating system allow the user to write assembly-language routines that can be called by FORTRAN programs or that can call FORTRAN subprograms.

Optimization. Although optimization is not required by the ANSI standard, Texas Instruments version of FORTRAN optimizes its output object code to produce compact, fast code. Since the 990 machine language is very powerful, many FORTRAN constructs can be expressed in machine language. FORTRAN optimizes four areas.

The compiler detects common arithmetic expressions and eliminates recalculation of these expressions. Basic external and intrinsic function calls with single arguments are also collapsed (only calculated once). For example, in the following statements the underlined parts are collapsed:

```
x = 3* SQRT (SAM+5)/(SAM+5)
       * SQRT (SAM+5)
Y = SQRT (SAM+5) * 6
```

Special cases for which a single machine instruction suffices implement that instruction instead of the usual FORTRAN expansion. For example:

INC (increment) is used for K=K+1.

Certain function references result in an in-line code compile to evaluate the function. For example, instead of generating a subroutine call, a FORTRAN code generates an object code:

```
IABS(I) generates MOV @I,A
                  ABS A
```

As far as possible, subscript expressions are preevaluated. Essentially, each linear subscript expression has a constant part and a variable part. The constant part is calculated by the compiler and does not need to be recalculated during program execution.

Data and Variable Types. FORTRAN recognizes types of data including integers, real numbers, double-precision numbers, complex numbers, logical functions, constants, fixed-point integers, and Hollerith strings. Integers and real numbers can have variable names implicitly defined. The other data types must have variable names that are explicitly declared in the program. Data can be handled as individual data elements or grouped in multidimensional arrays. Table 10 lists the data types and variable types allowed in FORTRAN and their respective range limitations.

Table 10. FORTRAN Numerical Data and Variable Types

Quantity	Range Restrictions
Simple Integer	-32768 to 32767
Extended Integer	-2147483648 to 2147483647
Real Number	10^{-78} to 10^{75}
Double Precision	10^{-78} to 10^{75}
Complex (c_1, c_2)	10^{-78} to 10^{75}
Fixed (±iQ±s)	
where:	
i = integer value	-32768 to +32767
s = scale factor	-31 to +31
Logical	True or False
Hollerith Strings	*

*Maximum number of characters is the number of bytes in the receiving field.

FORTRAN Library. The FORTRAN library includes the basic external and intrinsic functions defined in the ANSI standard as well as several extensions including the ISA-recommended extensions for executive functions, process I/O, bit manipulations, time and date, and procedures for file access and control of file contention (ISA-S61.1-1975 and S61.2-1976).

The following library routines satisfy the ISA-recommended extensions (S61.1-1975). Some of the extensions are basic external functions.

Start a Program. The START subroutine allows the user to specify a specific time delay before beginning a specified task.

Start a Program at a Specified Time. The TRNON subroutine allows the user to specify a specific time to start execution of a program.

Delay Continuation of a Program. The WAIT subroutine allows the user to specify a time delay before continuing with the execution of a program sequence.

Digital Input. The DIW subroutine allows the user to input sets of bits from the communications-register-unit (CRU) interface and store those bits in a specified array.

Latched Digital Output. The DOLW subroutine allows the user to output bits of information to the CRU. The bits can be latched in either the set or the reset condition.

Momentary Digital Output. The DOMW subroutine allows the user to output bit groups to the CRU. The bit groups consist of momentary digital output signals. Specified bits are reset after a delay.

Obtain Date. The DATE subroutine allows the user to determine the correct calendar date if the system date was initialized correctly.

Obtain Time. The TIME subroutine allows the user to determine the correct time of day if the system time was initialized correctly.

Analog Data Handling. The analog data-handling subroutines include optimal A to D and D to A conversion modules (AISQW, AIPDW, and AOW) to sample points in an order determined by the A/D hardware, to sample points in an order determined by the user, and to convert digital values in an order specified by the user.

The following library subroutines satisfy the ISA FORTRAN procedures (S61.2-1976) for file access and control of file contention:

- Create a file (CFILW)
- Delete a file (DFILW)
- Open a file (OPENW)
- Close a file (CLOSEW)
- Read a file (RDRW)
- Write a file (WRTRW)
- Modify file access (NODAPW).

Pascal

Texas Instruments version of Pascal for the 990 computer is a general-purpose language well suited for a variety of applications. Originally designed as a language for teaching a systematic concept of programming, Pascal is straightforward to learn and to use. Its readability makes the language especially useful when programs must be maintained by users other than the original author.

A popular application of Pascal is the development of systems software. The Pascal compiler is itself written in Pascal as are a number of other 990-system software modules. Pascal is also ideal for scientific or engineering applications that traditionally are written in FORTRAN or ALGOL. Its general-purpose structure is even useful for many business problems, although Pascal is seldom found in this application.

Pascal Hardware. The minimum system for Pascal is the Model 4 System. The Model 4 System includes a 990 processor, 128K-byte memory, 10M-byte disk drive, 911 VDT terminal, single-bay desk enclosure, and DX10 multiuser disk-system software license. Expansion capabilities can provide large amounts of memory, multiple 50M-byte disk drives, and a variety of additional standard 990 peripherals.

Components of Pascal. The Pascal system consists of five major components:

- Nester utility
- Configuration processor
- Pascal compiler
- Pascal run-time library
- Reverse assembler.

The nester utility generates source code indented in a standard format to improve readability. The configuration processor supports the separate compilation of nested program modules. The Pascal compiler with optimizing features produces linkable object modules. The Pascal run-time library provides operating-system interface. The reverse assembler optionally produces assembly-language source files or listings. Figure 8 shows these software components in the programming cycle.

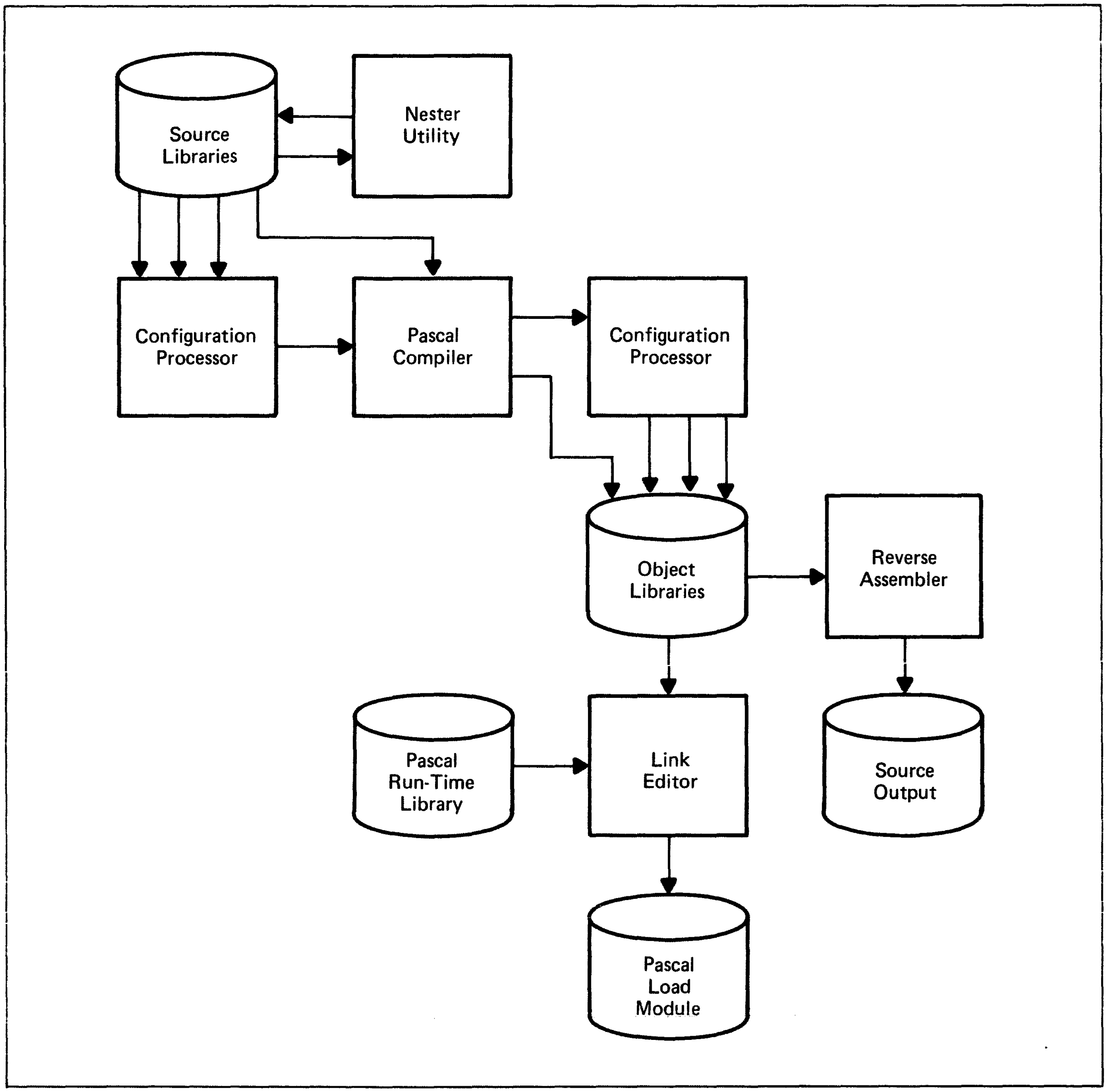


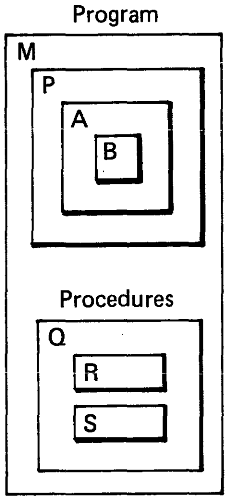
Figure 8. Pascal Programming Cycle

The object license for the Pascal software provides a complete package including all of the software components with the exception of the link editor, which is included with the license for the DX10 operating system. The DX10 system provides a powerful multiuser environment for Pascal.

Features of Pascal. Some of the more significant features of Pascal include:

- Block-structured format that directly supports structured programming concepts (figure 9)
- Stack allocation of variables for each routine
- Recursive routines
- User-defined data structures that are adaptable to data used in application
- User-defined data types and type checking
- Excellent bit-manipulation capability.

Comparison with Standard Pascal. Texas Instruments version of Pascal is closely compatible with the standard Pascal as defined by Jensen and Wirth in their *Pascal User Manual and Report.* Major modifications of, and extensions to, the



Objects Defined in Block:	Are Accessible in Blocks
M	M, P, A, B, Q, R, S
P	P, A, B
A	A, B
B	B
Q	Q, R, S
R	R
S	S

Figure 9. Pascal Block Structure

Pascal defined in that document are described in the following paragraphs.

Modifications. The precedence of Boolean operators has been altered to that used in ALGOL and FORTRAN. The CASE statement has an OTHERWISE clause, and subrange case labels are permitted. The WITH statement has a more reliable form, although no upward compatibility from Pascal has been lost. The FOR statement has a control variable that is local to that statement. The use of the GOTO statement has been restricted. PUT, GET, and the file-buffer variable have been deleted from the language; the standard functions READ and WRITE have been generalized to nontext files. Program parameters are no longer used for declaring files. User-defined functions may not have side effects.

Extensions. Data objects of type REAL can have a minimum precision. Structured jumps out of structured statements and procedures are allowed using the ESCAPE statement. Both array and set types can be "dynamic" in that their bounds can be fixed during program execution. ASSERT statements are included to aid program testing. Common variables, which have global extent and scope as dictated by ACCESS declarations, have been added to the language. External procedures and functions can be declared and communicated using a variety of linkages. FIXED and DECIMAL standard types have been added. Files can be declared as RANDOM, and standard functions that operate on random files are provided. The type structure can be explicitly overridden using type transfer. Fuller type-checking is included on procedure and function parameters.

Sort/Merge

The DX10 operating system supports an optional Sort/Merge package that can be accessed in several ways. The SCI provides commands to access Sort/Merge in batch or interactive mode. COBOL, FORTRAN, and BASIC programs can interface with Sort/Merge by using the CALL statement. Both sort and merge support record selection, reformatting on input, and summarizing on output.

Ascending key order, descending key order, or an alternate collating sequence can be specified. Any number of keys can be specified as long as their total length is less than 256 characters. The merge process supports up to five input files. The sort process allows full, address, key, and summary sorts.

Program-Development Tools

In addition to a comprehensive set of utilities that operate in conjunction with the DX10 operating system, Texas Instruments provides four major program-development tools: interactive text editor, macro assembler, link editor, and debug package.

Interactive Text Editor

The DX10 operating system provides an interactive text editor that operates from either a record-oriented interactive device or from one of the VDTs. Edit operations are initiated by edit keys or by command. When operations are initiated by commands, parameters are necessary and are prompted by the DX10 system. Edit operations allow modification, insertion, and deletion of entire records or of character strings within records.

Macro Assembler

The DX10 system assembler is the most powerful of the 990 family assemblers. In addition to accepting the standard 990 assembly-language instructions, the macro assembler is extended to include a macro facility, support for FORTRAN common segments, and conditional assembly. The macro facility provides character-string manipulation, accesses binary values in the symbol table, and supports macro-definition libraries. The relocatable object code produced by the assembler can be partitioned into segments. Common blocks, program segments, and data segments are assigned separate location counters. The link editor collects segments of the same type in contiguous memory areas. A sequence of assembly-language statements can be conditionally processed depending on the value of the assembler arithmetic or logical expression. Directives are provided to specify the amount of information in the assembly listing.

Link Editor

The DX10 system link editor accepts relocatable object code generated by the assembler or high-level-language compiler and combines the individual modules into a linked load module. References among the modules are resolved to their correct values. Common blocks, program segments, and data segments are collected, and each segment type is assigned its own contiguous area of memory.

The link editor accepts control statements that specify the use of shared procedures and the use of overlay structures. Available options include generating a load map, searching a set of object libraries to automatically resolve unresolved values, and producing a partial link that leaves external references to be resolved at a later time. Output of the link editor may be installed as an object file or may be written directly into memory image in a program file or DX10 image file.

Debug Package

The debug package is an interactive, symbolic debugging program for assembly-language tasks running under the DX10 operating system. It operates from either an interactive VDT or an interactive hard-copy terminal such as the Model 733 ASR Data Terminal. The debugger allows the display and modification of arithmetic-unit registers, workspace registers, and memory and provides the controlled execution of a task. In the run mode, a task may be halted or started; new breakpoints are set to halt the task when a breakpoint is encountered, thus allowing program debugging in near real time. In the simulate mode, a task's execution is analyzed between each instruction. Trap conditions, which interrogate the program counter or memory content, can be specified.

Communications Software

The 3780 emulator communications-software packages provide the 990 family of computers with a means of remote-job-entry (RJE) communications with an IBM 360/370 host computer or another 3780 emulator-equipped 990 computer.

Communications consists of exchanging data files between master and slave stations over leased point-to-point or switched telephone lines (figure 10).

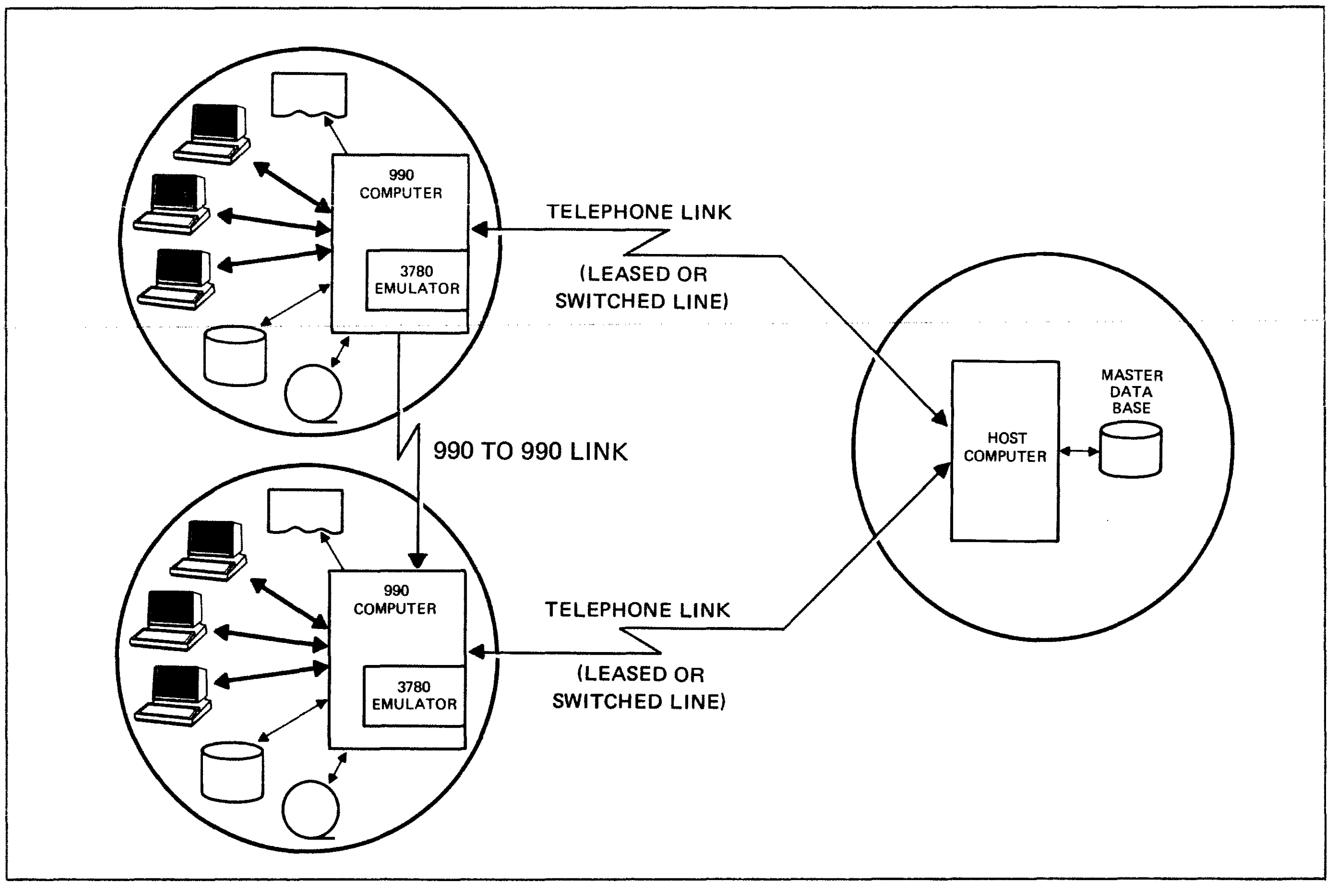


Figure 10. Typical Application of 3780 Emulator in Distributed-Processing Environment

Using the 3780 emulators, 990 computer systems can serve as satellite and/or central stations in distributed-processing networks or can be used to handle remote-job or batch-data entry for processing by a host. Remote stations can be dialed manually or automatically with an optional auto-call unit and internal modem. They can also be operated in an unattended mode as a called station in a distributed network.

Texas Instruments 3780 emulator communications software emulates the operation of the IBM 3780 Data Communications Terminal. However, unlike the IBM 3780, the source and destination of the transferred files are not restricted to the card reader/punch and line printer. Any file, input device, or output device available to the user's system can be used for input or output.

Two 3780 emulator software packages are available: the DX10 3780 Emulator for operation under the DX10 system software on DS990 systems and the TX990 3780 Emulator for operation under the TXDS/TX990 software on FS990 systems.

Features of 3780 Emulator

Line communications follow the same binary-synchronous-communications (BSC) procedure for point-to-point exchanges used by IBM 3780 communications terminals. The 3780 emulator supports many of the features of the IBM 3780 and provides several additional features.

Data Transmission. Transmission is performed in half-duplex mode at line speeds of up to 9600 bits per second (bps). The transmission code is EBCDIC, although the conversion of ASCII data to EBCDIC for transmission and back to ASCII upon reception is transparent to the user. The BSC protocol provides transmission-error checking and basic control of the data link. Retransmission is automatically requested if errors are detected.

Switched Network Control. Unattended operation capability is provided. A 3780 emulator will automatically answer an incoming call on a dial-up communication line and disconnect the line connection after transmission.

Host- or Slave-Station Operation. A primary (host) transmitting station can select an output device on a secondary (slave) receiving station, create a file, and send data to that device or file.

Space Compression/Expansion. Consecutive space characters can be suppressed from nontransparent data transmission and replaced by a control character and a space count; this improves throughput space. Characters are regenerated upon reception.

EBCDIC Transparency. In transparent transmissions, EBCDIC control characters are accepted as data without performing their control function. However, binary files cannot be transmitted or received.

Line-Printer Emulation. Vertical-forms control (VFC) permits vertical-carriage-control formatting of printer data by a transmitting station including single space, double space, triple space, space suppression, top of form, and vertical tab. The remaining twelve-channel VFC is emulated by fixed-line spacing. Horizontal format control provides a horizontal-tab function allowing deletion of spaces in formatted lines; this increases throughput.

Automatic Line Turnaround. The 3780 emulator automatically switches to "receive" after transmitting; this minimizes line time.

Multirecord Transmission. Multiple record blocks can be transmitted; this improves throughput by avoiding line turnaround at the end of each record.

Device Independence. Any file or system device can be specified to transmit or receive data. The pathname can be specified in the data stream. The 3780 emulator performs the emulation required to transmit any user file as if the file originated from the IBM 3780 card reader.

Auto-Call Operation. Remote stations can be dialed automatically with an optional Texas Instruments auto-call unit and internal modem.

Variable-Length Records. Up to 256 characters per record can be transmitted.

Comparison with IBM 3780

Features supported by the 3780 emulator that are not supported by the IBM 3780 include:

- Device independence
- Auto-call operation
- Variable-length records

Several IBM 3780 features are not provided by Texas Instruments version of the 3780 emulator. Multipoint operation is not supported, and there is no physical card punch. Data directed to the card punch can be redirected to any system output device or file.

Control

File exchanges using a 3780 emulator are referred to as communication sessions. Communication sessions are opened by initiating the 3780 emulator; they are controlled through interactive operator-interface commands/system exchanges or through execution of a predefined set of commands stored in a command file. Each transfer can include up to three files or text character strings. Any number of these transfers can be generated.

A user configuration table of predefined (default) transmission parameters is supplied as part of the 3780 emulator. The user can change any configuration parameter during system installation or as the first step in initiating a communication session.

During data transmission, the 3780 emulator starts the transfer of data from the input device/file to an input buffer. From the input buffer, the data is moved to one of two 512-byte communications-blocking buffers. When the communications buffers are filled, the data is queued for transfer to the communication line. Output-data transfer continues until an end-of-file is reached, an unrecoverable error occurs, or a processor interrupt is received from the host computer.

During communication sessions, the 3780 emulator always looks for incoming data when not transmitting. Data is received from the line into the alternating communications buffers and then is transferred to intermediate buffers for system output.

Data can be transmitted to and received from the RJE portion of IBM 360/370 host systems. The IBM host RJE terminal-control software supported includes HASP, ASP, and JES 3. The host can also be a 990 computer running under the DX10 or TXDS system software.

System Requirements

The DX10 3780 Emulator package operates under the DX10 system software (Release 3.1 or later) with the DS990 hardware configuration described in table 11.

The TX990 3780 Emulator package operates under the TXDS or TX990 software (Release 2.2 or later) with the FS990 hardware configuration described in table 12.

Components of 3780 Emulator

The TX990 3780 Emulator consists of a single loadable task containing the communications device service routine, emulator functions, operator interface, and all tables and buffers. The TX990 3780 Emulator software license includes an executable object module (used if no configuration parameter changes are required), concatenated object modules for all but the configuration module, and source data for the user-defined configuration-table

Table 11. Hardware Configuration for DX10 3780 Emulator

Computer system:	990/10 Minicomputer with a minimum of 128K bytes of memory	911 Video Display Terminal system console		
Communications interface hardware including one of the following:	Communications Interface Module, Synchronous Modem*, and optional Auto-Call Unit	Bell Data Set Interface for non-TI modems** (Bell 201A, B, C, or 208A, B compatible)		
Input device including one or more of the following:	DS10, DS25, DS31, or DS50 Disk Files	804 Card Reader	733 ASR Cassette Tape	979A Magnetic Tape
Output device including one or more of the following:	DS10, DS25, DS31, or DS50 Disk Files	810, 2230, or 2260 Printer	733 ASR Cassette Tape	979A Magnetic Tape

*Customer-furnished DAA is required (Bell Data Coupler CBS for automatic terminals with RS-232-C interface or equivalent).

**Customer-furnished modem required options include (1) internal timing, (2) without new sync, (3) without ACU, and (4) EIA option.

Table 12. Hardware Configuration for TX990 3780 Emulator

Computer system:	990/4 or 990/10 Computer with a minimum of 48K bytes of memory	911 Video Display Terminal or 733 ASR/743 KSR system console	
Communications interface hardware including one of the following:	Communications Interface Module, Synchronous Modem*, and optional Auto-Call Unit	Bell Data Set Interface for non-TI modems** (Bell 201A, B, C, or 208A, B compatible	
Input device including one or more of the following:	FD800 Floppy-Disk Files	804 Card Reader	733 ASR Cassette Tape
Output device including one or more of the following:	FD800 Floppy-Disk Files	810, 2230, or 2260 Printer	733 ASR Cassette Tape

*Customer-furnished DAA is required (Bell Data Coupler CBS for automatic terminals with RS-232-C interface or equivalent).

**Customer-furnished modem required options include (1) internal timing, (2) without new sync, (3) without ACU, and (4) EIA option.

module. No operating system generation is required for installation of this emulator. In this environment, the communications interface cannot operate at a shared interrupt level with any other system device.

The DX10 3780 Emulator consists of a system-resident, device service routine (DSR) including DSR tables and buffers and an overlaid loadable task including emulator functions, operator interface, and the remaining tables and buffers. The DX10 3780 Emulator software license includes object modules and source modules for configuration of the DSR and tasks. An operating system generation is necessary for installation of this emulator.

Customer Information

The customer must provide Texas Instruments with the following information before purchase of the 3780 emulator communications software. This information will ensure that integration is practical.

- Host computer to be interfaced.
- Host operating system and 3780 telecommunications software (HASP, ASP, JES, etc.).
- Modem type and telecommunications-line network type used. Customer is responsible for installing telecommunications lines and any other non-Texas Instruments equipment (modems, DAAs, etc.) before communications-software installation and for ensuring its proper operation.
- Any other information that might prove helpful for installation.

DS990 Hardware

DS990 Hardware

The DS990 hardware supports the features of the DX10 operating system. All DS990 systems are built around the Model 990/10 Minicomputer with a minimum of 128K bytes of error-checking-and-correcting (ECC) memory. The 128K-byte processor supports the DX10 operating system in a dual-disk drive system with one or more Model 911 Video Display Terminals (VDTs). The versatility of the 990 processor input/output (I/O) structure is extremely well-suited to the custom computer-system concept of the DS990 system. Figure 11 shows the standard and optional DS990 hardware connected to the TILINE bus and communications-register-unit (CRU) interface of the 990 processor. The high-speed, bidirectional TILINE bus interfaces the processor with memory and high-speed peripherals. The standard disk drives (Model DS10, Model DS25, and Model DS50 Disk Systems) hold a complete copy of the DX10 operating system, selected application programs, and related data bases. An optional disk drive on the TILINE (Model DS31 Disk System) and floppy-disk drive on the CRU (Model FD800 Floppy-Disk System) allow the user to partition the application software or data base to fit the requirements of a particular business. The optional Model 979A Magnetic-Tape Transport allows the user to maintain application data on nine-track, IBM-compatible magnetic tape.

While the TILINE bus can transfer large blocks of data with one command from the 990 processor, the CRU interface handles data ranging in size from a single bit up to a word with one command. The standard and optional 911 VDTs and the optional data terminals on the CRU (Model 733 ASR and Model 743 KSR Data Terminals) are the operator's interactive interface with the DX10 operating system. The optional printers (Model 810 Printer and Model 2230 and Model 2260 Line Printers) and the Model 804 Card Reader provide the user with a complete selection of I/O media for the DS990 system. The optional interface modules provide a facility for controlling and monitoring special-purpose equipment that may be connected to custom DS990 systems.

The optional I/O expansion chassis, shown in figure 11, extends the 990 main chassis in the larger DS990 systems to accommodate additionally selected options.

990 Processor

The coordinated 990 family of computers offers the prospective user a variety of choices that ranges from the TMS 9900 microprocessor to the DS990 integrated hardware/software systems. The common architecture and upward-compatible instruction set permits programs to be transported from smaller members of the family to the Model 990/10 Minicomputer.

Features of the 990/10 minicomputer, as supplied with a DS990 system, include:

- Advanced memory-to-memory architecture with programmable workspaces for fast, efficient processing of multiple interactive tasks
- Large, versatile instruction set with 16-bit-word-oriented and 8-bit-byte-oriented instructions and with hardware multiply and divide
- Sixteen vectored interrupt levels
- TILINE, parallel, asynchronous bus for high-speed communication among the central processor unit (CPU), memory, and I/O controllers
- Direct access up to 64K 8-bit memory bytes
- Program-controlled hardware mapping for memory-addressing capability up to 2048K bytes with isolation between multiple tasks
- Extended operation (XOP) codes for hardware or software implementation of special functions
- Privileged-mode instructions that protect the operating system from errant application programs
- Line-synchronized, real-time clock input
- Command-driven CRU for medium- and low-speed bit, byte, and word I/O operations
- Permanent read-only-memory (ROM) loader
- Thirteen-slot chassis that provides mounting space, cooling air, dc power, and signal interconnections for 990 CPU boards, memory boards, peripheral-device controllers, and other full-sized or half-slot 990 interface boards. The chassis includes a programmer panel for operator interface with the 990 CPU via a firmware utility program.

