

CONTROL DATA

SOFTWARE AND HARDWARE PRODUCT DESCRIPTIONS



Product Resource Manual


CONTROL DATA CORPORATION

MARCH 1979



Date: _____

To: Proposal Administration Department HQW09G

From: _____ Facility: _____

Subject: Software/Hardware Descriptions Manual (March 1979)

My comments and suggestions to make the subject manual more effective are:

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241
MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY
CONTROL DATA CORPORATION
8100 34th Avenue South
Minneapolis, Minnesota 55440



ATTN: Proposal Administration (HQW09G)

FOLD

FOLD

STAPLE

STAPLE

FOREWORD

This document contains descriptions of Control Data Corporation selected software and hardware products for use when preparing proposals. It does not include all products available from CDC, but rather it includes descriptions of software and hardware products most often used in proposals. The majority of these descriptions are general and do not describe any one product version.

The purpose of providing these descriptions is to improve efficiency in proposal preparation by minimizing time and cost. It is suggested that you modify these descriptions as necessary to fit your precise proposal needs rather than copying them exactly as presented. It is not the intent of this manual to promote the use of "boilerplate" which is generally an unacceptable practice, especially for Government customers.

This document represents the collective efforts of personnel from within the Computer Systems Marketing Group (HQW09G), the Systems Division, the Technical Publications Group (ARH220), as well as the many inputs and suggestions received from you – the users. We wish to thank each of you for your contributions to the development of this document and look forward to receiving your continued support and recommendations.

CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1	OPERATING SYSTEMS	
	CDC CYBER 1000	1-1
	MSOS (CDC CYBER 18)	1-7
	NOS (Abbreviated Version)	1-14
	NOS	1-19
	NOS/BE	1-43
	RTOS 3 (CDC CYBER 18)	1-48
	SCOPE 2.0	1-50
2	SOFTWARE PRODUCTS	
	ALGOL 60 5	2-1
	APEX III (Mathematical Programming System)	2-3
	APL 2	2-6
	BASIC	2-8
	CAS 1.0 (Conversion Aids System)	2-10
	CAS 2.0 (Conversion Aids System)	2-11
	CCP (Communication Control Program)	2-13
	CDCS (CYBER Database Control System)	2-16
	CID (CYBER Interactive Debug)	2-18
	COBOL 5	2-20
	COMPASS	2-23
	CRM (CYBER Record Manager)	2-25
	CYBER Cross System Software	2-26
	CSSL (Continuous System Simulation Language)	2-30
	CYBER 170/70 Model 76 Station Software	2-31
	DBU (Data Base Utilities)	2-32
	DDL (Data Description Language)	2-33
	DISSPLA	2-34
	DMS-170 (Data Management System)	2-36
	DNS (Distributed Network System)	2-37
	Eight-Bit Subroutines	2-38
	FORM (File Organizer and Record Manager)	2-39
	FORTRAN Database Facility (FDBF)	2-40
	FORTRAN Extended 4	2-41
	FORTRAN Extended and FORTRAN Common Library	2-42
	FORTRAN 4 to 5 Conversion Aid	2-43
	GPSS V (General-Purpose Simulation System)	2-44
	IAF (Interactive Facility)	2-45
	IMSL (International Mathematical and Statistical Library)	2-47
	Maintenance Package for NOS	2-48
	MSL (Math Science Library)	2-49
	NAM (Network Access Method)	2-50
	Network Products Software	2-52
	NPS (Network Products Stimulator)	2-54
	PDS/MAGEN	2-55
	PERT/TIME	2-56
	PL/I	2-57

<u>Section</u>	<u>Title</u>	<u>Page</u>	
2 (Cont'd)	QUERY/UPDATE	2-58	
	RBF (Remote Batch Facility)	2-59	
	RPG II System	2-60	
	Sanders Graphics 7 Post-Processor Under TIGS	2-62	
	JIMSCRIPT II.5	2-63	
	SIMULA	2-65	
	SORT/MERGE	2-66	
	SPSS (Statistical Package for the Social Sciences)	2-68	
	TAF (Transaction Facility)	2-70	
	Tektronix 401X Post-Processor Under TIGS	2-72	
	TEXTJAB	2-73	
	TIGS (Terminal Independent Graphics System)	2-74	
	TOTAL	2-77	
	TOTAL/ATHENA	2-80	
	TOTAL Extended	2-82	
	TOTAL Universal	2-84	
	TRS (Tape Reservation System)	2-90	
	UNILOT (Universal Plotting)	2-91	
	UNISTRUC	2-94	
	UPDATE	2-97	
	XEDIT	2-99	
	3	COMPUTER MAINFRAMES	
		CDC CYBER 18/Model 5 Batch Terminal Controller	3-1
CDC CYBER 18/Model 10M Computer System		3-6	
CDC CYBER 18/Model 17 Computer System		3-11	
CDC CYBER 18/Model 20 Processor		3-13	
CDC CYBER 18/Model 30 Time-Share System		3-16	
CDC CYBER 70/Model 71 Computer System		3-20	
CDC CYBER 70/Model 72 Computer System		3-27	
CDC CYBER 70/Model 73 Computer System		3-34	
CDC CYBER 70/Model 74 Computer System		3-41	
CDC CYBER 70/Model 76 Computer System		3-48	
CDC CYBER 170/Model 171 Computer System		3-52	
CDC CYBER 170/Model 172 Computer System		3-58	
CDC CYBER 170/Model 173 Computer System		3-65	
CDC CYBER 170/Model 174 Computer System		3-72	
CDC CYBER 170/Model 175-100 Computer System		3-78	
CDC CYBER 170/Model 175-200 Computer System		3-85	
CDC CYBER 170/Model 175-300 Computer System		3-92	
CDC CYBER 170/Model 176 Computer System		3-99	
CDC CYBER 170 Series 700 Computer System Model 720		3-105	
CDC CYBER 170 Series 700 Computer System Model 730		3-112	
CDC CYBER 170 Series 700 Computer System Model 750		3-119	
CDC CYBER 170 Series 700 Computer System Model 760		3-125	
CDC CYBER 1000 Hardware	3-131		
4	HARDWARE PRODUCTS		
	405 Card Reader	4-1	
	415 Card Punch	4-2	
	580 Train Printer	4-3	
	580 Train Printer with PFC	4-5	
	595 Train Cartridge	4-7	
	596 Train Cartridge	4-8	
	667 Magnetic Tape Transport	4-9	
	669 Magnetic Tape Transport	4-10	
	677 Magnetic Tape Transport	4-11	
	679 Magnetic Tape Transport	4-12	
751 Display Terminal	4-13		
752 Display Terminal	4-14		

<u>Section</u>	<u>Title</u>	<u>Page</u>
4 (Cont'd)	753 Non-Impact Printer	4-17
	755 Impact Slave Printer	4-18
	756 Display Terminal	4-20
	777 CYBER Graphics Terminal/774 Digigraphics IV Console	4-22
	819 Disk Storage Unit	4-24
	844 Disk Storage Unit	4-25
	881/883 Disk Packs	4-26
	885 Fixed Module Drive	4-27
	1711 Teletypewriter	4-28
	1713 Teletypewriter	4-29
	1811-1 Conversational Display Terminal	4-30
	1827-30/31 Line Printer	4-31
	1827-60 Line Printer	4-32
	1828-1 Card Reader/Line Printer Controller	4-33
	1829-30 Card Reader	4-34
	1832-4 Magnetic Tape Controller	4-35
	1833-1 Storage Module Drive Interface	4-36
	1833-3 Storage Module Drive Control Unit	4-37
	1843-1 Communications Line Adapter	4-38
	1860-3 Magnetic Tape Transport	4-39
	1860-92 Magnetic Tape Transport	4-40
	1865 Flexible Disk Drive	4-41
	1867-10 Storage Module Drive	4-42
	1867-20/21 Storage Module Drive	4-43
	1875-1 Breakpoint Controller	4-44
	1875-2 Breakpoint Panel	4-45
	255X Network Processing Unit	4-46
	2550-101 6671/6676 Emulation Controlware	4-52
	2554-16/32 Memory Expansion	4-53
	2556-11 Loop Multiplexer Expansion	4-54
	2558-3 Communications Coupler	4-55
	2558-4 Emulation Coupler	4-56
	2560 Series Communication Line Adapter (Sync)	4-57
	2561 Series Communication Line Adapter (Async)	4-59
	2563 Series Communication Line Adapter	4-60
	3270/8271 Transfer Switch Subsystem	4-61
	3446 Card Punch Controller	4-62
	3447 Card Reader Controller	4-63
	6671-3 Data Set Controller	4-64
	6673 Data Set Controller	4-65
	6676 Data Set Controller	4-66
	6681 Data Channel Converter	4-67
	6683 Satellite Coupler	4-68
	7012 Display Console	4-69
	7021-2X Magnetic Tape Controller	4-70
	7021-3X Magnetic Tape Controller	4-71
	7030 CYBER Extended Core Storage	4-72
	7054 Mass Storage Controller	4-74
	7152 Mass Storage/Magnetic Tape Controller	4-75
	7154 Mass Storage Controller	4-76
	7155 Fixed Module Drive Controller	4-77
	7622 Magnetic Tape Controller	4-79
	7639 Mass Storage Unit Controller	4-80
	7654 Mass Storage Controller	4-81
	7681 Data Channel Converter	4-82
	7683 Satellite Coupler	4-83
	7880 Mass Storage Subsystem	4-84
	10312 CDC CYBER 170 Memory Increment	4-86
	10313 CDC CYBER 170/Model 175 Memory Increment	4-87

<u>Section</u>	<u>Title</u>	<u>Page</u>
4 (Cont'd)	10314 CDC CYBER 170 Peripheral Processor Unit Increment	4-88
	10318-1/2 Extended Core Storage Coupler	4-89
	10377-1 CYBER 176 Peripheral Processor Increment	4-91
	10380 Compare/Move Unit	4-92
	10381 Data Channel Converter	4-93
	10400-10401 CLA Cables	4-94

SECTION 1

OPERATING SYSTEMS





CYBER 1000 SOFTWARE

The CDC CYBER 1000 Data Communications System is specially designed for communications processing either as a stand-alone message switch or as a distributed network system which allows multiple host computers and terminals to transfer and accept data to and from each other, or both. The design is an outgrowth of many years of experience with systems requiring a great degree of reliability, expandability, and freedom from interruption.

The CDC CYBER 1000 software is available in two different versions. The first version, a stand-alone message switch, was developed in order to use the CDC CYBER 1000 as a highly protected message switch. The system performs a store-and-forward function using an additional hardware component — a very reliable fast disk system. The system stores traffic and programs redundantly on the prime and copy sides of the fast disk system to provide greater reliability. An input message is acknowledged by the system only after it is stored on the fast disk system, thus guaranteeing delivery to its destination. Additional features allow the processing of messages stored only in central memory (memory switch).

The second version, called DNS (Distributed Network System), was developed to use the CDC CYBER 1000 as an efficient distributed network node with very high throughput and short response times. The DNS software permits geographically separated terminals and general-purpose computers to be integrated into a single network. The CDC CYBER 1000 relieves hosts and terminals of virtually all data communications network responsibilities. The DNS software does not require the fast disk system hardware. Message assurance, if required, is provided by means of an end-to-end protocol.

In either of the above versions, the software and the hardware package required to support a common category of terminals is known as a TIP (Terminal Interface Package).

TERMINAL INTERFACE PACKAGE

The terminal interface, or TIP (Terminal Interface Package), consists of hardware and program modules which provide a communication line interface between a CDC CYBER 1000 exchange unit and a terminal. The TIP is structured to accommodate the electrical and logical characteristics of the terminal. In general, CDC CYBER 1000 TIPs may service lines which have the following characteristics:

- Single or multi-station
- Simplex, half-duplex, or full-duplex
- Synchronous or asynchronous

TIPs also provide for a variety of line speeds, codes, and supervisory procedures. Any particular TIP, of course, provides for a defined set of these characteristics. For terminating a particular set of lines, one set of TIP hardware is required for each line and one TIP program is required for each different line type.

TIPs generally perform these typical functions:

- Polls stations on a line
- Analyzes responses to a poll
- Calls stations on a line
- Analyzes responses to a call
- Passes data and supervision to and from the common base to and from the line/station device
- Detects incoming supervision characteristics as specified by the common base
- Generates appropriate supervision characters as specified by the common base
- Assembles characters on input
- Disassembles characters on output
- Detects faulty conditions and transmits appropriate line/device status to the common base

The base system software features are individually discussed under the PMX/DNS descriptions which follow.

Off-Line Software

The off-line software consists of a program production system made up of a number of system support programs and a user application program. The systems support programs include an assembler, job processor, debugging utility routines, library edit, modular system builder, and off-line diagnostic and maintenance programs. Together they provide for program development, system modifications, and maintenance. The off-line user application programs include a USASI basic standard FORTRAN compiler and a FORTRAN processor that controls loading and execution of the FORTRAN programs. The FORTRAN compiler is intended for the development of nonreal-time programs. The assembler can also generate off-line, stand alone programs for off-line execution.

MESSAGE SWITCHING SYSTEM DESCRIPTION

The CDC CYBER 1000 System uses total redundancy of programs, tables, and message traffic, based on in-transit storage requirements, in order to provide complete backup while operating in a shared-load environment. Each processor in the system backs up its neighbor so that real-time automatic switchover is accomplished in the unlikely event that a component failure does occur.

On-line configuration, in the real-time environment, allows operations personnel to add, delete, or modify circuits and stations without having to stop message processing.

The dual CDC CYBER 1000 System operates on the concept of load-sharing. These are two processors, each of which typically services half the network under normal conditions. Each circuit is hard-wired to line controllers (one for each processor) in the system. In addition, some circuits of each terminal type

are assigned to each processor, so that the software in the two processors is identical. This is done to provide an assured "switchover" in the event of a processor failure. No physical switching is involved since the lines are already hard-wired to both processing systems; there is only a logical action required for the operating processor to service the additional lines.

Since the alternate processor is servicing the same types of lines and performing the same message processing tasks, there is no question of processor capability in servicing new lines.

The line controllers normally scan only the lines associated with their primary processor, and are idle for an equivalent time. This idle time is reserved for the situation in which one processor has failed, and the remaining processor services all the circuits of the network. In this event, the line controller's scan is extended. During the primary phase of its scan, the controller attends to its original lines; then, during the secondary (or idle) phase of its scan, it attends to those of the processor which failed.

A monitor checks the status of both processors. Upon detecting a failed processor system, this monitor commands the line controller to extend its scan, and commands the operating processor to service all its lines.

CDC CYBER 1000 MESSAGE SWITCHING SOFTWARE FEATURES

PMX, which is the software developed for switching functions, has these important capabilities:

- Message Formats
- Routing
 - Single
 - Multiple
- Restart and Recovery
- Magnetic Tape Outstation
- On-Line Configurator
- Rotary Circuit Service
 - Group
 - Implied
 - Special
- Message Priority
- Report and Controls
- History/Journal
- Retrieval
- Intercept
- Code Translation
- Message Queue Extension
- Test Message
- Overload Protection
- Address Stripping
- Terminal Interface Packages
- Line and Station Queueing
- Input Message Ender Stripping
- Date-Time Group Assignment
- Network Directory Reports
- Header Error Handling Procedures
- Message Protection
- Message Enders
- Message Sequence Numbering/
Verification
- Hold Queue Service
- Input/Output Character Counting
- Abnormal Termination of Input/Output
Message Service

TIPs SUPPORTED BY PMX

The following is a list of standard TIPs supported by PMX. Additional TIPs are developed as required to meet the growing needs of users.

- M28 TTY, 67 wpm, ¼ speed
- M28 TTY, 67 wpm
- M28 TTY, 100 wpm
- M28 TTY, 67 wpm, ½ speed
- ATT 83B 2/3
- ATT 81D1 TTY, 100 wpm
- ATA/IATA Low-Speed, Processor-to-Processor
- M35 TTY, FW, 100 wpm
- M37 TTY, FW, 150 wpm
- ATT 85A1, 110 Baud
- CYBER Console M35/M33
- TWX Dial-In/Output
- TELEX
- CDC 711-10, Synchronous, ASCII
- ATA/IATA Medium-Speed, Synchronous, ASCII
- ATA/IATA Medium-Speed, Synchronous, SABRE Code
- Dial BSC, Synchronous, ASCII
- Point-to-Point BSC, Synchronous, ASCII
- Dial BSC, Synchronous, EBCDIC
- Point-to-Point BSC, Synchronous, EBCDIC
- Dial Control Module
- Model 40 Dial-In/Out
- Model 40 Dedicated

DNS DESCRIPTION

The CDC CYBER 1000 DNS System can function as either a remote satellite concentration system or as a "front-end" communications subsystem with a host processor system.

When implemented as a "front-end" system, the communications processor has the primary responsibility for isolating, to the largest extent possible, the host computer (hardware and software) from the communications network. The communications processor relieves the host of all the high-overhead communications processing tasks required to properly administer and service multiple communications lines and varieties of terminal types. The front-end concept allows the terminal network to be physically and logically independent of a given host, thus providing a flexible, open-ended network system in which future additions of terminals and hosts are easily obtained.

The front-end/host computer approach provides the ideal combination of equipment to facilitate a sustained high message throughput and short response time, together with enhanced system reliability and message integrity.

In the remote DNS node configuration, the communications processor provides all of the necessary hardware and software to perform as an intelligent concentrator which interfaces with a diverse set of terminals, remote hosts and communications processors, and other remote nodes. Communication with other remote nodes or front-end nodes is accomplished via high-speed lines.

CDC data communications specialists, using a systems approach, construct the data communications system which provides the best solution to the specific requirements of each customer. Multiple points of concentration, called nodes, are defined to provide the most economically and technically feasible solution.

The DNS supports session concepts which are fundamental to the design of many contemporary networks in which responsibilities are shared by a host processor and data communications processor. This support includes:

- Distribution of application status changes to the entire network
- Verification of terminal "log-ons" to an application
- Routing of messages from a "logged-on" terminal to the application
- Routine of messages from an application to a terminal or another application
- Notifications to terminals and applications when an existing session must be broken

A terminal may establish a session with any application resident in any host in the network. A host may communicate with any terminal or any other host in the network. Applications may be open at different hosts at different times.

CDC CYBER DNS SOFTWARE FEATURES

DNS software supports the following features:

- Multiple terminal types
- Code translation
- Data integrity
- Multiple communication processors at a node
- Multiple hosts
- Multiple nodes
- Fully integrated communication networks
- System integrity†
- Distribution of status changes to affected communications processors and hosts
- Implied and address routing

†Failure of a host, communications processor, or an entire node causes the failed component to be isolated but does not otherwise impact the functional system.

- Dynamic selection of best path to destination
- System status visibility and control
- Remote supervisory control
- Load sharing
- Overload protection
- Message assurance
- Message sequence numbering

HOST INTERFACES SUPPORTED BY DNS

The CDC CYBER 1000 interfaces on a channel basis with host processors utilizing the CDC Micro Programmable Communications Controller or Control Board (MPCC or MPCB).

The following types of interface are currently supported:

- Four line 2701 SDAII communications controller emulation
- IBM 2803/2804 magnetic tape controller emulation
- IBM 3704/3705 communications controller emulation (under development)

Additional host interfaces are developed to meet user needs as required.

TIPs SUPPORTED BY DNS

- Console TIP – Provides interface to the DNS supervisory console
- CDC CYBER Interconnect – Full-duplex, fully transparent multiblock message transfer used to interface the CDC CYBER 1000 communications processors
- ISO R1745 – Half-duplex, controlled message transfer in accordance with ISO recommendation 1745
- IBM 970 Banking Terminals
- Teletype Corporation 2550 Cluster Controller
- Burroughs TC500 Banking Terminals
- NCR AT 770 Cash Dispenser
- IBM 3275 Information Display System
- IBM Binary Synchronous Communication (BSC) – Half-duplex, point-to-point, or multipoint
- Teletype Corporation Model 40/4

Additional TIPs are developed to meet user needs as required.

For additional information on this subject, please contact Marketing Support, Control Data Corporation, Communications Systems Division, 3285 East Carpenter Avenue, Anaheim, California, 92806, phone: (714) 630-2022.



CYBER 18 MSOS

The CDC CYBER 18 Mass Storage Operating System (MSOS) is a multiprogramming system designed to support a variety of on-line control and data acquisition applications requiring dedicated system utilization, batch processing, and program checkout features in a real-time environment.

MSOS regulates all multiprogramming on the basis of the priority level assigned to a particular operation whether the operation is program execution or input/output operations. Sixteen levels of hardware interrupts and program execution states are available. When an input/output operation is initiated from any area in core storage, the requesting program may be suspended pending completion of the input/output operation. Once scheduling of input/output occurs, any other program with a priority equal to or less than that scheduled by the suspended program, can be put into operation. Therefore, up to fifteen program execution of input/output operations can be performed concurrently. In compliance with these features, MSOS allows for any maximum number of user defined input/output or program execution operations to be queued for processing.

FEATURES

The CDC CYBER 18 MSOS provides the following features.

- Optimum required throughput and response
- Parallel debugging, batch processing, and application program execution
- Peripheral processing that allows maximum utilization of all devices
- Configuration capabilities to maximize core utilization requirements
- Automatic program scheduling through interrupts
- Status and error logging of system hardware and software detectable errors
- On-line system modification
- System modularity
- Relocatable binary loader; this is a nonresident module stored in the system library
- Breakpoint program and recovery program for debugging unprotected programs
- On-line debug program for accessing both protected and unprotected core to change, inspect, and copy core and mass storage locations
- Library editing program to alter the system and program libraries
- Engineering file program which logs hardware errors detected by the system and prints a report of these errors upon demand
- File manager to create and maintain both sequential and indexed files

MONITOR

The monitor is the overall real-time executive program for CDC CYBER 18. It serves as an interface between all programs and the hardware. The monitor allocates use of the central processor and the input/output equipment to various programs on a priority basis.

The monitor assigns time on a priority basis: real-time programs that must be executed within a time limit run at high priority levels; nonreal-time programs run at the lowest priority levels.

On-line programs (process programs), the monitor, the job processor, the library editing program, and the recovery program run in protected areas of core. Batch programs (job processing) run in unprotected core. This ensures that errors in the unprotected programs do not destroy the on-line system. However, unprotected programs may make use of protected routines, such as input/output drivers, through requests to the monitor. Unprotected programs are assisted in making monitor requests by the comprehensive program protect system. Programs running in unprotected areas are not allowed to transfer control to locations in protected areas. When unprotected programs make monitor calls, a memory protect violation interrupt occurs. The interrupt service program determines the nature of the memory protect violation and then examines the monitor request for validity. It rejects the request, if in error, and permits the monitor to honor the request if the request is valid.