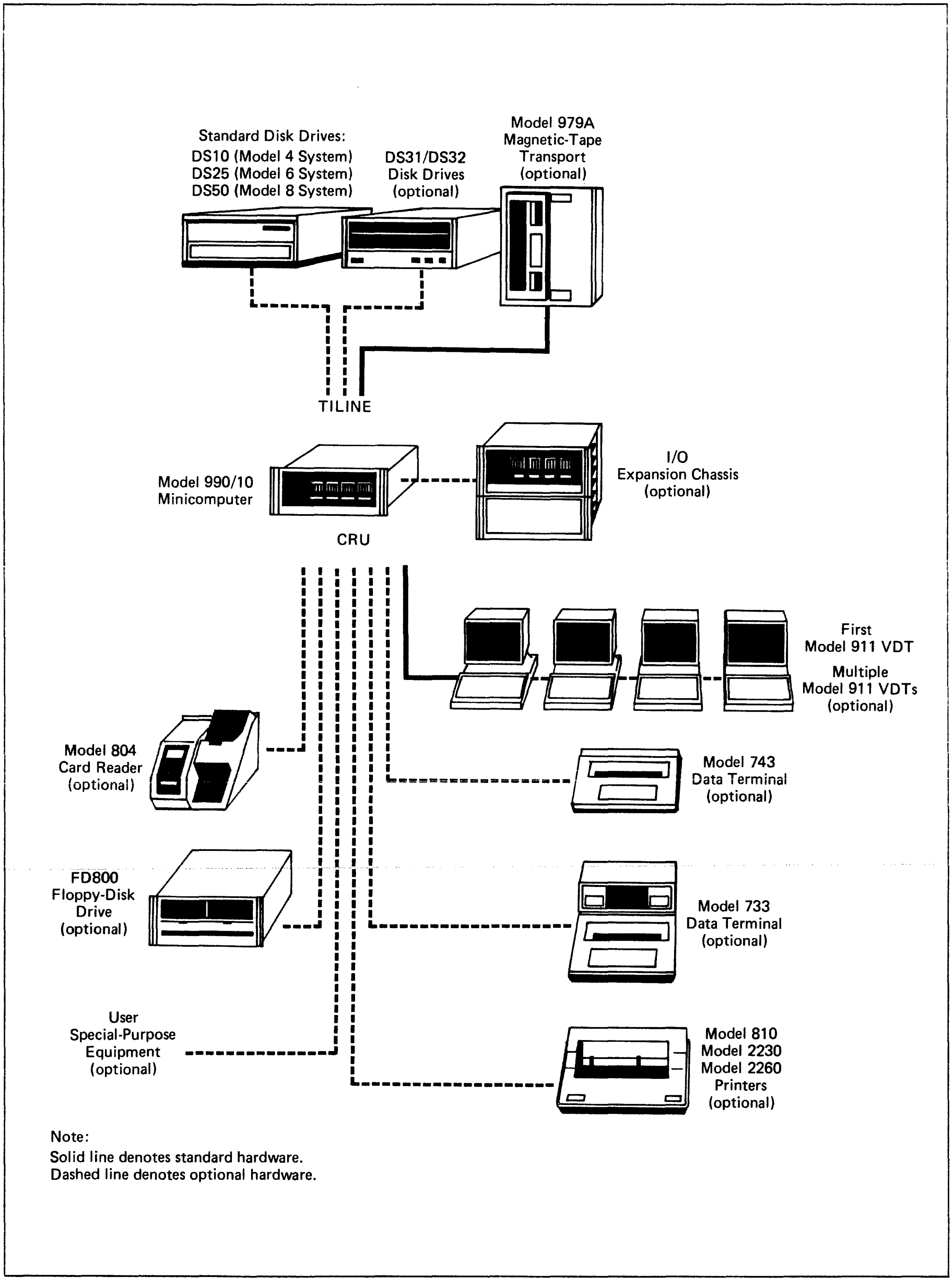


Figure 11. DS990 Standard and Optional Hardware

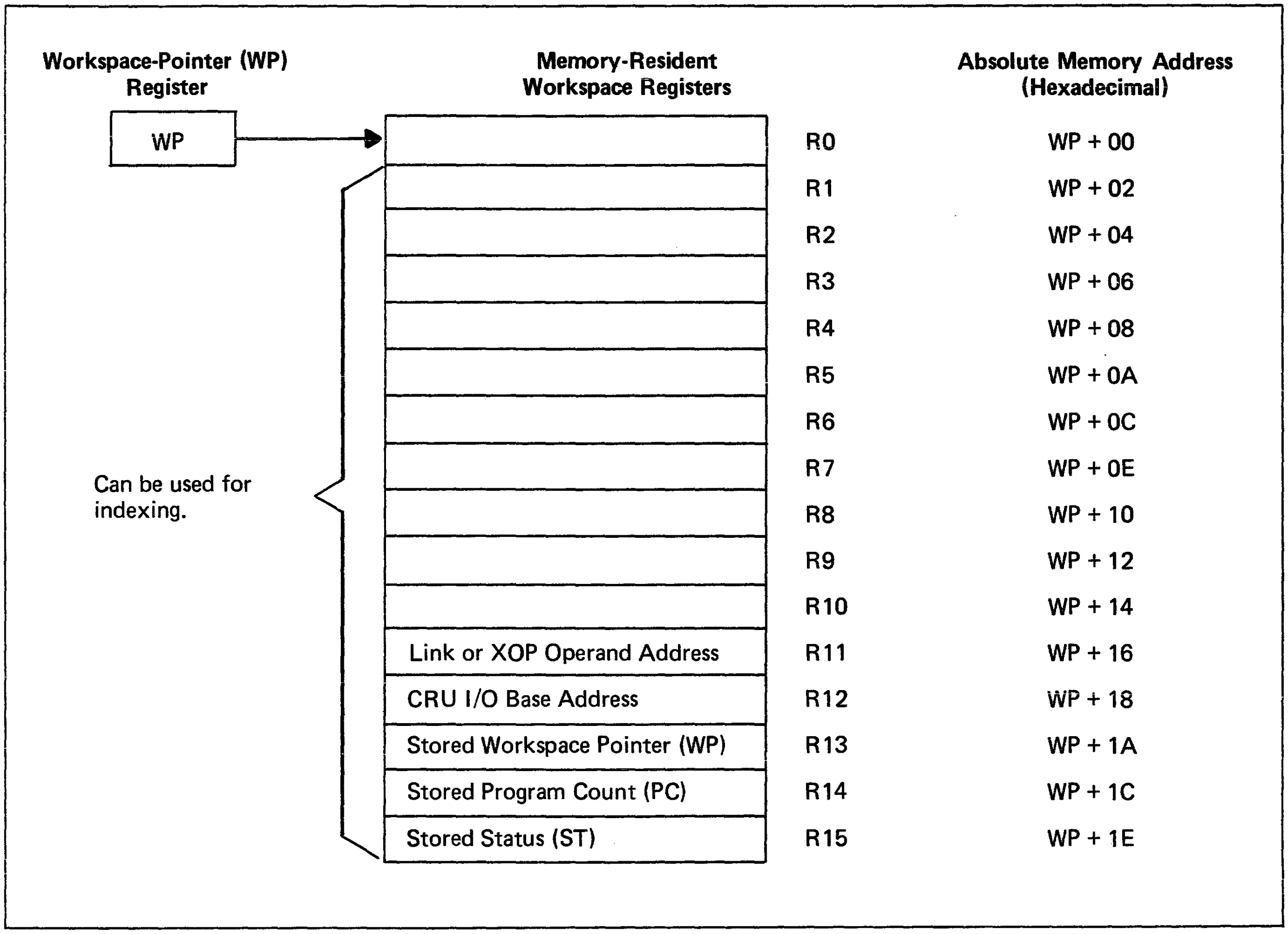


Figure 12. Workspace Pointer and Registers

990 Architecture

The 990 family of computers uses an advanced memory-to-memory architecture, which offers the convenience and speed of a register-to-register architecture without the instruction-overhead penalties. The 990 processor uses 16-word blocks of memory as workspaces (figure 12). Each memory location in a workspace is assigned a workspace-register number and is treated like a 16-bit, general-purpose hardware register. Any 16-word block in general memory can be used as a workspace. A workspace pointer, stored in a hardware register, locates the currently active workspace. The workspace concept makes it virtually impossible to run out of registers; any number of workspaces can be defined in memory and reached by modifying the workspace pointer. A dedicated workspace usually is assigned to each task or subroutine in the program.

A context switch occurs when the program suspends execution of a task, stores the intermediate results, executes another task, and usually returns to the initial task. Interrupt processing, subroutine calling, and multiple-task interleaving are typical examples of context switches. The workspace organization of the 990 processor greatly reduces the instruction and memory-access overhead associated with a context switch. For example, if the 990 processor used fifteen hardware registers, fifteen store cycles would be required to save all the possible temporary operands before the context change. By exchanging the status, program count, and workspace pointers, the 990 processor can perform a complete context switch in three store cycles and two fetch cycles without losing operands or data from either task. After the context switch, the hardware workspace-pointer register contains the memory address of a new workspace; the previous program count, status, and workspace pointer are stored in registers 13-15 of the new workspace.

The simplicity of context switching in the 990 processor and the data-handling capability of the high-speed TILINE data bus make it possible to efficiently support a large, versatile, multitasking operating system such as the DX10 system.

Interrupt Structure

The 990 processor has sixteen interrupt levels, numbered 0-15. Interrupt level 0 has the highest priority, and level 15 has the lowest priority. Interrupt levels 0-2 are reserved for the power-restored, power-failure-imminent, and error interrupts. The remaining interrupt levels are available for assignment to peripheral devices and controllers. The DX10 operating system assumes standard (default) interrupt assignments to the various I/O controllers, but these assignments can be altered when a DX10 system is generated and installed.

Enabling interrupt levels is dynamically managed by the DX10 operating system. An interrupt mask in the 990 status register identifies the levels that are enabled. The level of the highest priority pending interrupt is continually compared to the interrupt mask. If the pending interrupt has a priority equal to or greater than the lowest priority enabled, the interrupt is processed. The interrupt mask is updated as part of the processing; so interrupts can be nested in order of priority.

990 Address Space

The 990 CPU uses a 16-bit address word in instructions that require memory read or write operations. The least significant bit (LSB) is used within the CPU to select the upper or lower byte of a 16-bit word. The remaining 15 address bits are available for on-board ROM-loader and expansion-memory addresses. The address space defined by this direct addressing scheme consists of up to 65,536 8-bit bytes or 32,768 16-bit words. The word addresses are located on even-byte boundaries.

The TILINE, which links the 990 CPU, memory, and high-speed peripheral-device controllers, uses a 20-bit address to describe an address space of 1024K 16-bit words. Memory mapping extends the CPU-addressing capability to cover the entire TILINE address space.

Three sets of hardware mapping registers (MAP0, MAP1, and MAP2) convert CPU addresses to TILINE addresses. Each set of mapping registers consists of three 16-bit bias registers and three 11-bit limit registers for a total of eighteen registers. The three limit registers divide a 32K-word address space

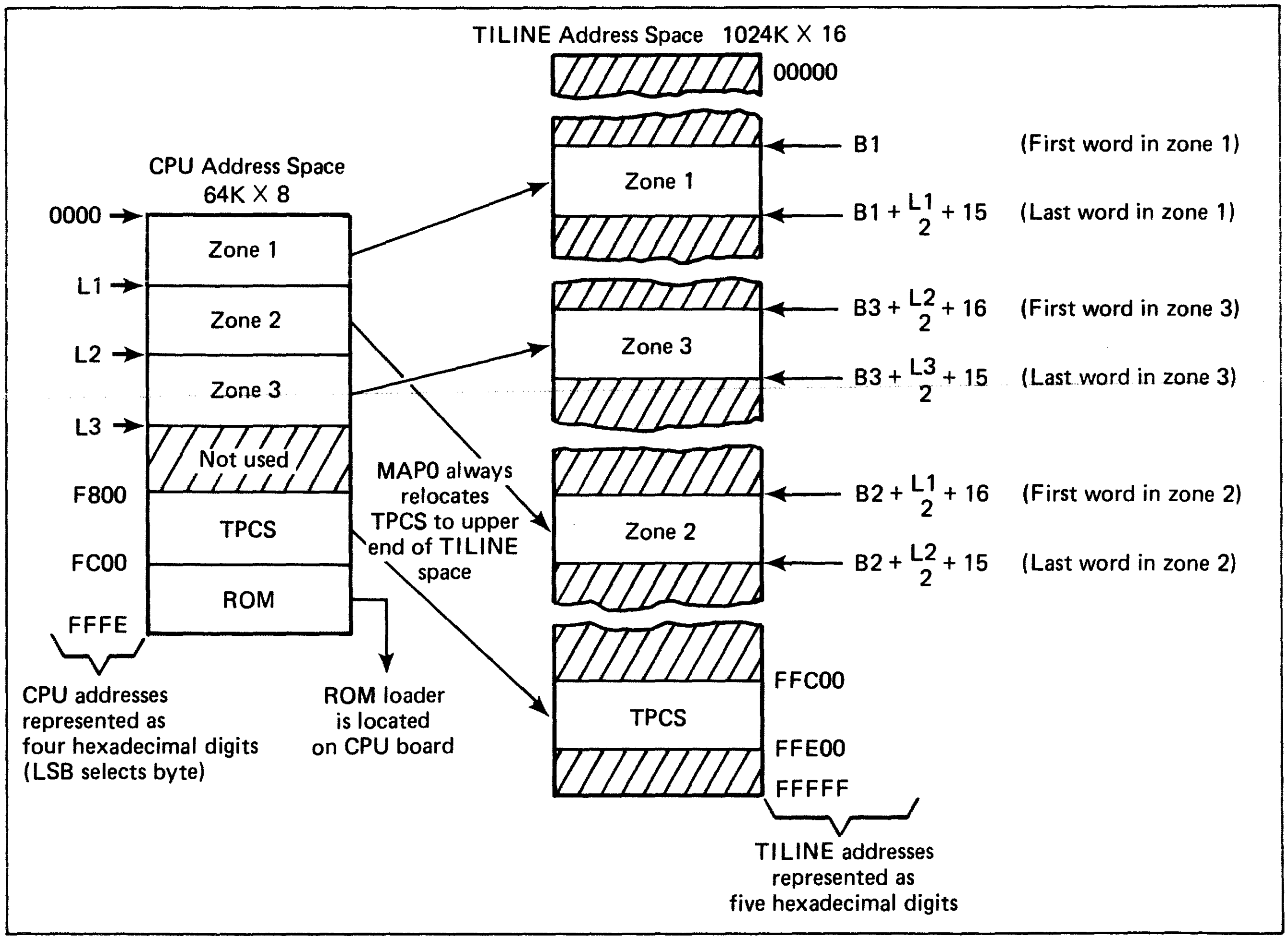


Figure 13. 990 CPU/TILINE Address Space with Map Option

into three zones on 16-word boundaries. The corresponding bias value is extended to 20 bits (by appending trailing zeros) and maps the zone into the 20-bit TILINE address space (figure 13).

The three mapping-register sets (each with three limit registers and three bias registers) are reserved by system software for specific functions. MAP0 is used by the DX10 operating system. Interrupts and XOP codes trap via MAP0. This map also reaches the TILINE peripheral control space (TPCS). MAP1 is the normal user-task map, and MAP2 is used for long-distance instructions that provide the user with access to additional memory space.

The memory-management functions of the DX10 operating system handle memory allocation and memory mapping automatically; so mapping is transparent to most users.

Instruction Set and Addressing Modes

The instruction set of the 990 computer family readily lends itself to efficient processing through simple and effective programming (Appendix B). The 990 instructions are divided into nine categories: arithmetic, logical, shift, compare, branch, load and move, control and CRU I/O, long distance (990 computer with mapping), and extended operations.

Many of the 990 instructions allow a choice of one of five addressing modes for one or both of the operands. The general addressing modes are (1) workspace-register addressing, (2) workspace-register indirect addressing, (3) workspace-register auto-increment addressing, (4) symbolic memory addressing, and (5) indexed memory addressing.

Workspace-register addressing specifies the workspace register that contains the operand. A workspace-register address is written as a term with a value between zero and fifteen and refers to the sixteen workspace registers identified by the current workspace pointer.

Workspace-register indirect addressing specifies a workspace register that contains the address of the operand. An indirect workspace register address is written as a term preceded by an asterisk. This form of addressing allows sequential processing of lists or arrays without a separate incrementing instruction.

Workspace-register indirect auto-increment addressing specifies a workspace register that contains the address of the operand. After the address is obtained from the workspace register, the workspace register is incremented. The workspace-register increment is one for byte operations and two for word operations.

Symbolic memory addressing specifies the memory address that contains the operand. An absolute address or a symbolic name contains the operand. An absolute address or a symbolic name can be used as the object of the symbolic address.

Indexed memory addressing specifies a memory address that contains the desired operand. The memory address is the sum of the contents of a workspace (index) register and a symbolic address.

Some 990 instructions imply an addressing mode other than these five general addressing modes. For example, all of the immediate instructions use the next word following the instruction as an immediate operand. Immediate instructions that require two operands derive one operand from the next word and one from a specified workspace register. The jump instructions use a program-counter relative addressing mode. The jump instruction includes a signed displacement value, and the resulting range of jump is within -128 to +127 words of the current program-counter value. The CRU single-bit I/O instructions use a displacement value relative to the contents of current workspace register R12. By convention, workspace register R12 contains a CRU base address, and individual CRU bits are located relative to that base address.

Extended Operation Instructions. Extended operation (XOP) instructions allow the user to expand the basic 990 instruction set by up to sixteen additional instructions. The 990 computer allows this expansion to be an additional logic board or software subroutines. A bit in the 990 status register steers XOP execution to the hardware processor (if present) or to a software XOP vector.

Software XOP vectors are similar to interrupt vectors, and software XOP processing is similar to interrupt service. The thirty-two memory-word locations immediately above the interrupt vectors are reserved for XOP vectors. Each vector consists of a workspace pointer and a program count, which are used for a context switch to one of the XOP subroutines. The initial workspace pointer, program count, and status are loaded into the new workspace; so the initial context can be recovered at the end of the XOP subroutine. The effective address of the XOP operand also is loaded into the new workspace.

The DX10 operating system reserves XOP 15 for supervisor calls, leaving XOP 0 - XOP 14 available for user definition. The operating-system generation program makes specific provision for user definition of XOP routines. These routines are incorporated when a DX10 operating system is generated and installed. The XOP instructions are global; that is, they can be used by any program that operates under DX10 control.

Macroinstructions. The macro-assembler program within the DX10 operating system allows the user to define macroinstructions that activate assembly-language subroutines. If a particular application program requires repetitive use of an instruction sequence, the sequence can be incorporated into a macroinstruction definition and activated by reference to the macroinstruction.

Macroinstructions differ from XOPs in that macroinstructions are defined at assembly time and may be local to a given application program. XOPs must be defined at DX10 system-generation (sysgen) time and can apply to all application programs.

990 CPU On-Board Loader

A loader is a short, simple program that contains the instructions required to read other programs from an input device to memory. The basic requirements for a loader are (1) that it not be altered by power interruption and (2) that it be either permanently resident in main memory or readily loaded by hardware action.

The 990 CPU is supplied with a 1024-byte loader program, which is permanently "burned" into two ROM devices. The loader is activated by the front-panel LOAD switch to initiate a bootstrap read of the DX10 operating system from a disk-storage system. The loader also can load the DX10 operating system from a 733 ASR terminal cassette, an 804 card reader, or a 979A magnetic-tape transport. The ROM loader includes a programmer-panel management utility, which controls the front-panel indicators and accepts panel-switch commands.

Purchasers of DS990 hardware-only systems may wish to use a different loader or to "burn" their own loader programs. Texas Instruments offers a variety of loaders as standard price-list items and offers a device kit, which consists of two 256-word by 8-bit programmable read-only memories. The maximum limit of two device kits (1024 bytes) is determined by socket space on the logic board.

Programmer Panel

The programmer panel is a CRU device, which contains logic, switches, and indicators that allow user control and observation of computer operations. Functions of the indicator lights and switches are defined by panel firmware included in the standard disk-loader ROM. The built-in programmer-panel logic includes a switch debounce timer. The programmer-panel logic is connected to the 990 CPU by a 20-conductor flat cable.

The three-position, key-operated POWER OFF, LOCK, UNLOCK switch at the left side of the panel controls the computer system's ac power and prevents the inadvertent halting of the computer. The key can be removed from the lock when the switch is in either the POWER OFF or LOCK position. When the switch is in the POWER OFF position and the key is removed from the lock, no unauthorized turn-on of the computer can occur. When the switch is in the LOCK position, the other programmer-panel switches are locked out and cannot start, stop, or otherwise affect computer operation. The programmer panel is operable, and manual control from the panel can be exerted with the key lock in the UNLOCK position.

990 Memories

DS990 systems are supplied with a minimum of 128K bytes of error-checking-and-correcting (ECC) 16K semiconductor memory. This random-access read/write memory (RAM) is required for operating-system storage, application-program and data storage, workspace-register I/O buffering, and execution scratch-pad areas. The DX10 operating system rolls programs and overlays between the disk and memory to manage the available memory space for best use and minimum execution time. Additional memory may be required in systems with multiple VDTs. However, the 990 system has a 2048K-byte maximum memory-addressing capability.

The devices used for 990 memory expansion are the 16K dynamic RAMs. The term RAM refers to read/write memory in which access time to any two storage locations is the same. A dynamic RAM requires periodic refreshing of the stored data; this is performed automatically by the memory-board logic. The refresh circuits steal one memory cycle each 15.5 microseconds. The refresh operation is transparent to the user.

The dynamic-RAM devices used in the 990 memory are the most reliable memory components available today. However, good system design requires some form of error correction for large memories. The DS990 system uses ECC memory for greater reliability. A unique 6-bit Hamming code is attached to each 16-bit word stored in memory. This code permits automatic (hardware) detection and correction of all 1-bit errors and detection of all 2-bit errors.

Automatic correction of 1-bit errors allows the system to operate normally despite a failure that would otherwise require immediate service. Repair of failed memory modules usually can be deferred to regular maintenance periods, thus reducing downtime.

The basic ECC memory consists of an ECC controller board with the necessary TILINE interface logic, refresh control, error-detection-and-correction logic, and 65,536 bytes of memory. The 65,536 bytes are organized into 32,768 22-bit words. Sixteen-bit words are read from or written to memory over the TILINE; byte selection is performed in the 990 CPU. The 6-bit Hamming code is internal to the ECC memory boards.

The ECC controller board can control an add-on expansion board in the adjacent slot via a top-edge interconnection. Switches on the controller board set the starting memory address anywhere in the TILINE address space on 16K-byte boundaries. Correctable and noncorrectable error indicators on each ECC controller board give a visual indication for servicing ease. Noncorrectable (2-bit) errors also send an interrupt to the 990 CPU.

The 128K-byte memory, which is standard for DS990 systems, consists of one 64K-byte ECC controller board and one 64K-byte ECC expansion board. Additional space is available within the chassis for future memory expansion. The amount of semiconductor memory needed in a system is determined by many factors. These factors include the number of simultaneously operated interactive terminals, the size and organization of the application programs, and the high-level languages selected. Refer to Appendixes C and D for additional information on memory-size selection.

990 TILINE

The TILINE is a multiuser, asynchronous, parallel data bus. This bus, which is capable of more than three million 16-bit word transfers per second, links the 990 CPU, memories, and high-speed peripheral controllers.

Bus communications are based on a master-slave relationship between TILINE devices. Slave devices, such as 990 memory, respond to a 20-bit word address and control signals by supplying or accepting a 16-bit word. Each word transfer is accompanied by an exchange of "handshaking" control signals.

Master devices, such as the 990 CPU, compete for access to the bus. A reservation scheme allows bus-access operations to partially overlap, reducing the overhead time to transfer control between masters. Conflicts between masters attempting to reserve the bus are resolved by a positional priority scheme based on chassis-slot location.

The high-speed "smart" peripheral controllers for the DS10, DS31, DS25, or DS50 hard disks and the 979A magnetic-tape system are TILINE devices that act as masters and as slaves at different times. They act as slaves when they accept a set of eight to sixteen setup and command parameters from the 990 CPU, and they act as masters when they perform the specified-record read or write operation between 990 memory and disk or tape drive.

Some applications of the 990 processor require more circuit boards than can physically fit in the standard thirteen-slot chassis. Both of the 990 I/O systems (the CRU and the TILINE) can be extended into additional chassis modules using the appropriate kit.

Communications Register Unit

The communications register unit (CRU) is the general-purpose I/O system for the 990 processor. This versatile, command-driven, synchronous I/O system has proved to be the most flexible system yet developed for low- and intermediate-speed applications. The CRU provides communications between the CPU and device controller for VDTs, data terminals, card readers, printers, the programmer panel, and other low- and medium-speed devices.

The CRU has two 4096-bit registers: one for input and one for output. Each bit is individually addressable. Direct addressing of CPU input and output bits is the key to the versatility of the CRU system. A single 990 instruction can read from (or write to) an individual bit or a group of up to 16 bits.

The 990 CPU performs partial decoding of the 12-bit CRU address to produce a group of module-select signals, which are hard-wired to the available half-slot locations in the chassis. This simplifies address decoding on the individual full-sized logic boards and imposes a positional addressing scheme. The CRU interface can be extended to additional chassis if system requirements exceed the available space for logic boards.

TILINE Peripherals

The TILINE peripherals are high-speed I/O systems which transfer data to and from 990 memory at rates that approach the instruction execution rate of the 990 computer. These systems include the DS31, DS10, DS25, and DS50 hard-disk systems and the 979A magnetic-tape system. All are used for nonvolatile mass-memory storage to supplement the semiconductor memory in the 990 computer.

Each of the TILINE peripherals consists of one or more electromechanical recording/reproducing devices (disk drive or tape transport), a recording medium (disk pack, disk cartridge, or magnetic tape), and a microprocessor-based "smart" controller, which occupies a full slot in the 990 chassis.

The 990 processor commands a data-transfer operation and specifies the parameters of the operation by sending a block of 8 control words to the controller. Upon receipt of this control block, the "smart" controller manages the entire data-transfer operation (which may involve several-thousand data words) without additional instructions. The controller handles the addressing, format conversions, signal transfers that are unique to the external device, and TILINE data transfers to and from semiconductor memory.

Every DS990 system has at least one high-speed hard-disk system to provide nonvolatile mass storage for the operating-system software, high-level-language compilers/interpreters, utilities, application programs, and user files. The DX10 disk-based operating system dynamically manages the 990 semiconductor memory by rolling programs and data from disk to memory as they are needed and then rolling them back to disk to conserve memory space for active tasks. Sections of the operating system itself are rolled in and rolled out as needed.

All of the hard-disk systems provide the following features:

- Automatic track switching across head and cylinder boundaries
- Automatic verification of track, sector, and format on every disk access
- Cyclic-redundancy (CRC) polynomial error checking of record headers and data
- Variable-record format (from one sector to full track).

The total amount of disk storage required by a system is determined by many factors. These factors include the size of the user's data base, file backup requirements, the complement of high-level languages selected, and system software requirements. Refer to Appendixes D and E for assistance in determining disk requirements.

The 979A magnetic-tape system is useful for recording and reproducing files that are sequential in nature. Typical applications of the magnetic-tape system include backup of critical files and programs, archival recording in an inexpensive format, transportation of programs and data between sites, and data reduction from sequential processes. The DX10 operating system can be transported on magnetic tape for transfer to a high-speed disk system, but the DX10 system cannot be executed directly from the tape system because of the sequential nature of magnetic tape.

Model DS31 Disk System

The Model DS31 Disk System, shown in figure 14, is a relatively small-capacity disk unit that stores up

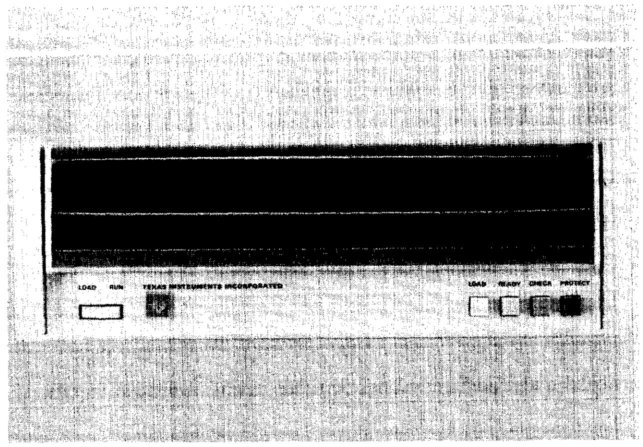


Figure 14. Model DS31 Disk System

to 2.8K bytes in the popular 2315-type, single-platter, removable cartridge. The cartridge is formatted into 203 tracks per surface for a total of 406 tracks. There are twenty-four sectors per track with 288 bytes of data storage per sector.

Single-track seek time is 15 milliseconds, average seek time is 70 milliseconds, and maximum seek time is 135 milliseconds. The average rotational latency (time for one-half revolution) is 20 milliseconds. The burst-transfer rate is 195K bytes per second.

The microprocessor-based "smart" controller occupies one full slot in the 990 chassis and interfaces with the TILINE parallel bus. The DS31 disk controller accepts an 8-word block of command and parameter words before initiating an operation and then independently executes the operation. This simplified control structure is convenient for users who design custom device service routines. It is transparent to users of the DX10 operating system.

As shown in figure 15, a single DS31 disk controller operates one to four disk drives. An overlapped seek capability allows other disks to seek to the next desired track while one of the disks is transferring data. This feature can significantly increase system throughput and is transparent to the DX10 user. The DS31 disk drive is a compact, front-loading unit in a rack-mounted chassis. A DS31 power supply, which mounts on the rear rails of the cabinet, supplies power for two disk drives.

The DX10 operating system is available on 2315-type cartridges, and the DS31 disk can be used as the system disk. However, Texas Instruments recommends the DS10 disk as the smallest system disk for use in general DX10-based systems because of the faster access and greater storage capacity.