These features permit the CDC CYBER 18 software system to execute real-time programs in response to interrupts or internal requests and to process off-line jobs on a time-available basis.

REQUEST ENTRY PROCESSOR

When a program encounters a monitor request, it transfers control to the appropriate request entry processor. The processor stores the registers of the requesting program, examines the request for conformity with system constraints, and transfers control to the required processor. A parameter list which accompanies each request defines the type of request, identity of the program, input/output devices required, priority, etc.

After a request has been threaded, control returns to the requesting program if no higher priority program is waiting to run. However, if the request has a higher priority, it is executed immediately.

INTERRUPT STACK AND SCHEDULER STACK

Interrupt Stack

The interrupt stack consists of the waiting list formed by partially executed programs that have been interrupted by higher priority programs and is ordered on a last-in, first-out (LIFO) basis.

Scheduler Stack

The scheduler stack is the waiting list of programs that have been requested by the scheduler request processor. It is ordered by priority on a first-in, first-out (FIFO) basis within each priority. When a program is taken off the list, the next program threaded becomes the top of the list.

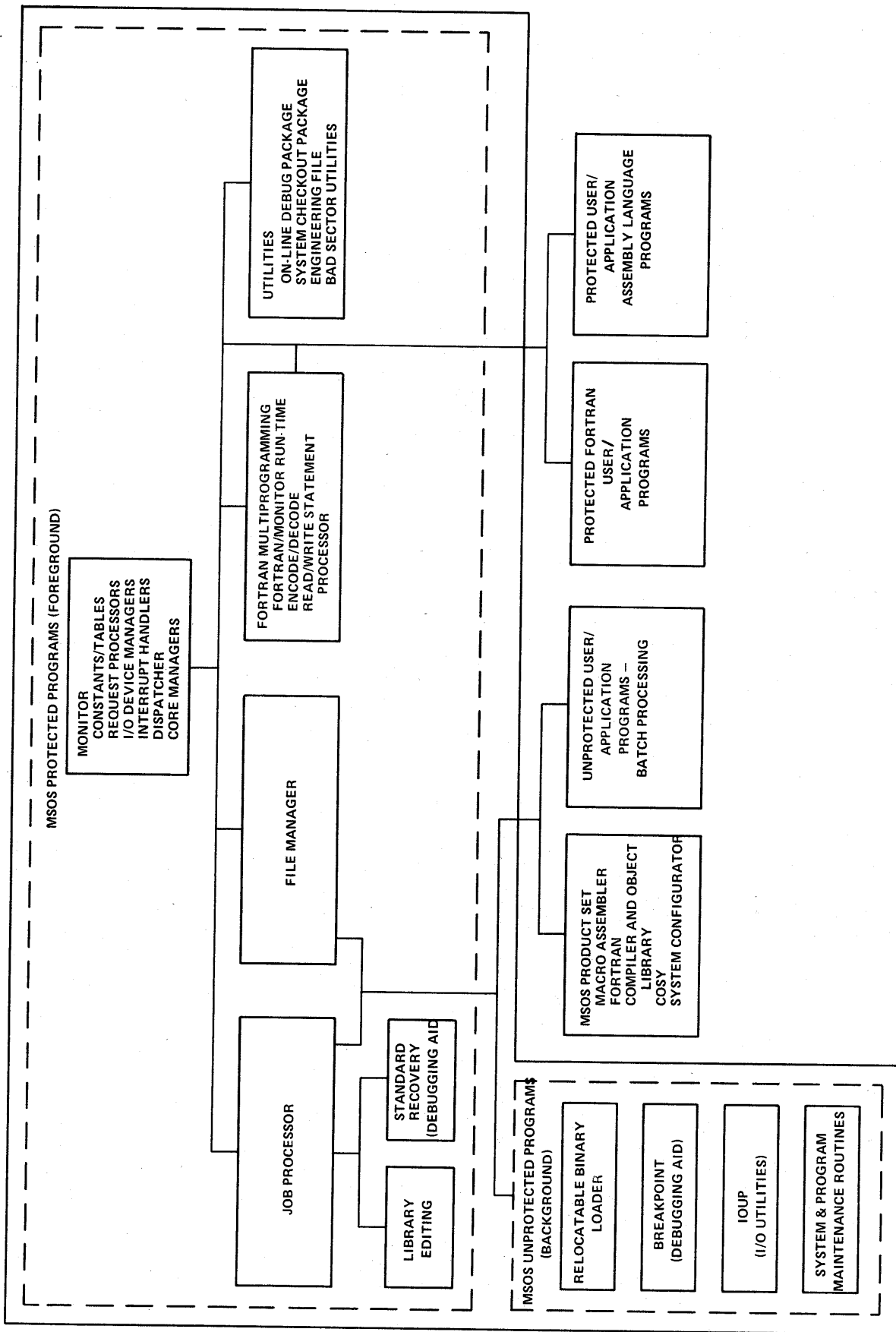
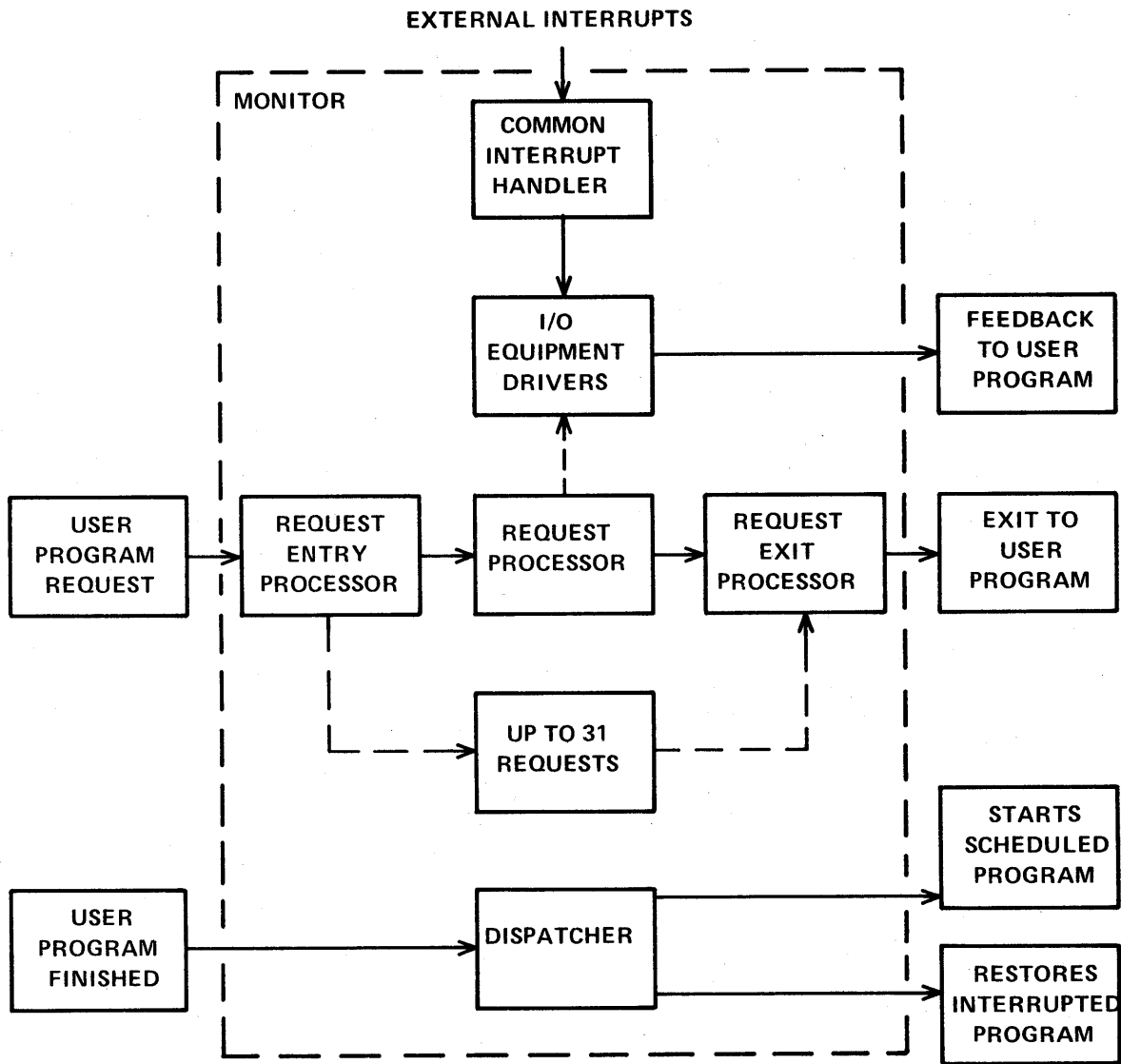


Diagram of Protected/Unprotected/Product Set/User Programs



Monitor Block Diagram for User Programs

COMMON INTERRUPT HANDLER

External interrupts transfer control of the computer to the common interrupt handler. When an interrupt is received on the computer, the hardware transfers control to the corresponding interrupt trap, saving the program address and overflow status and the associated interrupt routine is entered.

DISPATCHER

Control transfers to the dispatcher when a program or program element terminates. The dispatcher selects the program with the highest priority from either the scheduler stack or the interrupt stack. If priorities in both stacks are equal, the program in the interrupt stack is selected. Control is given to the selected program at the specified priority level and address location.

INPUT/OUTPUT DRIVERS

Each device in the system is associated with a device driver which is the only piece of software that gives direct commands to the device. The driver controls execution of a request passed to the monitor by a user program.

Each driver has three entries: initiator, continuator, and timeout. Functionally, the initiator initializes the working storage and initiates input/output on an idle device, the continuator drives the device to perform the actual requested task and timer entry is entered only on device hang-up on systems with a diagnostic timer.

JOB PROCESSOR

The job processor is a system program which monitors the unprotected core programs. This program is mass storage resident, stored in run-anywhere form in the system library, and is read into protected core by operator request. It allows programs to run in the background (unprotected core) when the system does not need the CPU or the background core area. The job processor runs under control of the monitor at a low priority level.

The job processor initiates and supervises the following programs running in or utilizing unprotected core.

- FORTRAN compiler
- Macro assembler
- COSY
- System configurator
- Off-line object programs
- Breakpoint
- Standard recovery } debugging aids
- Relocatable binary loader
- Library editing
- Input/output utilities
- System and program maintenance routines

The job processor is initiated by input through the console.

CDC CYBER 18 MASS STORAGE FORTRAN

CDC CYBER 18 Mass Storage FORTRAN will compile all basic FORTRAN source decks. If a compile time option is exercised, it becomes a superset of ASA Basic FORTRAN (X3.4/4). The compiler is a four pass compiler producing relocatable binary code and extensive diagnostics. Control card options can be used to give the user:

- Source listing (FORTRAN)
- Object code listing (Assembly)
- Run anywhere object code. This option allows a program in absolute form to be placed anywhere in memory
- Cross reference map

MACRO ASSEMBLER

The Macro Assembler for the CDC CYBER 18 family of computers provides a versatile symbolic language for writing source programs which it translates into object programs. The source programs are written with symbolic machine instructions, pseudo instructions, and macro instructions. Macro definitions may be defined by the user within the source program, or they may be placed on a separate macro library.

The Macro Assembler is a two pass assembler producing relocatable binary output as well as listings with extensive diagnostics.

DEBUGGING AIDS

The CDC CYBER 18 MSOS has three routines to aid in debugging programs. The three routines are described in this section in the order listed.

On-line debug package		Protected and unprotected core
Breakpoint program	}	Unprotected core
Recovery program		

On-Line Debug Package

The On-Line Debug Package (ODP) allows the programmer to access both protected and unprotected core in order to change core and mass storage locations, and to execute debugging functions while the system is running in an on-line state.

Breakpoint Program

Jobs can be controlled from the comment device with the breakpoint program. A software breakpoint is set initially to a core address in the command sequence of a program, and the program is interrupted prior to execution of the instruction stored at the breakpoint location. A software breakpoint may be set at any location in unprotected core.

Recovery Program

The programmer may, with the recovery program, determine the state of core and mass storage at the end of job execution.

INPUT/OUTPUT UTILITY PROGRAM

The Input/Output Utility Program (IOUP) program for the CDC CYBER 18 computer family enables the user to perform, via requests entered at the standard input or comment device, peripheral operations simultaneously in the background during normal foreground processing. Since the IOUP program operates under CDC CYBER 18 MSOS, the operating system's drivers must be available for all input/output equipment used in the IOUP program.

The following operations can be performed:

- Card to card
- Card to magnetic tape
- Card to paper tape
- Paper tape to printer
- Paper tape to paper tape
- Paper tape to magnetic tape
- Paper tape to card and printer
- Paper tape to card
- Magnetic tape to card
- Magnetic tape to printer
- Magnetic tape to card and printer
- Magnetic tape to magnetic tape
- Magnetic tape to paper tape

LIBRARY EDITING

The library editing program allows the user to:

- Add a program or file to the program library
- Remove a program or file from the program library
- Replace a program or file with another in the program library
- Replace allocatable core or partition core program in the system library
- Combine several relocatable binary programs in an absolute binary record and output this record on the binary output device
- Transfers information between peripheral devices and/or job processor file manager files
- Set request priorities for system directory programs



NETWORK OPERATING SYSTEM (NOS) (Abbreviated Version)

The Network Operating System (NOS) provides time-sharing, remote batch, transaction processing and batch processing capabilities for users of the CDC CYBER 170, CYBER 70, and 6000 computer systems. NOS is capable of providing this support whether the computer system is configured singly, in a multiple main-frame array, or as an element of a network.

The key to the ability of NOS to provide interactive processing while at the same time supporting local and remote batch processing, lies in the modular structure of the operating system. Selection from among four subsystem executive modules allows a user (either during initial installation or a future expansion) to specify an NOS configuration capable of up to six concurrent modes of operation.

The six modes of operation, and the subsystem module(s) required to support operation, are as follows:

- Remote batch (RBF) – the submission of jobs from a remote batch terminal for batch execution
- Local batch (BATCHIO) – the submission of jobs from the card reader or tape unit at the central site
- Time-sharing (IAF) – supporting an array of interactive terminals during execution
- Deferred batch (IAF) – the submission of batch jobs from an interactive terminal; this mode makes all batch resources available to interactive terminal users
- Conversation batch (IAF) – the submission of jobs in batch mode, along with data and control cards, from an interactive terminal
- Transaction processing (TAF) – support of transaction-oriented terminals with multiple users maintaining and accessing data bases

This capability of supporting a remote terminal network, in addition to and concurrent with batch processing, makes the speed and computing power of the computing system available to many more users.

FEATURES

The NOS system offers excellent response time for large numbers of interactive users, while simultaneously providing batch processing on a local or remote basis. Salient features of the NOS system include:

- Support of large numbers of time-sharing terminals
- Fast compilation rate; FORTRAN compiles at 12,000 statements per minute with the capability of compiling directly to core for execution, thereby eliminating storage and local time.
- High-performance USASI-standard COBOL compiler
- Dynamic allocation of central memory which assures maximum usage of system resources
- Complete hardware protection is provided to running programs by means of hardware relocation and storage field length registers

- An ability to link jobs using the job control language
- Object codes which can be saved and later executed
- Permanent file utilities
- A capacity of up to 23 jobs in central memory as determined by their size
- Text editor which permits both line and character string manipulation
- Security – using a password mechanism and a user validation file, a detailed security procedure is available
- Recovery – recovery is available for both the system and the user in the event of system malfunction or terminal line disconnect
- Accounting – a hierarchical resource accounting system is available utilizing user numbers, project numbers and charge numbers.
- Application library – an extensive application library is available to terminal and batch users
- Scheduler flexibility – the scheduler can be tuned to meet any foreseeable job balancing requirements
- Low system overhead – CPU overhead is low and requires a minimal amount of memory
- Tape labels (ANSI Standard) – implementation of tape label processing for ANSI-compatible labeled tapes (including multifile, multivolume labeled tapes)
- The MODIFY system which allows for maintenance of program source libraries.

Time-sharing via the Interactive Facility (IAF) is the most widely known mode of NOS system operation. Time-sharing has been defined as having an operating system which is capable of supporting many interactive terminals concurrently. The user's definition of time-sharing is having a terminal which can be used to enter source code line-by-line. The abilities to edit that source code, communicate with the program when it enters execution (either entering data or monitoring the job's progress), and to receive the output at the terminal.

The interactive programming languages which are available to the time-sharing terminal user are:

- FORTRAN
- BASIC
- APL
- ALGOL 60
- Other†

†Any source code "file" may be constructed, edited, and submitted for execution via NOS. This executing program can query the terminal for input, and can output data to the terminal.

Remote Batch

Remote batch processing is the ability to submit jobs to the system from a batch device located away from the central site. The jobs submitted are executed as batch jobs with the output returned to the originating device or optionally to the central site. These jobs are normally separated from the jobs submitted at the central site and given higher priority.

Since the jobs are processed as batch jobs, the complete product set which is available to the batch user is also available from the remote batch terminal.

Local Batch

Local batch refers to the class of user jobs which are submitted at the central site. Input for these jobs will come from the local on-line printers. The jobs submitted in batch mode are run to completion and do not permit interaction with the user.

The product set available for this class of jobs is the entire product set of the system, including the special interactive products such as FORTRAN Extended with T/S option and BASIC which can use standard input and output files instead of terminal input/output. A file of source code created at the time-sharing terminal can be executed in batch mode under the time-sharing compiler without conversion.

Deferred Batch

Deferred batch refers to the ability of the time-sharing user to submit jobs to be run in batch mode. This feature offers a tremendous expansion of time-sharing capabilities. The user is no longer restricted by the interactive terminal's characteristics or the defined time-sharing product set. Since deferred batch jobs submitted by the time-sharing user run in the batch environment, there are no constraints on the type or peripheral equipment that can be used. While the job is initiated from the interactive terminal, the output can be directed to a remote batch station of the central site rather than being returned to the interactive terminal. This feature truly gives the user both features – batch processing and the inherent time-sharing capability associated with the terminal.

Conversational Batch

Conversational batch processing not only affords an interactive terminal user all the same processing capabilities and resources as deferred batch, but provides the added feature of being able to receive the output at the originating terminal.

Transaction

Transaction processing allows a terminal user to initiate pre-written programs. Guided by the program, this user may interrogate or update a data base, process the data or send messages to other terminals. A programmer writes these programs, called tasks, and stores them for access by the TAF executive. A single copy of a task can be used by many users and up to 31 different tasks can be multiprogrammed at one time. Two data managers are available: the TAF data manager and a version of Cincom's TOTAL. A task may use either or both data managers. Only one copy of each data manager is used, regardless of the number of terminals or tasks in use.

The programmer is not concerned with the number of users of the task; TAF takes care of this. Nor is the programmer concerned with the terminal network interface or terminal configuration. A terminal name is specified to designate the origin or source of data and all further interfacing is done by the NAM product.

Applications for TAF include message switching, data base query and update, or multiple uses of computational programs, or a mix of these. Tasks may be written in COBOL, FORTRAN, or COMPASS.

PERMANENT FILES

A key feature of the NOS system is its permanent file capability. With this feature the user can securely save a file containing a program or data and access it at some future date. The major advantages of the NOS permanent file system are flexibility, ease of user, and security.

Two basic types of permanent files are available to the NOS user: indirect and direct access. Indirect access files are usually smaller files. The term indirect access means exactly what it implies — that when a user accesses the file, the user receives a working copy of the file which can be executed or modified. Thus, any changes the user makes to the file are retained only if the working file is saved. Direct access files, on the other hand, are files on which changes are made directly without the use of a working copy. For this reason, a write interlock feature is provided to prohibit more than one user from attempting to write on the file at the same time. Because of their generally large size, direct access files are often used for large data bases.

SECURITY

All users are validated for system resource usage through procedures which are required at log-in or job submittal. A validation file containing a list of resource limitations for each user is maintained by the system. The user may be required to provide a proper user number, charge number, and password to gain access to the system. His resource utilization is then monitored to ensure he remains within authorized limitations.

Because of the password system employed by NOS, a user's permanent files can be protected from unauthorized access. In addition, the user can specify that a file password be submitted before a file can be accessed. This increases the protection afforded a user's files. Finally, several levels of permission can be granted to other users. This enables the user who creates the file to prevent other users from damaging its contents.

User file protection includes private, semi-private, and library categories. Access permissions include write, modify, append, read, read with append, read with modify, and execute.

NOS CONTROL LANGUAGE

Another significant feature available to the user is the CYBER Control Language (CCL). CCL allows the programmer to transfer control and perform arithmetic and test functions within the control card record. These control language statements are similar to COBOL statements. The special control language statements provided are SKIP, BEGIN, DISPLAY, SET, IF, and FILE. These statements are combined with constants, arithmetic operators, relational operators, Boolean operators, and functions to form CCL expressions. In addition, a series of symbolic names is used to reference values that pertain to the job process.

RECOVERY

The NOS system is designed to permit recovery in the event of a system malfunction and to permit the terminal user to recover files whenever the system is restarted following such a malfunction. These two features are invaluable in the event of system failure. In addition, a checkpoint/restart capability is available for both jobs and the operating system.

ACCOUNTING

All accounting information is recorded in the accounting dayfile and is associated with user identification numbers. The accounting dayfile is completely separate from the console dayfile and contains terminal log-in and log-out times along with total CPU time. The accounting dayfile can be listed off-line.

OPERATING SYSTEM REQUIREMENTS

Listed below are approximations of central memory utilization by the various positions of the CDC CYBER system software.

Operating System	-	7,000 to 8,000 words
Remote Batch Facility	-	7,000 to 9,000 words
Local Batch Executive	-	7,000 to 10,000 words
Network Access Methods	-	13,400 words
Interactive Facility	-	5,500 to 6,000 words
Magnetic Tape Executive	-	850 words
Transaction Facility	-	6,000 to 15,000 words



NETWORK OPERATING SYSTEM (NOS)

EVOLUTION OF THE NETWORK OPERATING SYSTEM (NOS)

Since the early 1960s, there have been at least two major operating systems for Control Data's large computers - 6000 Series, CYBER 70 and currently CYBER 170 computers. This limited diversity, plus the hardware compatibility of these product lines, has allowed Control Data to evolve operating systems and software products of superior design and proven quality.