The DS31 disk system is most useful for program-development applications with multiple users independently developing program files. Multiple DS31 disk drives allow multiple users to insert and

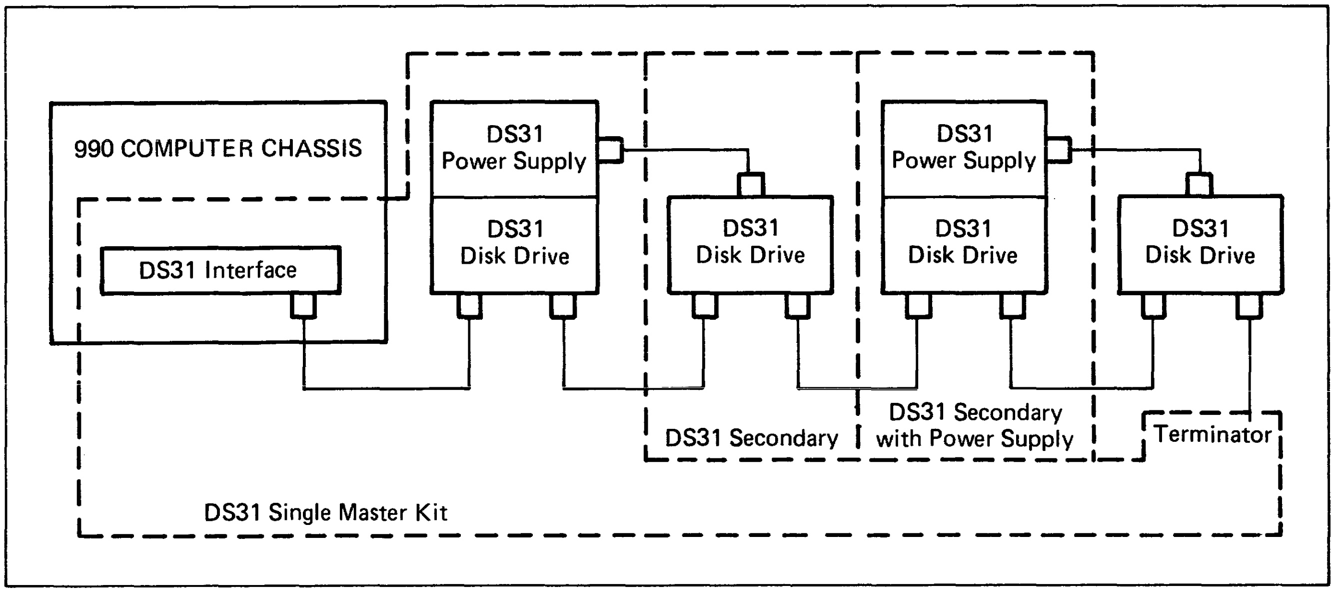


Figure 15. DS31 Controller with Four Disk Drives

remove disk cartridges without interfering with each other and without halting and reloading the operating system. The 2315-type cartridge is a convenient size (both in physical dimensions and in storage capacity), is easily stored, and is economically priced.

Service organizations that provide batch services, such as bookkeeping and subscription maintenance, to multiple customers also can benefit from adding DS31 disks to a standard DS990 Model 4, Model 6, or Model 8 System. Individual DS31 cartridges for each customer provide a convenient and economical means of keeping individual customer data separated. Individual cartridges are easily identifiable and transportable.

Model DS10 Disk System

The Model DS10 Disk System, shown in figure 16, provides mass storage for the DS990 Model 4 System. A DS10 disk drive uses two platters for data recording. One of the platters is built into the disk drive; the other is contained in a 5440-type, removable disk cartridge. Each platter has a formatted storage capacity of 4.7M bytes for a total storage capacity of 9.4M bytes per disk drive. The disk format is 288 bytes per sector, twenty sectors per track, and 816 tracks per platter (408 tracks on each surface).

The nonremovable and disk-cartridge platters share a common rotating-spindle and moving-arm read/write head carriage. Single-track seek time is 7.5 milliseconds, average seek time is 35 milliseconds, and maximum seek time is 60 milliseconds. Average rotational latency at 2400 rpm is 12.5 milliseconds.

The DS10 disk controller occupies a full slot in the 990 chassis and interfaces with the TILINE parallel bus. The two disk platters in a disk drive are treated as separate logical units; that is, a separate 8-word block of control and parameter words is used to control transfers to each platter. There is no automatic "spillover" of data from one platter to the other. Because of the common head carriage, independent overlapped seeks are not available with a DS10 disk system. A DS10 disk controller can operate two DS10 disk drives for a total of 18.8M bytes on four platters.

The DX10 operating system is available on a 5440-type cartridge for the DS10 disk drive. Most users will find it convenient to transfer the operating system to the nonremovable disk so it serves as the

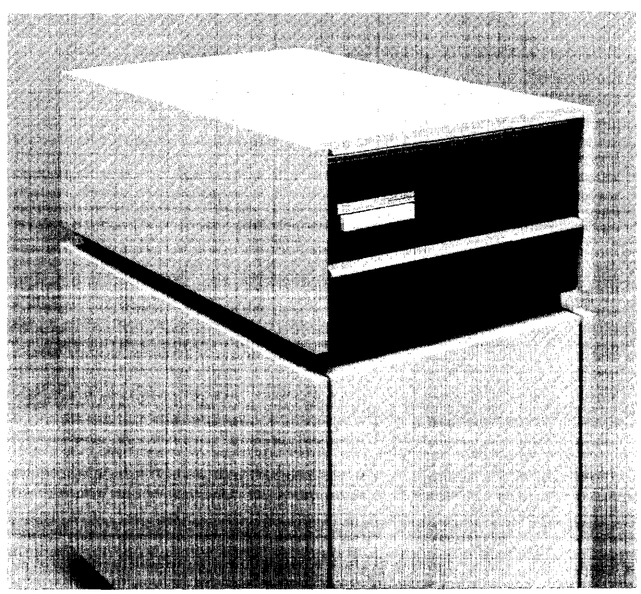


Figure 16. Model DS10 Disk System

system disk. This frees the removable cartridge for user programs and data.

Installing and removing a 5440-type cartridge requires access to the top of the disk drive. The DS10 disk drive is slide-mounted in a basic DS990 Model 4 System. An optional "quietized" pedestal is available for the DS10 disk drive. Texas Instruments recommends this pedestal for low-noise, office-type environments and for applications that require frequent cartridge changes.

If a single DS10 disk drive is used, as in the basic DS990 Model 4 System, the operating system must be stopped and reloaded when disk cartridges are exchanged. This creates an inconvenient situation for multiple users doing program-development work. Additional disk drives are recommended for such environments. The controller has a built-in capability for the addition of a second DS10 disk drive. The addition of a DS31 disk controller and one or more DS31 disk drives may be beneficial.

Model DS25 and Model DS50 Disk Systems

The Model DS25 and Model DS50 Disk Systems, shown in figure 17, are physically similar high-speed and high-capacity mass-storage systems. A DS25 disk drive provides 22.33M bytes of storage, and a DS50 disk drive provides 44.6M bytes of storage. Both

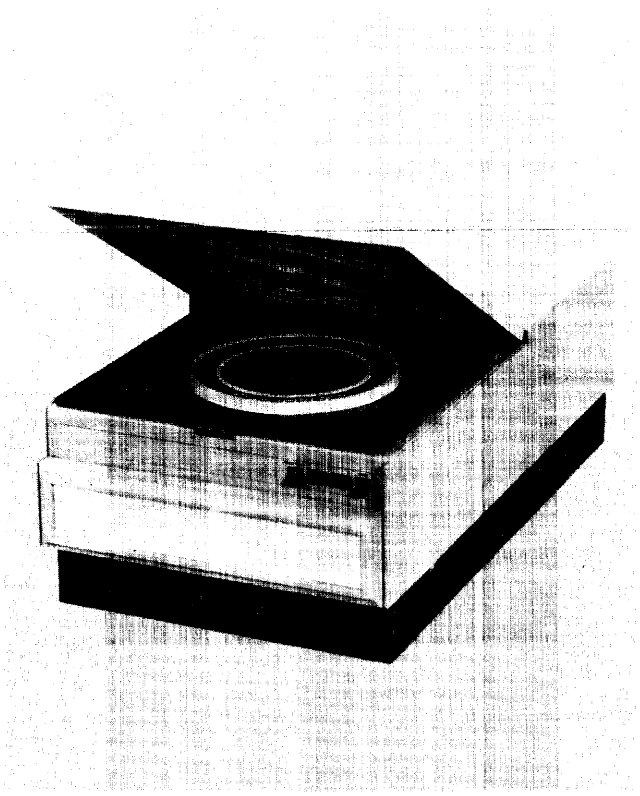


Figure 17. Model DS25/DS50 Disk System

drive types use a high-quality, certified 3336-type removable disk pack. The disk pack has three recording platters and two protective outer platters. The three recording platters provide five recording surfaces and one prerecorded servo surface.

The DS25 disk system provides 408 tracks per surface for a total of 2040 tracks. The DS50 disk system provides 815 tracks per surface for a total of 4075 tracks. For either unit, tracks are divided into thirty-eight sectors with 288 bytes per sector. The sectors are logically interlaced for better system loading and throughput characteristics.

Single-track seek time is 6 milliseconds, average seek time is 30 milliseconds, and maximum seek time is 55 milliseconds. Average rotational latency at 3600 rpm is 8.3 milliseconds. The disk pack is a high-quality, certified, and error-mapped pack per Texas Instruments source specifications. The DX10 operating system has specific provisions for error-mapped disk packs. Certified error-free packs are also available.

The disk controller occupies a full slot in the 990 chassis and interfaces with the TILINE. A single disk controller can operate up to four DS25 disk drives or four DS50 disk drives, but DS25 and DS50 disk drives cannot be intermixed on the same controller. The basic DS990 Model 6 System is supplied with two DS25 disk drives on separate pedestals. Expansion from 44.66M bytes of storage to 67M or 89.32M bytes requires only one or two additional disk drives and connecting cables. The DS990 Model 8 System is supplied with two DS50 disk drives on separate pedestals. Expansion from 89.2M bytes of storage to 133.8M or 178.4M bytes requires only one or two additional DS50 disk drives and connecting cables.

The DX10 operating system is supplied on a DS25 disk for the DS990 Model 6 System and on a DS50 disk for the DS990 Model 8 System. DX10 updates will be written on a customer-supplied disk during the active period of the software subscription. This disk must meet the certification and error-mapping specifications of the disks supplied with the DS25/DS50 disk drives.

The storage capacity of these large disks make the DS990 Model 6 and Model 8 Systems ideal for applications that require large data bases or large numbers of programs on-line for interactive operations.

The DX10 operating system enables multiple users to simultaneously develop software at interactive terminals. If possible, each software developer should have an individual dedicated disk pack so programs under development cannot be overwritten by errant programs. The DS31 disk, which is available as an option, provides an economical, transportable, and easily stored vehicle for an individual user's work. As the work of individual users comes to function, it can be integrated onto a DS25 or DS50 disk pack.

A DS25 or DS50 disk pack also can be dedicated to an individual user, but there are two drawbacks. First, a certified disk pack represents a significant expenditure if there are fifteen or twenty individual disk packs involved. Second, it is not desirable to have a large mass of work on a single disk when unproven programs are added. A safer development strategy calls for the development of reasonably sized program modules, which are backed up on off-line disks or tapes at each stage of development and integration. This strategy lends itself perfectly to structured or "top-down" programming concepts, which form a rational basis for large-software system design.

Model 979A Magnetic-Tape Transport

The Model 979A Magnetic-Tape Transport, shown in figure 18, cannot dynamically run the operating system. Magnetic tapes can be used for storing, transporting, and backing up the DX10 operating system, but the sequential nature of magnetic-tape files prevents the random access necessary for system execution. However, with the DX10 operating system loaded onto a system disk, the magnetic-tape system is fully supported for sequential files. The magnetic-tape system is a viable addition to the DS990 system for users who require large amounts of data storage/backup or who deal primarily with sequential files. The DX10 operating system is available on magnetic tape for initial system loading.

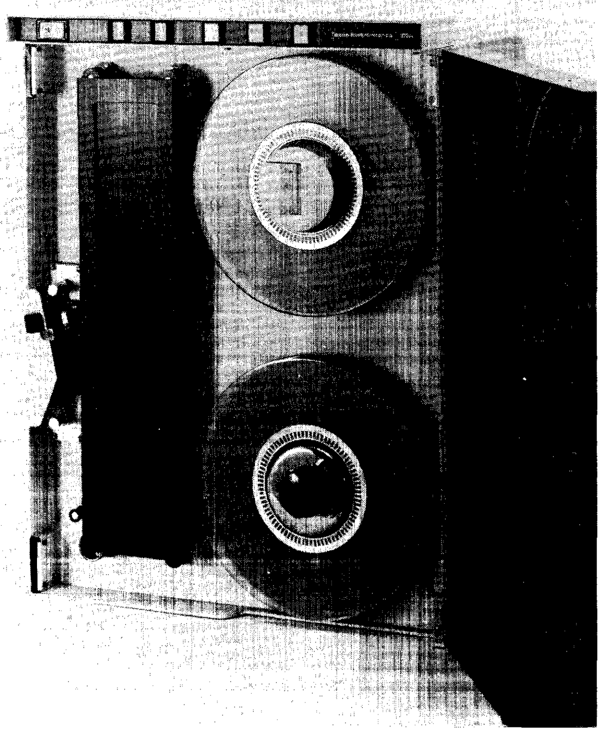


Figure 18. Model 979A Magnetic-Tape Transport

The 979A magnetic-tape transport is available in 800-bits-per-inch, nonreturn-to-zero-format and 1600-bits-per-inch, phase-encoded-format versions. Both versions use industry-compatible nine-track tape formats with 12.7-millimetre (1/2-inch) tape on 267-millimetre (10-1/2-inch) reels. The transport operates at 953 millimetres per second (37.5 inches per second) and features vacuum-column buffering for gentle tape handling.

Two models of single-board controllers are available for the 979A transport. The 800-bpi (only) controller can operate up to four 800-bpi transports. The 1600-bpi controller can operate any mix of 800- and 1600-bpi transports; again, the maximum capability is four transports.

CRU Peripherals

Many I/O devices do not require the high-speed, parallel data-transfer capability of the TILINE. The command-driven CRU system provides communications between the 990 processor and the low- to medium-speed peripheral units (i.e., the data terminals, VDTs, printers, floppy-disk storage units, 990 programmer panels, card readers, communications devices, and special-purpose process-control and instrumentation devices).

While the operations of these devices may appear rapid, they are slow with respect to the instruction execution rate of the 990 computer. This low data rate (compared to the computer clock rate) allows these devices to be efficiently serviced by a single multiplexed, serial data channel (the CRU).

Each device has an associated controller that performs the data format and signal-level conversions peculiar to that device. The controller logic board can occupy a full-slot or half-slot location in the 990 computer chassis or CRU expansion chassis.

Model 911 Video Display Terminal

A minimum DS990 system has a Model 911 Video Display Terminal (VDT) and a dual-port controller (to allow expansion to two VDTs). Additional VDTs and controllers can be easily added to the system. The DX10 operating system has been optimized for rapid response in multiterminal applications. The VDT, shown in figure 19, is used for data entry, display, and interactive program control.

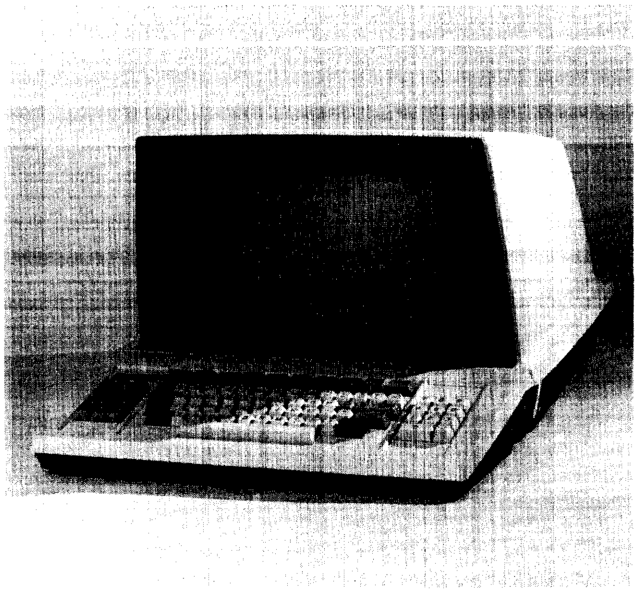


Figure 19. Model 911 Video Display Terminal

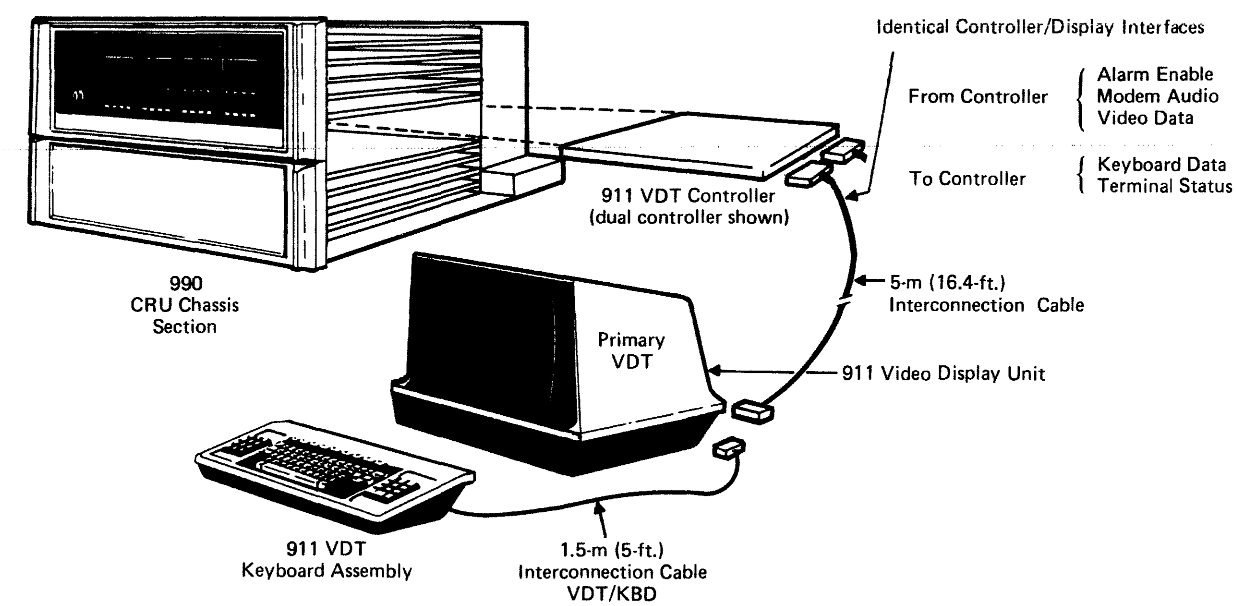
The 911 VDT provides the following features:

- High-resolution display screen
- Instant data display
- Uppercase and lowercase ninety-six-character set (keyboard is full ASCII)
- 1920-character display (twenty-four lines by eighty characters)
- Separated keyboard (cable-connected)
- Standard ten-key numeric pad
- Special-function bit that accompanies each character in memory to specify high or low display intensity or, as a software flag, to indicate a protected field
- Programmable-function keys.

The basic VDT kit, shown in figure 20, consists of a display unit, keyboard assembly, controller circuit board, and connecting cables.

The 911 VDT features a 305-millimetre (12-inch) diagonal, high-resolution screen, housed in a light-gray console. Driving circuitry for the display unit consists of all solid-state components mounted on a circuit board next to the tube. Natural convection cooling without a fan ensures the quiet operation necessary for an office environment.

Bright, easily readable, Gothic-font, uppercase and lowercase alphabetic, numeric, and special (+, -, *, etc.) symbols are formed as 5 x 7 dot-matrix characters. Optional graphic symbols use the entire 7 x 10 character space. The screen capacity is either twelve horizontal lines of eight characters per line (960 characters) or twenty-four horizontal lines of eighty characters per line (1920 characters). A special-function bit accompanying each character in memory specifies high or low display intensity or, as a software flag, indicates a protected display field. Cursor blinking is implemented through software selection.



Note:
Optional extension cables are available in lengths up to 610 metres (2000 feet).

Figure 20. Basic VDT Kit

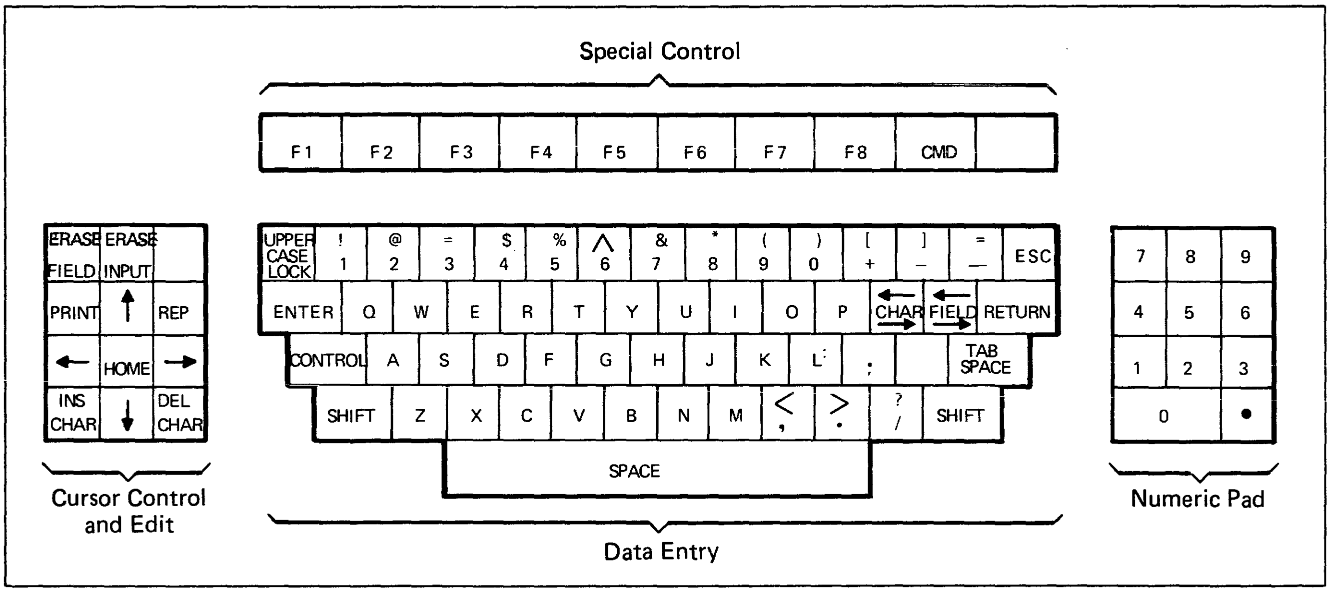


Figure 21. Model 911 Keyboard

For maximum versatility in keyboard positioning and system operation, the keyboard is separate from the display unit of the 911 VDT. The full ASCII 128-character set is extended in capability with thirty-two additional codes for special characters and functions. The eighty-eight keys, shown in figure 21, are arranged in four groups, consisting of data-entry, cursor-control and edit, numeric, and special-control keys.

Data-entry keys have a conventional keyboard arrangement. Pressing data-entry keys in the unshifted mode initiates transmission of lowercase, serial ASCII characters. Pressing either the SHIFT or UPPERCASE LOCK key simultaneously with the data-entry keys initiates transmission of uppercase ASCII characters. Pressing the CONTROL key simultaneously with a selected special symbol or function key initiates serial transmission of the special symbol or function. The cursor-control and editing keys permit the operator to edit and modify entries. The numeric-entry keys are arranged in the accounting industry's standard ten-key pad. Pressing the REPEAT key simultaneously with any data-entry key causes the data entry to be repeated ten times per second.

The REPEAT, SHIFT, UPPERCASE LOCK, and CONTROL keys do not send a code to the computer. Each of the other eighty-three keys has an associated key switch and electronics to produce the full ASCII 128-character set and extended code capabilities when pressed individually or in conjunction with the SHIFT, UPPERCASE LOCK, or CONTROL keys.

The 911 VDT's N-key rollover feature ensures that if more than one key is pressed, the codes for these keys are sent to the computer in the order the keys are pressed without loss of data.

The 911 VDT controller is implemented on a full-sized printed-circuit board that plugs into the chassis of a 990 computer or an expansion chassis. The controller interfaces the keyboard and display unit with the 990 CRU I/O bus. The controllers are available in either single- or dual-port versions. Use of dual-port controllers in multiterminal installations conserves available chassis space.

The controller receives data for display from the computer and provides a self-contained display-image memory. Keyboard data from the VDT is transferred from the controller to the computer, which determines appropriate action (such as write character to display-image memory, move cursor, or edit text) and stores this information in a self-contained display-image memory. The controller reads the data from its memory and generates the ASCII codes for the ninety-six standard display symbols, including uppercase and lowercase alphabetic, numeric, and special-symbol (+, -, *, etc.) characters. The controller can generate thirty-two additional symbols with optional logic or can generate custom symbols and international-language fonts as another option. The controller rewrites (refreshes) the image on the screen fifty or sixty times per second, keeping the image uniformly bright and free of flicker.

Any change in the memory-stored data due to a keyboard entry or computer output is instantly reflected in the display. The computer can fill the display screen with a completely new display (1920 characters) in less than 40 milliseconds. This eliminates the time-consuming line-by-line buildup of displays that is characteristic of less-advanced terminals. Instant display is especially convenient for applications in which display is organized into multiple pages for presentation to the operator. The operator can rapidly leaf through the pages without waiting for the terminal to present the selected display.

Device service routines for the 911 VDT are included in the DX10 operating system. The number of VDTs and the CRU base addresses of the VDT controllers (determined by chassis slot) are specified when the DX10 system is generated and installed. Each additional VDT expands the memory-resident section of the DX10 operating system by about 1500 bytes; this is in addition to the memory space required by the application program itself. If the memory space available for application programs is relatively small, the DX10 operating system must perform an excessive number of disk roll-in/roll-out operations, slowing overall system-response time.

Model 810 Printer

The Model 810 Printer, shown in figure 22, is a multicopy impact printer, which features Texas Instruments unique wire-matrix printhead and microprocessor-controlled bidirectional printing. Speed of the 810 printer is 150 characters per second. Throughput, which is determined by average line length, varies from sixty lines per minute for full 132-character lines to 440 lines per minute for ten-character lines.

Maximum throughput is achieved by line buffering and look-ahead techniques. As the printhead reaches the end of one line, the printer's built-in microprocessor examines the upcoming line and calculates the time to print with left-to-right motion and with right-to-left motion. The print direction is selected for minimum head travel and print time.

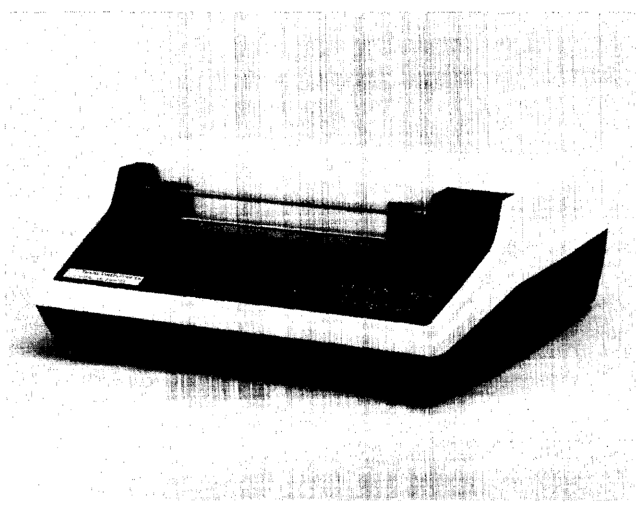


Figure 22. Model 810 Printer

Features of the 810 character printer, as supplied for use with the DS990 systems, include:

- Full ASCII character set with both uppercase and lowercase letters.
- Compressed-character printing. A full 132-column width can be printed on a 203-millimetre (8-inch) line length.
- Vertical-forms control (VFC). Up to eight programmable vertical-forms programs may be resident in nonvolatile storage. These programs can be entered from the 990 computer or from switches in the printer.
- Sprocket-type paper drive with adjustable width from 76 to 381 millimetres (3 to 15 inches).
- Multicopy forms with up to six parts.

The 990 computer controls the printer and sends data at a 4800-baud burst rate via the interface module to the printer for printing. The interface is a full-duplex, RS-232-C circuit, half-sized printed-circuit board that installs in a CRU slot of the 990 computer. The interface module receives serial data and commands from the CRU, converts them to EIA levels, and sends them to the printer. The interface module also receives status from the printer and routes these signals to the computer. The half-sized interface board can be installed in either side of a 990 CRU slot or in a 990 I/O expansion-chassis slot.

Optional extension cables, which are available in increments up to 305 metres (1,000 feet), provide versatility in locating a printer to fit the customer's application.

The 810 printer is a compact, tabletop unit that can be placed on top of the DS990 desk or on a customer-supplied table. Operating with the DX10 operating-system software, the 810 printer provides reliable hard copy at a very low price.

To operate a printer, software routines must provide initialization, character transfer, and end-of-data reporting. Stand-alone software programs for printer operation can be easily designed and implemented; however, output to the printer (other than operational tests) is normally accomplished by calling the device service routine of the DX10 operating system.

Model 2230 and Model 2260 Line Printers

The Model 2230 and Model 2260 Line Printers, shown in figure 23, are free-standing, heavy-duty, drum-type impact printers, which are ideally suited for applications requiring high throughput and

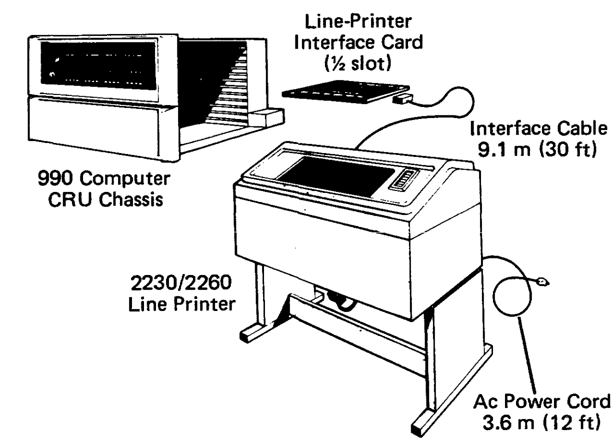


Figure 23. Model 2230 and Model 2260 Line Printers

consistent print quality. The 2230 and 2260 line printers print 300 and 600 lines per minute, respectively, using a sixty-four character, 136-column format. These line printers produce a clear, crisp, straight printout on single- to six-part continuous forms in widths of 102 to 425 millimetres (4 to 16.75 inches).

The 2230 and 2260 line printers are ideal for data processing and business applications. The line printers allow easy access for forms loading, ribbon changing, normal maintenance, and repair. A fault-indicator panel provides easy identification of operator-correctable problems. The self-contained line printers virtually eliminate operator and environmental discomfort by reducing acoustical noise to less than 79 dbA*.

Reliability and maintainability are also important features of the 2230 and 2260 line printers. The line printers consist of the reliable Mark IV print hammer, servo-controlled paper and ribbon-feed system, and rugged print drum. The electronics and major subassemblies are of modular construction to aid in rapid fault isolation and repair. The quick access and functional packaging result in a minimum mean-time-to-repair. An internal self-test provides a built-in means for exercising the line printer off-line by allowing the selection of various test patterns and line formats.

*The unit dbA refers to decibels with respect to the "A" weighting curve from the American National Standard Specifications for sound-level meters (S1.4-1971).

Standard features of the 2230 and 2260 line printers include the following:

- ASCII sixty-four-character set with numbers, uppercase letters, and most commonly used symbols
- Tape-controlled vertical-format unit (VFU), consisting of a paper-tape reader and associated electronics, to enable handling of a variety of form lengths and to allow rapid paper slewing within individual forms
- Static eliminator to ease feeding and stacking problems caused by low-humidity conditions. This extends the operating range to 10 percent relative humidity.
- Floor-mounted paper receptacle to facilitate paper collection and stacking
- Sprocket-type multiple-forms capability with adjustable width from 127 to 406 millimetres (5 to 16 inches).

The interface module for the 2230 or 2260 line printer is a small printed-circuit board that occupies a half slot in the 990 chassis. Print data and control signals are transmitted to the interface board on the CRU bus. Printer status is routed from the interface card to the processor over the CRU bus.

Device service routines for the 2230 and 2260 line printers are incorporated in the DX10 operating system and are activated by simplified logical-unit-number (LUNO) references. Stand-alone service routines for the line printers are easy to design and write.

Model 733 ASR Data Terminal

The Model 733 ASR Data Terminal, shown in figure 24, is a *Silent 700* automatic send/receive, full-duplex data terminal, which is available for use with the DS990 systems. Texas Instruments unique solid-state thermal printhead gives virtually silent printing of eighty-column lines at thirty characters per second. A typewriter-style, limited ASCII keyboard allows operator entries from the 733 ASR data terminal. A twin-tape cassette unit enables off-line editing and cassette read/write/copy. The interface board requires a half slot in the 990 chassis and interfaces with the CRU bus.

The loader ROM supplied with the 990 CPU will load programs from a cassette in the 733 ASR data terminal. This capability is useful for entering and testing programs that are not compatible with the operating system. For example, disk diagnostics can be loaded via a 733 ASR data terminal.

The DX10 operating system is not available in tape-cassette form, but device service routines in the

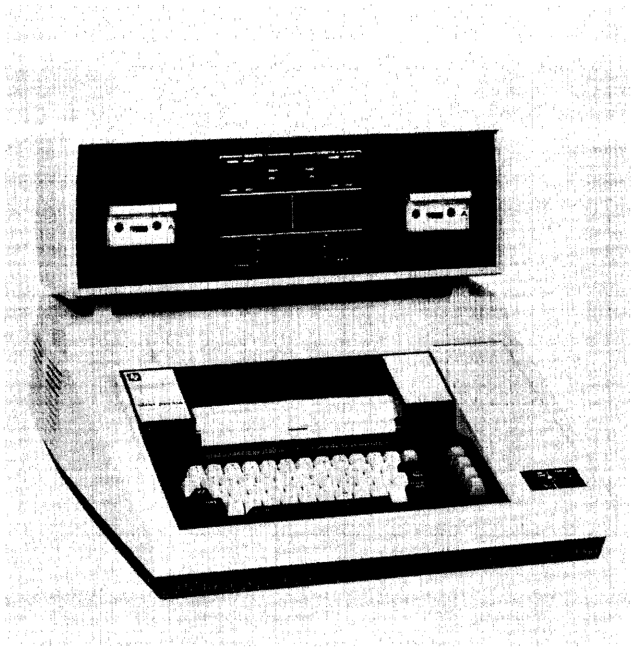


Figure 24. Model 733 ASR Data Terminal

DX10 system support the use of the 733 ASR data terminal as an I/O terminal.

Model 743 KSR Data Terminal

The Model 743 KSR Data Terminal, shown in figure 25, is a *Silent 700*, full-duplex data terminal, which is available for use with the DS990 system. The solid-state thermal printhead gives virtually silent printing of eighty-column lines at thirty characters per second. A sixty-four-character subset of ASCII forms the printer character set. A typewriter-style, limited ASCII keyboard allows operator entries from the 743 KSR data terminal. The interface board requires a half slot in the 990 chassis and interfaces with the CRU bus.

Device service routines for the 743 KSR data terminal are included in the DX10 operating system. For users who have occasional need for hard-copy output, the 743 KSR data terminal is the most economical means of obtaining hard copy from the DS990 system. For users who need multiple copies, full ASCII printing, forms entry, or higher speed printing, the 810 printer is remarkably economical.

Model 804 Card Reader

The Model 804 Card Reader, shown in figure 26, is a tabletop, light- to medium-duty, 400-card-per-minute reader that takes standard-sized, eighty-column punched or marked cards. Card capacity is 1000 in the input hopper and 500 in the output hopper. A fiber-optic read station, which is cleaned automatically by each incoming card, provides excellent accuracy and reliability. The interface board occupies a half slot in the 990 chassis and interfaces

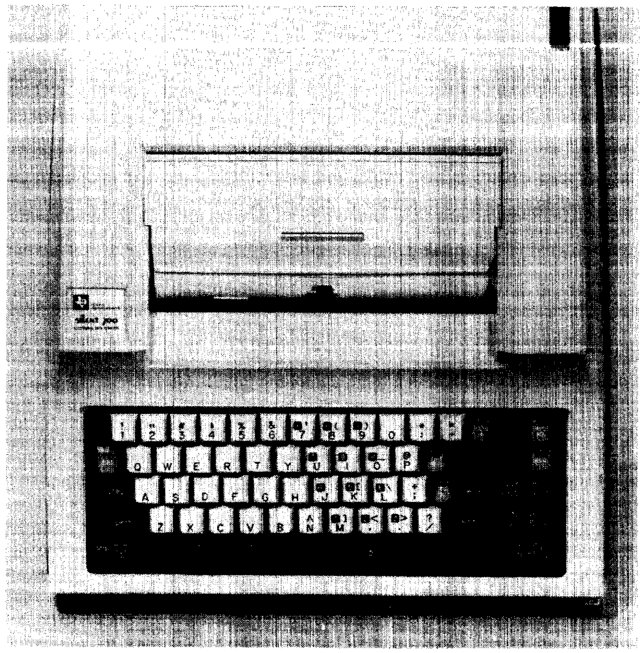


Figure 25. Model 743 KSR Data Terminal

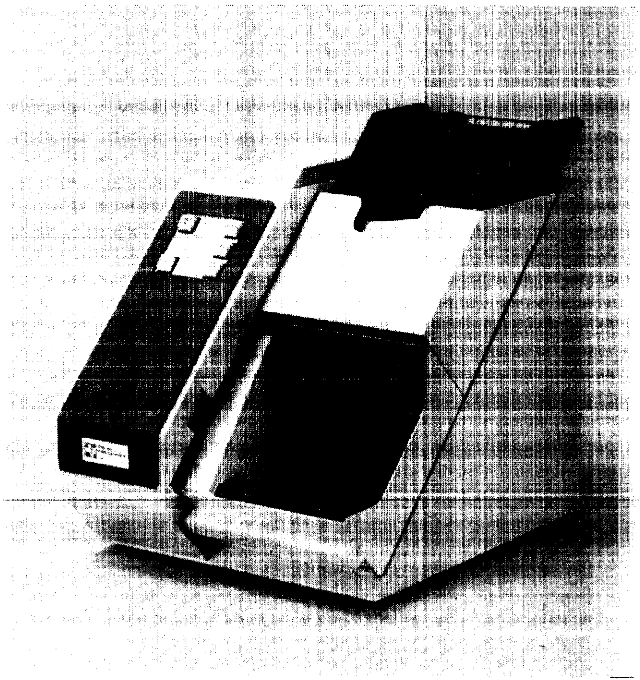


Figure 26. Model 804 Card Reader

with the CRU bus. The DX10 operating system includes device service routines for the 804 card reader. Punched cards are still the most commonly used input media for batch-processed jobs.

Model FD800 Floppy-Disk System

The Model FD800 Floppy-Disk System, shown in figure 27, serves as the prime mass-storage device for FS990 systems. The TX990/TXDS operating system executes from a floppy disk. The FD800 floppy-disk

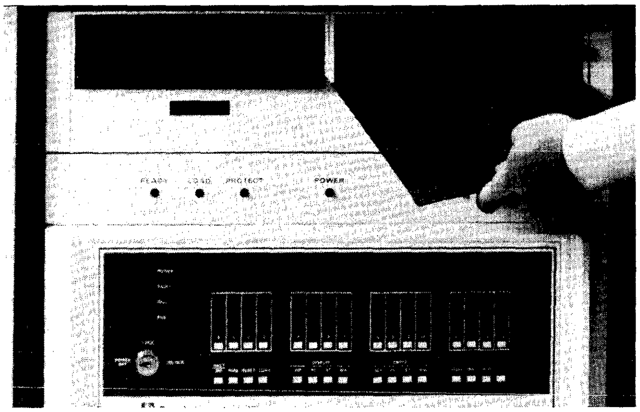


Figure 27. Model FD800 Floppy-Disk System

system does not have the storage capacity, speed, or TILINE interface necessary to execute the DX10 operating system or to support DX10 logical files. However, the FD800 floppy disk can be used as an I/O device for physical records in the same sense as a printer, keyboard, or ASR terminal. Original equipment manufacturers who develop software for FS990-based applications or for intelligent terminals may wish to use the software-development power of the DX10 software in a DS990 system. With Texas-Instruments-supplied utilities, a floppy disk can be added to the DS990 system to transport developed programs to the smaller system.

FD800 floppy disks feature seventy-seven tracks with twenty-six sectors and 128 bytes per sector for a total capacity of 256K bytes per diskette. Adjacent-track seek time is 8 milliseconds, and average rotational latency is 83 milliseconds. The floppy-disk controller uses the low- to medium-speed CRU interface rather than the high-speed TILINE. The average controller-to-host data-transfer rate is 5K bytes per second. This transfer rate is determined by the CRU I/O activity and the application software.

The floppy-disk interface is a full-sized card that plugs into the 990 chassis (the CRU bus) and interfaces with up to four floppy-disk drives. The chassis unit consists of a chassis with power supply that accommodates two floppy-disk drives.

Communications Equipment

The 990 communications modules available for the DS990 system are shown in figure 28. They include the 990 communications interface module, a choice of an asynchronous or synchronous modem, and an accessory auto-call unit. The communications interface module can be used with Bell data sets, which include modems and data-access arrangements.

The 990 communications interface module provides an RS-232-C interface with full modem control signals for synchronous and asynchronous modems. Baud rates of 75, 110, 150, 200, 300, 1200, 2400, 4800, and 9600 meet almost any communication requirement. Character size is selected from 5 to 9 bits with programmable parity (odd, even, or none). Other features include a line-break detection/generation, 250-millisecond timer, programmable SYN, DLE stripping, false-start-bit detection, stop-bit selection (1, 1-1/2, or 2 stop bits), and programmable self-test.

The 990 communications interface module requires a half slot in the 990 chassis and interfaces with the CRU bus. The *Model 990 Computer Communications System Installation and Operation Manual* also covers the 990 asynchronous modem and the 990 synchronous modem.

The 990 asynchronous-modem kit provides a Bell-202-equivalent (1200-baud) modem with auto-answer, capable of a full-duplex operation over a four-wire private line or a half-duplex operation over a DDD network. The modem provides a loop-back for the

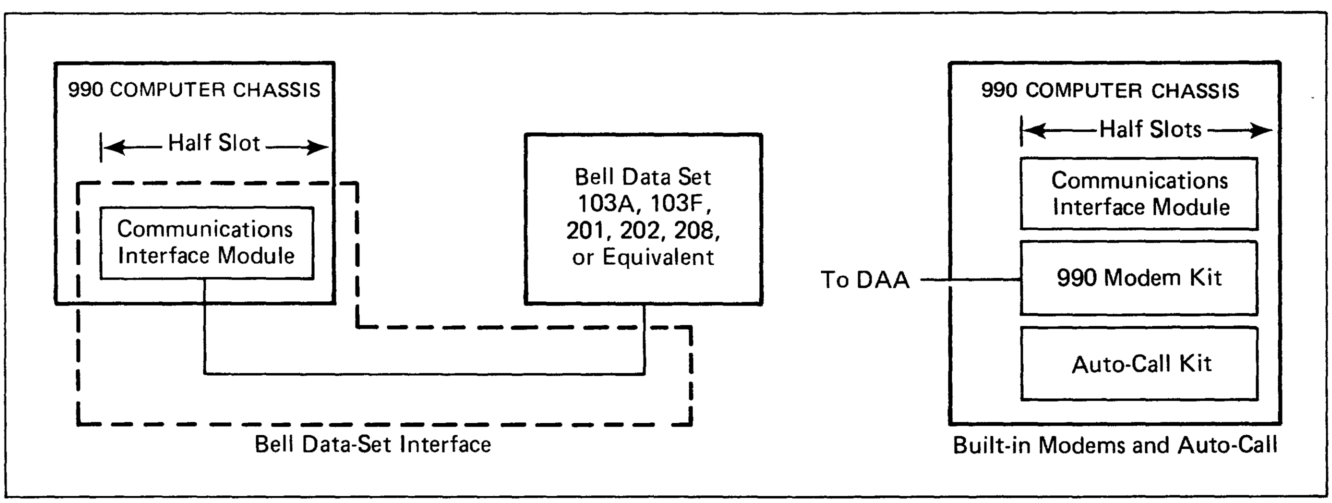


Figure 28. 990 Communications Modules

test. The module requires a half slot in the 990 chassis and interfaces with a 990 communications interface module (not included) via the top-edge connector and cable. The modem must be adjacent to the 990 communications interface module in the chassis.

The 990 synchronous-modem kit is similar to the 990 asynchronous-modem kit, except this kit provides a Bell-201C-equivalent (2400-baud) modem for synchronous communication. The modem provides an internal clock and loop-back for a self-test.

The auto-call kit provides for CPU calling via dial pulse or tone signals to telephone-switching circuitry. The CPU test of an auto-call module is provided by access to internal states. The module plugs into a half slot in the 990 chassis and interfaces with a synchronous or asynchronous modem by a top-edge cable. The auto-call module must be adjacent to a modem in the chassis.

A communications-software package is available from Texas Instruments. This package requires considerable customizing to meet the needs of an individual customer.

Special Interface Devices

Special interface devices include units that are not supported by device service routines under the DX10 operating system. They are not supported because they are application dependent or because they interface with equipment that is not part of Texas Instruments standard line of I/O devices. Use of these peripheral devices will require the user to prepare a device service routine and link it to the DX10 operating system or to the appropriate application program.

EIA Data Modules

The Electronics Industries Association (EIA) data modules provide a general-purpose, 16-bit, parallel I/O interface between the 990 processor and any external device(s) that requires EIA signal levels. The jumper-wire option card provides options of sixteen inputs and sixteen outputs or fifteen inputs, fourteen outputs, and one maskable interrupt.

An interrupt request can be polled regardless of the state of the interrupt mask. A module requires a half slot in the 990 chassis and interfaces with the CRU bus.

TTL Data Modules

The transistor-transistor-logic (TTL) data modules provide two-way parallel interface between the 990 processor and devices that require TTL interface signals. The jumper-wire option card provides options for sixteen inputs and sixteen outputs or fifteen inputs, fourteen outputs, and one maskable interrupt.

An interrupt request can be polled regardless of the state of the interrupt mask. A module requires a half slot in the 990 chassis and interfaces with the CRU bus. Top edge mates with a ribbon-cable connector.

TTY/EIA Terminal Interface Module

The Teletypewriter/Electronics Industries Association (TTY/EIA) terminal interface module interfaces 990 processors with terminal devices, such as VDTs and printers, that use EIA-standard RS-232-C or 20-milliampere TTY current-loop interfaces. (Modems should be interfaced via the communications interface module because several data-set control signals are omitted from the TTY/EIA module.) The TTY/EIA module occupies a half slot in the 990 chassis and interfaces with the CRU bus. The following baud rates can be selected by jumper wires on the card: 110, 300, 1200, 4800, and 9600.

32-Bit Input/Transition Detection Module

The 32-bit input/transition detection module monitors up to thirty-two TTL input lines. Any or all of the lines can be read (sixteen at a time) by the 990 CPU. The module can generate an interrupt when a transition occurs on any line and can supply the address and current state of the interrupting line. A programmable mask determines which lines generate an interrupt on transition, or the entire board can be masked. The module requires a full slot in the 990 chassis, although it physically occupies only a half slot. The module uses all thirty-two CRU channels dedicated to a full slot. The top edge mates with two ribbon-cable connectors on the termination-panel cables.

32-Bit Output-Data Module

The 32-bit output-data module provides thirty-two buffered output lines; each line can be addressed as a single line or as a member of a group of two to sixteen lines. Open-collector transistors on the output can sink 200 milliamperes and hold off up to 50 volts dc. This half-sized module requires a full slot in the 990 chassis because it uses all thirty-two CRU channels dedicated to the slot. The module interfaces with two ribbon-cable connectors on the termination-panel cables.

Digital-I/O-Termination-Panel Module

The digital-I/O-termination-panel module provides the necessary space for terminating sixteen digital I/O channels. The panel consists of sixteen cells, one per channel. Each cell provides pads for installing resistors, capacitors, diodes, jumper wires, and optical isolators. Connection to digital-input or digital-output cards is by means of a ribbon cable. Connection to external devices is by means of terminal strips, one per cell. The kit contains a panel, schematic for the panel, rack-mounting kit, assembly drawing for the rack-mounting kit, and cable. When used with the 32-bit input module or the 32-bit output module, two kits per module are required to terminate all available channels. This kit is not compatible with the sixteen I/O TTL data modules.

D/A Converter Modules

The D/A converter modules generate one to four analog output signals. Output voltage (or current) of each channel is independently commanded via a CRU to any value in output range within a 12-bit resolution. For the voltage option, the output stage regulates voltage into 2K-ohm or larger resistance. For the current option, the output stage regulates current into 500-ohm or smaller resistance. Switch-selected output ranges are as follows:

Voltage Range	Current Range
±10.24 V	±20.48 MA
±5.12 V	±10.24 MA
0 to 10.24 V	0 to 20.48 MA
0 to 5.12 V	0 to 10.24 MA

The maximum settling time to one-half LSB is 10 microseconds for voltage output. The slew rate is 10 volts per microsecond. A module requires a half slot in the 990 chassis and interfaces with the CRU bus.

A/D Converter Modules

The A/D converter modules provide sixteen to sixty-four analog input lines. When commanded by the CPU, a module reads a channel and represents the voltage as a 12-bit binary number. One input line is used for single-ended channels; a pair of input lines is used for differential channels. The maximum throughput rate is 27,500 single-ended channels per second. On-board switches select a ±10.24, ±5.12, 0 to 10.24, or 0 to 5.12 voltage range. A module requires a half slot in the 990 chassis and interfaces with the CRU bus.

990 Chassis Considerations

DS990 computer systems are supplied with the 990/10 minicomputer shown in figure 29. The minicomputer is slide-mounted in the DS990 desk. The 990/10 minicomputer is packaged in the thirteen-slot chassis shown in figure 30. The chassis provides dc power, cooling fans, mounting slots, and interconnections for the 990 CPU boards, memory, and peripheral controllers/interfaces. The integral front panel provides a keylock control and a complete programmer panel.

Each chassis slot has two connectors (P1 and P2), which mate with edge connectors on the horizontally mounted logic boards. Most interconnections between logic boards are made by an etched backpanel that links the chassis-slot connectors. The backpanel etch continues the control signals from slot 1 to all other slots in the chassis. It also distributes dc power. Some logic boards, such as the two 990 CPU boards or the ECC memory controller and ECC array boards, require additional interconnections. These board pairs, which are installed in adjacent slots, use cables or special jumper boards on top-edge (outside) connectors. Cable connections to external devices (printers, terminals, and disk drives) are also made at top-edge connectors on the logic boards.

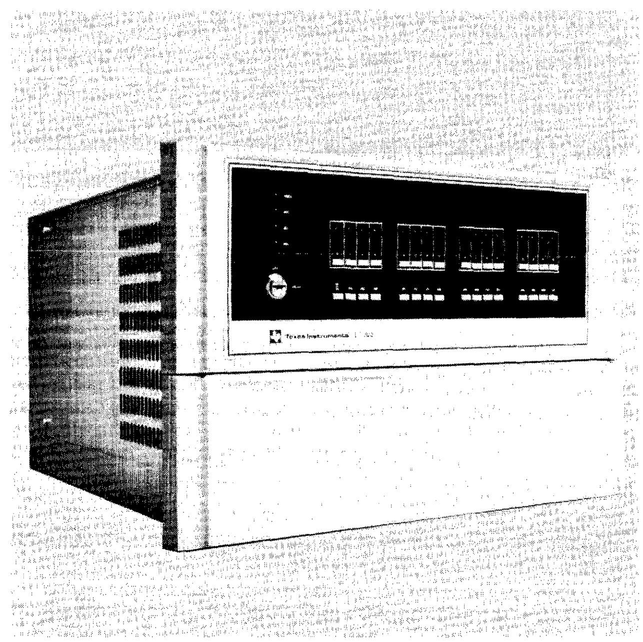


Figure 29. Model 990/10 Minicomputer

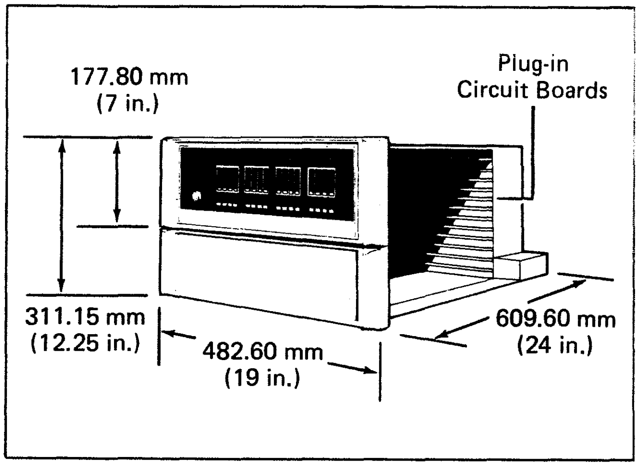


Figure 30. Model 990/10 Minicomputer Thirteen-Slot Chassis

A fixed CRU base address is assigned to each half-slot location, except the first slot location. Interrupt wires from each half-slot location (except slot 1) are routed to an interrupt jumper connector. Interrupt assignments and CRU-base-address assignments must be known at the time of DX10 operating-system generation. Default values that correspond to the standard DS990 configuration are assumed by the sysgen program. Interrupt assignments, CRU base addresses, and board locations are entered on a chassis map when the system is assembled.

Standard Chassis Configuration in DS990 Systems

Figure 31 is a standard configuration chart for the 990 chassis. The chart shows which logic boards are installed in each of the thirteen slots. CRU base addresses and interrupt assignments are omitted. The two 990 CPU boards occupy the first pair of slots. The basic 128K bytes of ECC memory occupy the next two slots. Memory is partitioned into one 64K-byte ECC controller board and one 64K-byte memory array board.

Slot 7 holds the memory-disk controller. The disk type is determined by the DS990 model number, as follows:

DS990 Model Number	Primary Disk Type	Maximum Drives Per Controller
4	DS10	2
6	DS25	4
8	DS50	4

Slot 8 is available for the addition of another disk controller, a 979A magnetic-tape controller, or another TILINE controller. Slot 9 holds the dual-port 911 VDT controller supplied with any DS990 system. Slot 10 and slot 6 are available for the addition of another 911 VDT controller; this raises the system capability to six VDTs. Slot 11 holds the floppy-disk controller if an FD800 floppy disk is selected. Slot 12 holds the optional half-slot printer and card-reader interface boards. Slot 13 holds the optional EIA interface, which is used for a 733 ASR or 743 KSR data terminal.

This standard configuration meets the most common user needs but remains within conservative chassis-design guidelines for dc-power consumption and heat dissipation. I/O expansion beyond this configuration usually requires an I/O expansion chassis. An additional memory array board can be added in slot 5.

Standard DS990 Configuration

1	990/10 AU2	
2	990/10 AU1	
3	ECC Memory Controller	
4	ECC Memory Array	
5	ECC Memory Array*	
6	911 VDT Controller (5 and 6)*	
7	Disk Controller	
8	Other TILINE Controller*	
9	911 VDT Controller (1 and 2)	
10	911 VDT Controller (3 and 4)*	
11	Floppy-Disk Controller*	
12	Printer*	Card Reader*
13		EIA Terminal*

*Options to basic DS990 system

Figure 31. Standard 990 Chassis Configuration Chart

Dc-Power Considerations

Direct-current operating power for the 990 computer is furnished by a chassis-mounted power supply. Operating voltages available are +5 (40-ampere maximum), -5, +12, and -12 volts.

Direct-current operating voltages are distributed to the logic boards by backplane wiring to the connectors at each slot location. Voltages that are critical to memory-refresh operations are distributed separately in slots 1-7. The distribution of these voltages on the memory boards also is split.

Selecting one of the standard configurations specified in this data sheet will assure that the power-supply ratings are not exceeded. Users who plan to use nonstandard configurations or plan to install custom logic modules should perform power-drain calculations. Standard charts to aid these calculations are included in the *990 Computer Family Catalog.*

CRU and TILINE I/O Expansion

If the standard thirteen-slot chassis does not provide enough available mounting slots or dc power to meet user requirements, expansion to one or more additional chassis becomes desirable. The TILINE bus and CRU bus can be extended individually into different chassis, or they can both be extended to the same chassis.

An expansion chassis is a thirteen-slot chassis with a 40-ampere power supply. Except for the front panel, it is identical to the main 990 chassis. The expansion chassis has a simplified front panel with keylock control. It does not have the operator-entry switches and data indicators featured on the programmer panel. The expansion chassis can be equipped with slides for rack mounting or with a dust cover for tabletop use.

CRU bus expansion requires a full-slot CRU expander board in the main chassis and a full-slot CRU buffer in each expansion chassis. The CRU buffer must be installed in slot 1 of the expansion chassis, leaving slots 2-13 available for CRU interface modules. Center card guides allow the division of full slots into half slots (up to a maximum of twenty-four). A single CRU expander can service two expansion chassis. The CRU data-transfer rate in the CRU expansion chassis is reduced to 250K bytes per second. A 3.7-metre (12-foot) cable from the expander to the buffer allows convenient positioning of the CRU expansion chassis.

TILINE bus expansion to a single chassis requires two full-slot TILINE couplers, one in the expansion chassis. The expansion chassis provides up to twelve full slots for memory or TILINE controllers. TILINE devices cannot be packaged on half-slot boards. TILINE transfers to expansion modules require an additional 0.5 microseconds to complete. It is electrically possible to chain TILINE expansion chassis, but the 0.5-microsecond penalty for a coupler pair makes expansion in a "star" more desirable. Each expansion chassis in the "star" requires a separate TILINE coupler in the main chassis. The 3.7-metre (12-foot) coupler-to-coupler cable allows convenient positioning of the expansion chassis.

In practice, the need for multiple expansion chassis is quite rare. Most user requirements can be met with no expansion or with a single combination CRU/TILINE expansion chassis.

Users with unusual I/O requirements should refer to the *Model 990 Family Chassis Data Sheet* and to data sheets for the individual I/O devices or should contact Texas Instruments for assistance. Appendixes F and G provide additional information on chassis and cabinet selection and planning. A list of the contents of the system kits, information on packing and shipping, information on site requirements, and lists of available documentation are provided in Appendixes H, I, J, and K, respectively.

Customer-Support Services

Customer-Support Services

Texas Instruments has specifically developed a broad range of customer services to accommodate the diverse needs of a growing customer base that represents all segments of the marketplace. It is Texas Instruments goal to be the complete source of all minicomputer products and services for our customers. Therefore, the services presented in this section cover all phases of computer-system project implementation. Services can be selected to best suit application and customer requirements. As unique requirements develop, please consult with the local Texas Instruments field sales engineer.

Texas Instruments customer services encompass the following areas: hardware installation, software installation, hardware maintenance, software update, education classes, and telephone hot line.

Hardware Installation

Installation of all standard 990 computer-family hardware products is available within the contiguous forty-eight United States. Hardware installation provides the following services:

- Proper unpacking of all equipment
- Verification of shipment completeness
- Signal-cabling connection between central processor unit (CPU) and peripherals
- Verification of hardware operational status
- Loading and execution of hardware-demonstration-test (HDT) software.

Technically skilled and trained personnel familiar with the technical and operational details of the equipment perform the installation in accordance with correct configuration and start-up procedures. This approach (1) minimizes installation and start-up time, (2) simplifies multisite concurrent installations and start-up at customer sites where no technical staff exists, and (3) ensures functional operation of system before on-line use.

Software Installation

Texas Instruments performs a DX10 system-generation (sysgen) operation in the factory for the items specified in the customer's purchase order (unless the hardware-only version is specified). After the hardware has been installed and checked at the customer's site, a Texas Instruments systems analyst will initiate operation of the DX10 operating system and verify that the factory-generated system runs properly at the site. The systems analyst will re-sysgen the system as indicated and verify proper operation of the DX10 operating system. This service applies only to a DS990 system with the licensed DX10 operating system and standard DS990 software options (such as DX10 FORTRAN or DX10 COBOL) purchased at the same time.

The service does not include linking customer-generated device service routines, application programs, or system modifications. It also does not apply if nonstandard equipment is interfaced with the system. Customers who will be attaching their own equipment are advised to let the Texas Instruments systems analyst check the unencumbered system and software before additional hardware is connected and the final sysgen is performed.

Hardware Maintenance

Three types of hardware-maintenance arrangements are available nationwide from Texas Instruments: maintenance-agreement service (basic, extended, and full coverage), on-call service, and fixed-price repair service. Service personnel are supplied with a spare-parts inventory locally stocked in accordance with the local base of system configurations and backed up by additional parts inventories at the district offices and the Texas Instruments Austin, Texas, factory.