By the 1970s two operating systems, SCOPE and KRONOS, had emerged as primary systems emphasizing different types of workloads. The SCOPE operating system has always been oriented toward a batch and scientific environment, while KRONOS was oriented toward a commercial/university time-sharing environment. Although each performed well in those environments, there were overlapping requirements and the need for a common system.

Control Data recognized this and by the mid-1970s had developed a base of common software products for both systems and had made the decision to go to a single standard system. The choice of system design was not simple since each had strong points in different capabilities. SCOPE had good batch support but KRONOS was superior in time-sharing. However, KRONOS batch support was not inferior to that of SCOPE and could thus offer better all-around support. Consequently, the KRONOS system design became the base of the Network Operating System (NOS). SCOPE support was continued, however, as the Batch Entry version and became known as NOS/BE.

Since then, the process of merging the desirable features of NOS/BE into NOS and extending capabilities has continued. Today, NOS represents the best set of features and capabilities for customers of Control Data's CYBER 170 and predecessor series computers which Control Data can offer. Control Data is committed to support this primary operating system into the future and on CYBER 170 successor products. The following description highlights many of the features and capabilities of NOS.

NETWORK OPERATING SYSTEM (NOS)

GENERAL CHARACTERISTICS

Overview

The Network Operating System (NOS) provides time-sharing, transaction processing, and batch processing capabilities for users of the CDC CYBER 170, CYBER 70, and 6000 series computer systems. NOS is capable of providing this support whether the computer system is configured singly, in a multiple-mainframe array, or as an element of a network.

The key to the ability of NOS to provide interactive processing while at the same time supporting local and remote batch processing lies in the modular structure of the operating system. Selection from among four subsystem executive modules allows the operator to dynamically specify a NOS configuration capable of up to six concurrent modes of operation. The six modes of operation, and the subsystem modules required to support each mode, are as follows:

- **Remote Batch (RBF)** – The submission of jobs from a remote batch terminal for batch execution.
- **Local Batch (BATCHIO)** – The submission of jobs from the card reader or tape unit at the central site.
- **Interactive (IAF)** – Supporting an array of time-sharing terminals and providing an environment for the development, testing and interactive use of programs.
- **Deferred Batch (IAF)** – The submission of batch jobs from an interactive terminal for detached execution. This mode makes all batch resources available to interactive terminal users.
- **Conversational Batch (IAF)** – The interactive execution of batch job control statements and data entered from an interactive terminal. This mode also makes batch resources available to terminal users but in a conversational mode.
- **Transaction (TAF)** – Support of transaction-oriented terminals with multiple users maintaining and accessing separate data bases via a library of transaction processing programs.

This capability of supporting a remote terminal network, in addition to and concurrent with local batch processing, makes the speed and computing power of the computing system available to a large number of users.

Advantages

Control Data believes NOS offers the customer distinct advantages through features that are often not found in other vendor's products. These features provide for the ease of use and high operational productivity for users of NOS systems. Control Data will provide stable system software which has been developed and proven over many years of use, and yet has been continually improved to provide the advantages of the latest software features in the industry. Most importantly, once delivered, the system is not forgotten. Control Data's Systems Division is highly responsive in solving the customer's unique needs. Control Data's field support provides timely response to customer identified deficiencies.

The Network Operating System supports the latest technology in mass storage, including the Mass Storage Subsystem (MSS). The MSS provides an automatic system for storing and accessing data on a magnetic tape cartridge medium. Using MSS, extremely large data bases can be retained on-line. MSS provides an extension to permanent file storage on disk, providing control of movement of files between disk and the MSS transparent to the user. Users will have the option of specifying preferred residence of files on disk or MSS.

In addition to MSS and disk, NOS supports another form of mass storage, that of Extended Core Storage (ECS). ECS provides up to 2 million 60-bit words of extended memory which is directly accessible to user programs and system routines. ECS provides for high speed block transfer and fast access to data files or programs, resulting in improved performance for many applications.

Another advantage Control Data offers is a comprehensive software accounting system. Some systems in the field today offer very little resource accounting, leaving the customer with a poor record by which to fairly charge users for resource utilization. The NOS accounting system offers a detailed hierarchical accounting structure to control projects.

In addition, NOS provides the system operator with an efficient and easily tuned system. NOS provides the keys by which the system may be fine-tuned to maximize the performance for each unique customer.

The operator of a NOS system has optimum control. System balancing of the central processor allocation to jobs is basically the adjustment of the scheduling processes. The concept of time slices and time slots to allocate time to the CPU and central memory provides for the balance of central processor utilization among jobs. The operator has control over this balance by having the capability to adjust the Central Processing Unit (CPU) priority set at job initiation, the CPU time slice, and the central memory time slice. In addition, the operator has control over upper and lower bounds for dynamic queue priority levels, as well as the increment at which the priorities are aged. The operator can also control the maximum number of jobs in memory and the maximum field length. The operator can control the relative service rates to jobs based on their job origin classification (time-sharing, local batch, and remote batch). On the other hand, the operator is not required to manually control the system, as NOS is a totally automatic multiprogramming system.

ACCOUNTING CONTROL

The NOS system provides a multilevel accounting system that records and accumulates detailed statistics on system and job usage.

The special system file, PROFILA, contains the information required to control a user's accounting and access to the system. This access is controlled not only by charge numbers and project numbers, but also by time in, time out, expiration for charge and project numbers, accumulated System Resource Units (SRUs), and up to eight accumulated resources defined by the installation.† In addition, all exercises of this access by individual users is written by the system to the accounting dayfile, thereby affording the customer a time-log as a basis for account billing.

PROFILA affords three levels of job accounting, charge number, project number and user number, as depicted in Figure 1.

- Charge Number

This is the primary division of the customer's job structure. It is a 1- to 10-character billing identifier. Charge numbers can only be entered onto PROFILA by an analyst using a system origin job or by a special accounting user, and their associated parameters may be changed according to the hierarchy of access. Associated with each charge number is a Master User, who administratively controls resources for each project and user. There is no logical limit to the number of charge numbers.

†Limit and accumulation fields for these eight installation-defined resources have been reserved in PROFILA and are checked for exhaustion of the corresponding resource by routine CHARGE. However, the system provides no facility to dynamically update the accumulation fields as these resources are being used.

- **Project Number**

This is an optional second level division of the charge number. It is a 1- to 20-character identifier of a particular customer project. The project number can be followed by time-access parameters to this project. The maximum number of project numbers is 4095 per charge number.

Users who have been declared master users can enter and change project numbers and their associated parameters.

- **User Number**

The third level is a 1- to 7-character identifier of the individual user who is allowed access to a designated customer category or in different ones. The maximum number of user numbers is 131,000 which can be optionally limited by an installation parameter.

The basic accounting unit for NOS is the SRU. The SRU is a measurement of the resources used by a job or a terminal session. The SRU algorithm combines measurements of the following resources into a single unit.

- Central memory field length
- ECS (Extended Core Storage - optional) field length
- CPU time
- Mass storage usage
- Magnetic tape usage
- Permanent file usage

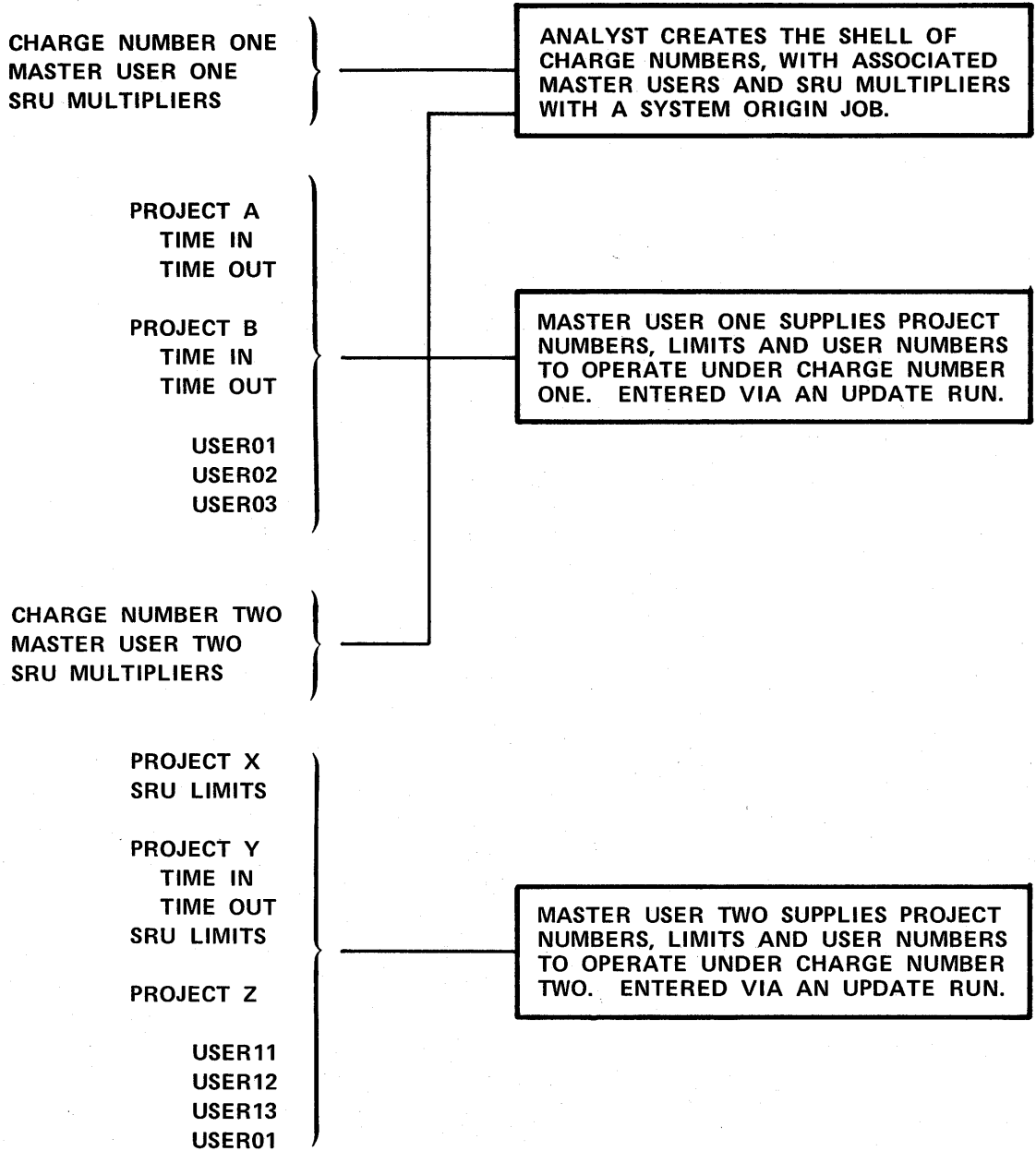


Figure 1. Representative Structure of PROFILA File

The SRU calculation is dynamic, that is, each time additional amounts of the above resources are utilized by the job or session, the SRU value is updated. Multiplier parameters are available to allow capability to vary the weighting of each category to the total SRU computation, thus providing installation flexibility.

The following summations of job activity are added to the end of the user's dayfile. This information is also issued to the associated account dayfile. The entries in the account dayfile also include the job name. Each entry is annotated with time-of-day.

- Application charge activity in kilounits.
- Permanent file activity in kilounits.
- Mass storage activity in kilounits.
- Magnetic tape activity in kilounits.
- Accumulated central processor time in seconds.
- SRU value in units for total job usage including CPU time, input/output activity, and memory usage.
- Lines printed in kilolines.

The following information is issued to the account dayfile only.

- Cards read in kilocards.
- Cards punched in kilocards.

In addition to these items of accounting information, many messages too numerous to delineate are included regarding system usage.

SECURITY CONTROL

NOS provides a high degree of security for system access and utilization through a system of validation and resource limitation controls. All users are identified by means of user number and password. A validated user's resource limitations are imposed through restrictions maintained in validation files.

Many of the security features of NOS are enforced through the job origin concept — only jobs of a given origin type can perform certain operations. A system origin job must be initiated from the system console, usually by a site analyst and may perform privileged tasks that a terminal or batch user may not perform. It is expected that the site exercise the constraints necessary to limit access to the operator console and the privileges associated with being system origin.

SYSTEM ACCESS

This subject covers user access to the system and the accounting (for subsequent billing) of the activities performed by the user.

The control of user access (validation) and user accounting is based on two system permanent files: VALIDUS (validation file) and PROFILA (project profile file). These files identify who can use the system and what system resources may be used and projects charged. These files can be manipulated through

special system jobs that require system origin. Since these files contain user and project numbers, the knowledge of these values should be limited to the user identified by these values and the site personnel that maintain these files.

The system provides for the suppression and secure entry of passwords, charges and project numbers. The user can use system mechanisms so that these values do not appear in hard copy form in his job dayfile or hard copy terminal output. Thus, these values can remain known only to the user and those responsible for maintaining them.

Security Count

To protect against a user who deliberately attempts to determine valid user numbers by attempting invalid USER statements when secondary USER statements are permitted or by submitting jobs with invalid USER statements, a security count is available in each user's validation file entry. The security count is decremented each time an invalid USER statement is detected. When the security count reaches a zero, the user will no longer be granted access to the system. Only the resetting of the security count by the operator will give the user access to the system again.

Validation Files

The system validation and project files are used to validate user access to the system. Validation defines and controls the following:

- Who can use the system
- What resources can be used (hardware and software)
- To what extent these resources may be used.

Access and resource usage validation is on a positive authorization basis. Thus the user must be known to the system for access and each resource must be specifically authorized and quantified where applicable.

Every user of the system must have a valid user number (if VALIDATION is enabled). In a batch environment this means that the statement following the job statement must be a USER statement.

If the user is required to be further validated for accounting purposes, a CHARGE statement must be the next statement in the job. The CHARGE statement causes the CHARGE routine to be loaded to validate the charge and project numbers to which the user is charging his activities.

Thus, the validation procedure allows the system to:

- Determine if a user is allowed to use the system.
- Charge the user for his resource usage.
- Restrict the user to certain resource usage, including denying access to the system when the project's SRU limit has been reached.
- Maintaining permanent files for the user and control access and security of them by validating the user's number against an access control list.

The following resources are individually validated for each user number.

- Time limit
- SRU limit
- Lines printed
- Cards punched
- Number of local files
- Central memory field length
- ECS field length (optional)
- Number of deferred jobs
- Number of permanent files
- Length of individual direct access file
- Total length of indirect access files
- Mass storage space
- Control statements processed
- Dayfile messages issued
- Output files disposed

FILE ACCESS

The permanent file subsystem has a variety of mechanisms whereby a user may control the access to his permanent files. These mechanisms include the file category (public, private, and semi-private) and the file password. The system provides the secure entry and suppressing of permanent file passwords so that these values do not appear on hardcopy output. The user may explicitly provide other users to access permissions to his permanent files by the use of the PERMIT control statement. Access can be restricted to Read, Write, Execute or various combinations thereof. In addition, the user can ensure local files are cleared from central memory after job step completion by using the PROTECT macro.

SECURE SYSTEM MEMORY

The Secure System Memory feature of NOS prohibits a user from accessing data in central memory after the program that brought the data into central memory has released storage, been storage moved, rolled out, completed, or aborted. Secure system memory involves prohibiting the dumping of privileged field length and clearing of memory when loading a new program or increasing the field length. This feature is invoked by a special entry point parameter for the program.

This feature prevents access to data that may be left as a residue from other jobs or job steps that have manipulated privileged data or files. For example, if the LIMITS command was aborted and the field length dumped, the dump might expose validation file data to unauthorized access if the secure system memory feature was not available.

MASS STORAGE PROTECTION

The philosophy of data security followed by NOS is that it should be impossible for a user to access the data that may be a residue in his own or another's field length. The secure system memory feature is just one example of this. Another area where data protection is done is the area of permanent file length errors. If a permanent file length error is detected, the file being retrieved is not given to the user as the data currently residing on the tracks specified for the file in error may contain someone else's data. When assigning mass storage to a job, only the area written upon may be accessed by the user.

OPERATING SYSTEM PROTECTION/MODIFICATION

The system may be operating under special modes in which system modification operations may be performed by authorized personnel. If the system is placed in DEBUG mode, then many of the controls on user operations are relaxed if the user has system origin privileges. If the system is placed in UNLOCKED mode, then certain console commands, including the ability to alter memory, are not prohibited. These special console modes should not be used during normal system operations as their function is to permit software debugging (DEBUG) and hardware maintenance (ENGINEERING) as well as protecting against accidental operator entries that may impact system operation and performance (UNLOCK). The NOS Operator's Guide, Publication Number 60435600, and NOS Installation Handbook, Publication Number 60435700, detail the console commands and installation parameters that enable/disable special console modes.

USER ACCESS PERMISSIONS

For each user there is a validation file entry. This contains a word with certain access permissions. The setting of various bits in the word allows the user to perform certain operations that the site may wish to control. Of particular note is the system origin privileges permission. The site should exercise caution in selecting which users may have system origin privileges, as this permission excuses the job from certain system controls if the system is running in DEBUG mode.

The site should take care in implementing their own access permissions so as not to remove the controls provided by NOS. NOS currently allows the following specified permissions:

- User may change his password
- User may use ACCESS commands from terminal
- User may create direct access permanent files
- User may create indirect access permanent files
- User may have system origin capability from any job origin if system is in DEBUG mode. User may also assign a device by specifying its EST ordinal (DEBUG not required). User may use on-line diagnostics if system is in ENGINEERING mode.
- User may access system library files
- User may request nonallocatable devices (for example, magnetic tapes)
- User may use the system without entry of charge or project numbers
- User may issue auxiliary device commands

- User may access special transaction functions
- User is not logged off because of timeout
- User has special accounting privileges
- User may use the System Control Point (SCP) facility

ERROR DETECTION AND RECOVERY

NOS provides extensive error checking, correcting, and recovery to maximize system up-time. Error recovery under NOS occurs at three different levels: automatic by hardware, automatic by the system drivers, and under operator/user control. Reliability was given a major consideration during design of the CDC CYBER 170 series systems. Examples of each type of error recovery are outlined as follows.

CENTRAL MEMORY ERROR CORRECTION/DETECTION

Forward error correction techniques include Single Error Correction/Double Error Detection (SECEDED) for central memory, correction of up to 11-bit bursts of error from the disk storage unit, and reread correction from the magnetic tape units. SECEDED is accomplished entirely in the hardware with no degradation whatsoever in performance. The disk controller continuously generates or checks a Cyclic Redundancy Code (CRC) and furnishes a correction vector to the system driver upon failure. The tape controller (which contains a microprocessor similar to the disk controller) can do correction on data, but rereads the data automatically in order to accomplish this.

Central memory may be reconfigured (degraded) upon failure of a memory area by switch deactivation of the failed memory bank. Upon a recovery deadstart, the system may then continue processing.

Similarly, non-critical failing peripheral equipment can be deactivated. The operator can logically turn-off a defective device while continuing system operation on remaining devices.

PERIPHERAL ERROR RECOVERY

Peripheral Processor (PP) memories, input/output channels, and all input/output components are protected by parity. Parity errors in PPs or channels are recorded in the status and control register which is a collection point for many kinds of hardware/software status. This register is monitored by the system software, for example, to report SECEDED errors to the error file or to notify the operator of impending interruption in the case of power failure. A system error log is maintained on mass storage and includes all hard and soft error events.

All PP software drivers of input/output devices are programmed to retry in the event of any kind of uncorrected error. The PP drivers can detect and differentiate between data failure and address failure to disk and implement different recovery measures. All this driver and system level recovery is transparent to the user, although the information is available to the user if desired.

The user can ask for write verify on disk and has the option to retry any failures that have exhausted the automatic retry facility of the system. Operators can reload individual PPs without downing the system and may change tape units and reinitiate jobs in concert with the system.

CHECKPOINT/RESTART

In terms of system recovery, NOS provides for four levels of deadstart, extending from complete recovery of the system state to complete reinitialization of the system. NOS provides for system checkpoint/restart, which recovers the system from the specific state at which it existed at checkpoint time.

In addition, an individual job checkpoint/restart capability exists. Users may checkpoint the status of their jobs at selected points for recovery in the event of job abort at some later time. This is especially useful for long duration jobs.

POWER FAILURE RECOVERY

The CDC CYBER 170 computer system includes a ride-through motor generator capable of providing continuous power to operate the system through intermittent power interruptions of 2.5 seconds or less. In the event of power failure or other environmental failure the system automatically begins a system checkpoint operation followed by a step mode condition. A message is displayed on the console: SHUTDOWN IMMINENT.

During system checkpoint, the system provides for job state preservation and writes the contents of central memory to mass storage. Step mode prevents any further central memory input/output operations and any further PP requests.

Following restoration of the power and environmental conditions necessary to resume operations, the message POWER/ENVIRONMENT NORMAL appears at the system console. Upon entering of the command UNSTEP, messages indicating time of power failure and contents of the error (Status/Control) register are entered into the error log along with time of return to normal condition. Processing then may be restarted.

It should be recognized that under certain circumstances sufficient time to fully checkpoint the system may not exist. In these circumstances only a partial recovery of central memory can be achieved. Even then, however, input/output queue, the job rollout queue, and permanent files are preserved for system restart.

OPERATIONAL CONTROL

SCHEDULING PRIORITIES

Each job origin type has a separate input queue to allow for priority biasing of one job origin type over another. This is accomplished by setting the entry level priority for each origin type (time-sharing, local batch, remote batch system origin) to the desired level. Figure 2 illustrates the priority scheme for job origin types. Priorities are dynamically aged upward for entries in each queue.