The range of service coverage, shown in tables 13 and 14, allows customers to select the best service plan for each application. Nationwide service personnel and computerized dispatching coordinate every effort to meet the needs of Texas Instruments customers.

Maintenance Agreement

The basic-coverage monthly rate for the maintenance-agreement service is shown for each 990 computer-family product in the itemized price list.

Table 13. Maintenance-Agreement Service

Coverage	Description
Basic Coverage	Service on equipment for eight hours during the period from 8:00 a.m. to 5:00 p.m., Monday through Friday, excluding holidays.
Extended Coverage	Service on equipment for sixteen consecutive hours during the period from 8:00 a.m. to midnight, Monday through Friday, and during the period from 8:00 a.m. to 5:00 p.m. on Saturday, excluding holidays.
Full Coverage	Service on equipment twenty-four hours per day, seven days per week.
Transportation	All travel by automobile and commercial surface and air carriers. This rate includes charges for travel time in transit as well as travel expenses.

Table 14. On-Call Service

Coverage	Description
Standard Shop Rate	Service on equipment during normal shop hours if the customer delivers equipment requiring service to a Texas Instruments service center.
Standard Field Rate	Service on equipment for up to eight hours during the period from 8:00 a.m. to 5:00 p.m., Monday through Friday.
Overtime Field Rate	Service on equipment exceeding eight hours during the period from 8:00 a.m. to 5:00 p.m., Monday through Friday. Service on equipment during the periods before 8:00 a.m. and after 5:00 p.m., Monday through Friday, and any time on Saturday.
Sunday and Holiday Rate	Service on equipment on Sundays or holidays (New Year's Day, Good Friday, Memorial Day, Independence Day, Labor Day, Thanksgiving Day, Friday after Thanksgiving Day, Christmas Day, Texas Instruments floating holiday).
Living Expenses	
Transportation Fees	All travel by automobile and commercial surface and air carriers.

Extended and full coverages are offered only where Texas Instruments can ensure adequate availability of personnel to maintain acceptable service and response time. Service calls resulting from failures or problems not the fault of Texas Instruments equipment will be separately invoiced at Texas Instruments then-current standard service rates.

The maintenance-agreement service covers all routine maintenance (including labor, travel, and material) except those customer responsibilities such as cleaning tape-transport heads, air filters, printheads, and other first-line maintenance items as specified in the equipment manuals. Customer care is equally important to ensure trouble-free operation and optimum performance levels.

On-Call Service

For customers who prefer service on an on-call basis (service performed outside the hours of an agreement and service for customers not covered by a service agreement), Texas Instruments customer engineers are available at Texas Instruments then-current standard service rates for labor, travel, and subsistence as described in the *990 Computer Family Price List*. Rates are subject to change without notice. On-call service outside local Texas Instruments office hours will be provided on a best-effort basis. Material will be charged at Texas Instruments then-current price.

Customers covered by Texas Instruments hardware-maintenance service are benefited in several ways. Highly skilled, expertly trained Texas Instruments customer engineers provide the maintenance service. Service work is performed at the board level to maximize system availability. All 990 computer-family products are designed and/or selected with reliability and board-level service as the primary guidelines. Maintenance-agreement-service customers have priority in both service scheduling and spare parts during peak periods of service requests.

Fixed-Price Repair Service

The fixed-price repair service is for those customers who stock their own spares of standard 990 computer-family products and service their own equipment. Factory fixed-price repair enables the do-it-yourself customer to receive a quality repair. All repairs are made by skilled technicians using test equipment and facilities especially designed for making component-level repairs. To ensure a quality repair, Texas Instruments inspects all work for material and workmanship before it is returned to the customer.

When an assembly is deemed acceptable for repair by Texas Instruments and no request has been made for the return of the same-serial-number assembly, Texas Instruments will, at its option, exchange or repair the defective assembly. Repair or exchange of most assemblies will be accomplished within fourteen days.

Fixed-price repair service offers the following services:

- Replacement assemblies will be of new or refurbished parts of equal quality and will be free of defects in material and workmanship for a period of thirty days from date of shipment to customer.
- Exchange assemblies are shipped from Texas Instruments upon receipt of defective part from customer.
- Complete factory unit testing of each assembly is performed in strict accordance with Texas Instruments material and workmanship specifications.

For an additional per part charge, two options are available. Upon receipt of written notice, Texas Instruments will repair and return to the customer the same-serial-number assembly. This option is applicable when the customer has made modifications to the assembly. If removals or repairs to customer-made modifications are necessary to test the returned assembly in accordance with Texas Instruments test specifications, labor and material will be charged at Texas Instruments then-current standard shop rate and an additional $25 will be charged for handling. The returned assembly should be tagged by the customer to specifically indicate that return of the same-serial-number assembly is required.

A forty-eight-hour, fixed-price expedite option is available for an expedite charge of $25 per part. This expedite option is for single-part emergency orders only and is subject to the availability of replacement assemblies. The customer calls the fixed-price repair center for an authorization number and provides shipping, billing, and purchase-order information. Subject to the availability of the replacement parts, Texas Instruments will provide the customer with delivery information. If parts are not available, Texas Instruments will provide an estimated date of availability.

Benefits of fixed-price repair service include:

- Lowest-cost equipment service when customer maintains an on-site spare-parts inventory and a technical staff
- Increased system uptime due to minimum time required for service response and exchange of defective assemblies
- Optional repair and return of customer-modified assemblies for an additional charge.

Software Update

The 990 licensed software includes a one-year software subscription service providing software updates in the year following purchase. The option for annual renewal of the subscription service is available for a sixty-day period after the initial one-year term lapses. The subscription service provides an updated version of the software and a one-year subscription service for customers who allow the subscription service to lapse.

The subscription service is provided on the media specifically listed for that product. The media will be provided by Texas Instruments except for the DS25 or DS50 disk. Software subscription updates on DS25 and DS50 disk media will be issued on disk packs supplied by the customer unless otherwise requested. Subscription updates on new DS25 or DS50 disk packs are available for additional charge by including an order for a new DS25 or DS50 disk pack.

Subscribers will have sixty days after notification of an impending software-update release to send in a disk pack for update. Texas Instruments will not be liable for loss or damage to the disk packs before arrival at Texas Instruments; thus, suitable arrangements should be made to protect their condition in transit. Disks will be subject to inspection by Texas Instruments, and disks found to be unacceptable will be returned to the customer. Customers will pay for postage to Texas Instruments only, and Texas Instruments will pay for return postage.

Subscribers are encouraged to remain under the software subscription service to receive the benefit of any additional features or improvements made available through new software releases. Support for noncurrent software releases will be limited to correcting any deficiencies deemed necessary by Texas Instruments and will be available for only six months from the date of the most recent release.

Education Classes

Texas Instruments offers regularly scheduled courses in programming and hardware maintenance to users of 990 computers. The courses are conducted at the Digital Systems Division Education and Development Center in Austin, Texas. Courses include both classroom lecture and laboratory projects. The following subjects are some of the regularly scheduled courses:

- Introduction to 990 Assembly Language
- Programming the 990/10 Using DX10
- 990 Hardware Maintenance
- CRU Peripheral Interface Kits Maintenance
- TILINE Peripheral Interface Kits Maintenance
- Introduction to the 990/9900 Computer Family
- TMS 9900 Software Development.

Texas Instruments educational classes benefit customers in several ways. Customer hardware and/or software technical staff receives applicable technical education and hands-on laboratory experience before hardware installation and operation. Technical understanding of Texas Instruments hardware and software before application-system design and development enhances successful project implementation.

Customer-Support Line (Hot Line)

The Customer-Support Line (hot line) is a telephone number for customers to use for direct contact with the factory. A selected staff of senior engineers and programmers provides technical assistance on all Texas Instruments computer-system products. The hot line makes available the software-design, hardware-design, and manufacturing-engineering expertise of the entire manufacturing facility.

Appendixes

Appendixes

The following eleven appendixes provide detailed information about the DS990 system. Appendixes include SCI commands, 990 instructions, memory requirements, recommended memory and disk configurations, disk requirements, 990 computer chassis and I/O expansion chassis planning, cabinet selection and planning, DS990 system kit contents, space, power, and environmental requirements, packing and shipping, and documentation. For further information, consult a Texas Instruments field sales engineer.

Appendix A SCI Commands

Table 15 lists all standard system command interpreter (SCI) commands available with the DX10 operating system. The SCI is a collection of more than 170 procedures that provide system functions ranging from setting the time of day or initiating compiles to backing up disks.

Appendix B 990 Instruction Set

The instruction set of the 990 computer family readily lends itself to efficient processing through simple and effective programming. Table 16 lists the instruction set of the 990 computer family.

Appendix C Memory Requirements

Table 17 is a memory-estimating form that gives system-memory requirements (in bytes) for all of the available DS990 options. Note that user-memory requirements are in addition to these requirements. Each line item in the form is explained in the following paragraphs.

Item 1. Operating System. The base portion of the DX10 operating system contains the anchors for the data structures that are necessary for the operating system to run.

Items 2-13. Physical-Device Support. These items cover the memory requirements for the anchors for the actual physical devices that are included in the system.

Item 14. Disk File-Management Task. The file-management task processes all file input/output (I/O) requests. With additional copies of this task, several file I/O requests can be processed concurrently, yielding faster average processing of each request. A suggested quantity for this item is two tasks. The quantity entered for this item is the answer to the sysgen question "FILE-MANAGEMENT TASKS: (2)".

Item 15. System Overlay Area. The DX10 operating system supports system overlays. The number of overlay areas affects the response time for nonmemory-resident functions. A smaller number of overlay areas will require more disk accesses to perform an operation that uses the system overlay area. A suggested minimum quantity for this item is two overlay areas. The quantity entered for this item is the answer to the sysgen question "OVERLAYS: (2)".

Item 16. Sum of the Base Operating-System Support (Items 1-15). This item represents the base operating-system table support (for devices) and other low-level support. This number does not include the table space for task support, additional I/O buffers, and intertask buffer area.

Item 17. Device I/O Support. The DX10 operating system provides common routines that support device I/O. This number includes the size of the routines necessary to support physical devices.

Item 18. IBM 3780 Communications I/O Support. Under the DX10 operating system, the IBM 3780 communications-emulator package requires a COMMCOM module in the Device I/O Support phase of the system. COMMCOM contains data structures and code and serves as an interface for the communications physical-device I/O support and the standard DX10 I/O modules. Details concerning the COMMCOM module are documented in the communications-package manuals.

Item 19. IBM 3780 Communications-Support Tables. Table areas are required for each IBM 3780 communications link supported by the DX10 operating system.

Item 20. SVC Support. The DX10 operating system provides routines that support supervisor calls (SVCs). This number includes the size of the routines that support the standard supervisor calls used by DX10 software.

Items 21-30. Physical Device Service Routines. These subtotals are the sizes for the standard DX10-supported devices. Only one copy of the necessary device service routine will be included in the system even when more than one physical device of the same type is connected to the system.

The communications device support for the DX10 operating system includes modules that control a communications link using the IBM binary, synchronous, communications-line discipline. They interface with standard DX10 I/O support via specialized modules included in the Device I/O Support phase of the DX10 operating system. One device-service-routine module will support multiple communications links of the same type. Refer to the communications-package manuals for additional details.

Item 31. Sum of SVC and I/O Support (Items 17-20). This total is a composite value that is derived by determining the largest value of items 24-35 and adding items 18-23. It includes the SVC and physical-device I/O support for a DX10 operating system having standard devices and supervisor calls.

Item 32. File-Management Support. The DX10 operating system supports four standard file types: relative-record, sequential, program, and image files. This item is the size required to support the standard files the DX10 system uses. Refer to item 33 for optional key-indexed file support.

Item 33. Key-Indexed Files. Optionally, the DX10 operating system supports key-indexed files. This item must be included to have key-indexed-file support. The answer entered for the sysgen question "KIF? (NO)" must be "yes" to include key-indexed-file logic in the newly generated system.

Item 34. DBMS 990. Detailed information on the data-base management system for 990 computers (DBMS 990) will be available in the next revision of this manual.

Item 35. Largest Physical Record on Disk. The DX10 operating system supports a default record size for blocked files. This item reflects the largest physical-record size for blocked files. This number also should include the largest logical record (task record) that is within the largest blocked physical record (disk record). The number entered for the sysgen question "BUFFER MANAGEMENT: (1K)" should only be the size of the largest blocked physical record.

Item 36. Sum of File-Management Support (Items 32, 33, 34, and 35). This item represents the memory size required for the DX10 standard disk-file support and also includes the support for specialized files.

Item 37. Size of Common. The task common area is a region of system memory accessible to all tasks. It is used for intertask communications and coordination and is directly controlled by the tasks using this area. This is an optional item. The size of this item is directly related to the tasks that will use this area. Therefore, it is difficult to estimate a useful size for this item without knowing the tasks that will use the area.

Item 38. Memory-Resident System Tasks. The DX10 operating system requires some tasks to be in memory at all times. These tasks perform specific functions necessary for the operation of the DX10 system. This item reflects the size of memory-resident tasks that the operating system uses. If the user adds any memory-resident tasks to the system program file, then the size of the user's task(s) should be added to this number.

Item 39. Size of the Largest Item (Items 31, 36, 37, and 38). The maximum size of the DX10 operating system depends on the separate sizes of the different parts but does not depend on the sum of these parts. Therefore, the DX10 operating system can be much larger than the limiting bound of 63,488 bytes described in item 40.

Item 40. DX10 Limit. The DX10 operating system is bounded by hardware limitations. The number given for this item is a maximum bound for a device service routine. For practical purposes, the number is the bound from which the remaining calculations can be made.

Item 41. Sum of Items 16 and 39. This total is a partial sum of the required DX10 operating system. This calculated subtotal includes all of the executable code necessary for the DX10 system to operate. It does not include the dynamic table space or buffer space that is needed to support any active task.

Item 42. Difference of Item 40 and Item 41. This positive subtotal is the size available to be allocated to table space, I/O buffers, and intertask. If this subtotal is a negative number, then items 1-41 must be recalculated, an arithmetic error has occurred, or the specified system has exceeded the allowable size.

Item 43. Additional I/O Buffers. The buffers for I/O operations through physical devices are allocated in the system table area. Additional space may be needed for these buffers if many Initiate I/O calls are made to the system. Buffers for special devices should also be included in the amount of additional buffer area. This number is the answer to the sysgen question "I/O BUFFERS: (0)".

Item 44. Intertask Buffers. The intertask communications buffers (other than common) are taken from the system table area. This number is the maximum amount of memory that the DX10 operating system can use for intertask communications buffers. This parameter must be defined when Sort/Merge is to be run on the system. This number is the answer to the sysgen question "INTERTASK: (100)".

Item 45. Active Foreground Tasks. An active foreground task in the DX10 operating system is a process that requires operator interaction but does not allow operator interaction with other tasks at the same time. This item is an estimate of the amount of system table area required to support an

Table 15. SCI Command Index

Command	Description	Command	Description
AA	Add Alias	HO	Halt Output at Device
AB*	Assign Breakpoint	HT	Halt Task
AF	Append File	IBMUTL	IBM Conversion Utility
AGL	Assign Global LUNO	IDT	Initialize Date and Time
AL	Assign LUNO	IF*	Insert File
ALGS	Assemble and Link Generated System	IGS	Install Generated System
AS	Assign Synonym	INV	Initialize New Volume
ASB*	Assign Simulated Breakpoint	IO	Install Overlay
AT	Activate Task	IP	Install Procedure
AUI	Assign User ID	IS*	Initialize the System
BACC***	Break Apart COBOL Compiler	ISL	Initialize System Log
BATCH**	Begin Batch Execution	ISO	Install System Overlay
BD	Backup Directory	IT	Install Task
BL	Backspace LUNO	IV	Install Volume
CC	Copy/Concatenate	KBT*	Kill Background Task
CD	Copy Directory	KEY**	CFKEY Key Specification
CF	Create File	KO	Kill Output at Device
CFDIR	Create Directory File	KT	Kill Task
CFIMG	Create Image File	LB*	List Breakpoints
CFKEY	Create Key-Indexed File	LC	List Commands
CFPRO	Create Program File	LD	List Directory
CFREL	Create Relative-Record File	LLR	List Logical Record
CFSEQ	Create Sequential File	LM	List Memory
CKS	Copy KIF to Sequential File	LS	List Synonyms
CL*	Copy Lines	LSB*	List Simulated Breakpoints
CM	Create Message	LTS	List Terminal Status
CSF	Create System Files	LUI	List User IDs
CSK	Copy Sequential to Key	MAD	Modify Absolute Diskette
CSM	Copy Sequential Media	MADU	Modify Allocable Diskette Unit
DA	Delete Alias from Pathname	MD	Map Diskette
DB	Delete Breakpoint	MFN	Modify File Pathname
DCOPY*	Diskette Copy/Restore Utility	MFP	Modify File Protection
DD	Delete Directory	MIR*	Modify Internal Registers
DF	Delete File	MKF	Map Key-Indexed Files
DL*	Delete Lines	ML*	Move Lines
DO	Delete Overlay	MLP	Modify LUNO Protection
DP	Delete Procedure	MM*	Modify Memory
DPB*	Delete and Proceed from Breakpoint	MOE*	Modify Overlay Entry
		MPE*	Modify Procedure Entry
DS*	Delete String	MPF	Map Program File
DSB*	Delete Simulated Breakpoint	MPI	Modify Program Image
DT	Delete Task	MR*	Modify Roll
DUI	Delete User ID	MRF*	Modify Relative to File
DXTX*	DX10 File to Diskette File	MRM*	Modify Right Margin
		MS*	Modify Synonym
EBATCH**	End Batch Execution	MSG	Send a Message
EC	Error Count	MT*	Modify Tab Settings
ENDKEY**	End CFKEY Specification	MTE*	Modify Task Entry
FB*	Find Byte	MTS	Modify Terminal Status
FL	Forward Space LUNO	MUI	Modify User ID
FS*	Find String	MVI	Modify Volume Information
FW*	Find Word	MWR*	Modify Workspace Registers

Table 15. SCI Command Index (Continued)

Command	Description
PB*	Proceed from Breakpoint
PF	Print File
PGS	Patch Generated System
Q	Quit SCI
QD*	Quit Debug Mode
QE*	Quit Text Editor
Q$SYN	Erase Secret Synonyms
RAL	Release all LUNOs
RD	Restore Directory
RGL	Release Global LUNO
RL	Release LUNO
RO	Resume Output at Device
RPGCONV	RPG II Diskette Conversion Utility
RPGEDIT	RPG II Source Editor
RS*	Replace String
RST*	Resume Simulated Task
RT*	Resume Task
RWL	Rewind LUNO
SAD	Show Absolute Diskette
SADU	Show Allocable Diskette Unit
SBS*	Show Background Status
SDT	Show Date and Time
SF	Show File
SIR*	Show Internal Registers
SIS	Show I/O Status
SL*	Show Line
SMS	Show Memory Status
SOS	Show Output Status
SP*	Show Panel
SPI	Show Program Image
SRF	Show Relative to File
ST*	Simulate Task
STI	Show Terminal Information
STS	Show Task Status
SV*	Show Value
SVS	Show Volume Status
SWR*	Show Workspace Registers
TGS	Test Generated System
TXCM	Compress Diskette File
TXCP*	Change Diskette File Protect
TXDF	Delete Diskette File
TXDX*	Diskette File to DX10 File
TXFD*	Format Diskette
TXMD*	Map Diskette
TXSF*	Set System File
UV	Unload Volume
VB	Verify Backup
VC	Verify Copy
WAIT*	Wait for Background
XANAL*	Analyze DX10 Crash File
XB*	Execute Batch SCI
XBB*	Execute Business BASIC
XBSM*	Execute Batch Sort/Merge
XCC	Execute COBOL Compiler
XCCF*	Execute COBOL Compiler in Foreground
XCP	Execute COBOL Program
XCPF*	Execute COBOL Program in Foreground
XCT	Execute COBOL Task
XCTF*	Execute COBOL Task in Foreground
XCU*	Execute 2.2 to 3.0 DX10 Conversion
XD*	Initiate Debug Mode
XE*	Initiate Text Editor
XES	Initiate Text Editor with Scaling
XFC	Execute FORTRAN Compiler
XFCF*	Execute FORTRAN Compiler in Foreground
XFT	Execute FORTRAN Task
XFTF*	Execute FORTRAN Task in Foreground
XGEN*	Execute GEN990—Auto-Sysgen Program
XHT	Execute and Halt Task
XLE	Execute Linkage Editor
XMA	Execute Macro Assembler
XPS	Execute Patch Synonym Processor
XRPGB	Bind RPG II Program
XRPGC	Execute RPG II Compiler
XRPGCF	Execute RPG II Compiler in Foreground
XRPGT	Execute RPG II Task
XRPGTF	Execute RPG II Task in Foreground
XSB*	Execute Scientific BASIC
XSM*	Execute Sort/Merge
XSMF*	Execute Sort/Merge in Foreground
XT	Execute Task
XTS	Execute Task and Suspend SCI
XTU***	Execute Transliteration Utility
XTUM***	Execute Transliteration Utility Using MIRA

* Foreground only

** Batch only

*** With source packages only

Table 16. 990 Instruction Set

Category	Instruction	
Arithmetic	Add Words	Increment
	Add Bytes	Increment by Two
	Add Immediate	Decrement
	Subtract Words	Decrement by Two
	Subtract Bytes	Absolute Value
	Multiply	Negate
	Divide	
Logical	AND Immediate	Set to Ones
	OR Immediate	Set Ones Corresponding
	Exclusive OR	Set Ones Corresponding, Byte
	Invert	Set Zeros Corresponding
	Clear	Set Zeros Corresponding, Byte
Shift	Shift Right Arithmetic	Shift Right Logical
	Shift Left Arithmetic	Shift Right Circular
Compare	Compare Words	Compare Ones Corresponding
	Compare Bytes	Compare Zeros Corresponding
	Compare Immediate	
Branch	Branch	Branch and Load Workspace Pointer
	Branch and Link	Return with Workspace Pointer
	Unconditional Jump	Jump if Equal
	Jump if Logical High	Jump if Not Equal
	Jump if Logical Low	Jump on Carry
	Jump if High or Equal	Jump if No Carry
	Jump if Low or Equal	Jump if No Overflow
	Jump if Greater Than	Jump if Odd Parity
	Jump if Less Than	Execute
Load and Move	Load Immediate	Move Byte
	Load Interrupt-Mask Immediate	Swap Bytes
	Load Workspace-Pointer Immediate	Store Status
	Load Memory-Map File	Store Workspace Pointer
	Move Word	
Control and CRU I/O	Reset	Set Bit to Logic Zero
	Idle	Set Bit to Logic One
	Clock Off	Test Bit
	Clock On	Load CRU
	Load or Restart Execution	Store CRU
Long Distance	Long-Distance Source	Long-Distance Designation
Extended Operations	(Not Defined)	

Table 17. DX10 Memory-Estimating Form

Item Number	Description	Module Size (Bytes)	x	Quantity	=	System Memory Subtotal (Bytes)	Total System Memory (Bytes)	Required (Yes/No)
1	Operating system	11300				11300		Yes
2	Disk(s)	170	x	____	=	____		Yes
3	Disk controller(s)	40	x	____	=	____		Yes
4	911 VDT(s)	140	x	____	=	____		No
5	913 VDT(s)	140	x	____	=	____		No
6	Line printer(s)	80	x	____	=	____		No
7	733 ASR with cassette(s)	290	x	____	=	____		No
8	743 KSR(s)	140	x	____	=	____		No
9	979A magnetic tape(s)	170	x	____	=	____		No
10	979A controller(s)	40	x	____	=	____		No
11	Diskette(s)	100	x	____	=	____		No
12	Card reader(s)	80	x	____	=	____		No
13	IBM 3780 communications link	120	x	____	=	____		No
14	Disk (file-management tasks)	380	x	____	=	____		Yes
15	System overlay area(s)	800	x	____	=	____		Yes
16	Sum of base operating-system support (Items 1-15)						____	
17	Device I/O support	1800				1800		Yes
18	IBM 3780 communications I/O support	2200	x	1	=	____		No
19	IBM 3780 communications-support tables	1250	x	____	=	____		No
20	SVC support	7100		____		7100		Yes
21	Disk	950		____			950	Yes
22	911 VDT	1700	x	1	=		____	No
23	913 VDT	1700	x	1	=		____	No
24	Line printer	300	x	1	=		____	No
25	733 ASR with cassette	1850	x	1	=		____	No
26	743 KSR	1000	x	1	=		____	No
27	979A magnetic tape	850	x	1	=		____	No
28	Diskette	1150	x	1	=		____	No
29	Card reader	550	x	1	=		____	No
30	IBM 3780 communications link	4200	x	1	=		____	No
31	Sum of SVC and I/O support (Items 17-20)						____	
32	File-management support	3950				3950		Yes
33	Key-indexed files	4230	x	1	=	____		No

Table 17. DX10 Memory-Estimating Form (Continued)

Item Number	Description	Module Size (Bytes)	x	Quantity	=	System Memory Subtotal (Bytes)	Total System Memory (Bytes)	Required (Yes/No)
34	DBMS 990	*	x	*	=	*	*	*
35	Largest physical record on disk	____	x	1	=	____		Yes
36	Sum of file-management support (Items 32, 33, 34, and 35)						____	
37	Size of common	____	x	1	=	____	____	No
38	Memory-resident system tasks	4300				4300	4300	Yes
39	Size of largest item (Items 31, 36, 37, and 38)					____		
40	DX10 limit	63488				63488		
41	Sum of items 16 and 39					____		
42	Difference of item 40 and item 41					____		
43	Additional I/O buffers	____	x	1	=	____		No
44	Intertask buffers	____	x	1	=	____		No
45	Active foreground tasks	1300	x	____	=	____		Yes
46	Active background tasks	1600	x	____	=	____		No
47	Installed disk(s)	400	x	____	=	____		Yes
48	System table size (Sum of items 45, 46, and 47)					____		
49	Total system table area (Sum of items 43, 44, and 48)						____	
50	Sum of Total System Memory column						____	

*Information on the DBMS 990 will be available in the next revision of this manual.

active foreground task. The actual requirement depends on the specific application environment and may be less than or greater than the estimated amount.

For full support of all terminals, the quantity specified for this item should be equal to the sum of the quantity of items 4, 5, 7, and 8, which is the number of terminals. A quantity less than this sum may be used, and the DX10 operating system will enforce this specified quantity. In systems where the terminal usage is less than 100 percent, specifying a quantity less than the total number of terminals on the system may be quite satisfactory. If the maximum number of terminals are active and a user attempts to activate another terminal, the DX10 operating system will reject the request with a message informing the user that he cannot be supported at this time.

The quantity entered for this item is the answer to the sysgen question "SCI FOREGROUND: (8)".

Item 46. Active Background Tasks. An active background task in the DX10 operating system is a process that is not a foreground process. This item is an estimate of the amount of system table area required to support an active background task. As with the estimate for the foreground task, the actual requirement depends on specific details of the background process. The quantity entered for this item is the answer to the sysgen question "SCI BACKGROUND: (2)".

Item 47. Installed Disks. The information necessary to access a disk when it is installed is retained in the system table area. For each disk installed, table space is required. The quantity entered for this item is the maximum number of disks that can be installed at any time.

Item 48. System Table Size (Sum of Items 45, 46, and 47). This is the dynamic table area from which the work areas for the active tasks in the system are taken. This item is a composite of estimates and is the answer to the sysgen question "TABLE SIZE: (1K)". For a better estimate, it is necessary to monitor the usage of the system table area on a running system. This can be done by using the SCI command SMS (Show Memory Status).

Item 49. Total System Table Area (Sum of Items 43, 44, and 48). This item is the total space allocated for the system work area. This item must be less than or equal to item 42. If this item is greater than item 42, then either items 1-39 need to be decreased or items 43-47 need to be decreased in size.

Item 50. Sum of "Total System Memory" Column. The DX10 operating system is structured in such a manner that the total size of the system can be much larger than 63,488 bytes of memory. The sum of the Total System Memory column is a reasonably close estimate of the total memory size used by the DX10 operating system. The rest of the memory on the system can be used by any disk-resident task.

Appendix D Recommended Memory and Disk Configurations

Table 18 gives the recommended memory and disk configurations for a software-development system, and table 19 gives similar information for application systems. The recommended disk size is not given because it depends on the size of the user's data base.

Table 18. Recommended Software-Development Configurations

DS990 Model	Number of Terminals	Recommended Disks	Recommended Memory (K Bytes)	Recommended Languages and Utilities
4	1-2*	DS10	128	All configurations support FORTRAN, COBOL, BASIC, RPG II, DBMS 990, Pascal, or Sort/Merge.
4	3-4	Dual DS10	128	
6	3-4	Dual DS25	128	
6	5-8	Dual DS25**	192	
6	9-12	Dual DS25**	256	
8	3-4	Dual DS50	128	
8	5-8	Dual DS50**	192	
8	9-12	Dual DS50**	256	

*These terminals are used concurrently.

**Additional small disks, such as DS31 disks, may be desirable to allow each user an individual disk.

Table 19. Recommended Application-System Configurations

Number of Terminals	DS990 Model	Recommended Disks	Recommended Memory (K Bytes)	Recommended Languages and Utilities
1-4	Model and disk depend on size and characteristics of application data base.		128	All configurations support FORTRAN, COBOL, BASIC, RPG II, DBMS 990, Pascal, or Sort/Merge.
5-8			192	
9-12			256	
13-16			384	

Appendix E
Disk Requirements

Table 20 gives the storage size (in megabytes and in sectors) of each of the basic disk packs. The table also shows the percentage of the disk pack that would be used by an installed DX10 operating system. Table 21 gives the disk-space requirements of each of the major software kits available for use with the DS990 system.

Table 22 gives disk-space requirements for typical files that are developed when a DS990 system is used for software development. This data allows the user of a software-development station to determine file requirements based on the expected level of activity at the installation. User-file requirements and software-storage requirements are added and compared to the storage available on the various disk packs. For uses other than software development, the user must base the estimate of user-file size on the specific application.

Table 20. Disk-Pack Sizes

Disk	Disk-Pack Size (M Bytes)	Disk-Pack Size (Sectors)	Percentage of System Disk Used by DX10 Object
DS31	2.8	9,744	65
DS10*	4.7	16,320	39
DS25	22.3	77,520	8
DS50	44.6	154,850	4

*Two disks per drive

Appendix F
Chassis Planning

Chassis planning involves three steps: selection of I/O and peripheral devices, planning of direct-current power, and assignment of controller logic boards to chassis-slot locations. The selection of peripheral devices depends on the application. The hardware descriptions supply information that is useful for device selection. For more detailed information on any given peripheral or I/O device, refer to the data sheet for that device.

Dc-power planning ensures that the power consumption of the logic boards remains within the ratings of the chassis power supply. This is seldom a problem with the rugged 40-ampere power supply in the thirteen-slot chassis. The memory boards and the high-speed disk and tape controllers draw more dc power than an average logic board; concentrations of these logic boards in one chassis can exceed the

Table 21. DX10 Software Sizes

Description	Disk Space as Installed (M Bytes)*	Memory (K Bytes)
Object	1.82**	52
FORTRAN Object	0.52***	32
COBOL Object	0.23†	32
BASIC Object	0.25	32
Business BASIC Object	0.25	32
RPG II Object	0.56††	37
Pascal Object	1.50	64
3780 Object	0.45	19
Sort/Merge Object	0.24	32/64

*A sector has 288 bytes.

**1.82M bytes include two terminals.

***0.52M bytes include 0.17M bytes for compiler and 0.35M bytes for run time.

†0.23M bytes include 0.12M bytes for compiler and 0.11M bytes for run time.

††0.56M bytes include 0.3M bytes for compiler and 0.26M bytes for run time.

Table 22. Software-Development Disk Requirements

Description	System Disk (Sectors)*	Secondary Disk (Sectors)
Create 100-line source file	0	15
Text edit 100-line source file with changes on 10 percent of the records	9 (Temporary file)**	15
Assemble typical 100-line program	18 (Temporary file)	4 (Object output)
COBOL compile 100-line program	0	4 (Object output)
FORTRAN compile 100-line program	20 (Temporary file)	6 (Object output)
Link edit 5K byte	20 (Temporary file)	26 (Linked object)

*A sector has 288 bytes.

**The DX10 operating system always allocates temporary work files to the system disk. Do not place so many user files on the system disk that temporary-file space becomes restricted.

power-supply ratings. Note that the standard DS990 chassis configuration, shown in figure 31, has been engineered to meet both dc-power and chassis-slot constraints. Detailed chassis planning is unnecessary for this configuration or a subset of it.

Chassis-layout planning determines the chassis-slot locations for the logic boards. Planning ensures that the selected configuration is realizable and workable with minimum chassis modifications. For example, a memory array board must be located adjacent to the memory controller because they are linked by short cables across the top edge. As another example, slots 1 and 2 are assigned to the 990 arithmetic-unit (processor) logic boards. Again, the standard DS990 chassis configuration of figure 31 has been preplanned to meet all chassis power and slot-location constraints. Detailed planning is unnecessary for this configuration.

Chassis-Power Planning

The Chassis Planning Form, shown in table 23, allows rapid calculation of dc-power requirements for standard Texas Instruments chassis, peripherals, and interfaces. The form uses shortcuts and simplifications that are completely safe for standard catalog items but are inadequate for special designs furnished by customers. For more information, refer to the *Model 990 Computer Family Chassis* data sheet.

To calculate dc-power requirements, perform the following steps:

1. Use a separate copy of the form for each chassis.
2. Every item in part 1 contains a chassis. The slots in part 1 are vacant slots in that chassis, and the power numbers are available current in amperes. Identify the chassis type, and underline that row. Copy the slots and power available onto line A.
3. Identify the modules in part 2 that will be plugged into the chassis. Enter the number of modules used in the Quantity column.
4. Working across the row for each item used, complete every box. Plan chassis slots = quantity x chassis slots per unit. Dc power plan +5 V main = quantity x +5 V main per unit, and so forth.
5. Add the Plan columns, and enter the total on the bottom line and on line B in part 1.
6. Compare the power requirements (line B) to the power available (line A). If the power and slots available exceed the requirement, then the plan is right.

   If line A +5 V Mem and +12 V Mem equal zero, then the chassis will have no standby power supply. The +5 V Mem will be connected to +5 V Main. In this case, add +5 V Mem and +5 V Main on line B, and enter the total on line C +5 V Main. Then, add +12 V Mem and +12 V Main, and enter the total on line C +12 V Main.

   If line A +5 V Mem and +12 V Mem do not equal zero but the power requirement exceeds the power available, then more memory will be on the standby power supply than the supply can support. Some modules must be removed from standby power. Identify the modules in part 2 to be moved. For these modules, subtract the +5 V Mem and +12 V Mem power from the plan, and add the same values to the planned +5 V Main and +12 V Main. Carry the corrections through the form, and enter them on line C.

   Compare the corrected power requirement (line C) to the power available (line A). If the power available exceeds the power requirement, then the plan is right. Otherwise, add one or more expansion chassis for CRU or TILINE expansion. In this case, use a separate copy of the form for each chassis to evaluate the power and slots for the new plan.

Table 23. Chassis Planning Form (Part 1)

Description	Vacant Chassis Slots Per Unit	Dc Power Per Unit Available Current in Amps +5 V Main	+5 V Mem*	+12 V Main	+12 V Mem*
DS990 Model 4 (13-slot chassis)	7	13.44**	0	2.74**	0
DS990 Model 6 (13-slot chassis)	7	11.44**	0	2.74**	0
DS990 Model 8 (13-slot chassis)	7	11.44**	0	2.74**	0
990/10 Minicomputer Chassis					
Mapping / *Panel* / *Standby Power*					
with / programmer / without	11	31.48	0	4.00	0
with / programmer / with	11	31.48	1.40	4.00	1.20
CRU expansion kit	12	39.50	0	4.00	0
TILINE expansion kit	12	38.80	0	4.00	0
A Available					
B Required (from Part 2)					
C Corrected power requirements					

*Available for memory only.

**Based on equivalent 4K-RAM memory sizes.

Chassis-Layout Planning

The standard chassis layout, shown in figure 32, and the blank chassis-layout forms, shown in tables 24 and 25, simplify layout planning for standard Texas Instruments chassis, modules, and peripherals. For more information, refer to the *Model 990 Computer Family Chassis* data sheet.

In general, any module can be plugged into any slot in the chassis with the following exceptions:

- Slot 1 is reserved for 990/4 AU, 990/10 AU2, or communications-register-unit (CRU) buffer board. (Slot 1 is uniquely wired to decode CRU strobes).
- Certain modules are interconnected by cables across the top edge and must be adjacent in the chassis. These include the 990/10 AU1 and 990/10 AU2, error-checking-and-correcting (ECC) memory controller and array board, communications interface module and modem, auto-dialer module and modem, and emulator and trace modules.
- Most modules generate interrupt signals that must be routed to the central processing unit (CPU) and must arrive on the interrupt level where the software recognizes them. Interrupt jumpers can be used to vary these assignments.
- Most CRU modules use decoded CRU addresses and consequently must be plugged into the chassis where the software attempts to address them.
- The first TILINE controller (hard disk or tape) should be located in slot 7 because the priority-signal (TLAG) path is specifically cut at slot 7 for this installation.

Use of standard configurations results in shorter delivery time than custom configurations. The documentation that accompanies the system explains how to alter or add to the configuration. Texas-Instruments-furnished operating-system packages include one object that can be linked for custom sysgen plus one or more linked-object systems for immediate operations. The system must be operational to perform a sysgen. In general, this means a chassis layout must match the standards shown in figure 32 until the custom sysgen has been completed. However, Texas Instruments will perform custom system generation at the factory for equipment specified on a DS990-system purchase order. For Texas Instruments to configure backplane jumpers, a chassis layout must be included with the purchase order.

Table 23. Chassis Planning Form (Part 2)

Description	Qty	Chassis Slots Required		Dc Power Per Unit Current Required in Amps				Dc Power Plan			
				+5 V		+12 V		+5 V		+12 V	
		Per Unit	Plan	Main	Mem	Main	Mem	Main	Mem	Main	Mem
TMS 9900 emulator kit		1		2.90	0	0.70	0				
Logic-state trace module kit		1		5.00	0	0	0				
PROM programmer kit		½		0.53	0	0	0				
EPROM memory module with 2K bytes		1		0.40	0	0.10	0				
Each additional 2K bytes EPROM		0		0.10	0	0.10	0				
ECC memory controller with 64K bytes		1		2.00	0.55	0	0.54				
ECC memory controller with 96K bytes		1		2.01	0.55	0	0.59				
ECC array with 64K bytes		1		0.30	0.40	0	0.80				
ECC array with 128K bytes		1		0.31	0.40	0	0.90				
ECC array with 192K bytes		1		0.32	0.40	0	1.00				
ECC array with 256K bytes		1		0.33	0.40	0	1.10				
Floppy-disk master kits*		1		3.00	0	0.20	0				
DS31 master kits*		1		4.50	0	0	0				
DS10 master kits*		1		6.00	0	0	0				
DS25 master kit*		1		8.00	0	0	0				
DS50 master kit*		1		8.00	0	0	0				
DS200 master kit*		1		8.00	0	0	0				
979A master kit, 800 bpi (800 b/25.4 mm)*		1		5.00	0	0	0				
979A master kit, 800/1600 bpi (800/1600 b/25.4 mm)*		1		7.00	0	0	0				
911 VDT kit, single-display controller		1		2.62	0	0.12	0				
911 VDT kit, dual-display controller		1		4.20	0	0.24	0				
743 master kit		½		0.38	0	0.02	0				
733 master kit		½		0.38	0	0.02	0				
810 Printer master kit		½		0.38	0	0.02	0				
2230/2260 Printer master kit		½		0.38	0	0.02	0				
804 Card Reader master kit		½		0.60	0	0	0				
TTY/EIA interface module		½		0.38	0	0.02	0				
990 communications interface module		½		1.50	0	0.10	0				
Bell data-set interface kit		½		1.50	0	0.10	0				
990 asynchronous-modem kit		½		0.15	0	0.15	0				
990 synchronous-modem kit		½		0.20	0	0.20	0				
16 I/O EIA data module		½		0.34	0	0.08	0				
16 I/O TTL data module		½		0.53	0	0	0				
D/A converter module, 1 channel		½		0.75	0	0	0				
D/A converter module, 2 channels		½		1.05	0	0	0				
D/A converter module, 3 channels		½		1.55	0	0	0				
D/A converter module, 4 channels		½		1.90	0	0	0				
A/D converter modules, 16 to 64 channels		½		0.95	0	0	0				
32-in transition detection module		1†		0.85	0	0	0				
32-out data module		1†		1.50	0	0	0				
CRU expander card coupler		1		0.90	0	0	0				
TILINE expander card		1		1.20	0	0	0				
Subtotal of Part 2 — to Line B in Part 1											

*These require no power for secondary kit.

†These modules occupy a half slot physically but require the full slot electrically.

Table 24. Blank 990-Chassis Layout Form

	P1 (Chassis Front)			P2 (Chassis Rear)		
Slot Number	Fixed CRU Base Address	Circuit Board	Interrupt Level	Fixed CRU Base Address	Circuit Board	Interrupt Level
1	N/A	990/10 AU2	N/A	N/A	990/10 AU2	N/A
2	02E0	990/10 AU1	N/A	02C0	990/10 AU1	N/A
3	02A0			0280		
4	0260			0240		
5	0220			0200		
6	01E0			01C0		
7	01A0			0180		
8	0160			0140		
9	0120			0100		
10	00E0			00C0		
11	00A0			0080		
12	0060			0040		
13	0020			0000		

Notes:

1. Interrupt levels 3-4 and 6-15 can be assigned via jumpers.
2. Interrupt-level assignments must be known before DX10 sysgen.
3. The main TILINE disk controller should be in slot 7 where TLAG etch is broken. Additional TILINE controllers should be in higher numbered slots.
4. The standard DS990 configuration should be followed as closely as possible to minimize hardware and software modifications.
5. Additional 911 VDT controllers can occupy slots 11-13 if the slots are not otherwise used.
6. The preferred location for a second printer controller is slot 12B if a card reader is not used. If a card reader is used, a second printer controller should be in slot 13A or 13B or in CRU expansion-chassis slot 12A.
7. On systems with a DS31 disk and a TILINE coupler, the DS31 controller should be in slot 10 with interrupt level 11.

For a custom layout, the floppy-disk controller should be located in slot 11 (even in a CRU expansion chassis), the disk controller should be located in slot 7 (even in a TILINE expansion chassis), and so forth.

Refer to the Chassis Slots Required column of the Chassis Planning Form to determine if a given module requires a full or half slot. Each slot occupied by one or two half-slot cards requires a chassis center-card-guide kit.

Appendix G Cabinet Planning

This appendix contains the information necessary for selecting enclosures and planning the placement of the equipment in the enclosures. It includes two major topics: enclosure descriptions and custom-cabinet layout.

Enclosures

DS990 systems are packaged in a coordinated line of enclosures that blend with modern office decor (figure 33). This line includes a single-bay pedestal, a single-bay desk, and a double-bay desk. Each of these units has a rugged, neutral-white work surface with a simulated walnut-grain comfort edge. The frames are textured charcoal gray, and the removable rear and side panels are light gray. Any blank panels are textured white. All of the enclosures feature standard 483-millimetre (19-inch) Electronics Industries Association (EIA) rack-mounting space for easy installation of equipment.

Also available is the 1.78-metre (70-inch) cabinet that can be used in a computer-room environment.

Single-Bay Desk. The single-bay desk has 622 millimetres (24.5 inches) of rack mounting space at the right side. The work surface is 1.37 metres (54 inches) long with a cable-access slot above the equipment bay. A 15-ampere circuit breaker with a protective guard is located in the 432-millimetre (17-inch) kneehole on the side of the equipment bay. The right side and rear panels are slotted to allow movement of cooling air through the equipment bay. A minimum of 305 millimetres (12 inches) of side clearance and 152 millimetres (6 inches) of rear clearance is required for convection cooling of the equipment bay.

The 115-volt ac power is supplied through a 4.6-metre (15-foot) cable that is rated at 20 amperes and has a standard 3-prong connector. The 230-volt system has a recessed locking connector (Hubbel 2625 or equivalent) mounted at the rear of the cabinet. The mating cable-mounting female connector (Hubbel 2623) is also supplied. The user supplies a 230-volt, 7.5-ampere service cable and mounts the connector to the cable.

Double-Bay Desk. The double-bay desk features a 1.82-metre (71.6-inch) work surface and two bays. Each bay has 622 millimetres (24.5 inches) of EIA rack-mounting space. Cable cutouts are provided above each equipment bay, and a cable path between the bays is concealed behind the kneehole. The desk consists of two equipment bays and the center kneehole and power-entry module. A protected 15-ampere circuit breaker in the kneehole provides easy operator control of alternating-current power. A minimum of 305 millimetres (12 inches) of side clearance and 127 millimetres (5 inches) of rear clearance is required for convection cooling of the cabinet.

The 115-volt ac power is supplied through a 4.6-metre (15-foot) cable that is rated at 20 amperes and has a standard 3-prong connector. The 230-volt

Table 25. Blank CRU/TILINE Expansion-Chassis Layout Form

	P1 (Chassis Front)			P2 (Chassis Rear)		
Slot Number	Fixed CRU Base Address	Circuit Board	Interrupt Level	Fixed CRU Base Address	Circuit Board	Interrupt Level
1	N/A	CRU Buffer	N/A	N/A	CRU Buffer	N/A
2	02E0	TILINE Coupler	N/A	02C0	TILINE Coupler	N/A
3	02A0			0280		
4	0260			0240		
5	0220			0200		
6	01E0			01C0		
7	01A0			0180		
8	0160			0140		
9	0120			0100		
10	00E0			00C0		
11	00A0			0080		
12	0060			0040		
13	0020			0000		

Notes:

1. CRU base addresses must be added to expansion-chassis base address as follows:
   First expansion chassis 200_{16}
   Second expansion chassis 400_{16}
2. Standard DS990 configuration should be followed as closely as possible to minimize modifications and speed factory processing.
3. Additional 911 VDT controllers can be installed in slots 11-13 of a CRU expansion chassis.

Standard DS990 Configuration

Slot		
1	990/10 AU2	
2	990/10 AU1	
3	ECC Memory Controller	
4	ECC Memory Array	
5	ECC Memory Array*	
6	911 VDT Controller (5 and 6)*	
7	Disk Controller	
8	Other TILINE Controller*	
9	911 VDT Controller (1 and 2)	
10	911 VDT Controller (3 and 4)*	
11	Floppy-Disk Controller*	
12	Printer*	Card Reader*
13		EIA Terminal*

Standard DS990 Configuration with Combination CRU/TILINE Expansion

Slot				Slot	
1	990/10 AU2		CRU Buffer	1	Available for CRU devices in CRU (only) expansion (slots 1–7)
2	990/10 AU1		TILINE Coupler	2	
3	ECC Memory Controller		ECC Memory Controller	3	
4	ECC Memory Array		ECC Memory Array	4	
5	ECC Memory Array*		ECC Memory Array	5	
6	911 VDT Controller (5 and 6)*		911 VDT Controller (13 and 14)	6	
7	Disk Controller		TILINE Controller	7	
8	TILINE Coupler		TILINE Controller	8	Available for TILINE devices in TILINE (only) expansion (slots 8–13)
9	911 VDT Controller (1 and 2)		911 VDT Controller (7 and 8)	9	
10	911 VDT Controller (3 and 4)*		911 VDT Controller (9 and 10)	10	
11	CRU Expander		Floppy-Disk Controller	11	
12	Printer*	Card Reader*	911 VDT Controller (11 and 12)	12	
13		EIA Terminal*		13	

*Options to basic DS990 system

Figure 32. Standard Chassis Layout

system has a recessed locking connector (Hubbel 2625 or equivalent) mounted at the rear of the cabinet. The mating cable-mounting female connector (Hubbel 2623) is also supplied. The user supplies a 230-volt, 7.5-ampere service cable and mounts the connector to the cable.

Single-Bay Pedestal. Single-bay pedestal has 622 millimetres (24.5 inches) of EIA rack-mounting space. A cable slot at the rear of the work surface provides an unobtrusive cable route from desktop equipment (such as a display terminal) to rack-mounted equipment. A cable plug fills the slot if desktop-to-bay cabling is unnecessary.

The side and rear panels are slotted to allow sufficient movement of cooling air through the equipment bay. A minimum of 305 millimetres (12 inches) of side clearance and 152 millimetres (6 inches) of rear clearance is required for convection cooling of the equipment bay. Two pedestals can be placed side-by-side if the intervening side panels are removed to allow unrestricted air flow.

The 115-volt ac power is supplied through a 4.6-metre (15-foot) cable that is rated at 20 amperes and has a standard 3-prong connector. The 230-volt system has a recessed locking connector (Hubbel 2625 or equivalent) mounted at the rear of the cabinet. The mating cable-mounting female connector (Hubbel 2623) is also supplied. The user supplies a 230-volt, 7.5-ampere service cable and mounts the connector to the cable.

Rack-Mounting Cabinet. For computer-room environments, Texas Instruments offers a functional rack-mounting cabinet with 1.60 metres (63 inches) of standard 483-millimetre (19-inch) EIA rack-mounting space. The removable door panels are light gray, and the frame is charcoal gray.

Cooling air is drawn into the cabinet through a large, washable aluminum filter on the rear door and is exhausted by an enclosed 600-cubic-feet-per-minute fan at the top of the cabinet. A minimum of 152 millimetres (6 inches) of rear clearance is required for proper air flow; a minimum of 152 millimetres (6 inches) of clearance above the exhaust fan also is required.

The cabinet can exhaust approximately 2000 watts or 6826 Btu/hour of heat with a clean air filter. When configuring cabinet layout, consider the obstructions that create dead air space, the start-up power for disk drives, and the maximum heat load within the enclosure.

Cables and ac power enter the cabinet through a panel on the rear of the cabinet. The signal-cable entry has a built-in adjustable strain-relief clamp. Ac power enters the cabinet via a recessed twist-lock connector and a 2.5-metre (10-foot) heavy-duty 3-wire power cable, requiring a National Electrical Manufacturers Association (NEMA) 5-20 power outlet. Power is controlled by a 20-ampere circuit breaker on the power-input panel. The ac power distribution strip is mounted in the cabinet. The circuit breaker allows a 200 percent overload for 2.5 seconds.

The 230-volt option replaces the single breaker with a dual breaker and replaces the 115-volt power cable with a recessed Hubbel 2625 locking connector. The customer must furnish a heavy-duty 3-wire supply cable with a Hubbel 2623 locking connector.

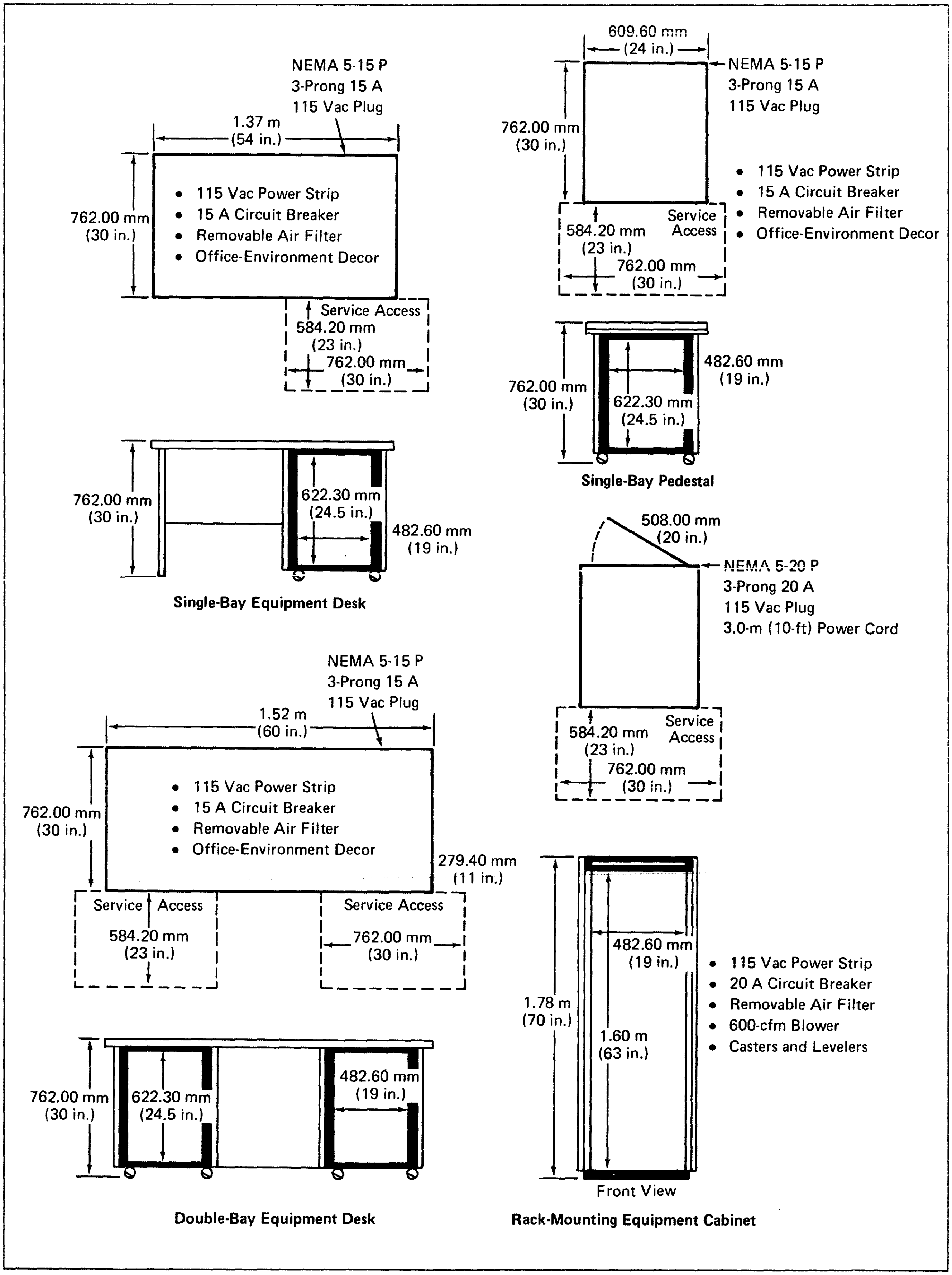


Figure 33. DS990 Enclosures

A 610-millimetre (24-inch) arc of clear area behind the cabinet is required to open the rear door. With casters installed on the cabinet, the bottom of the door is 133-millimetres (5.25 inches) above the floor. If the rear door is not to be opened, a smaller clearance can be provided. The standard front-to-rear mounting dimension for rails is 610 millimetres (24 inches).

Disk-Mounting Pedestal. The DS10 disk drive can be rack mounted in a desk enclosure, or it can be mounted in a freestanding "quietized" pedestal, as shown in figure 34. The pedestal mount offers several advantages including (1) easy access to the top-loading disk drive at a convenient height, (2) reduction in audible noise due to the acoustic-suppression materials in the pedestal, and (3) freeing of desk-enclosure space for mounting other equipment.

The DS25 and DS50 disk drives are usually supplied in a freestanding pedestal, as shown in figure 35. The pedestal places the top-loading drive at a convenient height for rapid disk changes.

Cabinet-Layout Worksheet

If user requirements cannot be met by a standard configuration, custom cabinet planning will be necessary. Figure 36 is a cabinet-layout worksheet. If additional assistance is needed, consult a Texas Instruments sales engineer.

Using the Worksheet. To plan a custom cabinet, perform the following steps:

1. Use a separate copy of the form for each cabinet.
2. List equipment in desired locations in the rack. Tick marks are at 1.75-inch intervals. (Texas Instruments standard equipment rack-mounting requirements, heat load, ac power, and starting current are shown in table 35.)
3. Verify that each item is at a convenient working height.
4. Attach a copy of the cabinet layout to the system purchase order. Texas Instruments will configure the cabinet as indicated. On the 1.78-metre (70-inch) rack-mounting cabinet, blank panels are supplied at no charge. (System, enclosure, and installation must be on one purchase order.)

Considerations in Cabinet Layout. In general, the cabinet layout should be designed for operator convenience. The CPU, TILINE, and CRU chassis should not be mounted in the left bay of a double-bay desk. There are no vents in the desk kneehole for the exhaust from these chassis.

The TILINE chassis should always be adjacent to the CPU in the same bay. This minimizes cable length, and therefore transfer time, between chassis.

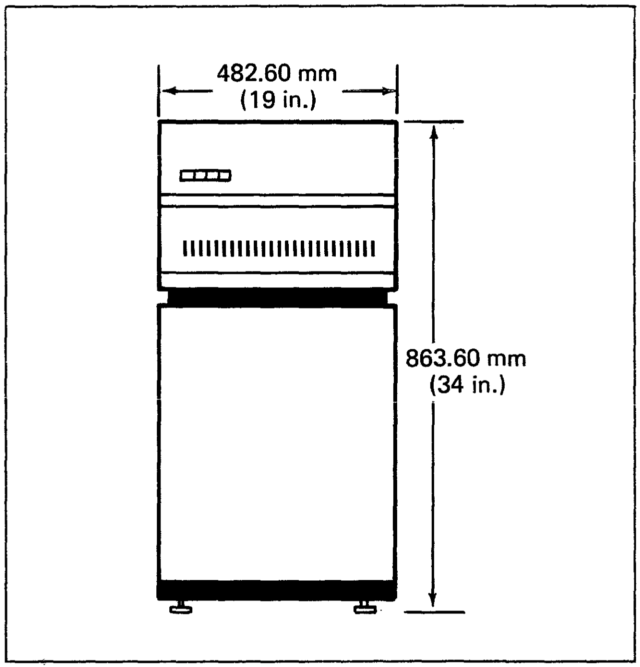


Figure 34. Dimensional Outline of Model DS10 Disk on "Quietized" Pedestal

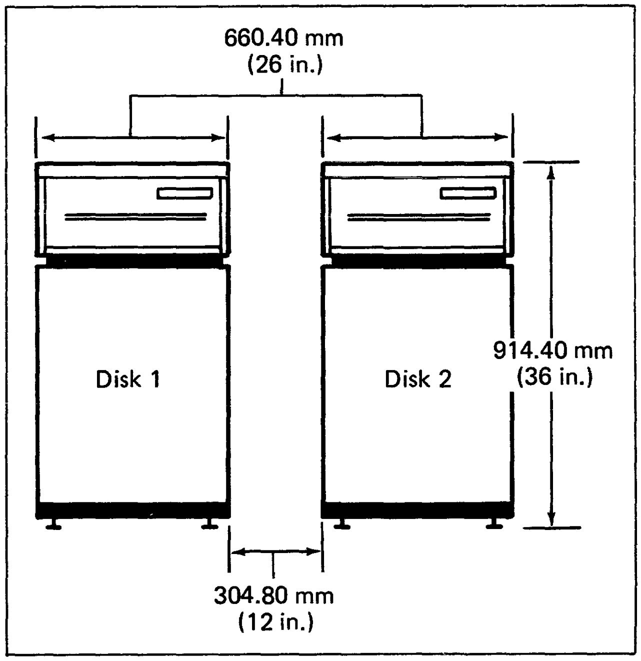


Figure 35. Dimensional Outline of DS25 or DS50 Disk on Pedestal

Also, the CPU, TILINE, and CRU chassis have a contoured front panel, which offers the best appearance when these chassis are adjacent.