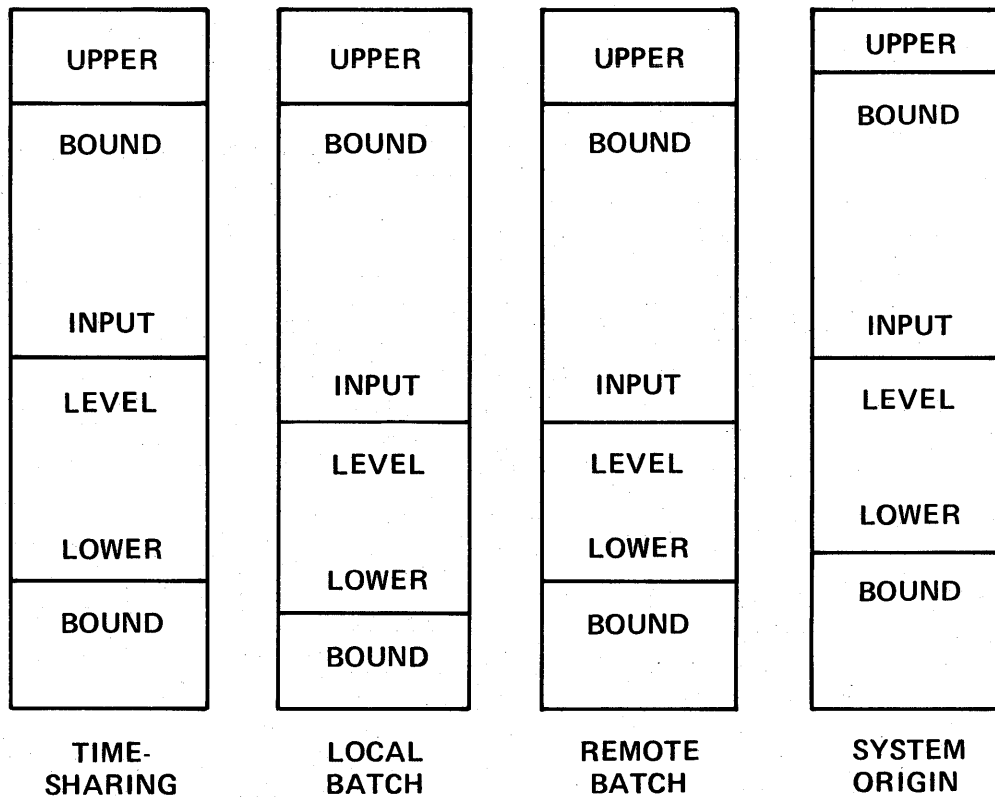


Figure 2. NOS Priority Scheme by Job Origin Type

NOS automatically controls the scheduling of up to 23 jobs in central memory in concurrent execution, with many additional jobs in various queues on mass storage. Queued jobs are assigned to "control points" in central memory on a priority basis. If no control points are available, queued jobs are assigned to memory when their priorities exceed priorities of jobs currently occupying control points. Low priority jobs are rolled out to mass storage, as required.

Interactive users are provided minimum response time to interactive commands (not involving execution or compilation) by assigning the Interactive Facility (IAF) subsystem to a control point with a special system level priority. IAF executes a job except it has the highest priority and is exempt from rollout or storage moves. This allows NOS the ability to provide rapid response time even when the system has a significant batch load.

The NOS system dynamically ages the individual job priorities, and schedules resources in an optimum manner while retaining the flexibility for complete operator control, when desired.

Tape units are assigned automatically by a resource scheduling algorithm. Volume Serial Numbers (VSNs) are prescanned and jobs are scheduled to prevent deadlock. If tape units/reels are unavailable, jobs are rolled out until they become available.

OPERATOR COMMUNICATIONS

The operator and the system communicate through the console keyboard and console CRT display screen. The system displays information about job and system status on the console screen. Data entered from the keyboard is also displayed. A permanent record of all system/console communication is retained on the system dayfile and may be printed at operator request.

The system console is normally controlled by the DSD program, however, other display programs may be temporarily invoked. The primary functions of DSD are to maintain a current display of system status and to process keyboard entries from the operator. At the console keyboard, the operator may assign equipment, exercise control over job scheduling and execution, initiate utility programs, and select displays. The CDC CYBER 170 series console keyboard contains a PRESENTATION CONTROL switch which allows the operator to display a left screen display only, a right screen display only, or both the left and right screen displays on a split screen.

These status displays take full advantage of the display console capabilities. All displays are structured in a meaningful arrangement with most layouts remaining static while the information changes according to activity. The rate of change is well above the perceived flicker rate for the human eye. Thus it is possible to observe the flow of system activity and even to correlate activities by their simultaneity or rhythm. For example a job entry may appear briefly in a queue display on one side of the screen only to reappear as a job at a control point on the right screen. Then while executing, job status characters will toggle through different settings such that they appear to be multiple characters overprinted, although the most frequent status will stand out. If the job must be rolled out it will vanish (or be replaced by another job) and reappear in the rollout queue. Such displays are uniquely suited to the visual capabilities of the human eye and thus maximize the delivery of status information.

On the converse, operator input via the console keyboard must by definition be relatively slow and serial in nature. This problem is minimized by using highly abbreviated entries, the use of certain keys for special functions in certain contexts, and finally by an interpretive keyboard concept. In this concept a partially entered command is checked against the directory of commands and when enough characters are entered to uniquely identify a valid command, the remainder of the command is supplied (except for parameters), and the displayed entry is ripple flashed indicating recognition. This plus several other operator convenience features quite literally allow the console operator to flip through a series of displays at the touch of a button.

Any of the DSD displays can be selected by the console command*:

xy

where x and y represent the letter designation of the displays; x appears on the left screen and y appears on the right. If x and y are identical, both screens display the same information. The displays available to the operator are:

<u>Letter Designation</u>	<u>Display</u>	<u>Description</u>
A	Dayfile*	Chronological history of system operations.
B	Job Status	Current status of all jobs assigned to control points.
C,D	Central memory	Contents of 32 central memory words (four selectable 8-word groups) in five columns of four octal digits with display code equivalents.
E	Equipment status	Status of peripheral devices.
F,G	Central memory	Contents of 32 central memory words (four selectable 8-word groups) in four columns of five octal digits with display code equivalents.
H	File Name Table (FNT)	List of FNT entries for all active files in the system.
I	BATCHIO status	Status of central site unit record devices.
J	Control point status*	Status of the specified control point.
K,L	CPU programmable*	Dynamic operator/CPU job communication.
M	ECS display	Contents of 32 60-bit words of ECS memory (four selectable 8-word groups) in five columns of four octal digits with display code equivalents.
N	File display	Contents of any file assigned to an FNT ordinal. Display is initially selected with the DISPLAY,xxx. commands.
O	Transaction status	Status of the transaction subsystem.
P	PP communications area	Current contents of PP registers.
Q	Queue status	Status of active input, output, and rollout queues.
R	RBF status	Status of remote batch operations
S	System control information	Parameters used to control job flow.
T	Time-sharing status	Status of time-sharing users.

*This display is control-point oriented. Paging forward and backward through the display for each control point is achieved with the + and - keys, respectively. The number of the control point also appears at the top next to the letter designator (for example, A5).

<u>Letter Designation</u>	<u>Display</u>	<u>Description</u>
Y	Monitor functions	List of all monitor mnemonics and codes.
Z	Directory	List of the letter designators and descriptions of all DSD displays.

The following DSD command allows the operator to preselect the left screen display sequence.

SET,ssss.

ssss Letter designating any four of the DSD displays listed. Note that four display identifiers must be specified.

Pressing the right blank key after this command is entered causes the first display specified to appear on the left console screen. Pressing the key again selects the second display. The next display in the specified sequence appears on the left console screen each time the right blank key is pressed.

NOS provides the system operator with a high degree of flexibility in controlling the system by entering changes to system parameters which affect overall performance or operating characteristics. Initially, these parameters are entered at installation time on the deadstart tape. Later, during operation the operator can change them via console keyboard entries.

The commands which may be entered at the console to control the system include:

- ACCOUNT - Enable or disable account number validation.
- VALID - Enable or disable user number validation.
- USERS - Enable jobs to use secondary USER statements (user may specify a different permanent file catalog during processing).
- AUTOROLL - Enable or disable automatic job rollout.
- BATCHIO - Enable or disable batch processing.
- DEBUG - Enter system modification mode.
- ENGR - Enter or clear system engineering mode (diagnostics).
- QUEUE - Specify queue priorities for input, output, and rollout queues for each job origin type.
- SERVICE - Specify the service limits for CPU priority, CPU time slice, and the maximum number of jobs, time-sharing users, field length of a specific job, field lengths of all jobs, size of permanent files, etc.
- DELAY - Specify system scheduling interval parameters, and threshold count for corrected single bit errors.

In addition to these parameters, a number of additional parameters may be controlled from the system console.

MONITORING PROCEDURES

Control Data recommends that the customer consider the periodic use of the Performance Analysis Services offered by Control Data Canada, Ltd., Professional Services Division (PSD). Standard services consist of using software monitors to measure both internal and external performance. Internal performance data consists of major resource utilizations (e.g., CP, CM, PP, channels), and external performance data consists of throughput rates, response times, job elapsed times, etc.

A large repertoire of tools have been developed and used by PSD performance analysts. The primary tools used in this service include:

- **HEMI (Hybrid Event Monitoring Instrument)**

A hardware/software monitor capable of providing utilization data for major resources (e.g., CP, CM, PP, channels). For most applications, the software monitor is sufficient.

- **DFA (Dayfile/Account file analyzer)**

A sophisticated dayfile/account file analyzer used for workload analysis and program evaluation.

- **Dialog Monitor**

A software monitor capable of measuring response times for all interactive commands.

Other tools available include:

- **PMP (Program Monitor)**

A program monitor used to optimize performance of individual programs.

- **CPM**

A system performance simulation model for lower CDC 6000s, CYBER 70s, and CYBER 170s. This model can be used in answering many system performance problems, and is especially useful in answering "what if" type of questions.

FILE SYSTEM

FILE DEFINITION UNDER NOS

The fundamental unit of data organization under NOS is a file. A user's interaction with the system is in terms of files.

- The job a user submits for processing is a file.
- Collections of data included with the job are transferred to the system in named packets; each packet is a file.
- Control statements in the job can designate a name as a file; this is an empty file to which other statements or programs can add data.
- Control statements in a job can copy all or part of an existing file; the copy is a file.
- The output from job processing is a file.

A file may be empty or as large as the user's resource limitations permit.

The user can subdivide a file into logical records and he can link several files together into a multifile file. The user does this subdividing either by means of parameters the user includes in programs or control statements that create the file, or the user can insert delimiters in the data he submits with a job.

The user subdivides a file into records when the user wants to access these subdivisions without accessing the entire file (for example, a master inventory is divided into parts categories). On the other hand, the user links several files into a multifile file when they fit into a single category he wants to access with a single reference (for example, student rosters from a number of classes are consolidated into one department roster).

All files are identified and accessed by a file name that is defined by the system or the user. The user names his file in the program or control statement that initiates its creation.

CATEGORIES OF FILES

There are two basic categories of files under NOS: **local (working) files** and **permanent files**.

Local Files

A local file has the following characteristics:

- It is created by the job being processed. It can be accessed only by that job.
- It is no longer accessible when job processing terminates (i.e., the file is closed and the space reallocated).

Since local files are available only to the job that creates them, they are said to be local to that job.

When local files are created at an interactive terminal, they are referred to as working files.

Permanent Files

Permanent files are saved at one terminal session (or job) and can be retrieved at subsequent terminal sessions (or jobs). Each permanent file that you create is identified by a name. In general, only the user who originally saves a file can retrieve, modify, or delete the permanent file. (Methods for granting permanent file access to users other than a file's creator are available.)

There are two types of permanent files, direct access and indirect access. When an indirect access file is retrieved, the system makes a copy of the permanent file available for use as a local file at the terminal or with a batch job; when a direct access file is retrieved, the permanent file in storage is itself available for use at the terminal or with a batch job.

Permanent files are protected by access restrictions through the user of user number, file name, and password validations. They are further protected by mode restrictions which the owner may apply (read only, read/write, read/append, append only, execute only, etc.). These restrictions apply to users who must be granted access permission by the owner. The owner may grant universal access (PUBLIC file or SEMI-PRIVATE file), or explicitly grant access to specific users (PRIVATE file).

FILE STRUCTURES

The nucleus module of the NOS file system is the CDC CYBER Record Manager (CRM) (see Figure 3). Among the options are several file organizations with various record types from which the user can select.

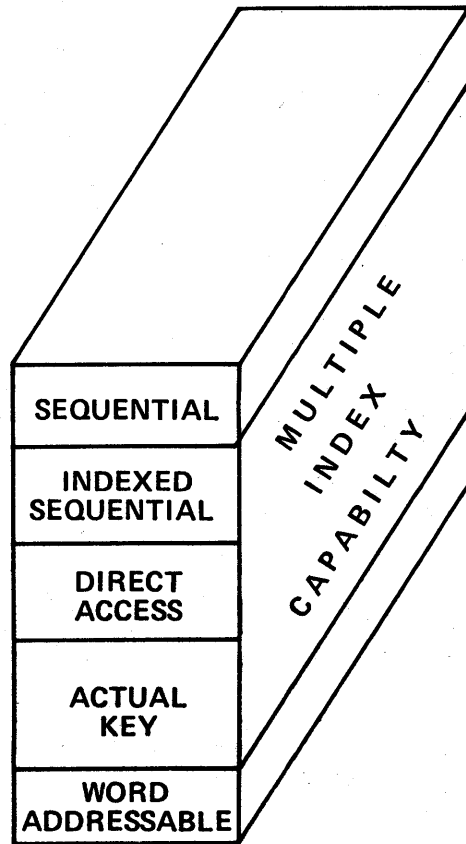


Figure 3. CRM File Processing Capabilities

The multiple-index capability supplements indexed sequential, direct access, and actual key file organizations by supplying multiple-access paths to data records, in addition to the primary key access path. The multiple-index capability allows records to be accessed both sequentially and randomly through separate alternate keys, as well as the primary record key. Figure 4 shows the placement of the keys in a multiple-index file. When the file is updated, the user need specify only the primary key of a record; all other index updating is performed automatically. Since it provides multidimensional access to records, the multiple-index capability is of particular value to the data base designer. Complex record structures (trees, networks, and so forth) can be created because a data value in one record can be used as a key to locate other records (Figure 5). There is no physical limit to the number of keys imposed upon a hierarchical structure.

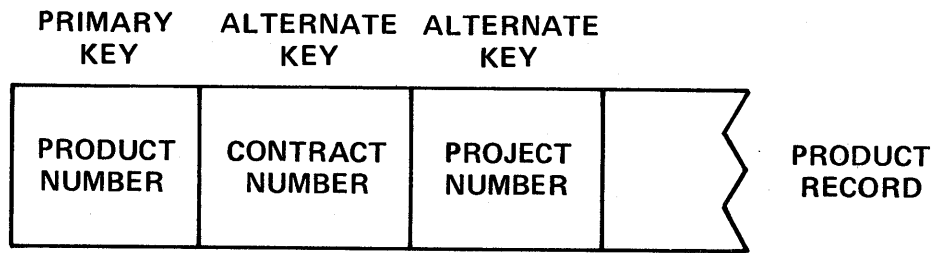


Figure 4. Multiple-Index File with Multiple Keys

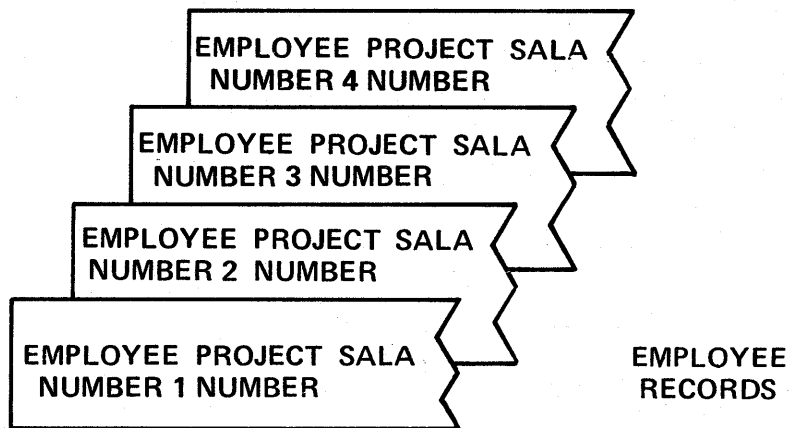
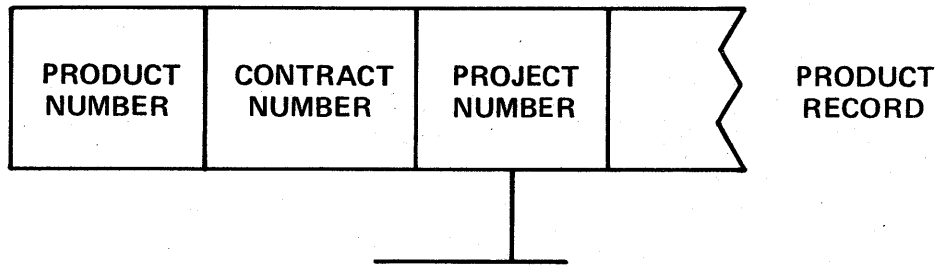


Figure 5. Establishing Record Relationships Using Multiple-Index Capability

Tape Files

NOS provides for ANSI labeled tape files and includes tape resource management functions to alleviate operator scheduling of tape units.

NOS tape file labels include a Volume Serial Number (VSN) identifier for automatic identification of the tape volume and for association with a particular job. A user job requiring two or more tapes is required to supply a RESOURC control card stating the maximum number and type of tape units needed. This is used by the system for automatic tape assignment and the prevention of deadlocks in scheduling tape units.

Tape labels provide for file generation (son, father, grandfather) handling through the use of a generation number on the label. In addition, the label field provides for a file expiration date. Tape label checking provides for positive tape access control through validation of the owner by user number and family name.

File Reliability

NOS provides for very high file reliability in the event of system malfunctions due to the fact that all critical files are retained on mass storage across all levels of system restart. The only files which may be lost are local files open during job execution. Local files are those which the user has created for temporary use during his job, and which have not been explicitly saved on permanent files on mass storage. The system preserves all files in input/output queues and rollout queues for restart in the event of failure.

SYSTEM GENERATION

The NOS operating system and its associated product sets may be generated and installed on a running system. While performing the modification to a running system it should, however, be relatively inactive at the time. Modification can be done while the system is in debug mode. Each product set of NOS is contained on a formatted program library tape. Installation procedures consist of nested procedure files (callable preprogrammed jobs) which accomplish installation. Each product is obtained from the tape by calling a procedure file named GENJOB. It is possible to set appropriate values in a call statement to GENJOB. One of the GENJOB parameters specifies whether or not to incorporate the certain installation parameters by setting keywords to appropriate values in a call statement to GENJOB. One of the GENJOB parameters specifies whether or not to incorporate the system modifications into the existing system – a process called SYSEDIT. This means a minimum disruption of normal work flow. Thus, it is possible to generate a version of the operating system without disrupting the system operation.

MULTIMAINFRAME OPERATION

The NOS Multiframe Module is an available software option which provides the capability for up to four CYBER 70/170 or 6000 series computers to share mass storage files (see Figure 6). Shared access through mass storage controllers is provided. Extended Core Storage is required in such a configuration to maintain tables which are used for multiframe operation and recovery. Failure of one system in the configuration does not affect operation of any other system. All systems operate with independent copies of NOS.

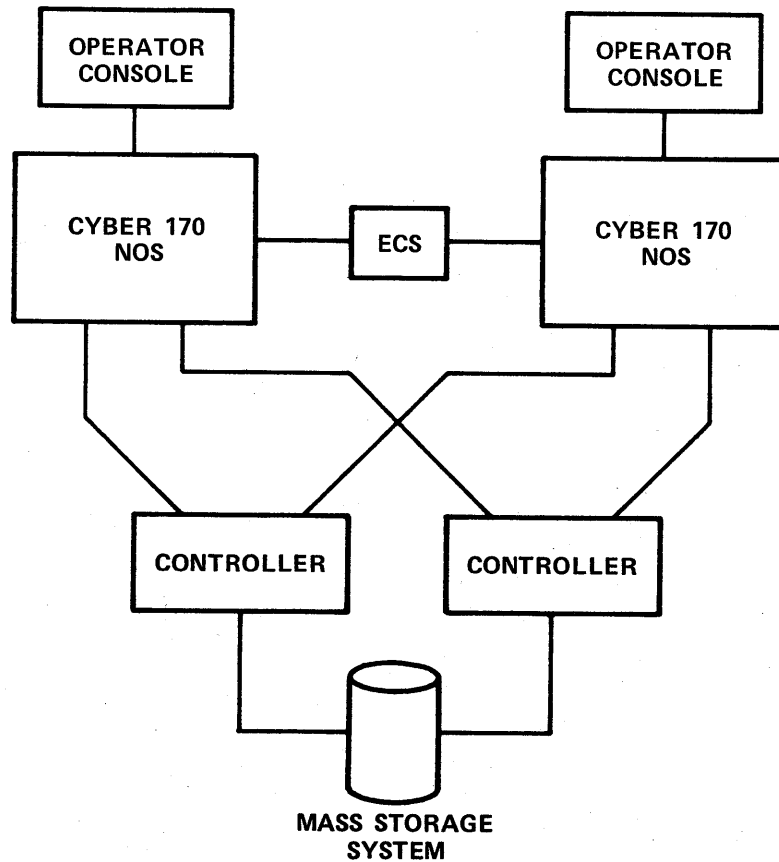


Figure 6. Multimainframe Configuration

NETWORK PRODUCTS SOFTWARE

The Control Data Network Products Software is based on an architecture that supports a wide range of configurations and applications. This software is supported by the NOS operating system. NOS Network Products Software includes the following products:

- CCP (Communication Control Program)
- NAM (Network Access Method)
- RBF (Remote Batch Facility)
- IAF (Interactive Facility)
- IAF (Transaction Facility)

FEATURES

Application Interface

The Network Access Method (NAM) provides a centralized and defined interface between system applications and the communications network. Remote Batch Facility, Interactive Facility, and Transaction Facility are system applications provided. Additionally, sites may prepare their own system applications to interface to NAM.

Routing

NAM controls both the logical and physical connections between terminals and applications. In the event of a communication line failure, NAM, in conjunction with the Communication Control Program (CCP), can establish alternate routing.

Security

Data paths are created only after proper access control procedures have been executed. Data paths are revalidated after a connection failure or when a connection is switched to a new application. Messages concerning all unauthorized access attempts are logged and made available to the computer center operations personnel.

Stimulator

A Network Product Stimulator (NPS) software product is provided as a tool to test and evaluate system performance under varying load conditions, and to verify proper operation of applications interfacing to NAM. It also provides an analysis of test results.

Network

A network of local and remote communication processors (CDC 2550 Series), interfacing to a wide range of terminals, can be constructed with the Network Products software. A bit-oriented protocol (HDLC/ADCCP based) is used to efficiently connect remote nodes to the central site communication processor. Additionally, terminals with various protocols (HASP Multileaving, Control Data Mode 4, Asynchronous, etc.) can connect to either remote nodes or central site communication processors.

Network Definition

The network is defined by use of a Network Definition Language (NDL). When processed at a central site, the NDL statements build a network configuration file and a local configuration file. These files are used to define the initial state of the communication processors and terminals which can then be modified as required during network operations.

Network Control

Network control is performed by the supervisory functions of NAM, which allow a network operator to enable, disable, reload, dump or status various network elements. Operator messages may be sent to all terminals or individual terminals.

Diagnostics

Network products provide in-line, on-line and post-mortem diagnostics. Diagnostics are also provided to allow verification of the correct operation of terminals connected to the network. The on-line and in-line diagnostics provide appropriate alarm messages to the operator.

Cross Support

The network processing units (CDC 2550 Series) within the network are supported by a CYBER Cross system which provides for the installation, maintenance and enhancement of the CCP software which is then downline loaded into local and remote communication processors. The CCP software has been developed using PASCAL, a powerful and effective high-level implementation language.

Virtual Terminal

Virtual terminal is a concept introduced with CDC NOS Network Products to minimize the interdependencies between the terminal user's applications program and the physical characteristics of the terminal. This concept defines a universal interface to application programs and is implemented in CCP. The Terminal Interface Programs (TIPs) in CCP have been implemented to provide translation between the physical terminal characteristics and the universal terminal interface.

Terminal Support

TTY and TTY-compatible asynchronous terminals at speeds to 9600 bits per second:

- TTY M33, M35, M37, M40
- CDC 751, 752, 756, 713
- Memorex 1240
- Tektronix 4010, 4014
- Hazeltine 2000

IBM 2741 Correspondence Terminals

CDC Mode 4 synchronous terminals at speeds to 9600 bits per second:

- CDC 711, 714
- CDC 200UT (BCD/ASCII)
- CDC 731, 732, 734
- CDC CYBER 18-05, 18-10, 18-20
- CDC 241, 777

HASP Multileaving Workstation at speeds up to 19,200 bits per second:

- **IBM 360/25 HASP Workstation or equivalent**

CDCCP Link Interface at speeds up to 19,200 bits per second:

- **CDC 2551 remote node or equivalent**



NOS/BE OPERATING SYSTEM

The CDC Network Operating System/Batch Environment (NOS/BE) operating system supports the CDC CYBER 170, CYBER 70, and the 6000 series computers.

NOS/BE combines with the unique architecture of the CDC CYBER series to implement distributive processing concepts of the CDC CYBER design. In this combination, the Central Processing Unit (CPU) is designed for high-speed computation, and the Peripheral Processors (PPs) provide the control of input/output and systems functions. Central memory provides storage for user programs as well as operating system programs and tables.

NOS/BE is controlled by a software monitor having functions distributed among CPU and PP programs. The monitor controls assignment of the CPU to user jobs in central memory and is responsible for switching the CPU between jobs as input/output is performed or as job priorities dictate. The monitor also directs assignment of PPs to input/output operations or system functions.

SYSTEM SCHEDULER

The integrated scheduler provides a coordinated approach to hardware/software resource allocation. NOS/BE allows up to 15 discrete user jobs to reside concurrently in central memory (control points) facilitating true multi-programming and multi-processing. Additional jobs in various stages of execution can be stored in Extended Core Storage (ECS) or other mass storage for swapping to central memory as resources become available. The scheduler provides for the dynamic allocation of control points and central memory between conversational time-sharing, interactive graphics, remote batch, and local batch processing jobs. The scheduler controls the swapping of jobs between central memory and ECS or system mass storage, taking into account resource requirements, availabilities, job classes, and job priorities. The scheduler also provides extensive flexibility in assigning job priorities, thereby permitting the more critical jobs to be processed faster. The scheduler dynamically monitors and allocates the many system resources effectively and efficiently.

SYSTEM DISPLAY

The system display, permanently assigned to one of the peripheral processors, serves as the communication linkage between the system and the operator. One or two console displays provide system monitoring information and displays of central memory. By means of the console keyboard, the operator can modify central memory contents and request the system to display information about the jobs in the system. One display for example, is the dayfile. All system messages and diagnostics, job steps, and operator requests are entered into the dayfile, scrolled through the dayfile display, and entered into a permanent file maintained on system RMS for accounting and archival purposes. This display enables the operator to follow the activity within the system. Another job status display provides the status of all control points. Additional displays list the jobs waiting in the input/output queues, allow for tape and disk-pack staging, and permit the operator to view the contents of key tables within the operating system.

OPERATION

The flow of jobs through the system can be summarized as follows. A job enters the system through a card-reader, tape drive, or remote terminal, and is placed in the input queue maintained on mass storage. The file is then loaded from mass storage into central memory for processing on the basis of its priority in relation to other jobs in the system, and resource availability to process the job. Execution of the job proceeds, time-sliced on the basis of its priority and job-class in conjunction with jobs at other control points. Job requests for input or output are processed by NOS/BE. A program-recall scheme exists to permit reassignment of the job's time-slices to the other control points while input/output is being performed. The integrated scheduler periodically reevaluates the priorities of all jobs requesting central memory.

Output generated by an executed job is collected on mass storage for disposition at specific request or by default following termination of the job. Output files, as they become part of the output queue, are commonly printed at the terminal or mainframe (in a multi-mainframe environment) from which the job was originally entered. Commands exist to permit routing of output to other terminals or mainframes, and a load-leveling of output jobs may take place where no specific destination is requested and a multi-mainframe environment exists.

LOADER

Under the NOS/BE operating system, three distinct loader modes are available: First, the ability to perform fast and efficient, single-module loading. Second, multimodule loading of simple (3-level) overlay structures by explicit program call. And third, multimodule or link loading of program segments that precedes automatically when an entry point of a segment not in core is referenced by an instruction in a segment that is in core. Thin link loading capability permits 4,093 levels of segmentation and is designed to facilitate loading of large jobs.

LIBRARY ORGANIZATION

The NOS/BE library organization allows logically independent, multiple libraries with associated directories to be permanent files or a local Rotating Mass Storage (RMS) file. System libraries are permanent files, structured for multiuser access; additional libraries (permanent or local RMS files) may be formed and identified as user libraries.

TAPE SCHEDULING

Includes a comprehensive set of automatic tape-assignment features and prescheduling capabilities. A control measure, called tape over-commitment, allows job groups to run which have a total tape requirement in excess of the number of units currently allocatable. This control measure guards against potential unit-assignment deadlock situations.

CONTROL POINTS

An installation can have up to 15 discrete programs in core concurrently; each a candidate for multi-programming under control of NOS/BE. This feature also provides flexibility which allows NOS/BE to be structured to meet the requirements of a particular installation.

PERMANENT FILES

A permanent file is a file created on RMS and cataloged so that its location and access permutations are always known to the system. Residency is established in such a manner that these files cannot be accidentally disturbed by any normal system activity or operation, by planned or unplanned interruption of the system session, or by unauthorized access if privacy controls have been imposed. Any file, regardless of its content, access method or file type can be made permanent if physical residence (at creation) has been directed to RMS and recognized by the system as supportive of permanent files. Permanent files can be dumped to tape for off-line archiving purposes or for maintenance of the system, and reestablished on RMS through operator action.

Permanent files are protected by the system from unauthorized access according to privacy controls imposed by the originator. Passwords associated with levels of file activity can be individually or collectively assigned to require their application by a user to gain permission to read the file, to alter or modify the file, to add to or extend the file, or to release (purge) the file.

DEVICE SETS

All configured RMS devices are grouped into device sets. Each device set is labeled with a unique setname, and is composed of one or more members. One member of each set is further designated as the master in that it holds all tables describing the set. Device sets are of two classes: public or private.

Public sets are defined and maintained by the installation and are mounted and available at all times. Public sets hold the input/output queues, the operating system file, the default permanent file base, and provide scratch space allocated to transient files.

Private sets are logically and physically dismountable and typically made available (mounted) by specific job request and operator action. Members are mounted as referenced by active jobs and may be dismounted (other than the master) by an operator type-in or a specific user dismount command at any point in the processing. Masters can be dismounted when no active job is referencing any file maintained on the set. Private sets are commonly used to hold permanent files generated in the same manner as normal (default) permanent files but having residence requested for the private device set by setname. Permanent files so created can thus be dismounted and moved to another computer in much the same manner as labeled tapes.

MULTI-MAINFRAME

A multi-mainframe software module is optionally available as an extension to single or stand-alone mainframe support under NOS/BE. The module is composed of three logical elements:

- Symmetric station
- Enhanced station
- Shared-RMS

The symmetric station provides the capability for two CDC 6000 or lower CDC CYBERs (e.g., CYBER 70/Models 71 through 74, or CYBER 170/Models 171 through 175) to be linked via a CDC 6683 Satellite Coupler pairs or Extended Core Storage.

The enhanced station can be described as that portion of the station code permitting the linkage of a lower CDC CYBER to an upper CDC CYBER (e.g., Model 76 or 176) or CDC STAR through a CDC 6683 or 7683 Satellite Coupler. Extended Core Storage cannot be configured on upper CDC CYBERs or CDC STAR. Its use as a link medium, in this case, is not supported.

Symmetric station and enhanced station can coexist and share common logic. The station module runs at a control point and is brought up and dropped as required by console operator commands.

Mainframes part of the multi-mainframe environment are denominated by three character logical identifiers. Each mainframe must have a unique host identifier to differentiate it, and may include one or more additional logical identifiers to permit levels of job and file manipulation. Jobs submitted into the complex can designate execution at a specific mainframe through use of these logical identifiers on their respective job cards. The design also covers the ability to route job output, to delay execution of certain jobs, and load leveling of input/output queues. Load leveling is only supported under symmetric station, however.

In addition to the transmission of queue files, the station handles a subset of operator commands and displays associated with multi-mainframe. It controls the exchange of permanent files maintained on RMS local to one mainframe and pertinent to a job executing on the connected mainframe.

Shared-RMS is the third element of the multi-mainframe module. In an environment containing two lower CDC CYBER/6000s, it is possible to configure CDC 7X54/844 RMS devices, equipped with appropriate dual-access hardware features, in such a manner as to physically share the allocatable disk space. This feature supports both dynamic space allocation to transient files as well as direct permanent file sharing. Dynamic space allocation is handled in a reserve/demand fashion; a "pool" of space is maintained on a shared 844 and allocated/retrieved on a threshold basis by mainframe. The sharing of permanent files resident on shared 844s is under the control of the permanent file manager with access interlocked by hardware reserve and flags set into the disk resident tables.

SUPPORT

NOS/BE supports the CDC 255X Network Processing Unit (NPU) front-end through its communications software. This provides support for interactive terminals, with distribution of communications input/output functions to the stations. Interface of this type promotes availability of peripheral processors with attendant capability in multiprogramming and overall throughput. Support provisions include all CDC CYBER and 6000 series systems.

An extensive product set of compilers is also available to the NOS/BE user. This product set permits the CDC systems to excel in a broad variety of data processing, data management, and scientific applications. Time-sharing and communications handling complement these capabilities. Any application can be written as a multiuser job under NOS/BE, and can be used interactively by many users simultaneously. NOS/BE also provides modularity which allows additional terminals to be supported in the future.

DATA MANAGEMENT

Several data management products have been grouped together to form the CDC DMS-170 Data Management System. The products available include:

- CDC CYBER Record Manager (CRM)
- Data Description Language (DDL)
- CDC CYBER Database Control System (CDCS)
- QUERY/UPDATE

CRM is the nucleus module of the DMS-170 system and provides common input/output routines for CDC CYBER host languages. This includes blocking and deblocking of records, label checking, and input/output error processing.

DDL is a self-contained language, used to produce a description of an entire data base and data known to specific programs. This language accepts schema and sub-schema source statements and performs conversion to object directories.

The CDCS module adds data base features to the conventional files of CRM. The CDCS provides features for data validation, encode/decode, derived items, and logging and data base utilities for recovery/restoration.

QUERY/UPDATE is a high-level, English-like conversational language for query and manipulation of data files organized under CRM. The QUERY/UPDATE also includes a report writer module, and utilizes features of CRM multiple indexing.

COMPILER AND APPLICATIONS SOFTWARE

An extensive product set of compilers and applications software is available under NOS/BE. Among the standard products are:

- COMPASS
- SORT/MERGE
- COBOL
- SYMPL
- FORTRAN Extended
- BASIC
- PL/1
- APT IV
- APEX III
- PERT/TIME
- Graphics Support
- TOTAL Universal
- SIMSCRIPT
- Math Science Library



RTOS 3 (CYBER 18 REAL-TIME OPERATING SYSTEM)

The CDC CYBER 18 Real-Time Operating System (RTOS) is a real-time multiprogramming operating system for the CDC CYBER 18 installations and resides in memory with no need for a mass storage device. Systems are tailored to user's hardware and job requirements.

RTOS is a compatible subset of the CDC CYBER 18 Mass Storage Operating System (MSOS 5). Up to sixteen program-priority levels are provided, and input/output requests are also processed on a priority basis.

The monitor minimizes the time for which interrupts are inhibited to provide fast interrupt response times. It contains a request processor to handle the following standard requests:

- READ
- WRITE
- FREAD
- FWRITE
- SCHDLE
- INDIR
- EXIT
- TIMER
- MOTION

Optional request processors are available to handle requests for: SPACE/RELEASE/CORE/STATUS. Other optional features include a protect processor for use while debugging user-written programs, a memory allocator, an alternate device handler, and a diagnostic timer.

All input/output drivers are compatible with RTOS 3 and MSOS 5.

Minimum hardware requirements:

- Computer with 32K bytes of memory. Memory is expandable to 64K bytes (memory addresses above 64K may be used only for application data)
- Console device (teletypewriter, CRT)
- Input device (card reader, magnetic tape)
- Output device (card punch, on CDC CYBER 18-17 only; magnetic tape) optional

An RTOS system with 32K bytes of main memory supports the assembler that is compatible with the MSOS Assembler (except that macros may not be used). Previously compiled FORTRAN or RPG II programs can be run under RTOS; however, programs can not be compiled under RTOS.

A job processor provides for loading (*L) and executing (*X) non-resident absolute programs such as the relocatable binary loader and the assembler. An optional job processor provides expanded load and execute features (*L, *X), schedule (*S), and terminate (*Z).

Optional modules allow for batch mode control (*U, *V), dumping core (*D), inserting values into core (*I), punching core (*P), magnetic tape motion (*REW, etc.), assigning logical units (*K), marking logical units up/down (*M, *N), and copying and/or converting data (*T). The optional job processor modules may be either memory resident or system library resident.

The system library resides on 9-track magnetic tape and contains system maintenance, debug, and utility programs. This includes programs in the RTOS product set (relocatable loader, assembler, system initializer, LIBEDT, SMART, SETUP, MTUP) and all optional job processor modules.

RTOS features an optional processor to load any of the programs from the system library by file name.

The Library Editor (LIBEDT) provides for maintenance of the system library labeled and unlabeled files in absolute or relocatable binary form on magnetic tape. A source tape editor (SETUP) and a relocatable binary tape editor (SMART) are also provided.

RTOS may be stored on 9-track magnetic tape or punched cards and installed or, if necessary, reloaded via a bootstrap loader.



SCOPE 2 OPERATING SYSTEM

SCOPE is a general purpose, multiprogramming operating system designed to support the CDC CYBER 70/ Model 76. SCOPE fully utilizes its resources, provides a data manager to provide a common input/output base, includes priority interrupts to allow responsiveness to real-time requests, and takes full advantage of the streaming, staging, and multiplexing capabilities of the Model 76.

FUNCTIONAL STRUCTURE

The SCOPE operating system consists of four functional parts. Three of these are working parts: the job supervisor, the interrupt handlers, and the system executive. The remaining part is called the system interchange which is the coordinating functions within the operating system. The system resident is structured to use overlays in order to keep the amount of small semiconductor memory (SSM) or central memory required to a minimum.

The interrupt handlers are unique to the software for the Model 76. They are small SSM-resident routines whose sole function is to empty the central memory input and output buffers into and out of larger buffers in large core memory (LCM). This is the only input/output with which the Model 76 is concerned. The system executive performs all task-oriented functions. The system interchange serves as the clearing house for servicing requests and controlling transfers between the other functional groups of the system monitor. The job supervisor performs all job-oriented functions. It handles all user requests, initiates the job, advances the job, terminates the job, and supports the data manager. The job supervisor has its own reference address and field length, different from the user job it is attached to, giving the job supervisor wider access privileges than the normal job. (The data manager, therefore, has the same access privileges.)

JOB MANAGEMENT

Job management is responsible for accepting, staging, initiating, advancing, and terminating all jobs entering the system. The SCOPE system provides a flexible job control language. The user will be able to control the flow within his job and among several jobs by means of the job control language. He may also request, by means of the job control language, the various system resources, such as magnetic tape units (staged or direct on-line), system mass storage (for data and scratch files), amount of central and large core memories, and SCOPE utilities, including those for using permanent files and for enhancing the performance of the job, such as COPY, REWIND, REDUCE, etc.

RESOURCE MANAGEMENT

Resource management is responsible for allocating and accounting for system resources including the central processor, large and small memories, system mass storage, and magnetic tape units. The multiprogramming, multiprocessing capabilities of the CDC CYBER 70 systems allow many jobs in the system at one time. Priority and time-slicing algorithms exist to designate which job gets access to a specific system resource

next. System resources are deployed by means of time slicing. There are three types of priorities: one for the central processor unit (CPU), one for the small semiconductor memory (SSM), and one for the large core memory (LCM). System resources are divided among programs on the basis of a fixed time quantum or multiple of this quantum. The time slice given to each program is an installation parameter. Allocation of system resources is always given to the highest priority job waiting to use that resource.

The resources required by a job are assigned at each step of a job as specified: the job may be required to wait for the resources before proceeding. In normal system operation, various jobs will exist in SSM and LCM which are able to use the CPU actively. Once a job gives up the CPU (for an input/output request, etc.) or the CPU is taken by the system (time-slice, interrupt, etc.), the highest priority waiting job in central memory is assigned to the CPU. Jobs residing in SSM are the first choice. Next is the highest priority job in LCM which is then swapped to SSM for execution. Finally, those jobs rolled out to mass storage are picked by the highest priority job waiting to get into LCM.

In the process of execution, a job may reside in SSM, LCM, or on mass storage. Changing the residence of the job is primarily the result of scheduling the CPU. Some other actions can force the change of job residence, for example, requesting additional resources, operator intervention, time-slice expiration, job completion, higher priority job, etc.

At job completion, all resources assigned are returned to the system for rescheduling. Files created during the job, including the job's OUTPUT file, are disposed of as specified by the user or in accordance with default specifications.

DATA MANAGEMENT

The input/output of the central system occurs in each of the three working elements of the central operating system. The logical input/output (or data manager) functions are carried out entirely in the job supervisor associated with each job. (A particular overlay associated with each job resides in the SSM job supervisor area and swaps out to LCM with the job if system conditions require swapping.) The physical input/output is distributed between the system executive and the interrupt handlers. At the highest level are the compilers, service routines, and user jobs, all of which perform their logical input/output functions through the data manager. At the next level below logical input/output is the physical input/output part of the central system which is handled by the queue managers and their associated interrupt handlers. Physical input/output is also distributed outward from the central system and occurs as a function of first level peripheral processors, as well as in input/output stations.

PERMANENT FILES

Permanent file management provides methods for declaring, protecting, and accessing files which are permanently assigned to mass storage. Files may be assigned to a system file set or to removable sets. Members of the system permanent file set are always on-line. Commonly associated files can be grouped into individual sets which can be requested as necessary during system operation.

Certain privacy features beyond sets are supported by the permanent file subsystem. Files are, of course, uniquely identified by name. There are two other identify features: cycle numbers, which permit the user to have multiple editions of the same file in the system, and the user ID, which permits the user to group all of his files together. File accessibility is controlled by optional turn-key and access mode passwords. If a given password is not stipulated at catalog time, then that type of permission is implied. The access permissions are independent.

LOADER

The LOADER is an operating system routine which provides high-speed transfers to contiguous locations in large core memory or small semiconductor memory from input or storage devices. This routine can be called either statically, by control cards, or dynamically, directly by object code.

FEATURES

The SCOPE operating system includes the following features:

- Multiprogramming using the RMS, LCM and SSM storage hierarchy
- Source language compatibility with CDC 6000, CYBER 70, and CYBER 170. series counterparts
- Job dependency
- Priority scheduling and CPU time-slicing
- Data manager
- Batch processing
- Multiple input/output stations
- Input/output multiplexing
- Remote station access
- Direct on-line tape support, on-line tape staging, and station based tape staging
- Station permanent file staging
- Permanent file staging between multiple Model 76s
- Real-time clock
- Permanent file support
- Direct access (word-addressable) files
- Job and file recovery
- Checkpoint restart capability
- Automatic systems diagnostics

- Comprehensive accounting data recovery

Using the above elements as a nucleus, SCOPE provides a common interface for streaming tasks between distributed magnetic tape, unit record, communication and computer stations. SCOPE supports the software products available on the computer stations with its own powerful software products and facilities. These are:

- COBOL
- SORT/MERGE
- COMPASS (Comprehensive Assembly Language)
- FORTRAN

SECTION 2

SOFTWARE PRODUCTS



ALGOL 60 5

The ALGOL-60 5 Compiler for the CDC 6000, CYBER 70, and CYBER 170 computer system is an extensive implementation of that described in the Modified Report on the Algorithmic Language ALGOL-60, which was published in the British Computer Society - Computer Journal, Volume 19, Number 4, November 1976.

The ALGOL-60 5 compiler supports the full standard and also provides additional facilities for the programmer to make optimum use of the computer system hardware in solving long, difficult, and complex mathematical calculations.

ALGOL-60 5 contains the following features:

- Variable identifiers may be of indefinite length. A procedure may have up to 255 parameters and strings can be as many as 131,071 characters in length.
- Automatic memory management is performed during program execution, as the size of the data stack changes.
- A standard or user-supplied circumclude may be selected when compiling a program. This circumclude consists of a prelude and postlude that serves to define the ALGOL program's environment; for the standard circumclude, it contains declarations for all standard procedures, such as math library functions and input/output routines.
- Additional arithmetic and relational symbols are permitted beyond those mentioned in the standard.
- Debugging features for subscript bound checking and traceback dump are provided.
- The procedure channel may be involved to define or redefine the characteristics of any ALGOL channel (file). Procedures CONNECT or DISCONT may be used to activate or deactivate terminal files during interactive processing.
- A segmentation facility enables large program modules to be broken up into smaller units and selectively loaded into memory as needed.
- An input operation which encounters an error displays the offending line, along with the diagnostic message.
- Additional input/output procedures are available for handling integer data.
- Two levels of code optimization are selectable.
- Compilation rates are extremely fast; a 100-200 statement program requires only a few seconds. Execution speed of ALGOL-60 5, as measured by the Wichman Mix benchmark, is some 30-40 percent faster than with the predecessor ALGOL 4. Memory required for execution is significantly reduced from the earlier version; a savings of 40 percent is not uncommon. Many of these performance improvements in ALGOL-60 5 are more readily recognized in large, lengthy application or research type programs.