The first disk drive should be as close to the CPU chassis as possible (after the TILINE and CRU expansion chassis) to minimize the cable length. Where possible, rack-mounted disks should be placed in the same enclosure. Pedestal-mounted disks should be immediately adjacent to the CPU enclosure with a 305-millimetre (12-inch) air space to the right side of the desk enclosure. Where multiple disk types are

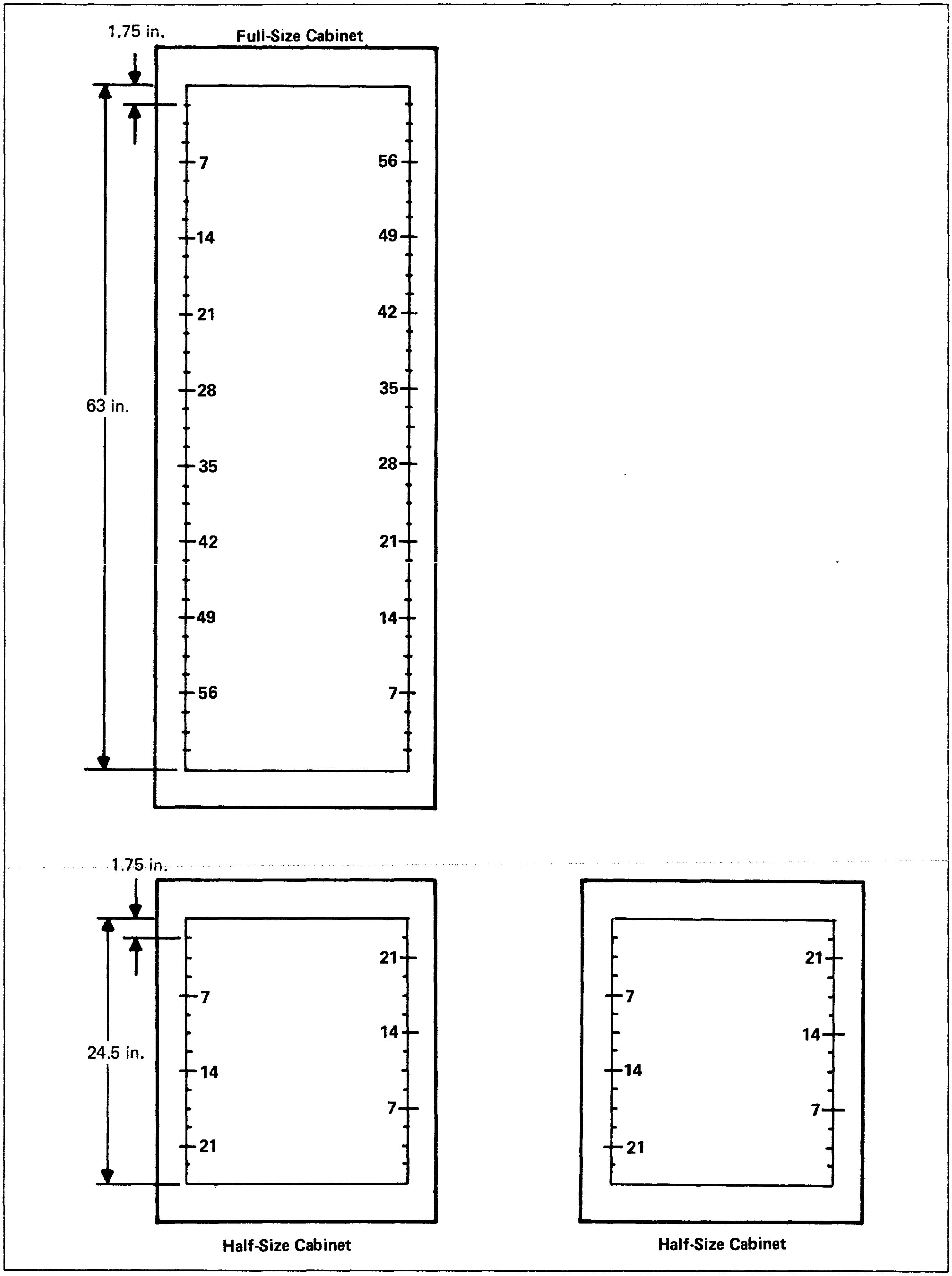


Figure 36. Cabinet-Layout Worksheet

present (DS31, DS10, DS25, or other), the first disk drive of each type should be as close to the CPU as possible.

The 979A magnetic-tape transports should be mounted in the top of a 1.78-metre (70-inch) cabinet. This is most convenient to the operator. Also, standard Texas Instruments maintenance rates assume this mounting for maintenance access.

The DS31-disk and floppy-disk drives are front-loaded and should be mounted at the top of desk-height enclosures or near the center of 1.78-metre (70-inch) racks. Desk or rack mounting is preferred over tabletop mounting for the floppy-disk drive.

Appendix H
DS990 System Kit Contents

This appendix consists of a series of tables that describes the basic DS990 system kits, including hardware, software, and manuals. These tables do not include part numbers. For additional information, refer to the *990 Computer Family Catalog.*

Table 26 gives a description for each of the basic DS990 systems. Variables in this table are the DS990 model number (Model 4, Model 6, or Model 8 System), enclosure type (single-bay, double-bay, single pedestal, or 1.78-metre (70-inch) cabinet), primary power (100 V, 50 Hz or 100 V, 60 Hz or 115 V, 60 Hz or 230 V, 50 Hz) and version (hardware/software or hardware only).

Table 27 is a consolidated parts list for DS990 Model 4 Systems. Table 28 is a consolidated parts list for Model 6 and Model 8 Systems. Table 29 is a consolidated parts list for the DS990 Model 4 Commercial System. Table 30 is a consolidated parts list for the DS990 Model 6 and Model 8 Commercial Systems.

Each system contains a software package. The software package for hardware/software versions contains all software and hardware documentation, DX10 operating-system software on the appropriate disk cartridge or disk pack, and blank disk cartridges or packs. The software package for hardware-only versions contains all hardware documentation, limited software documentation, and blank disk cartridges or packs. Tables 31 and 32 list the contents of these two packages. Note that the tables describe only the basic system complement (excluding options such as COBOL, FORTRAN, BASIC, RPG II, DBMS 990, and Pascal).

Software options are described in table 33. This table gives the software licenses and specifies the media on which the object code is supplied. Ownership and all rights (except those granted under the license agreement) are retained by Texas Instruments. Hardware options are given in table 34. For additional information, refer to the *990 Computer Family Price List.*

Appendix I
Equipment Specifications

Table 35 is a summary of standard DS990 equipment specifications that are relevant to site preparation and planning. Data supplied by the table includes (1) input ac-power requirements (root-mean-square (RMS) magnitude, frequency, and power consumption), (2) air-conditioning data (heat dissipation, operating temperature range, and operating humidity range), and (3) physical planning data (weight, chassis dimension, and cable length).

Required physical clearances for equipment access and unobstructed cooling air flow are described with the enclosure options. Figure 37 summarizes these space requirements.

Appendix J
Packing and Shipping

Basic DS990 systems are shipped with the desk-mounted equipment installed. Texas Instruments shipping requirements for these systems specify the use of "air-ride" vans for domestic shipping. The rugged packing crate/shipping pallet used for DS990 systems is shown in figure 38. Dimensions and weights are noted on the figure. Internal arrangement of the packing crate is shown in figure 39 (single-bay desk) and figure 40 (double-bay desk).

An Open Me First packet is attached to the exterior of each DS990 desk-system crate. The contents of this packet should be carefully read before the crate is unpacked. Contents of the packet include (1) *DS990 Installation and Service-Request Information Sheet,* (2) *Digital Systems Division Field Service Brochure,* (3) *DS990 Desk Enclosure System Unpacking/Installation Procedure,* and (4) reminder of the customer's legal responsibilities with respect to Texas Instruments licensed software.

The packing of separate disk drives can vary because the vendor's packing is used. Figures 41 and 42 are typical packing configurations for a rack-mounted disk drive and a pedestal-mounted disk drive, respectively. The shipping weight is approximately 91 kilograms (200 pounds) for the DS10 disk drive and 127 kilograms (280 pounds) for the DS25/DS50 disk drive. The shipping weight with the pedestal is 181 kilograms (400 pounds) for the DS10 disk drive and 191 kilograms (420 pounds) for the DS25/DS50 disk drive.

Table 26. Basic DS990 System Kit Configurations

System	Description
DS990 Model 4 System	Single-bay desk kit with software (115 V, 60 Hz)
	Single-bay desk kit with software (230 V, 50 Hz)
	Single-bay desk kit with software (100 V, 50 Hz)
	Single-bay desk kit with software (100 V, 60 Hz)
	Single-bay desk kit, hardware only (115 V, 60 Hz)
	Single-bay desk kit, hardware only (230 V, 50 Hz)
	Single-bay desk kit, hardware only (100 V, 50 Hz)
	Single-bay desk kit, hardware only (100 V, 60 Hz)
	Double-bay desk kit with software (115 V, 60 Hz)
	Double-bay desk kit with software (230 V, 50 Hz)
	Double-bay desk kit with software (100 V, 50 Hz)
	Double-bay desk kit with software (100 V, 60 Hz)
	Double-bay desk kit, hardware only (115 V, 60 Hz)
	Double-bay desk kit, hardware only (230 V, 50 Hz)
	Double-bay desk kit, hardware only (100 V, 50 Hz)
	Double-bay desk kit, hardware only (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with software (115 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with software (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with software (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with software (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit, hardware only (115 V, 60 Hz)
	1.78-m (70-in.) cabinet kit, hardware only (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit, hardware only (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit, hardware only (100 V, 60 Hz)
	762-mm (30-in.) cabinet kit with software (115 V, 60 Hz)
	762-mm (30-in.) cabinet kit with software (230 V, 50 Hz)
	762-mm (30-in.) cabinet kit with software (100 V, 50 Hz)
	762-mm (30-in.) cabinet kit with software (100 V, 60 Hz)
	762-mm (30-in.) cabinet kit, hardware only (115 V, 60 Hz)
	762-mm (30-in.) cabinet kit, hardware only (230 V, 50 Hz)
	762-mm (30-in.) cabinet kit, hardware only (100 V, 50 Hz)
	762-mm (30-in.) cabinet kit, hardware only (100 V, 60 Hz)
DS990 Model 6 System	Single-bay desk kit with two DS25 disk systems and software (115 V, 60 Hz)
	Single-bay desk kit with two DS25 disk systems and software (230 V, 50 Hz)
	Single-bay desk kit with two DS25 disk systems and software (100 V, 50 Hz)
	Single-bay desk kit with two DS25 disk systems and software (100 V, 60 Hz)
	Single-bay desk kit with two DS25 disk systems, hardware only (115 V, 60 Hz)
	Single-bay desk kit with two DS25 disk systems, hardware only (230 V, 50 Hz)
	Single-bay desk kit with two DS25 disk systems, hardware only (100 V, 50 Hz)
	Single-bay desk kit with two DS25 disk systems, hardware only (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems and software (115 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems and software (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems and software (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems and software (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems, hardware only (115 V,60 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems, hardware only (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems, hardware only (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS25 disk systems, hardware only (100 V, 60 Hz)

Table 26. Basic DS990 System Kit Configurations (Continued)

System	Description
DS990 Model 8 System	Single-bay desk kit with two DS50 disk systems and software (115 V, 60 Hz)
	Single-bay desk kit with two DS50 disk systems and software (230 V, 50 Hz)
	Single-bay desk kit with two DS50 disk systems and software (100 V, 50 Hz)
	Single-bay desk kit with two DS50 disk systems and software (100 V, 60 Hz)
	Single-bay desk kit with two DS50 disk systems, hardware only (115 V, 60 Hz)
	Single-bay desk kit with two DS50 disk systems, hardware only (230 V, 50 Hz)
	Single-bay desk kit with two DS50 disk systems, hardware only (100 V, 50 Hz)
	Single-bay desk kit with two DS50 disk systems, hardware only (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems and software (115 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems and software (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems and software (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems and software (100 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems, hardware only (115 V, 60 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems, hardware only (230 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems, hardware only (100 V, 50 Hz)
	1.78-m (70-in.) cabinet kit with two DS50 disk systems, hardware only (100 V, 60 Hz)

Table 27. DS990 Model 4 System Consolidated Parts List

Description	Quantity
990/10 CPU with mapping in 13-slot chassis with programmer front panel and 40 A power supply	1
990/10 disk ROM loader	1
990/10 memory controller (64K bytes)	1
990/10 memory array (64K bytes)	1
990/10 memory interconnect	1
DS10 master kit	1
Model 911 VDT dual controller (1 tube)	1
Chassis slide kit	1
44-mm (1.75-in.) blank panel	1
Center card guide	2
Single-bay desk or 762-mm (30-in.) cabinet or 1.78-m (70-in.) cabinet or Dual-bay desk	1
DX10 software (if applicable)	1
Installation (if applicable)	1

Appendix K Documentation

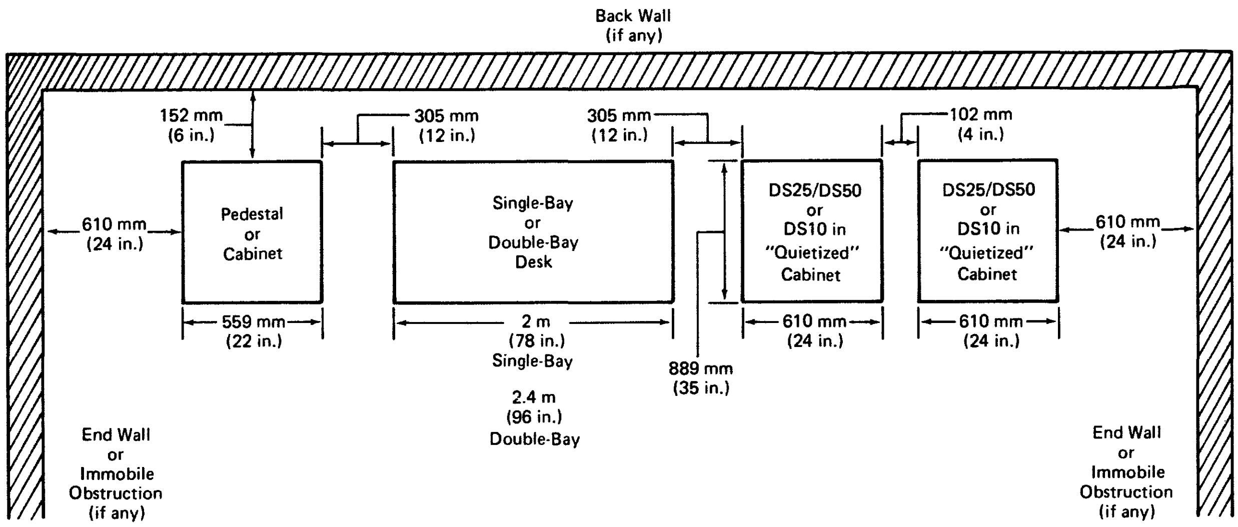
All 990 computer-family products are fully documented in hardware and software manuals.

Hardware Manuals

The 990 computer-family hardware is fully documented in one or more of three types of manuals: installation and operation, field maintenance, and depot maintenance manuals. Table 36 is a complete listing of hardware manuals for DS990 systems, standard options, and other 990 family catalog items.

Installation and Operation Manual. An installation and operation manual presents specific information regarding the correct procedures and site preparation required for the successful installation of the hardware peripheral. An overview of the physical, electrical, and operational characteristics of the peripheral (including both the hardware device and its associated computer interface) are included. In addition, the software requirements are detailed to assist the customer who intends to develop custom application-software drivers as opposed to using the standard Texas Instruments operating-system(s) device service routine(s).

Installation and operation manuals are included in each hardware-device kit with the following exceptions. The 990 memory boards and 990/10



Notes:

1. These are rough minimums.
2. Room should be allowed to pull the equipment approximately 1000 millimetres (39 inches) from the wall for service access.
3. Measurements to the end wall are with respect to the last equipment unit on that end of the DS990 system.

Figure 37. Minimum Equipment Spacing for DS990 Systems

arithmetic-unit boards are covered in the *Model 990/10 Computer System Hardware Reference Manual.* This manual is supplied in any shipment that includes a 990/10 minicomputer. TILINE coupler and CRU expander and buffer boards also are covered in the *Model 990/10 Computer System Hardware Reference Manual.* The synchronous and asynchronous modems and the communications interface module are combined in the *Model 990 Computer Communications System Installation and Operation Manual.* This manual is supplied with the communications interface module.

Field Maintenance Manual. The field maintenance manual outlines the preventive maintenance procedures required to maintain the computer and/or peripherals in good operating condition and presents equipment-malfunction troubleshooting techniques. In general, the maintenance procedures resolve equipment problems at the assembly or board-swap level.

Field maintenance manuals are separately purchased items. Customers who plan to do their own field-level maintenance also should have *Model 990 Computer Family Maintenance Drawings, Volume I — Processors* and *Volume II — Peripherals.*

Depot Maintenance Manual. Depot maintenance manuals present detailed electrical and mechanical data to allow circuit-level diagnosis and resolution of equipment malfunctions. These manuals are separately purchased items. Customers who plan to do depot-level maintenance in their own facility should carefully consider the amount of special- and general-purpose test equipment required. This information is included in the depot maintenance manuals. Customers who plan special-purpose modifications to Texas Instruments equipment should have copies of the depot maintenance manuals because Texas Instruments maintenance and warranty agreements do not cover customer modifications.

Customers who are interested in equipment details at the depot-maintenance-level also should have *Model 990 Computer Family Maintenance Drawings, Volume I — Processors* and *Volume II — Peripherals.*

Software Manuals

The 990 computer-family software is fully documented in one or more of the following types of manuals: system operation guide, user guide, programmer's guide, reference card, and installation procedure. At least one complete set of applicable manuals is supplied with each software package. Table 37 is a complete listing of manuals for software products that are compatible with the DX10 operating system.

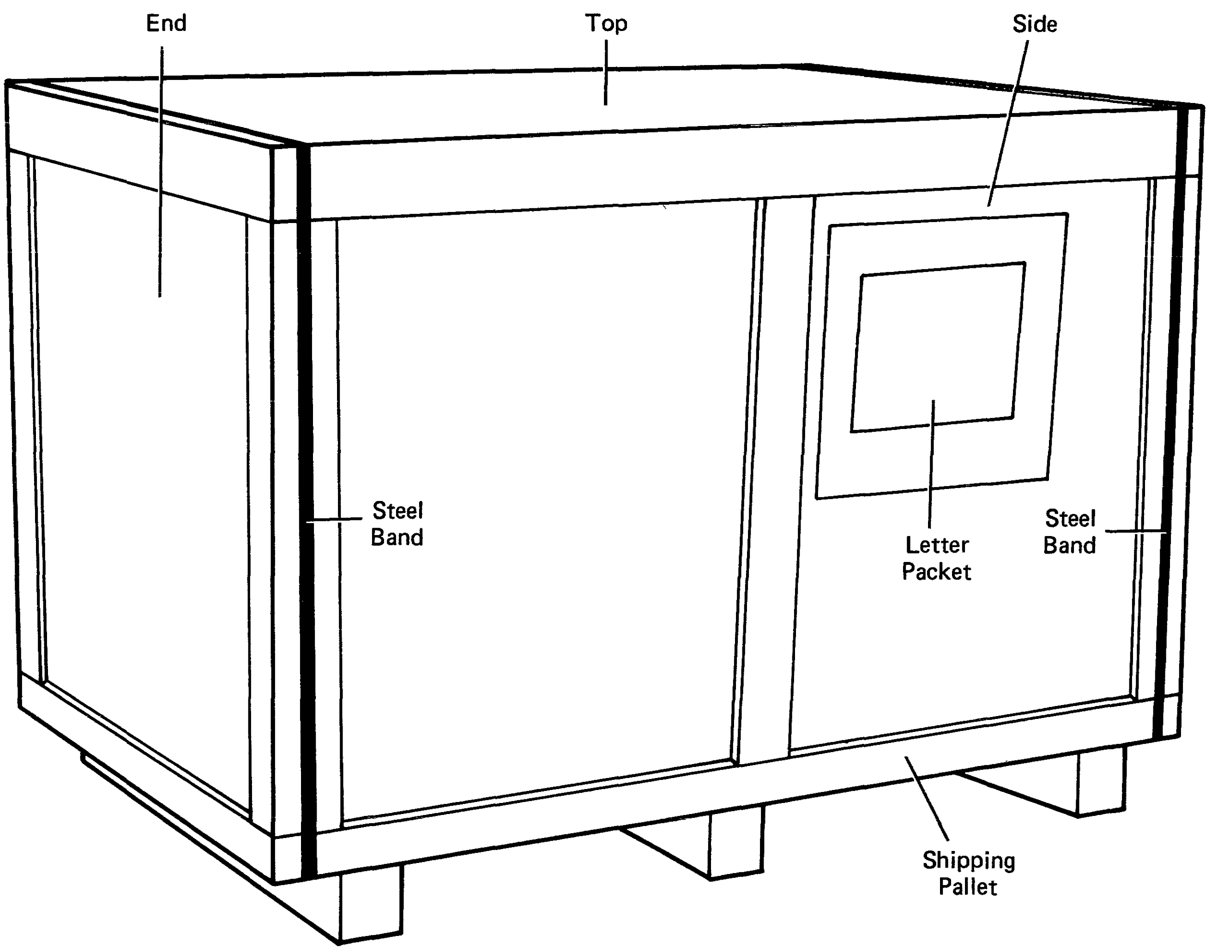
System Operation Guide. A system operation guide is provided with each of the packaged systems for the 990 computer family. This manual links the hardware components of the system with the software that accompanies it and describes the concepts required to effectively use the system. This information includes installation instructions, procedures for verifying that the system is operating effectively, plus operating instructions for using each of the software packages in conjunction with the hardware included with the system.

User Guide. A user guide provides information about individual software packages that are not ordinarily used in conjunction with a specific system. Each guide contains a description of the functions and capabilities of the package as well as detailed instructions for effectively using the package.

Programmer's Guide. A programmer's guide provides complete, detailed coverage concerning an operating system or programming language. These guides provide all the information an experienced user requires to interface with the 990 computer family through the subject medium.

Reference Card. The handy pocket-sized reference card condenses the essential information necessary to program the computer. These cards list the instruction set and give formats for the different instructions as well as summarize other helpful concepts.

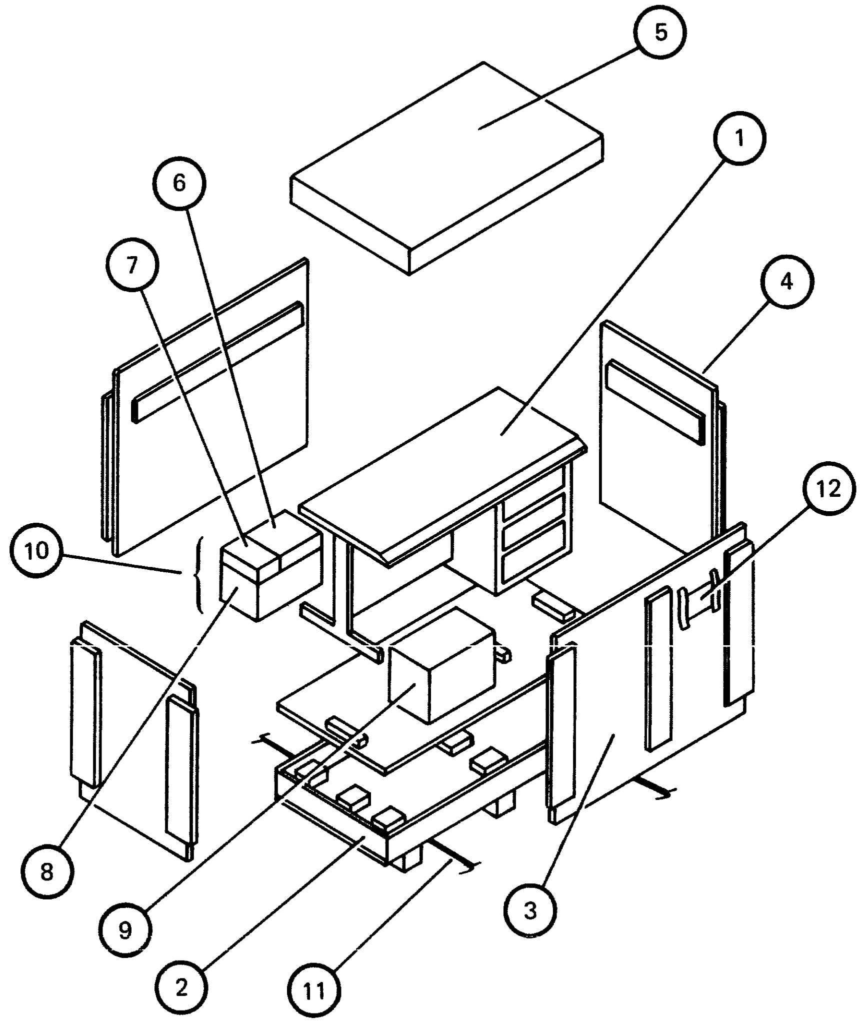
Installation Procedure. The installation procedure supplies the information necessary to initially install a software package into a system.



Dimensions: 1022 mm (40.25 in.) Height
978 mm (38.5 in.) Width
1.49 m (58.5 in.) Length (Single-Bay Desk)
1.90 m (74.75 in.) Length (Double-Bay Desk)

Weight: 250 kg (550 lbs) Single-Bay Desk
295 kg (650 lbs) Double-Bay Desk

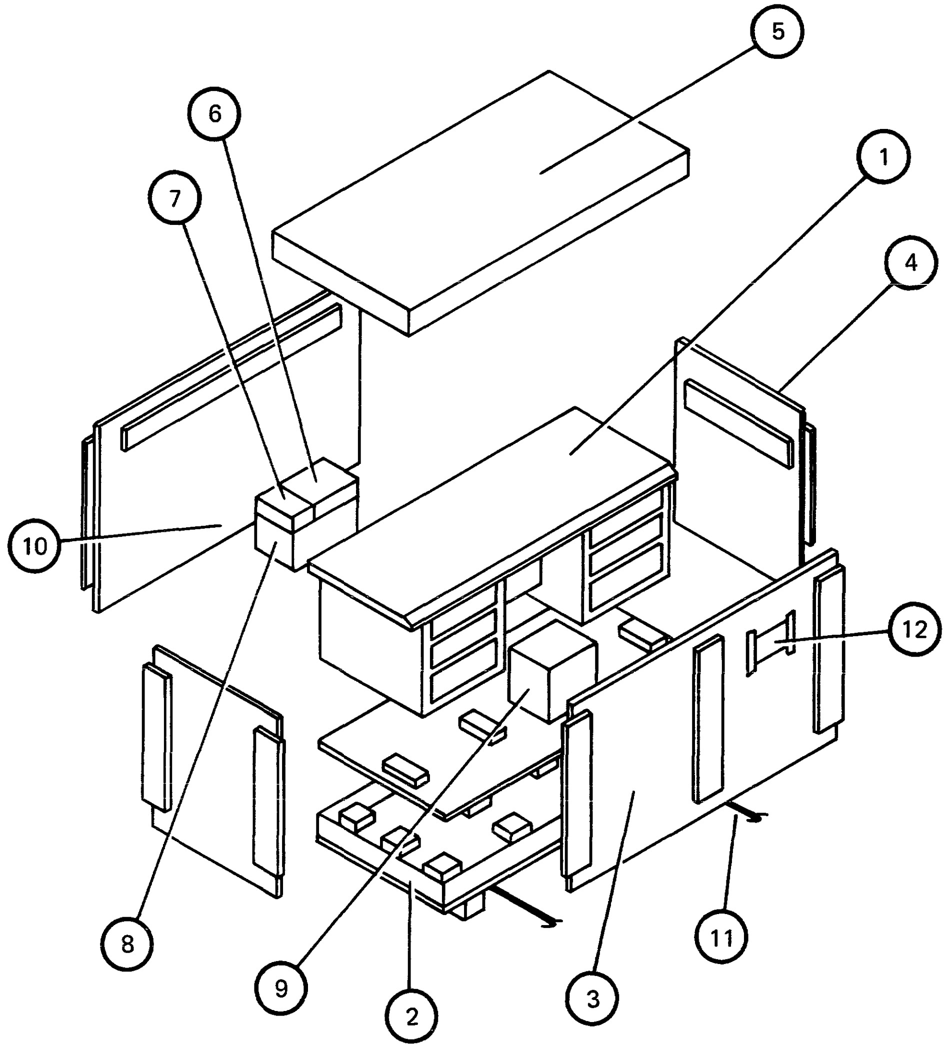
Figure 38. DS990 Desk-System Packing Crate



Legend:

1. Single-bay desk assembly
2. Bottom, container
3. Side, container
4. End, container
5. Top, container
6. Miscellaneous options
7. Software options
8. Keyboard options
9. VDT options
10. Avis strap
11. Strapping, steel
12. SP, letter packet

Figure 39. Exploded View of Single-Bay-Desk Packing Crate



Legend:
1. Dual-bay desk assembly
2. Bottom, container
3. Side, container
4. End, container
5. Top, container
6. Miscellaneous options
7. Software options
8. Keyboard options
9. VDT options
10. Avis strap
11. Strapping, steel
12. SP, letter packet

Figure 40. Exploded View of Double-Bay-Desk Packing Crate

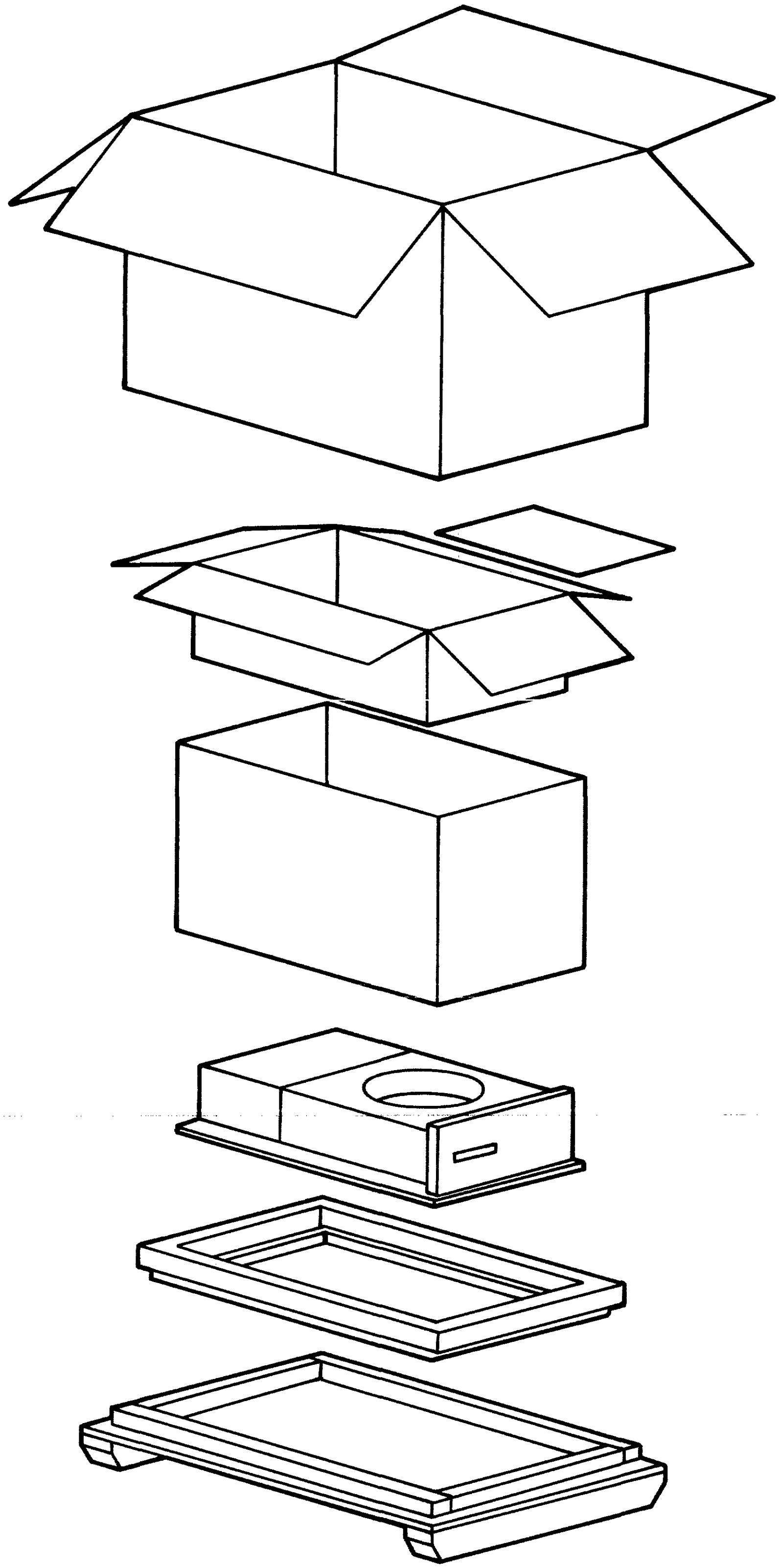


Figure 41. Typical Rack-Mounted-Disk-Drive Shipping Configuration

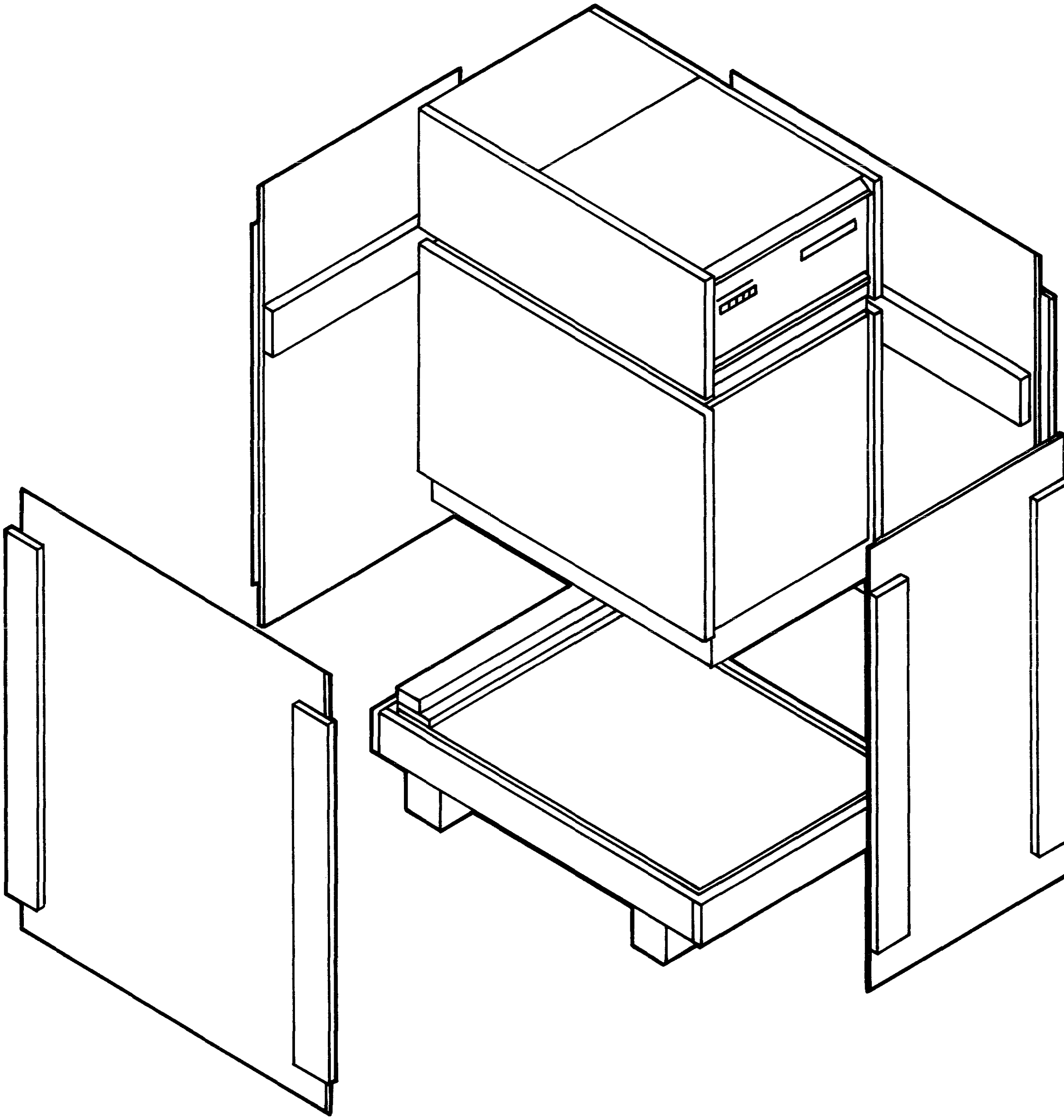


Figure 42. Typical Pedestal-Mounted-Disk-Drive Shipping Configuration

Table 28. DS990 Model 6 and Model 8 Systems Consolidated Parts List

Description	Quantity
990/10 CPU with mapping in 13-slot chassis with programmer panel and 40 A power supply	1
990/10 disk loader	1
990/10 memory controller (64K bytes)	1
990/10 memory array (64K bytes)	1
990/10 memory interconnect	1
DS25 master kit	1
DS25 secondary kit or DS50 master kit	1
DS50 secondary kit	1
Model 911 VDT dual controller (1 tube)	1
Chassis slide kit	1
44-mm (1.75-in.) blank panel	1
266-mm (10.5-in.) blank panel	1
Center card guide	2
Single-bay desk or 1.78-mm (70-in.) cabinet	1
DX10 software (if applicable)	1
Installation (if applicable)	1

Table 29. DS990 Model 4 Commercial System Consolidated Parts List

Description	Quantity
DS990 Model 4 System	1
Model 911 VDT monitor and keyboard	1
Model 810 Printer	1
Sort/Merge	1
COBOL or RPG II	1
Single-bay desk or Dual-bay desk or 1.78-m (70-in.) cabinet or 762-mm (30-in.) cabinet	1

Table 30. DS990 Model 6 and Model 8 Commercial System Consolidated Parts List

Description	Quantity
DS990 Model 6 or Model 8 System	1
Model 911 VDT monitor and keyboard	1
Model 911 VDT dual controller kit	1
Model 2230 Line Printer kit	1
32K extra bytes on memory controller	1
Sort/Merge	1
COBOL	1
DBMS 990	1
Single-bay desk or 1.78-m (70-in.) cabinet	1

Table 31. DS990 Basic Software-Package Contents for Hardware/Software Versions

Description	Quantity
Three-ring binders for system documentation	7
Blank T25/50 disk pack (Model 6 and Model 8 Systems)	2
or	
Blank 5440-type disk cartridge (Model 4 System)	1
DX10 operating-system object code on either 5440-type disk cartridge for Model 4 System or on T25/50 disk pack for Model 6 and Model 8 Systems	1
990/TMS 9900 assembly-language coding-form pads	2
Software-trouble report-form pad	1
DX10 software-subscription registration form (Model 4 System)*	1
or	
DX10 software-subscription registration form (Model 6 and Model 8 Systems)*	1
Diagnostic-operational-control-system (DCOS) diagnostic object code on either 5440-type disk cartridge for Model 4 System or on T25/50 disk pack for Model 6 and Model 8 Systems	1
990 hardware demonstration test (HDT)	1
990 diagnostic operating procedure	1
Model 990 Computer Diagnostic Handbook	1
Model 990 Computer DS990 Disk System Installation and Operation	1
Model 990 Computer Model 911 Video Display Terminal Installation and Operation	1
Model 990 Computer DX10 Operating System Installation Guide	1
Model 990 Computer DX10 Operating System Release 3 Program Description Documents	2
Model 990 Computer DX10 Operating System Release 3 Reference Manual	
Volume I — Concepts and Facilities	2
Volume II — Production Operation	2
Volume III — Application Programming Guide	2
Volume IV — Developmental Operation	2
Volume V — System Programming Guide	2
Volume VI — Error Reporting and Recovery	2
Model 990 Computer TMS 9900 Microprocessor Assembly Language Programmer's Guide	2
Model 990 Computer Link Editor Reference Manual	2
990 Computer Family Systems Handbook	1
Model 990 Computer Model DS10 Disk System Installation and Operation (Model 4 System only)	1
Model 990 Computer Model DS25/DS50 Disk System Installation and Operation (Model 6 and Model 8 Systems only)	1

*The customer must fill out and sign the DX10 software license agreement and subscription registration forms and return them to Texas Instruments within ten days of receipt of the DX10 software.

Table 32. DS990 Basic Software-Package Contents for Hardware-Only Versions

Description	Quantity
Three-ring binders for system documentation	2
Blank 5440-type disk cartridge (Model 4 System) or	1
Blank T25/50 disk pack (Model 6 and Model 8 Systems)	2
990/TMS 9900 assembly-language coding-form pads	2
Software-trouble report-form pad	1
Model 990 Computer DS990 Disk System Installation and Operation	1
Model 990 Computer Model 911 Video Display Terminal Installation and Operation	1
Model 990 Computer TMS 9900 Microprocessor Assembly Language Programmer's Guide	1
Model 990 Computer Model DS10 Disk System Installation and Operation (Model 4 System only)	1
Model 990 Computer System Model DS25/DS50 Disk System Installation and Operation (Model 6 and Model 8 Systems)	1

Table 33. Optional DS990 Software, Object Format

Description
DX10 FORTRAN License* (Category B)**
DX10 COBOL License* (Category B)
DX10 BASIC License (Category A)
DX10 Business BASIC License (Category A)
DX10 RPG II License* (Category B)
DX10 Sort/Merge License (Category A)
DX10 Pascal License* (Category B)
DX10 DBMS 990 License (Category A)
Media
The following media supports the software listed:
DS31
DS25
DS50
Tape, 800 bpi (800 b/25.4 mm)
Tape, 1600 bpi (1600 b/25.4 mm)
DS10
DS25/DS50 Add On†

*The run-time portion requires a category A software license.

**All licensed software remains the property of Texas Instruments. Payment of a license fee and execution of a license agreement allows the customer to use the software. Licenses fall into two categories. Category A software license is purchased once, and there is no additional charge for use on additional CPUs. Sublicensing fees are not charged for sublicensed category A programs. Category B software license must be purchased once for every CPU on which the software will reside. Sublicense charges equal to the initial license fee are charged on each sublicensed program. Unlicensed software is purchased without licensing restrictions. No subscription service is provided. However, updates can be purchased.

†Two or more software products are supplied on the same disk pack.

Table 34. DS990 Hardware Options

Category	Description
990 Memories	Error-checking-and-correcting memory controller, 64K bytes Error-checking-and-correcting memory controller, 96K bytes Error-checking-and-correcting memory array, 64K bytes Error-checking-and-correcting memory array, 128K bytes Error-checking-and-correcting memory array, 192K bytes Error-checking-and-correcting memory array, 256K bytes
Mass Storage	FD800 single-floppy master kit FD800 dual-floppy master kit FD800 single-floppy secondary kit FD800 dual-floppy secondary kit FD800 rack-mounting slide set FD800 tabletop cover
	DS31 single master kit with power supply DS31 secondary kit with power supply DS31 secondary kit
	DS10 single master kit (rack-mounting cabinet) DS10 secondary kit (rack-mounting cabinet) DS10 single master kit ("quietized" pedestal cabinet) DS10 secondary kit ("quietized" pedestal cabinet)
	DS25 master kit DS25 secondary kit DS50 master kit DS50 secondary kit DS25/DS50 mounting pedestal
	DS200 master kit DS200 secondary kit
	979A master kit, 800 bpi (800 b/25.4 mm) 979A secondary kit, 800 bpi (800 b/25.4 mm) 979A master kit, 800/1600 bpi (800/1600 b/25.4 mm) 979A secondary kit, 1600 bpi (1600 b/25.4 mm)
Peripherals	Model 911 VDT kit, 1920-character, single controller, 1 display and keyboard Model 911 VDT kit, 1920-character, dual controller, 1 display and keyboard Model 911 VDT kit, 1920-character, dual controller, 2 displays and keyboard 911 controller-display extension cable, 15.2 m (50 ft.) 911 controller-display extension cable, 30.5 m (100 ft.) Model 911 VDT expansion display and keyboard
	Model 743 KSR master kit Model 733 ASR master kit
	Model 810 Printer master kit Model 810 Printer extension cable, 15.2 m (50 ft.) Model 810 Printer extension cable, 30.5 m (100 ft.)
	Model 2230 Line Printer master kit Model 2260 Line Printer master kit
	Model 804 Card Reader master kit

Note: Table 34 is continued on page 98.

Table 35. Standard DS990 Equipment Specifications

Item	Input Voltage Options	115 V Power Consumption (Steady State)	Heat Dissipation	Operating Temperature Range
DS990 Model 4 System				
Basic System with single-bay desk (no options)	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	1115 W*	3806 Btu/hr*	16°C-32°C*
Basic System with double-bay desk (no options)	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	1115 W*	3806 Btu/hr*	16°C-32°C*
Basic System with 1.78-m (70-in.) cabinet	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	1115 W*	3806 Btu/hr*	16°C-32°C*
Basic System with 762-mm (30-in.) cabinet	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	1115 W*	3806 Btu/hr*	16°C-32°C*
DS990 Model 6 and Model 8 Systems				
Basic System with single-bay desk (no options)	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	2013 W*	6872 Btu/hr*	16°C-32°C*
Basic System with 1.78-m (70-in.) cabinet	115 V, 60 Hz 230 V, 50 Hz 100 V, 50 Hz 100 V, 60 Hz	2013 W*	6872 Btu/hr*	16°C-32°C*
Model 990 Computer 13-Slot Chassis with 40 A power supply	115/230 V, 50/60 Hz 100/200 V, 50/60 Hz	560 W (700 VA) maximum	1911 Btu/hr	0°C-50°C†
Model 911 Video Display Terminal	100 V, 50/60 Hz 115 V, 50/60 Hz 220 V, 50/60 Hz	115 W	393 Btu/hr	0°C-40°C
Model DS10 Disk Drive	100 V, 50/60 Hz 115 V, 50/60 Hz 220 V, 50/60 Hz	440 W††	1502 Btu/hr	16°C-32°C

*These specifications are based on 4K-RAM technology.

**Weight specifications will be available in the next revision of this manual.

***This weight does not include the shipping weight of two Model 6 or Model 8 System disk drives.

†The upper limit is decreased 2° C for every 762 metres (2500 feet) of altitude.

Operating Humidity Range (Noncondensing)	Weight	Chassis Dimensions (H x D x W)	115 V Power Cable Length	990 Interface-to-Device Cable Length
10-80%*	250 kg (550 lbs) in crate	784.86 mm x 863.60 mm x 1.37 m (30.9 in. x 34 in. x 54 in.)	4.6 m (15 ft)	N/A
10-80%*	295 kg (650 lbs) in crate	784.86 mm x 863.60 mm x 1.82 m (30.9 in. x 34 in. x 71.6 in.)	4.6 m (15 ft)	N/A
10-80%*	**	1.78 m x 812.80 mm x 584.20 mm (70 in. x 32 in. x 23 in.)	3.1 m (10 ft)	N/A
10-80%*	**	784.86 mm x 863.60 mm x 658.80 mm (30.9 in. x 34 in. x 27 in.)	4.6 m (15 ft)	N/A
10-80%*	250 kg*** (550 lbs) in crate	Same as Model 4 System plus two DS25/DS50 disk drives	4.6 m (15 ft)	N/A
10-80%*	**	1.78 m x 812.80 mm x 584.20 mm (70 in. x 32 in. x 23 in.)	3.1 m (10 ft)	N/A
0-95%	15.9 kg (35 lbs)	311.15 mm x 660.40 mm x 482.60 mm (12.25 in. x 26 in. x 19 in.)	1.8 m (6 ft)	N/A
5-95%	20.4 kg (45 lbs)	348.74 mm x 660.40 mm x 508.00 mm (13.73 in. x 26 in. x 20 in.)	1.3 m (6 ft)	Increments to 610 m (2000 ft)
10-80%	68 kg (150 lbs)	261.87 mm x 778.00 mm x 481.08 mm (10.31 in. x 30.63 in. x 18.94 in.)	1.8 m (6 ft)	4.6 m††† (15 ft)

††Because of the heavy start-up current, the disk drives require a separate electrical service and cannot use the power strip in the desk enclosure. DS25/DS50 disk drives are started one at a time to limit the current inrush. The electrical service for the disk drives must have fuses or circuit breakers that are rated to withstand the following currents for a maximum of eight seconds.

Disk Drive	Voltage	Start-up Current	Running Current
DS10	120 V	10 A	4.6 A
DS25/DS50	100-127 V	30 A	7.5 A
DS25/DS50	208-220 V	13 A	4.5 A

†††The DS10 daisy-chain (drive-to-drive) cable length is 1.8 metres (6 feet).

Table 35. Standard DS990 Equipment Specifications (Continued)

Item	Input Voltage Options	115 V Power Consumption (Steady State)	Heat Dissipation	Operating Temperature Range
Model DS25/DS50 Disk Drive	110 V, 50/60 Hz 115 V, 50/60 Hz 220 V, 50/60 Hz	669 W*	2283 Btu/hr	16°C-38°C
Model 810 Printer	100/120/220/240 V, 50/60 Hz	200 W	683 Btu/hr	5°C-40°C
Model 2230/2260 Line Printer	110 V, 50/60 Hz 115 V, 50/60 Hz 230 V, 50/60 Hz	Model 2230– 525 W Model 2260– 680 W	Model 2230– 1792 Btu/hr Model 2260– 2321 Btu/hr	10°C-38°C
Model 733 ASR Data Terminal	110 V, 50/60 Hz 115 V, 50/60 Hz 230 V, 50/60 Hz	260 W	888 Btu/hr	10°C-40°C
Model FD800 Dual Floppy Disk	100 V, 50/60 Hz 115 V, 50/60 Hz 230 V, 50/60 Hz	200 W	682 Btu/hr	10°C-38°C
Model 979A Magnetic-Tape Transport	115 V, 50/60 Hz 230 V, 50/60 Hz	422 W†	1440 Btu/hr	10°C-32°C
Model DS31 Disk Drive (without power supply)	See power supply	184 W	628 Btu/hr	16°C-32°C
DS31 Power Supply	105-240 V, 50/60 Hz	220 W	450 Btu/hr (one drive)	16°C-32°C
Model 804 Card Reader	115/208 V, 60 Hz 100/240 V, 50 Hz	300 W	1024 Btu/hr	10°C-38°C

*Because of the heavy start-up current, the disk drives require a separate electrical service and cannot use the power strip in the desk enclosure. DS25/DS50 disk drives are started one at a time to limit the current inrush. The electrical service for the disk drives must have fuses or circuit breakers that are rated to withstand the following currents for a maximum of eight seconds.

Disk Drive	Voltage	Start-up Current	Running Current
DS10	120 V	10 A	4.6 A
DS25/DS50	100-127 V	30 A	7.5 A
DS25/DS50	208-220 V	13 A	4.5 A

Operating Humidity Range (Noncondensing)	Weight	Chassis Dimensions (H x D x W)	115 V Power Cable Length	990 Interface-to-Device Cable Length
10-80%	100 kg (220 lbs)	266.70 mm x 812.80 mm x 452.12 mm (10.5 in. x 32 in. x 17.8 in.)	2.4 m (8 ft)	4.6 m ** (15 ft)
5-90%	25 kg (55 lbs)	203.20 mm x 508.00 mm x 654.05 mm (8 in. x 20 in. x 25.75 in.)	1.8 m (6 ft)	9.1 m*** (30 ft)
30-90%****	Model 2230– 154 kg (340 lbs) Model 2260– 168 kg (370 lbs)	1130.30 mm x 838.20 mm x 660.48 mm (44.5 in. x 33 in. x 26 in.)	3.7 m (12 ft)	9.1 m (30 ft)
10-90%	25.4 kg (56 lbs)	508.00 mm x 660.40 mm x 558.80 mm (20 in. x 26 in. x 22 in.)	2.4 m (8 ft)	9.1 m (30 ft)
20-80%	20.4 kg (45 lbs)	177.80 mm x 635.00 mm x 482.60 mm (7 in. x 25 in. x 19 in.)	2.4 m (8 ft)	1.8 m (6 ft)
20-80%	61 kg (135 lbs)	622.30 mm x 419.10 mm x 482.60 mm (24.5 in. x 16.5 in. x 19 in.)	1.4 m (4.5 ft)	4.4 m (14.5 ft)
20-85%	19.5 kg (43 lbs)	175.26 mm x 581.66 mm x 482.60 mm (6.9 in. x 22.9 in. x 19 in.)	N/A††	3.1 m ††† (10 ft)
20-80%	11.3 kg (25 lbs)	82.55 mm x 431.80 mm x 203.20 mm (3.25 in. x 17 in. x 8 in.)	1.7 m (66 in.)	N/A
10-90%	25 kg (55 lbs)	431.80 mm x 469.90 mm x 362.20 mm (17 in. x 18.5 in. x 14.26 in.)	1.8 m (6 ft)	6.1 m (20 ft)

**The DS25/DS50 disk drives use a bus cable and up to four radial cables. The bus cable can be extended by a daisy-chain cable. This cable length is 2.4 metres (8 feet) or 4.6 metres (15 feet). However, the 4.6-metre (15-foot) controller-to-drive radial cable determines the maximum controller-to-drive distance.

***The 810 printer I/O cable is available in increments up to 305 metres (1000 feet).

****The 2230/2260 line printer operating humidity range can be extended to 10-90 percent with the optional static eliminator.

†The 979A magnetic-tape-transport start-up current is approximately 10 amperes.

††A separate DS31 power supply provides operating voltages for up to two DS31 disk drives. The power supply is usually mounted on the rear EIA rack-mounting rails. The power-supply-to-drive cable length is 2 metres (6.5 feet).

†††Up to four DS31 disk drives can be daisy chained on a common bus. Drive-to-drive cable length is 1.2 metres (4 feet).

Table 34. DS990 Hardware Options (Continued)

Category	Description
Interfaces and Communications Equipment	TTY/EIA terminal interface module
	TTY/EIA data-terminal interface cable
	990 communications I/F module (synchronous and asynchronous)
	Bell data-set interface kit
	990 asynchronous-modem kit
	990 synchronous-modem kit
	Auto-call kit
	16 I/O EIA data module
	16 I/O EIA data module with high/low interrupt
	16 I/O EIA data module with low/high interrupt
	16 I/O TTL data module
	16 I/O TTL data module with high/low interrupt
	16 I/O TTL data module with low/high interrupt
	16 I/O TTL data module with pull-up (3K)
	D/A converter kit, 1 channel
	D/A converter kit, 2 channels
	D/A converter kit, 3 channels
	D/A converter kit, 4 channels
	A/D converter kit, 16 channels
	A/D converter kit, 32 channels
	A/D converter kit, 48 channels
	A/D converter kit, 64 channels
	32-bit input/transition detection kit
	32-bit output-data kit
	Digital-I/O-termination-panel kit with 3.0-m (10-ft.) cable
	Digital-I/O-termination-panel kit with 4.6-m (15-ft.) cable
	Digital-I/O-termination-panel kit with 6.1-m (20-ft.) cable
Other	CRU expansion master kit
	CRU expansion secondary kit
	TILINE expansion kit
	TILINE interface kit
	Single-bay pedestal
	Rack-mounting equipment cabinet

Table 36. Hardware Manuals

Model 990 Computer DS990 Disk System Installation and Operation
Model 990/10 Computer System Hardware Reference Manual
Model 990 Computer Model FD800 Floppy-Disk System Installation and Operation
Model 990 Computer Model DS31/DS32 Disk System Installation and Operation
Model 990 Computer Model DS10 Disk System Installation and Operation
Model 990 Computer Model DS25/DS50 Disk System Installation and Operation
Model 990 Computer Model 979A Magnetic Tape System Installation and Operation
Model 990 Computer Model 979A Tape Transport Installation and Operation
Model 990 Computer Model 911 Video Display Terminal Installation and Operation

Model 990 Computer Model 743 KSR Data Terminal Installation and Operation
Model 990 Computer Model 733 ASR/KSR Data Terminal Installation and Operation
Model 990 Computer Model 810 Printer Installation and Operation
Model 990 Computer Model 2230 and 2260 Line Printers Installation and Operation
Model 990 Computer Model 804 Card Reader Installation and Operation
Model 990 Computer TMS 9900 Emulator and Buffer Modules Installation and Operation
Model 990 Logic State Trace Data Module Installation and Operation
Model 990 Computer PROM Programming Module Installation and Operation
Model 990 Computer TTY/EIA Terminal Interface Installation and Operation

Model 990 Computer Communications System Installation and Operation
Model 990 Computer Pulse/Tone Auto Calling Unit Installation and Operation
Model 990 Computer Analog/Digital Converter Module, Digital/Analog Converter Module Installation and Operation
Model 990 Computer 32 Input/Transition Detection Module, 32-Bit Output Data Module, and Digital Input/Output Termination Panel Installation and Operation
Model 990/10 Computer System Field Maintenance Manual
Model 990 Computer Peripheral Equipment Maintenance Manual
Model 990 Computer Diagnostics Handbook
Model 990/10 Computer System Depot Maintenance Manual
Model 990 Computer Model FD800 Floppy-Disk Controller Depot Maintenance Manual

Model 990 Computer Models DS31/DS32 Disk Controller Depot Maintenance Manual
Model 990 Computer Model 804 Card Reader Interface Module Depot Maintenance Manual
Model 990 Computer Model DS25/DS50 Disk Controller Depot Maintenance Manual
Model 990 Computer Model 979A Magnetic Tape Controller Depot Maintenance Manual
Model 990 Computer Model 979A Tape Transport Maintenance Manual
Model 990 Computer Model 911 Video Display Terminal Depot Maintenance Manual
Silent 700 Electronic Data Terminals Model 743 KSR Model 745 Portable Maintenance Manual
Silent 700 Electronic Data Terminals Models 732/733 ASR/KSR Maintenance Manuals
Model 990 Computer TMS 9900/9980A Emulator and Buffer Modules Depot Maintenance Manual

Model 990 Computer AMPL Logic State Trace Date Module Depot Maintenance Manual
Model 990 Computer PROM Programming Module Depot Maintenance Manual
Model 990 Computer TTY/EIA Terminal Interface Module Depot Maintenance Manual
Model 990 Computer Communications Interface Module Depot Maintenance Manual
Model 990 Computer Automatic Calling Unit Depot Maintenance Manual
Model 990 Computer 16 Input/16 Output EIA Data Module Depot Maintenance Manual
Model 990 Computer 16 Input/16 Output TTL Data Module Depot Maintenance Manual
Model 990 Computer Analog/Digital Converter Module, Digital/Analog Converter Module Depot Maintenance Manual
Model 990 Computer Family Maintenance Drawings
- *Volume I — Processors*
- *Volume II — Peripherals*

Table 37. Software Manuals

Model 990 Computer TMS 9900 Microprocessor Assembly Language Programmer's Guide

Model 990 Computer Programming Card
(Order multiples of twenty copies)

Model 990 TMS 9900 Assembly Language Coding Form

Model 990 Computer DX10 Operating System Release 3 Reference Manual

- *Volume I — Concepts and Facilities*
- *Volume II — Production Operation*
- *Volume III — Application Programming Guide*
- *Volume IV — Developmental Operation*
- *Volume V — System Programming Guide*
- *Volume VI — Error Reporting and Recovery*

Model 990 Computer DX10 Operating System Release 3 Programming Card
(Order multiples of twenty copies)

Model 990 Computer DX10 Operating System Installation Guide

Model 990 Computer Link Editor Reference Manual

Model 990 Computer Terminal Executive Development System (TXDS) Programmer's Guide

Model 990 Computer Microprocessor Prototyping Laboratory System (AMPL) Operation Guide

Model 990 Computer AMPL System Tutorial

AMPL Reference Card
(Order multiples of twenty copies)

Model 990 Computer FORTRAN Programmer's Reference Manual

*Model 990 Computer DX FORTRAN Installation**

Model 990 Computer DX10 Operating System Release 3 COBOL Programmer's Guide

*Model 990 Computer COBOL Installation for DX10 Release 3**

Model 990 Computer Scientific BASIC Programmer's Guide

*Model 990 Computer Scientific BASIC Installation for DX10 Release 3**

Model 990 Computer Business BASIC Programmer's Guide

*Model 990 Computer Business BASIC Installation for DX10 Release 3**

Model 990 Computer SORT/MERGE User's Guide

*Model 990 Computer SORT/MERGE Installation for DX10 Release 3**

Model 990 Computer Report Program Generator (RPG II) Programmer's Guide

*Installation procedures are part of applicable software kit.

Sales and Service Offices of Texas Instruments are located throughout the United States and in major countries overseas. Contact the Digital Systems Division, Texas Instruments Incorporated, P.O. Box 1444, Houston, Texas 77001, or call (512) 258-7305, for the location of the office nearest to you.

Texas Instruments reserves the right to change its product and service offerings at any time without notice.

TEXAS INSTRUMENTS
INCORPORATED

DIGITAL SYSTEMS DIVISION
P.O. BOX 1444 • HOUSTON, TEXAS 77001

Printed in U.S.A

TI-577-5M-6/78