- An extensive cross reference listing capability. An alphabetical list of all identifiers used in the compiled program unit is provided. Associated with each identifier is a variety of information concerning where and how it is being used.
- Optionally, line numbers can be used with ALGOL source for ease of use with the time-sharing editors. The line numbers are referenced in all diagnostics, object code listings, and the cross reference listing.

Non-standard features furnished by previous Control Data ALGOL compilers that continue to be supported in ALGOL-60 5 are:

- Alternate character representation is allowed, for example, two periods . . represent the colon :
- COMPASS interface macros are supplied, however they utilize different calling sequences and are not compatible with the older macros.
- Data residing in ECS/LCM storage may be accessed through the procedure READECS and WRITEECS.
- Both sequential and word-addressable input/output facilities are present.



APEX III MATHEMATICAL PROGRAMMING SYSTEM

The APEX III Mathematical Programming System operates on CDC 6000, 7000, CYBER 70, and CYBER 170 series computer systems. It provides a comprehensive, highly efficient system for the solution of a wide variety of linear and mixed-integer programming problems. Because of its efficient solution algorithms, and exceptionally compact matrix representation, APEX III provides extremely fast solutions, often entirely in core.

APEX III has been designed to solve both standard linear programming and Mixed Integer Programming (MIP) problems. The MIP option allows the user to solve problems in which some of the variables are restricted to integer values.

LINEAR PROGRAMMING FEATURES

- MPS format input
- Three levels of control programs
- Solution ranging
- BCDOUT
- Matrix revision
- Matrix reduction and recreation
- Indirect coefficients
- Easy interface for FORTRAN report writers
- Equation listing
- SAVE and RESTART
- Efficient use of extended core storage, large core memory, or disk as auxiliary storage
- BASISIN and BASISOUT

MIXED INTEGER PROGRAMMING FEATURES

- Supports binary and integer variables and special ordered sets of Type 1 and Type 2
- Allows single control card SOLVE option
- Uses branch and bound algorithm
- Uses Driebeck/Tomlin penalties
- Best projection used for branch selection

- **TREEIN** verb available for priority and cascade-type facilities
- Same branch, both branch, and forced branch candidate problem selection options
- User-controlled interrupts, including time limit, mode and iteration limit, integer solution, and near optimal solution
- **MIXFIX** verb to fix integer variables and allow sensitivity analysis on continuous variables
- Provisions to search for alternate optimal integer solutions
- **SAVE/RESTART** capability at any interruptable point in problem solution
- Allows user-written **FORTTRAN** control program or worker subroutines

PROCESSING OPTIONS

The **APEX III** system provides a number of processing options corresponding to the complexities of the linear programming solution, the sophistication of the user, and the type of output required.

Single SOLVE Card

The customer can employ **APEX III** by submitting a single control card. Since most linear programming problems do not require extensive program options, there is no need to closely monitor the system's processing. By utilizing the parameters on a single **SOLVE** card, **APEX III** establishes numerous standard solution procedures which handle most conventional linear programming problems.

Control Programs

Some customers have highly complex problems which require more specialized handling. **APEX III** accommodates these users by processing the problem according to instructions contained in a user-written **APEX III** control program. This permits the user to alter the standard solution procedures the programming system normally follows, and allows him to follow the course of solution, obtain extra output, and modify the rules that the optimization algorithm uses in its solution.

APEX III as a Callable Subroutine

The customer may have a mathematical problem which is exceptionally complex and requires a customized programming system. Because **APEX III** utilizes an extremely efficient solution algorithm, it may be used to optimize the problem, while the user's own routines handle the remainder of the processing. In this case, **APEX III** can be called by another program to function as a subroutine.

Flexible Output Options

APEX III can generate complete reports which detail every row and column or partial reports which generate only selective variables. In addition, the program can list the counts of nonzero coefficients of each vector in the linear programming model. Output can include a listing of all rows, all columns, all bounded variables, or a combination of these items. Additional options govern the amount of printed output for those using the single SOLVE card and for those using a control program.

TYPICAL APPLICATIONS

APEX III provides solution alternatives for problems which encompass more variables and options than can be effectively handled through manual methods. Typical applications include:

- General assignment problems
- General investment problems
- Production scheduling and control
- Process optimization
- Blending
- Inventory control
- Transportation and distribution
- Advertising media selection

APEX III MODULES

The APEX III system is comprised of the following modules:

- **APEX-III Out-Of-Core System**
A high performance linear programming system with an in-core primal optimizer. Provides a basic linear programming system capability. In addition to the capabilities of the basic system. Provides an out of core capability utilizing disk.
- **APEX-III Mixed Integer Programming Option**
Provides a matrix reduction (reduce) capability to the APEX-III package including regeneration of solution to the original problem.
- **APEX-III Matrix Reduction Option**
Provides a matrix reduction (reduce) capability to the APEX-III package including regeneration of solution to the original problem.
- **APEX-III Parametrics**
Provides parametric RHS and parametric OBJ capabilities for APEX-III package.



APL 2 A PROGRAMMING LANGUAGE

CDC CYBER 170 series APL is a mathematical programming language for the systematic treatment of complex algorithms.

Programs written in APL are executed by an interpreter which runs on CDC CYBER 170 series computers under control of the Network Operating System (NOS).

The APL 2 language consists of a large set of primitive (i.e., predefined) functions for manipulating data and performing mathematical computations. The notation used in describing the syntax is very compact. A single APL character represents the function desired.

Unlike functions in other programming languages, most primitive functions in APL are defined for general arguments. While scalar or single-value arguments are possible in special cases, the arguments are typically array data structures, and the functions operate in a predefined manner on these structures as an entity. All data in APL is handled in the form of arrays.

Control Data's implementation of APL incorporates all of this language's advantages. APL is easily learned, programming tasks are performed rapidly, programs are one-half to one-fifth as long in APL as in other programming languages, and APL has the ability to perform array operations (i.e., matrix algebra).

Control Data APL offers two additional advantages: a standard-terminal interpreter, and access to all NOS time-sharing features.

In the past, APL implementations have required the use of a terminal with a special APL character set on its keyboard due to the descriptive power of special symbols instead of English words. Control Data's APL can translate standard terminal keyboard characters into special APL characters. Equipped with a table of character equivalents, a user can access APL from any terminal supported by NOS, either with or without the special APL keyboard. Terminals supported by NOS include the CDC 713, teletypewriter, IBM 2741, Tektronix 4013 (direct-view storage tube), and other compatible devices.

The principal component of the APL system is a conversationally interactive interpreter designed for, but not limited to, time-sharing terminal operation. The interpreter operates as a separate subsystem under the Network Operating System. Thus, the user can avail himself not only of APL but also of all other NOS time-sharing features.

Once access is gained to APL, expressions keyed on a terminal are evaluated and results, if requested, are displayed immediately.

In addition to this ability to operate the system as a sophisticated desk calculator, the following features enable it to be operated as a complete programming system:

- A user may define his own APL functions in terms of APL expressions using previously defined or existing functions.
- A user library facility enables APL functions and previously input or processed data to be stored for subsequent use or for interchange with other users.
- Extensive diagnostics, debugging aids, and editing facilities enable the APL programmer to be extremely productive.
- A variety of terminal types can gain access to the APL system and exchange data and programs.
- Batch users may also employ the system in batch mode.



BASIC

BASIC is an acronym for beginner's all-purpose symbolic instruction code. It is a programming language that was designed to enable nonprogrammers to solve problems on a computer. However, it has now gained widespread acceptance, and is used by programmers and nonprogrammers alike because of its simplicity and directness.

The BASIC language was originally designed and first implemented at Dartmouth College. The language implemented in BASIC for the CDC 6000, the CYBER 70, and the CYBER 170 series computers closely follows this original implementation.

In addition, the following features that are not in Dartmouth BASIC are implemented:

- Any BASIC identifier can be used as the name of an array variable
- An array variable can have three dimensions
- The word LET in the assignment statement is optional
- Multiple assignment statements are implemented
- File input/output is provided
- The ON ATTENTION statement provides BASIC programs with a means to execute special processing routines when user interrupts occur.
- A number of string and substring manipulation functions (as defined in the proposed ANSI BASIC standard)
- Structured programming constructs such as IF THEN ELSE

BASIC Compiler

The standard CDC BASIC compiler is compatible with the original Dartmouth BASIC. It operates either in batch mode as a normal job or in terminal mode from a terminal. However, it is expected that the system is primarily operated in terminal mode. Therefore, the compiler's normal mode of operation is to compile to core and execute. In batch mode, however, there are options to produce load-and-go files, binary decks, etc.

The BASIC compiler and run routines are written in COMPASS. No additional hardware is required over and above that needed by the operating system. Syntactically incorrect input programs can create diagnostics, but do not result in catastrophic action by the compiler. (An abort back to the operating system is not considered catastrophic.)

The terminal capability provided by BASIC when operating under terminal mode allows the user to interact with an executing BASIC program. Thus, the user creates a program using the interactive editing features, then types a command to give control to BASIC. The program is then compiled and diagnostics returned to the terminal. The user can then modify the program using the interactive editor. If BASIC is able to compile the program without errors, it is immediately executed and the results returned to the terminal. The user can use the input statement in the BASIC language to give control to the terminal during the execution of a program. In addition, BASIC gives control to the terminal in the event of certain execution errors.

BASIC provides greater capability than the Dartmouth BASIC in the following areas:

- Diagnostics are, in general, more comprehensive
- Compilation and execution speeds are significantly faster
- Limitations on program size are significantly reduced



CAS 1.0 CONVERSION AIDS SYSTEM

The CDC 3000L to CDC CYBER Conversion Aids System (CAS), Version 1, provides the user with a means of converting MASTER batch jobs for FORTRAN and/or COBOL applications programs and sequential data files associated with these programs.

The Conversion Aids System consists of:

1. A language conversion subsystem containing language conversion processors to handle the conversion of:
 - ANSI and MS FORTRAN and COBOL to FORTRAN Extended Version 4.0 and COBOL Version 4, respectively.
 - MASTER control cards to SCOPE control card records.
 - MASTER or MS SORT directives to CDC CYBER SORT/MERGE directives.
 - COSY compile directives to UPDATE compile directives.
2. Independent file conversion processors which convert:
 - FORTRAN sequential unformatted binary files.
 - COBOL sequential data files.
 - COSY source libraries to UPDATE source files.



CAS 2.0 CONVERSION AIDS SYSTEM

The Control Data 3000L to Control Data CYBER Conversion Aids System (CAS), Version 2, provides the user with a means of converting MASTER batch jobs for FORTRAN and/or COBOL applications programs and sequential data files associated with these programs.

The Conversion Aids System consists of:

1. A language conversion subsystem containing language conversion processors to handle the conversion of:
 - ANSI and MS FORTRAN and COBOL to FORTRAN Extended Version 4.0 and COBOL Version 5 respectively.
 - COSY compile directives to UPDATE compile directives.
2. Independent file conversion processors which convert:
 - FORTRAN sequential unformatted binary files.
 - COBOL sequential data files.
 - COSY source libraries to UPDATE source files.

CAPABILITIES

1. CAS software:
 - Provides options to convert batch jobs or separate source programs.
 - Provides options at the program and subprogram level for FORTRAN and COBOL conversion.
 - Converts ANSI FORTRAN V2 and ANSI COBOL V3 source programs to FORTRAN Extended V4 and COBOL V5 source programs. These programs may be in a batch job, Update source file, or stand-alone card decks.
 - Converts multiple jobs and/or programs during one conversion run.
 - Detects input and flags for manual analysis, or deletes input having no equivalent or which is not applicable to the NOS system.
 - Produces input listings, if required, and output listings of the converted source text. These listings include information indicating conversions performed, requirements for manual analysis, deletions, data maps, and conversion summaries.
 - Produces a converted file or card deck.
2. CO-UP provides the following capabilities:
 - Conversion of COSY tape files created under MASTER
 - Conversion of COSY tape files for either 3170, 3300, or 3500 system
 - Output files in Update source format residing on tape, mass storage, or card deck media
 - Conversion of complete COSY files or selected decks via user-supplied directives
 - Dynamic determination of input files formats (MASTER) and compression modes (3170, 3300, or 3500)
 - Insertion of *DECK deckname Update directives where deckname is dynamically derived from the COSY input file
 - Replacement of periods (.) appearing in decknames by hyphens (-) on the *DECK directive due to the special usage of periods as delimiters by the Update utility

3. The FORTRAN LCP software does the following:

- Converts statements from the ANSI FORTRAN source programs to equivalent statements in FORTRAN Extended that preserve the computational effect. A description of the translations is given under FORTRAN Differences and LCP Processing.
- Detects and flags for manual analysis statements that have no equivalent in FORTRAN Extended or that cannot be accurately or unambiguously translated.
- Produces a source listing, if required, and a target listing of the FORTRAN Extended program(s). These listings include informative messages indicating conversions made and/or requirements for manual conversions, and a count of each.
- Produces a FORTRAN Extended program file or card deck as specified by the user.



CCP COMMUNICATIONS CONTROL PROGRAM

The Communications Control Program (CCP) is the composite software and firmware that is executed by the 255X Network Processing Unit (NPU). It includes versatile software segments to permit communications with a wide variety of terminals, as well as the CDC CYBER 70/170 or 6000 series host computing systems to which it is connected.

HIGHLIGHTS

- Interfaces to host operating system
- Provides for circuit characteristic detection and control
- Provides terminal and line protocol routines
- In conjunction with associated host software, provides for circuit configuration control
- Gathers statistical data relevant to communications activities for storage at the host level
- Provides on-line hardware diagnostics permitting terminal circuit failure detection and isolation
- Provides data transfer management functions for information flowing to and from the host processor, to and from terminals.
- CCP 3.1 auto-recognizes asynchronous terminals on 110, 134.5, 150, and 300 baud circuits. Auto recognition support of 600 and 1200 baud circuit is added.
- HASP post print support provides post print carriage control for HASP multileaving terminals.

DESCRIPTION

The Communications Control Program is functionally divided into three main segments: Base System which includes the Multiplex Subsystem software and firmware; Interface software including the Host Interface Program and Terminal Interface Programs; and the Network Communications software.

BASE SYSTEM SOFTWARE

The Base System Software is the "kernel" upon which the Communications Control Program is based. The Base System not only performs its functions of an operating system, but also provides control for the flow of information between the network and the CDC CYBER 70/170 or 6000 series host processor. Base system functions include:

- Monitor
- Test utility package
- Initialization
- Buffer management
- List services

- **Queue management**
- **Interrupt handler**
- **Timing services**
- **Console driver**
- **Reports and alarms driver**
- **On-line diagnostics**
- **Command interpreter**

Included in this Base System is the Multiplex Subsystem software and firmware which has the following functional responsibilities:

- Processes each input character and distributes to associated by-line buffers
- Provides a point-of-interface for special processing of each character, depending on the state of the line and terminal being serviced
- Distributes characters to hardware line adapters from by-line output buffers
- Provides a logical interface to other software modules for all communications commands and status
- Dynamic allocation of table space and line management provides capability to connect a large number of lines.

INTERFACE PROGRAMS

Host Interface Programs

The function of interfacing the 255X/NPU to the CDC CYBER 70/170 or 6000 series computer is delegated to the Host Interface Program (HIP). It is the responsibility of the HIP to communicate with the host in a specified host protocol format.

The HIP receives data from the other elements of CCP; formats it according to host protocol; and, with the appropriate address information about which terminal sent the data, passes it to the CDC host processor. On output data from the host, the HIP formats the data for proper internal processing and passes it to other elements of CCP for proper disposition to the correct terminal.

Terminal Interface Programs

Terminals vary widely in the protocols which they use to communicate. The CCP software supports a wide array of terminal systems through the use of terminal support software called Terminal Interface Programs (TIPs).

The function of a TIP is to provide the interface between a given terminal type, following a specific protocol, and the other elements of CCP which follow a standard protocol for all data flow (whether it be destined for terminal or host).

The TIP makes allowances for those terminal features which comprise the terminal protocol, such as code length and type, line and message framing conventions, equipment timing restrictions, etc.

NETWORK COMMUNICATIONS SOFTWARE

The Network Communications Software is comprised of all programs which are not part of the Base System or are not Interface Programs. Included are modules which provide network services for the CCP, such as message routing, procedures for system configuration, and statistics accumulation, as well as common routines in support of the Terminal Interface Programs (TIPs).



CDCS
(CYBER DATABASE CONTROL SYSTEM)

Control Data's modular approach to data base systems permits users to construct data bases using only the modules that provide maximum benefit at the lowest cost in terms of complexity and programming effort. A simple data base application can thus be programmed entirely in COBOL, or COBOL can be used in conjunction with QUERY/UPDATE. As applications become more complex, however, and particularly as the uses for the data available in the computer system become more diversified, the need arises for more centralized control over the format of the data base itself, as well as over its access. Problems of data integrity and security, and the need to coordinate the varied data usage forms found in different software products, necessitate provision of a standard data base interface module in a sophisticated data management package.

On the human level, centralized control is usually brought about by a data administrator, so that data base usage by various applications is coordinated and the compatibility of the results achieved by these applications is ensured.

On the software level, an analogous, though not identical, role is played by CDC CYBER Database Control System (CDCS), which acts as a centralized controller to monitor and interpret data base access requests from applications programs. CDCS ensures data integrity and security by preventing incompatible uses of data by different applications, and makes these applications more convenient by translating various individual input/output data description formats to compatible terminology. CDCS further enhances compatibility across product lines by conforming in large measure to CODASYL's data base control system recommendations.

CDCS uses schemas provided by the Data Definition Language (DDL) compiler to describe the data bases under its control (multiple, independent data bases can be managed at the same time). Each application program has a subschema describing its view of the data base it is accessing in syntactical and sematical terms; consistent with the language the program is written in (both the FORTRAN and COBOL languages are supported). Included in these subschema descriptions may be descriptions of logical records formed from pieces of physical records taken from more than one area ("file"); this capability is obtained using the READ RELATION construct. In order to read, write, add, delete, lock and unlock records, data manipulation languages, or DML's, are supported by COBOL and FORTRAN which are, again, syntactically natural to COBOL and FORTRAN, respectively.

CDCS uses the schemas and subschemas, together with program DML requests, to determine which calls should be made to the CDC CYBER Record Manager (CRM). Before initiating CRM activity, CDCS will, where appropriate:

- check that all element-level locking conditions are met (deadlock situations are automatically detected and resolved)
- check that area-level security checks are satisfied
- check that all CONSTRAINT conditions, which ensure that owner-member relations are properly maintained, are satisfied

- invoke any available site-supplied data base procedures (normally for the purposes of data validation, integrity or security processing)
- perform any indicated data item transformation, and
- perform any indicated data item transposition.

Normally, CDCS then proceeds to initiate CRM activity. By taking advantage of CRM's multiple file and multiple index processing capabilities, CDCS can provide both network and hierarchical data structuring possibilities for its users, as well as limited relational accessing possibilities. Because CDCS runs as a centralized system function, it provides the ability for multiple programs to operate concurrently on the same data bases.

CDCS provides before and after image logging services for both logical records and physical data blocks. When these log files are manipulated by the Data Base Utilities (DBU), full recovery capabilities to roll back or restore data bases are available, whether in the case of application failures or system failures.

Because CDCS operates through the CYBER Record Manager, it is possible – while retaining full security control – to have applications using CDCS and others only using the CYBER Record Manager share access to the same data files (at different times of the day, of course). This flexibility can allow a site to gradually convert its applications to use a centralized data base control system.



CID (CYBER INTERACTIVE DEBUG)

CDC CYBER Interactive Debug (CID) is a package to aid in program debugging by allowing the user to interactively monitor and control the execution of a program. Its basic features are independent of any source language and require no special compilation procedures. These features include the ability to:

- Set breakpoints at specific locations in the user program which cause execution to be suspended at that point.
- Set traps in the user program which cause execution to be suspended upon the occurrence of specific events (such as a STORE into a designated location or the loading of an overlay).
- Interrupt and restart the program from the terminal.
- Display or alter values of program variables and then resume execution.
- Build complex sequences of basic debugging commands which may include loops and conditionals and may be attached to traps or breakpoints for automatic execution.
- Save a debugging environment (traps, breakpoints, and debug command sequences) created during one debugging session for use at a later session.

CID is designed primarily for interactive usage, but may also be used in batch mode. Debugging commands used for input in a non-interactive run are placed on the file DBUGIN. Most commands may be used in either mode, but many provide more dynamic help in the context of interactive usage. Erroneous or questionable commands are skipped in non-interactive mode. Control passes to the next command on the file DBUGIN.

CID has been designed to incorporate a variety of source language dependent modules. Use of such modules requires that compilation be done in debug mode, but provides the user with an extension to the normal set of debugging commands which closely reflects the syntax of the source language.

In addition, such special compilation causes symbol and statement tables produced at compilation time to be retained for use with the debugging package. This allows the user to specify variables by name rather than by address. Locations for traps or breakpoints may be specified directly by line number or statement number. Default formats for display commands automatically vary to reflect the type of the specified variable.

The CID module for FORTRAN provides for such FORTRAN-like constructs as:

PRINT statements:

```
PRINT *, X,Y,Z,  
PRINT *,(ARRAY(I), I=1,10)
```

ASSIGNMENT statements:

```
I=4  
X=Y
```

and IF statements:

```
IF(K.NE.7) PRINT *, ARRAY(K)
```

which can be executed during program suspension or included in sequences attached to traps or breakpoints.

The CID module for BASIC provides for such BASIC-like constructs as LET, PRINT, MAT PRINT, IF and GOTO, which can also be executed during program suspension or included in sequences attached to traps or breakpoints.

A CID run is initiated by typing the command DEBUG(ON) or DEBUG immediately prior to the loading of the user program. A normal load sequence will cause CID to be automatically loaded along with the user program and will add approximately 3,000 words to the programs' normal central memory requirements.

At execution, control transfers immediately to CID, which issues the message:

```
CYBER INTERACTIVE DEBUG  
?
```

and pauses for user response. The user may set traps or breakpoints as desired, read in debugging commands from a local file, or display or modify variables in the user program.

A GO or EXECUTE command causes CID to transfer control to the user program, returning with an appropriate message when a trap or breakpoint is encountered. Default traps occur on program completion or abnormal termination.

After a trap or breakpoint occurs the user may issue additional debugging commands of any type, use GO or EXECUTE to return to the user program (unless a trap was due to program termination), or type QUIT to terminate the debugging session and return to the operating system.

Legal but questionable debugging commands causes CID to issue a warning message to the user before processing the commands. The user may type OK or ACCEPT to cause execution of the commands, REJECT to cause the command to be ignored, or any other legal command.

A HELP feature is available for use at any time during the debugging session. HELP provides general information about debugging options available to the user and HELP, along with a specific debugging command, provides details about the specified command.

The user program performs as it would under normal circumstances, handling input and output from the terminal or other files as designated in the source code.



COBOL 5

The COBOL 5 compiler is designed as an integral part of the CDC CYBER 170 and 70 computer systems, operating in harmony with the other software facilities, and complementing the systems' hardware architecture.

HIGHLIGHTS

Some of the highlights of the COBOL 5 compiler include:

- Input/output operations, during both program compilation and program execution, all use the facilities of CDC CYBER Record Manager. No unique-to-COBOL input/output is used or needed.
- The management of dynamic table areas in memory during compilation and of record areas, buffer areas, etc., during execution, is done through the facilities of the CDC CYBER Common Memory Manager. Thus, the memory requirements of the various system components such as the loader, record manager, SORT, and COBOL program are centrally coordinated to avoid conflicts.
- COBOL programs can be combined with other language programs to form composite programs because COBOL object modules, even from segmented programs, use standard CDC CYBER Loader formats and facilities and standard calling sequences and data types.
- The COBOL switch condition feature is interfaced with the CDC CYBER console sense switch facility so that the operator, using standard actions, can manipulate the COBOL switches.
- COBOL 5 is certified by the US Federal Government's National Compiler Testing Service as a "High-Level" COBOL.

Throughout the system, care has been taken to ensure that all parts function together in the most efficient manner.

The COBOL 5 compiler produces highly efficient object code that is closely matched to the performance capabilities of the machine. Register usage is optimized, and any redundant subscript calculations are eliminated. Particular attention has been given to generating the best possible code for high-frequency elements such as the MOVE, IF, PERFORM, and simple arithmetic statements.

To make more efficient use of computer memory, the compiler employs the technique of automatic spilling to disk, or to ECS if available, of internal compiler tables. In this way, even very large programs can be compiled in the minimum field length. If a larger field length is made available to the compiler, table spilling is reduced with a concomitant increase in compile speed. The compiler will automatically increase field length required until the user or system supplied maximum is reached.

FEATURES

COBOL 5 fully implements and strictly conforms to the 1974 ANSI COBOL Standard (X3.23-1974)†. In addition to the full range of ANSI features, there is a powerful set of CDC extensions. However, for

†The ANSI COBOL Communications Facility is not provided.

those users who wish to ensure that their programs are completely ANSI standard, a compiler option will cause use of these extensions to be diagnosed as nonstandard.

In addition to the full set of ANSI features, a number of significant extensions are provided in the COBOL 5 compiler. As part of an overview of the entire compiler, the paragraphs below describe separately the major new ANSI features and CDC extensions. COBOL 5 contains the following:

- Complete set of data movement statements. These are the MOVE, MOVE CORRESPONDING, STRING, UNSTRING, INSPECT, and INITIALIZE statements. The last four are new and powerful additions to COBOL. They are especially useful for input editing, character manipulation, message processing and text editing applications.
- Complete set of arithmetic statements. These include the ADD, ADD CORRESPONDING, SUBTRACT, SUBTRACT CORRESPONDING, MULTIPLE, DIVIDE, and COMPUTE statements. Rounding and size error detection are provided for each. The arithmetic operators in the COMPUTE statement are addition, subtraction, multiplication, division, exponentiation, and unary plus and minus. Expressions can be enclosed in parentheses, as desired, to control the order of evaluation.
- Complete set of record ordering statements. These are SORT, MERGE, RELEASE, and RETURN. Input and output procedures can be defined, or the input and output files can be named. Nonsequential files can be sorted or merged as well as sequential files. The collating sequence for the sort or merge can be explicitly specified.
- Full set of input/output statements. These are the OPEN, OPEN EXTEND, CLOSE, READ, WRITE, REWRITE, DELETE, and START statements. The ACCEPT and DISPLAY statements are also available for low volume data transfers. Several file organizations are available to suit a wide range of applications. These are the sequential, relative, indexed, direct, actual-key, word-address, and data base subschema files. File access can be sequential, random, or dynamic (a combination of sequential and random). Error declaratives are available as well as a file status register. Files may be unlabeled or have ANSI standard labels.
- Tables of up to three dimensions are provided. Tables may be fixed length or variable length. The SET statement is available for manipulating indexes. The SEARCH and SEARCH ALL statements are provided for table searching. Either subscripting or indexing can be used for table accessing.
- Complete set of branching statements. These include GO TO, PERFORM, GO DEPENDING, and ALTER. Implicit branching to out-of-line declarative procedures and to SORT/MERGE input/output procedures is also provided.
- Complete Report Writer facility. Report Heading, Report Footing, Control Heading, Control Footing, Page Heading, Page Footing, and Detail Report groups are available. Summing, crossfooting, and rolling forward of sum counters is provided. The INITIATE, GENERATE, TERMINATE, SUPPRESS, and USE BEFORE REPORTING statements are provided.
- Programs may be segmented to reduce memory requirements.
- Independently compiled subprograms may be accessed. The CALL and EXIT PROGRAM statements provide transfer of control to and from the subprograms. Data declared in COMMON is available to both the main program and its subprograms. Alternatively, data may be passed to subprograms with a parameter list mechanism. The ENTER statement is also provided for interfacing to non-COBOL subprograms.

- The IF statement provides a complete conditional facility. IFs may be nested to any depth. Compound, complex, and abbreviated conditions are available. Relational, class, sign, switch status, and level 88 conditions are provided. SIZE ERROR, AT END, INVALID KEY, OVERFLOW, and END OF PAGE condition tests are also available as options on the arithmetic and input/output statements.
- The COPY statement allows model file descriptions, record descriptions, procedures, etc., from one or more libraries to be incorporated into a source program during compilation. A powerful REPLACING facility enhances the usefulness of this feature.
- A complete and powerful debugging facility is provided whereby the programmer can specify debugging algorithms with standard COBOL statements. Both data items and procedures can be monitored. This monitoring can be turned off without recompiling the program.

HUMAN INTERFACE

Besides ensuring that the COBOL 5 compiler conforms to the 1974 ANSI standard and contains an extensive set of features, great care has been taken to make this compiler a high-quality product. The human interface aspects of the compiler have been carefully designed to make COBOL 5 a pleasure to use.

Particular care was taken in designing the compiler response to incorrect COBOL programs. Diagnostic messages, rather than being cryptic or vague, are complete and precise so that errors can be corrected without recourse to the reference manual. Each error is diagnosed only once. Secondary or propagated effects of an error are clearly indicated. Clauses and statements within a division are fully analyzed, even when they are out of context. For example, a BLOCK CONTAINS clause is completely checked for errors even though it is encountered after the period that terminates the File Description.

Diagnostics can be routed to a file separate from other listings, if desired, and can optionally be formatted for viewing on a terminal screen. Trivial errors, secondary or propagated errors, advisory messages, and non-ANSI messages can be individually selected or suppressed as desired. Line and column numbers of errors are given exactly, rather than approximately. The wording of messages is consistent, courteous, and precise.

A complete cross reference listing is produced. Both referenced and unreferenced names are listed as well as the usual data names and procedure names.

The source program listing shows both line numbers and column numbers on each page for easy correlation with cross reference and diagnostic listings.

A listing showing the generated code can be selected, if desired. This listing shows assembler mnemonics for the generated instructions and clearly indicates the relationship of the generated code to the COBOL source statements. Optionally, the generated code can be output to a file and processed by the COMPASS assembler.

A data map listing can also be selected. This listing shows the location, size, and usage of data items.

The vertical spacing of all listings can be set at either six or eight lines per inch. At eight lines per inch, less paper is used. This is important from a conservation point of view. The listing can be turned on or off within the program body. All page ejects except the normal end of page control can be turned off.



COMPASS

COMPASS is the assembly language for the CDC 6000, CYBER 70, and CYBER 170 series computer systems. The COMPASS assembler provides a versatile and extensive symbolic language for generation of object code. COMPASS is designed to facilitate the efficient utilization of all computer resources while giving maximum flexibility during program construction.

The COMPASS language allows all hardware functions of the central and peripheral processors to be expressed symbolically. In addition, many pseudo instructions are included which the programmer can use to control the assembly process itself. The source program can also include definitions of, and references to, macro and micro instructions. An extension to the macro facility permits user definition of instructions having machine language formats.

COMPASS executes under the control of the operating system and operates on a basic computer system configuration. COMPASS executes as a central processor program and produces a binary file used to establish an executable program. COMPASS also produces a coded output file which is normally printed. This file contains a listing of each source statement and object code generated. Pseudo instructions provide the programmer with the option of obtaining a variety of detailed or brief listings with or without special titles on each page. Errors detected by the assembler are noted on the listing and a cross-reference table is produced if requested.

FEATURES

Some of the features of the COMPASS system include:

- Free-field source statement format
- Control of local and common areas to facilitate interprogram communication
- Address arithmetic permits extensive use of constants, symbolic addresses, and arithmetic expressions for addresses
- Constants can be designated in integer, floating-point, and display code notation
- Macro coding — assembly of sequences of instructions
- Micro coding — substitution of sequences of characters
- Thorough language diagnostics — source program errors included an output listing
- Extensive listing control — assembly-time control of list content
- Remote assembly — defers assembly of defined coding sequence until later in the assembly

- Concatenation — two nonadjacent character strings can be combined to form a single character string
- Symbol qualification — associate a symbol qualifier within a qualified sequence to render the symbol unique to the sequence



CRM (CYBER RECORD MANAGER)

The CDC CYBER Record Manager (CRM) is the key to the flexibility and modularity of the CDC DMS-170 Data Management System. In addition to ensuring data structure compatibility, CRM controls all input and output operations for the supportive operating systems; for FORTRAN Extended, COBOL, ALGOL, QUERY/UPDATE and SORT/MERGE; and for numerous other current Control Data software products. Because of CRM's flexibility, many different applications can use the same file without recreating that file. By using the FILE control card, the user can specify alternate data structures for a file to be processed by a previously compiled FORTRAN Extended or COBOL program. A master mailing list file created through COBOL and statistically analyzed through FORTRAN Extended, for example, could later be viewed as a portion of a data base file with Data Definition Language (DDL) descriptions and CDC CYBER Database Control System (CDCS) processing, then accessed with QUERY/UPDATE.

In addition to providing indirect file handling capabilities through other software products, CRM is of direct aid to more advanced programmers in creating custom-tailored applications to suit specific installation needs. IBM System 360/370 files, ANSI COBOL data formats, and CDC file organizations are supported by CRM; all can be read and reformatted in CRM format with the FORM utility and the 8-bit subroutines package.



CYBER CROSS SYSTEM SOFTWARE

The CDC CYBER Cross System provides software support for the Communication Control Program (CCP), permitting the user to effectively and efficiently utilize the user-programmability features designed into CCP. The CDC CYBER Cross System consists of a series of programs that execute on a CDC CYBER 170 Computer System and provides for the installation, modification, and maintenance of the CCP.

The following modules support the Cross System:

- **PASCAL** – a high-level compiler patterned after ALGOL 60, which eases the burden of program implementation. Using PASCAL, the implementer defines the task in statements which are processed by the compiler to yield a variable number of actual program instructions.
- **Formatter** – a program which reformats PASCAL output into an object code format compatible with output of the communications processor's macro assembler.
- **Macro Assembler** – assembles communications processor macro-level source programs and produces relocatable binary output. Source programs are written with symbolic machine, pseudo and macro instructions.
- **Micro Assembler** – this assembler provides the language necessary for writing a micro program. The assembler translates symbolic source program instructions into object machine instructions.
- **Library Maintenance Program** – used to maintain a library of object programs as output by the PASCAL compiler, MACRO assembler, and the MICRO assembler.
- **Link Editor** – the link phase accepts object program modules and generates a memory-image load file suitable for downline loading into the 255X Network Processing Unit. The edit phase also introduces initial data for variables within the memory image load file.

OVERVIEW OF PASCAL

PASCAL is a high-level language patterned after ALGOL 60. The PASCAL source language consists of two essential parts:

- Description of actions to be performed
- Description of data on which actions are performed

Actions are defined by statements; data is described by declaratives and definitions.

Data

Data is represented by values of variables. Every variable occurring in a statement must be introduced by a variable declaration which associates an identifier and a data type with that variable. The data type defines the set of values which may be assumed by that variable.

Basic data types are the scalar types. These include integer, Boolean, and char. Values are denoted by numbers for type integer, by identifiers for type Boolean, and by triple marks for type char. The set of values for type char is the CDC 63-character set. The programmer may define an enumeration type. The definition indicates an ordered set of values (i.e., introduces identifiers standing for each value in the set). A type may also be defined as a subrange of a scalar type by indicating the smallest and the largest value of the subrange.

Structured types are defined by describing their components and by indicating a structuring method. In PASCAL, there are three structuring methods available: array structure, record structure, and set structure.

In an array structure, all components are of the same type. A component is selected by an array selector and its type is indicated in the array-type definition, and which must be scalar. It is usually of the type integer.

In a record structure, the components (called fields) are not necessarily of the same type. A record selector is an identifier uniquely denoting the component to be selected. These component identifiers are declared in the record type definition.

A record type may be specified as consisting of several variants. This implies that different variables, although said to be of the same record type, may assume structures which differ in a certain manner. Differences may consist of a different number and different types of components. The variant which is assumed by the current value of a record variable is indicated by a component field which is common to all variants and is called the tag field. The part common to all variants may consist of one or more components, including the tag field.

A set structure defines the set of all subsets of values contained in its base type. Base type must be a scalar type, and will usually be a programmer-defined scalar type or a subrange of the type integer.

In contrast to explicitly declared variables, some variables may be generated by an executable statement. Such a generation yields a pointer which subsequently serves to refer to the variable. The pointer may be assigned to other variables of type pointer. Every pointer variable may obtain pointers which identify variables of the same type T only, and it is said to be bound to this type T. However, it also may contain the value "nil," which points to no variable.

Actions

The most fundamental statement is the assignment statement. It specifies that a newly computed value be assigned to a variable (or component of a variable). The value is obtained by evaluating an expression. Expressions consist of variables, constants, sets, operators and functions operating on the denoted quantities and producing new values. Variables, constants, and functions are either declared in the program or are standard entities. PASCAL defines a fixed set of operators which are subdivided into the following groups:

- Arithmetic operators of addition, subtraction, multiplication, division, sign inversion, and computing the remainder
- Boolean operators of union, conjunction, and negation

- Set operators of union, intersection, and set difference
- Relational operators of equality, inequality, ordering, set membership, and set inclusion. The results of relational operations are of type Boolean. Ordering relations apply only to scalar types.

The procedure statement causes execution of the designated procedure. There are statements which specify sequential, selective, or repeated execution of their components. Sequential execution of statements is specified by the compound statement, conditional or selective execution by the IF statement and the CASE statement; and repeated execution by the FOR statement, the REPEAT statement, and the WHILE statement. The IF statement serves to make the execution of a statement depend on the value of a Boolean expression, and the CASE statement allows for the selection among many statements according to the value of a selector. The FOR statement is used when the number of iterations is known beforehand, and the REPEAT and WHILE statements are used otherwise.

A compound statement can be given a name, and be referenced through that name. The statement is then called a procedure, and its declaration, a procedure declaration. Such a declaration may additionally contain a set of variable declarations, type definitions, and further procedure declarations. The variables, types, and procedures thus declared can be referenced only within the procedure itself, and are called local to the procedure. Their identifiers have significance only within the program test which constitutes the procedure declaration and which is called the scope of these identifiers. Since procedures may be declared local to other procedures, scopes may be nested. Entities which are declared in the main program (i.e., not local to some procedure) are called global.

A procedure has a fixed number of parameters, each of which is denoted within the procedure by an identifier called the formal parameter. Upon an activation of the procedure statement, an actual quantity must be indicated for each parameter which can be referenced from within the procedure through the formal parameter. This quantity is called the actual parameter.

There are three kinds of parameters: value parameters, variable parameters, and procedure or function parameters. In the first case, the actual parameter is an expression which is evaluated once. The formal parameter represents a local variable to which the results of this evaluation is assigned before execution of the procedure (or function). In the case of a variable parameter, the actual parameter is a variable and the formal parameter stands for this variable. Possible indices are evaluated before execution of the procedure or function. In the case of procedure or function parameter, the actual parameter is a procedure or function identifier.

Functions are declared analogously to procedures. The only difference lies in the fact that a function yields a result which is confined to a scalar type and must be specified in the function declaration. Functions may therefore be used as constituents of expression. To eliminate side-effects, assignments to nonlogical variables should be avoided within function declarations.

FEATURES

One of the primary design goals of the communication control program is to provide a straightforward, economical method for Control Data and/or the customer to add application-dependent logic and new

features to the system. Techniques used to accomplish this design goal include:

- Use of a higher level implementation language, PASCAL. The CCP software has been implemented using PASCAL.
- Clearly defined points of interface. In anticipation of requirements for adding application-dependent software, clearly defined Points Of Interface (POI) to the CCP software have been identified to allow access to and modification of data flow in the system. POI definition, rules, and protection are provided.
- Use of standard system subroutines. A number of standard subroutines are provided with CCP. These may be used to augment customer changes.
- All Level 1 PASCAL procedures generate separate object code modules which permit macro-level assembly programs to call Level 1 procedures. Conversely, it is possible to call, within the PASCAL code, a macro-level assembly program.
- All global data variables within the CCP are generated as entry symbols such that assembled macro-level programs may reference them as externals.
- The library maintenance facility permits the collection of separately compiled procedures which may then be used to generate a CCP system. This facility allows a stepwise, modular development of 255X systems without the need for recompiling the entire system.
- The link editor facility provides for the declaration of overlay areas and the specification of modules which may reside within those areas as overlays.
- Macro-level assembly modules may be generated with object text compatible with the macro-level code. This permits the micro-level modules to be linked and loaded at the macro level, prior to its transfer to micro memory.



CSSL (CONTINUOUS SYSTEM SIMULATION LANGUAGE)

CSSL is a digital computer language designed to facilitate the representation and simulation of continuous dynamic systems. The aim of the language is to provide a problem-oriented vehicle for the representation of continuous dynamic systems that can be modeled by sets of ordinary differential equations. It provides a simple and straightforward programming tool for the novice user or for the programmer with a simple, straightforward task. Features of CSSL which satisfy the above include:

- Permitting a clear form for model description statements
- Providing a set of operators that are capable of handling most problems involving differential equations in a simple manner.
- Providing for built-in integration
- Providing a set of problem-oriented diagnostics for both compilation and execution errors
- Automatic sorting of operations to assure proper order of the calculations
- Permitting both expression-based and block-oriented representation statements

In addition to the foregoing approach of a simple and obvious programming tool, CSSL provides additional flexibility for the skilled programmer faced with a major programming task. To this end, CSSL achieves this sophisticated approach by:

- Acting as an adjunct to FORTRAN, providing procedural programming capabilities
- Allowing the structure (or flow) of a program to be programmable, and not be confined to one rigid scheme
- Allowing the operator set to be open-ended, thus permitting simple procedures for generating programmer-defined operators

Many of the features of CSSL are greatly enhanced by the addition of the CSSL macro processor. This processor, besides providing for conditional programming techniques, satisfies an ultimate goal of flexible expansion of the language, in anticipation of changes in the areas of graphical display, hybrid computing, and conversational mode programming.



CYBER 170/70 MODEL 76 STATION SOFTWARE

A CDC CYBER 170, CYBER 70, or the 6000 series system used as a station provides the Model 76 user with great flexibility of input/output support while minimizing overhead normally associated with such capabilities. The station software package allow the Model 76 to have access to most of the capabilities of the station's operating system, product set, and communications capabilities.

FEATURES

- Model 76 system operational control
- Provides all unit record support processing for the Model 76.
- Provides remote job entry and flexible output file disposition direction.
- Permits jobs to be directed for processing on the Model 76 or on the station itself.
- Provides magnetic tape file staging to/from Model 76 mass storage.
- Permits members of the station based permanent file set to be staged to or from the Model 76.
- Provides interactive processing capability.
- Provides complementary and back-up computing power for the Model 76.



DBU (DATA BASE UTILITIES)

The data base utilities supplement the data management services furnished by CDC CYBER Database Control System (CDCS). They constitute a set of tools required by the data administrator to maintain the integrity of the data base. The means to guard against loss of all or part of the data base, to monitor data base usage, and to perform data base recovery in an emergency situation are inherent in the data base utilities.

A log file of entries, which contains pertinent information on input/output operations processed, can be produced for a data base file by CDCS through the logging facility. The recover and restore utilities depend on the logging feature because they require a historical record of all user interactions with a data base file. The log file actually serves a dual purpose: the log entries function as input to the recover and restore utilities and secondly, the log file provides a method of gathering usage statistics for analysis of data base areas.

The recover and restore utilities are called off-line to accomplish recovery or restoration processing. Selective recovery or restoration by area, user, checkpoint, and date and time is possible.

Several utilities that support CDC CYBER Record Manager (CRM) are applicable to the data base environment. FORM, a file management utility, can be used to manipulate records and reorganize files. Utilities associated with indexed sequential files include SISTAT and ESTMATE; utilities for key analysis and creation of direct access files are also provided. The IXGEN utility can generate an index file for alternate key access of an existing indexed sequential, direct access, or actual key file.

In addition to the data base utilities and CRM utilities, the operating systems provide related utility routines that are useful in data base management for dumping, loading, and copying files.



DDL (DATA DESCRIPTION LANGUAGE)

The content and structure of individual items in the data base are described with the COBOL-like statements that constitute the Data Description Language (DDL). These statements are then translated by the DDL compiler into a form usable by other CDC DMS-170 Data Management System components.

Usually a single person or group, known as the data administrator, is responsible for preparing and maintaining the data base. The description of the actual data base, called a schema, describes in detail the internal structure of the entire data base, its storage formats, the data access methods that can be used, the access or modification constraints, and so forth.

The data base descriptions employed by application programs and QUERY/UPDATE users are also prepared by the data administrator; these descriptions are called subschemas. To achieve data independence and flexibility, it is advantageous to describe the same data in the data base in more than one way, thus both schema and subschemas are used. The subschemas are written in DDL dialects similar to the language in which specific applications are to be written. COBOL subschemas are written using COBOL/DDL, FORTRAN subschemas are written using FORTRAN/DDL, and QUERY/UPDATE subschemas are written using QUERY/UPDATE/DDL.

Subschemas need describe only the sections of the data base required for a specific application or user. Since the subschemas are described by the data administrator, access by other users to unrequired sections of the data base can be restricted; thus both efficiency and a measure of security are ensured.



DISSPLA

DISSPLA is a versatile and highly flexible integrated software system capable of producing virtually any graphic data display. Developed by Integrated Software Systems Corporation (San Diego, California), DISSPLA provides a convenient method to utilize a plotter for a wide range of output, including data verification, business graphs, and sophisticated three-dimensional model displays. The package is available in the United States, Canada, Europe and Australia through CDC's CYBERNET data services networks.

Operational on CDC CYBER 70 and 6000-Series Computer Systems, DISSPLA consists of a powerful package of FORTRAN subroutines called from the user's application program to produce the desired graphic displays. Thus, the DISSPLA system permits full use of FORTRAN to formulate the data to be displayed. Each subroutine communicates to others through common blocks, enabling DISSPLA not only to ensure that user instructions are legitimate, but also to transfer parameter information . . . saving the user repetitious entry of information.

FEATURES AND OPTIONS

Included in DISSPLA is a set of the most used axis systems:

- Log
- Linear
- Polar
- Month and user labeled

Grids are available on all axis systems. Chain-dotted, dotted, dashed and chain-dashed grids and data curves can be used. The user can also define his own interrupted line pattern and length.

DISSPLA enables the user to project lines, curves, and surfaces in a three-dimensional plot. Three-dimensional surfaces may be drawn with or without their hidden lines removed. A three-dimensional object can be viewed from an arbitrary point in space, and a projection obtained of the object from an arbitrary perspective above or below it. Three-dimensional axes and surface interpolation are also provided.

DISSPLA provides simplified page layout, with right-justified text, legends, labels, and centered underlined headings.

Other features include eight standard alphabets with 13 different fonts, seven standard interpolation modes, smoothed curves for noisy data, sixteen standard map projections, pie graphs and bar charts, clear and comprehensive documentation, and extensive debugging aids.

APPLICATIONS

DISSPLA produces a UNIPLOT neutral picture file which can be input to the UNIPLOT post-processor.

Typical applications include:

- Management information charts
- Marketing studies
- Research and development plots
- Inventory level charts
- Cost projection and evaluation
- Worldwide distribution analyses
- Stockholder reports
- Project execution monitoring
- Flip charts
- Speed data display
- Visual debugging
- Structural and electrical analysis

In the United States and Canada, customers may use the DISSPLA package in either of two ways:

(1) over-the-counter at more than 60 CYBERNET computer centers or remote service centres located in major metropolitan areas coast-to-coast or (2) from private terminals located in their own offices, linked into the nationwide CYBERNET network.

Internationally, customers may access the program either over-the-counter or from remote terminals linked to CDC centers in Paris, France; Frankfurt, Germany; The Hague, Netherlands; Stockholm, Sweden; and Sydney and Melbourne, Australia.

DISSPLA is supported jointly by Integrated Software Systems Corporation and Control Data, and is fully documented.

For more information, please contact your nearest Control Data sales representative, or write:

CYBERNET Services
CONTROL DATA CORPORATION
P.O. Box 0 HQW05H
Minneapolis, Minnesota 55440



DMS-170 DATA MANAGEMENT SYSTEM

Control Data's DMS-170 provides an all-encompassing data base capability. The user is not required to make an enormous commitment when the consequences for the particular installation are not yet investigated. Through Control Data's modular approach, the potential for highly sophisticated applications lies latent in the wide range of solutions offered by their data base management software technology. An installation can grow from simple data base applications that use only CDC CYBER Record Manager (CRM) with COBOL or FORTRAN to sophisticated applications of great intricacy.

The record and file structure compatibility provided by CRM enables implementation of the most fundamental level of the data base concept, in a streamlined manner that facilitates easy conversion from one philosophy of management to another. Data creation can take place through many other Control Data software products in addition to those specifically designed for data management.

The Data Definition Language (DDL) alleviates the need of the applications programmer to make burdensome and potentially troublesome decisions concerning data formats each time a program is written, and provides computer systems personnel with greater operational control. Interrogation and updating are handled with maximum convenience through QUERY/UPDATE and the data management facilities of COBOL and FORTRAN; through these languages programmers of widely varying expertise are permitted immediate data base access for file manipulation. The CDC CYBER Database Control System (CDCS) functions as a centralized control module to assist in efficient access and ensure file integrity and security. As specific data base applications develop, their definition is the responsibility of the data administrator.



DNS (DISTRIBUTED NETWORK SYSTEM)

The Distributed Network System (DNS) concept is based on a sharing of resources that allows computing power to be distributed to many users at reduced cost. DNS permits a distributed complex of computers, terminals, and communications lines to operate as a single information-processing facility. DNS applications include engineering firms, credit bureaus, manufacturing companies, educational institutions, government and civil agencies as well as private citizens. Ready access to this information and computing pool eliminates the need for data files, program replication and redundant computer and terminal use.

A unique feature of DNS is the integration of unlike computer systems, large and small. Several unlike computer systems can engage in a direct data and services interchange. Likewise, thousands of multi-vendor terminals can communicate with multi-vendor hosts or terminals located thousands of miles away. A network of minicomputer systems can also realize large economies of scale when united under DNS.

DNS extends the power and flexibility of both terminals and host computers. Specially designed communications processors relieve the host of all network traffic tasks, leaving more memory, more processor capability and extensive software resources. In addition, services can be changed dynamically to suit specific individual needs.

Users who do not have large-scale or specialized computers on-site can access them via the network...users with on-site computers can access other off-site computers for their special data processing needs.



8-BIT SUBROUTINES

The 8-Bit Subroutine package is a group of user-callable subroutines for processing 8-bit data in the CDC CYBER 170, CYBER 70, and 6000 series computers. This product is intended for use under NOS 1.0, KRONOS 2.1, and SCOPE 3.4 operating systems. The subroutines are accessed by FORTRAN CALL and COBOL ENTER statements and by COMPASS return-jump instructions. Input/output operations are under the control of CDC CYBER Record Manager (except the routine COPY8P).

Some of the main features of this package are:

- Convert IBM 360/370 sequential 8-bit tape or card files to CDC internal format, maintaining 8-bit significance where necessary.
- Perform data moves, comparisons, packing and expanding of the converted data in which 8-bit significance has been maintained.
- To copy an IBM 360/370 print file to a 595-6 or 596-6 extended print file maintaining upper and lower case characters via the 95-character ASCII character set (only when this character set is supported by the operating system).
- IBM 360/370 8-bit sequential tape files may contain EBCDIC character string data, ASCII character string data, bit stream data, or arithmetic data in IBM format.
- The stand-alone program COPY8P provides the capability to copy an IBM 360/370 print file into a CDC-compatible print file without loss of 8-bit (upper and lower case) significance.



FORM (FILE ORGANIZER AND RECORD MANAGER)

File Organizer and Record Manager (FORM) is an integrated group of general-purpose routines for file and record manipulation, with enhancements to allow applications to be built by its use. In its simplest use, it is a data file copy program. Through the use of free-form parameters, however, the user can invoke more sophisticated functions that recognize data field values and relationships. At this level, FORM acts as a query/retrieval processor, report writer, file builder, file splitter, or file/record reformatter.

FORM provides parameterized routines for manipulating bit, character, and byte strings for the purpose of moving, scanning, comparing, inserting, and extracting without regard for word boundaries. These routines are made available to the COMPASS/FORTRAN/COBOL user for use independent of FORM.

FORM's functions are available to the user from control cards and from programs written in FORTRAN Extended, COBOL, and COMPASS. String manipulation can be used as subroutines or as macros; are not available through control cards.

FORM is designed in modular fashion to allow for expansion; OWNCODE exits provide the user with the ability to further specialize FORM's functions to particular applications.

FORM can be used at different levels of sophistication. Using the basic module only, full or partial files can be duplicated. Facilities are included for initial file positioning, reblocking, run statistics, and error processing. In this mode, FORM operates only at the file/logical record level, and has no knowledge of the structure or content of the data. This is the typical utility copy function.

At an intermediate level, FORM becomes a tool for use by applications programs. In this mode, additional knowledge of logical record format allows functions of foreign data format conversion, reformatting of records, record sequencing, and print formatting.

Using the full capabilities of FORM, some applications can be entirely processed by FORM functions. Free-form parameters minimize the complexity of coding at any level of usage.

FORM includes an extensive conversion capability for transforming IBM System/360 or 370 tape formats to CDC CYBER 70/170 tape formats and vice versa. This major module of FORM is modular in itself and lends itself to the addition of routines for converting other formats as they become desirable. It will accommodate a single 360 or 370 input or output tape. The tape converter routines are both a module of FORM and an independent package. When used as a module of FORM, the conversion takes place prior to any other consideration of the input logical record, or as the last function before output of a logical record.



FORTRAN DATABASE FACILITY (FDBF)

The CDC FORTRAN Database Facility (FDBF) 1.0 operates under control of the Network Operating System (NOS) on a CDC CYBER 170 computer system.

The FORTRAN Database Facility provides FORTRAN users access to DMS-170. FDBF consists of three components; a FORTRAN subschema compiler, a preprocessor to the FORTRAN compiler, and a set of routines to provide the interface between a FORTRAN application program and CYBER Database Control System (CDCS).

The major features are:

- A FORTRAN Data Description Language to:
 - describe all or part of a database.
 - redefine characteristics of data items.
 - rename and reorder data items.
 - restrict occurrences of relations defined in database.

- A FORTRAN Data Manipulation Language to:
 - create, retrieve, and modify records in a database.
 - satisfy privacy controls for files.
 - establish exclusive use of database files.
 - synchronize recovery actions with CDCS.



FORTRAN EXTENDED 4

The FORTRAN Extended compiler for the CDC CYBER 170 series computer system provides for expressing mathematical and scientific problems in a familiar mathematical notation. The compiler is designed to provide the complete capability of the FORTRAN language as described by the ANSI publication, FORTRAN X3.9-1966. Several additional features are provided which add considerable power and flexibility.

The FORTRAN Extended compiler operates under control of the operating system. It utilizes the operating system's multiprogramming features to provide compilation and execution within a single job operation, as well as simultaneous compilation of several programs. The object code produced by the compiler is comparable with code written by an experienced COMPASS programmer. Sophisticated techniques (e.g., on-line evaluation of some functions, critical path analysis of instruction sequences) are used to produce highly optimized object jobs which take full advantage of the speed and compute power of the computer systems.

Debugging aids implemented with FORTRAN Extended provide a variety of features to aid in program checkout. Typical compiler output consists of a source code listing including easily understood diagnostics listed by statement numbers. The programmer can also obtain a cross-reference table of symbols used or a complete listing of COMPASS code generated by source statements.

Some of the features of the FORTRAN Extended system include:

- Interactive terminal control
- Use of OCTAL and HOLLERITH data in language elements
- Use of MASKING expressions statements
- Two branch ARITHMETIC and LOGICAL IF
- Use of IMPLICIT and LEVEL specification statement
- Various utility subprograms, e.g., DUMP, TRACE
- Provides for PRINT, PUNCH, BUFFER OUT, NAMELIST, ENCODE, and DECODE input/output statement
- Comprehensive diagnostics
- Flexible subroutine structure
 - multiple entry points to subroutines
 - return to one of several statements in main program through multiple return statements
 - mixed FORTRAN and COMPASS coded routines
- Use of overlays to reduce amount of storage required



FORTRAN EXTENDED 5 AND FORTRAN COMMON LIBRARY

The FORTRAN Extended Compiler for the CDC CYBER 170 Series computer system provides for expressing mathematical and scientific problems in a familiar mathematical notation. The FORTRAN Extended Compiler 5 (FTN5) and the FORTRAN Common Library 5 (FCL5) together implement the full FORTRAN 77 language described in the ANSI publication, FORTRAN X3.9-1978. Several extensions to the standard language are provided which add considerable power and flexibility.

The FORTRAN Extended compiler operates under control of the operating system. It utilizes the operating system's multiprogramming features to provide compilation and execution within a single job operation, as well as simultaneous compilation of several programs. The object code produced by the compiler is comparable to code written by an experienced COMPASS programmer. Sophisticated techniques (e.g., on-line evaluation of some functions, critical path analysis of instruction sequences) are used to produce highly optimized object jobs which take full advantage of the speed and compute power of the computer systems.

Many of the extensions to the previous standard FORTRAN (as defined in ANSF publications FORTRAN X3.9-1966) implemented by Control Data in FORTRAN 4 (FTN4/FCL4) also exist in FORTRAN 5 (FTN5/FCL5). Some of the extensions Control Data employed in FTN4/FCL4 are not utilized by FTN5/FCL5 because of source program portability or because they are in conflict with the new standard. Other FTN4/FCL4 features have become redundant by new standard features. All together, some 55 items of the FTN4/FCL4 language do not exist or operate differently in FTN5/FCL5.

FTN5 supports three modes of compilation. OPT=0 (equivalent to TS mode in FTN4) offers fast compilation with little regard to optimizing generated code and is used primarily in an environment where many people are compiling small programs. OPT=1 offers reasonable compilation speed with moderate regard to optimization of generated code. OPT=2 offers slower compilation with high concern for optimized generated code.

Newly recognized forms in FTN5/FCL5 include Charter as a type statement, Parameter and Save as specification statements, If-Then-Else as control statements, and Open, Close, and Inquire as input/output statements. FTN5/FCL5 offers extensions to the syntax for some of these forms. Control Data extensions continued from FTN4/FCL4 into FTN5/FCL5 also include such complete statement forms as Encore, Decode, and Namelist, plus minor additions to other statements described in the previous standard.

FTN5/FCL5 and FTN4/FCL4 can coexist in a running system. FTN5/FCL5 includes support for Control Data CYBER Interactive Debug and Post-Mortem Dump. FCL5 must be used in conjunction with FTN5 to provide execution time support for the new statement forms.

A conversion tool is available to modify existing source programs for processing by FTN5/FCL5.



FORTRAN 4 TO 5 CONVERSION AID

The FORTRAN 4 to FORTRAN 5 Conversion Aid makes most of the required conversions automatically and issues messages when a conversion is necessary but requires human intervention. The Conversion Aid can produce a complete replacement source deck, changed cards only, or changed cards plus attendant Update/Modify directives. Listing options exist to clearly indicate each conversion mode.

Source language programs accepted by FORTRAN 4 (FTN4) are not universally accepted by FORTRAN (FTN5) since several language features present in FTN4 are in direct conflict with the standard as defined in ANSI publication X3.9-1978. Other features of FTN4 are redundant in view of the new standard features and were removed in the interest of clean language definition or compiler size.

For each program unit processed, the FORTRAN 4 to 5 Conversion Aid produces a FORTRAN comment indicating complete conversion, plus a message indicating whether the converted program unit remains as acceptable input to FORTRAN 4.



GPSS V/6000 — GENERAL-PURPOSE SIMULATION SYSTEM

GPSS V/6000[†] is a general-purpose simulation system for modeling and analyzing systems in engineering and management science. Operational on CDC CYBER 170, 70, and 6000-Series Computer Systems, GPSS V/6000 is available worldwide through Control Data's CYBERNET computer services networks.

This program provides an effective means of testing and evaluating the performance of existing and proposed system configurations under various operating conditions. Any physical process which can be modeled in flow chart form (such as message switching in a teleprocessing network or the flow of customer traffic through a supermarket) can be readily simulated using GPSS V. Now one of the most widely used simulation languages available, GPSS V represents many years of development and implementation efforts in a wide variety of industries and businesses.

GPSS V establishes mathematical models and provides simulation results for subsequent analysis, thus ensuring rapid solutions to a wide variety of business problems. The system is tailored for the practical planner — it is easy to learn and use, and requires no previous programming experience. GPSS V allows the user to describe and study many logical and procedure-oriented processes whose complexity makes an accurate performance evaluation otherwise impossible. This enables the user to investigate and evaluate proposed changes to existing policies, methods and operations without capital investment or disruption of existing operations. In a matter of minutes, the computer can evaluate weeks or even years of simulated activity.

Programs written for processing by earlier versions of GPSS will be handled properly by GPSS V/6000. In addition, GPSS V/6000 is upward-compatible with other versions of GPSS on non-CDC systems.

Among the many areas of applications of GPSS V are:

- Computer system modeling
- Airport ground support
- Tanker car and fleet routing
- Transportation modeling
- Pipe fabrication modeling
- Materials inventory modeling
- Pipe line loading
- Bank paper work flow
- Job shop production scheduling
- Power plant fueling logistics
- Nuclear projects scheduling
- Post office mail handling
- Logic circuitry design
- Warehouse locating
- Lumber mill modeling

[†]Developed by Northwestern University



IAF (INTERACTIVE FACILITY)

The Interactive Facility (IAF) is a Network Application Program under the Network Operating System (NOS) that provides the terminal user with a wide range of time-sharing capabilities. Although a large number of users may be simultaneously accessing IAF, the software that controls the system assures the complete separation of each user's activities. Thus, each user may conceptually imagine that he/she is the only user of the system.

HIGHLIGHTS

- Provides user identification and validation through log-in procedure. A user name and password sequence is used to identify the user to the system, to validate that this user is an approved user, and to specify specific system access limits defined for this user. These access limits define which system resources and how much of these resources are available to the user. The resources include applications, central memory, central processor, mass storage, magnetic tape, and unit record peripherals.
- Maintains control of all information files created by the users. Prior to creating a file, the user names it and may subsequently call for it by name at any time. IAF assumes that any file belongs exclusively to the user creating it, and will deny any other user's request to access it. However, a user may specify that his/her file be made available to other users and further may specify the level of access granted to other users. The levels include execute only, read, extend, modify read and allow modify, and write. The owner of the file may also determine when and which alternate users have accessed his file.
- Provides for control of all interactive sessions from remote terminals through a comprehensive set of system commands. These include commands for terminal control, job processing, file handling, editing, etc.
- Allows lengthy time-consuming command sequences to be consolidated to a one-line user-oriented call through the interactive use of procedure files. Full procedure file capabilities and features are available to the interactive terminal user including parameter substitution.
- Ease of job or information file entry and modifications by use of the IAF program editor or text editor. The program editor provides an effective mechanism for inputting and modifying program files through the use of statement line numbers. The text editor employs English-like mnemonics to provide a comprehensive capability for the manipulation of character and string text information. The edit commands operate on the entire file without special reformatting of the file or having the file memory resident.
- Dynamic interaction with and control of user jobs during execution. This provides for data input to and output from a running application program and also provides the capability to control job execution. If desired, the user can submit the job for batch execution which disassociates the execution of the job from the terminal and allows the terminal user to continue with other session activities.
- User control of terminal characteristics such as line width, page length and half-/full-duplex if the defined default values do not meet user requirements.

IAF, in conjunction with NOS, provides the user with a powerful and flexible time-sharing system designed with the user in mind and yet one which optimizes the use of system resources. The user interacts with IAF through simple English-like commands which relate effectively with the action performed. This allows the new or novice user to become productive with minimal training and to progress as the user's needs and experience dictate. Regardless of the expertise of the time-sharing user, the system makes efficient use of resources and provides the full power of the CDC CYBER 170 computer system to the time-sharing user.



IMSL
(INTERNATIONAL MATHEMATICAL AND
STATISTICAL LIBRARY)

The International Mathematical and Statistical Library (IMSL) is a collection of FORTRAN subroutines and function subprograms in the areas of mathematics and statistics. It is one of the most widely used statistical and scientific packages and is rapidly becoming an accepted industry standard by universities. It operates under control of the Network Operating System (NOS) on CDC CYBER 70/170 series machines.

A list of the FORTRAN subroutines and function subprograms included in the current library edition is given below.

- Analysis of Experimental Design Data
- Basic Statistics
 - Data Screening; Transgeneration
 - Elementary Classical Inference
- Categorized Data Analysis
- Differential Equations; Quadrature; Differentiation
- Eigenanalysis
- Forecasting (Box-Jenkins); Econometrics; Time Series
- Generating and Testing of Random Numbers
- Interpolation; Approximation; Smoothing
- Linear Algebraic Equations
- Mathematical and Statistical Special Functions
 - Probability Distribution Functions
 - Special Functions of Mathematical Physics
- Nonparametric Statistics
 - Analyses of Variance
 - Binomial or Multinomial Bases
 - Hyper (or Multihyper) Geometric Bases
 - Kolmogorov-Smirnov Tests
 - Randomization Bases
 - Other Bases
- Multivariate Statistics
 - Factor Analysis
 - Principal Components Analysis
- Regression Analysis
 - Linear Models
 - Special Nonlinear Models
- Sampling
- Utility Functions
 - Error Detection
 - Special I/O Routines
- Vector, Matrix Arithmetic
- Zeros and Extrema; Linear Programming



MAINTENANCE PACKAGE FOR NOS

Maintenance tools for NOS are provided on several release tapes. These maintenance tools are divided into two categories: SYMPL compiler and maintenance programs.

SYMPL

SYMPL (Systems Programming Language) is designed to facilitate systems programming; it does not contain some features normally found in higher level languages, such as complex arithmetic and input/output capability. Instead, it contains features particularly suited to systems programming — bit manipulations, based arrays, and an elementary macro capability. It produces code optimized for efficient register and functional unit usage, particularly oriented toward the CYBER/6600 type mainframe.

The SYMPL compiler is written mainly in SYMPL; only the system interface routines are in COMPASS. Thus an absolute binary of SYMPL is necessary for installation if changes are to be made to the source. SYMPL has no installation parameters. Before SYMPL can be installed, NOS, COMPASS, CYBER Record Manager and the FORTRAN Extended object library must have been installed.

In addition to the SYMPL compiler, the following products are also provided with the NOS Maintenance Package:

- Customer Engineering diagnostics
- Program library for CDC 881/883 Disk Pack Formatting
- Miscellaneous maintenance tools

The miscellaneous maintenance tools include binaries for:

- Time-sharing stimulator
- Dayfile sort program
- P-register analyzer
- Diagnostic routines
- Status/control register maintenance programs
- Deadstart dump interpreter
- CPU debugging routine
- Network Communication Supervisor dayfile analyzer
- Program library for SIFT



MSL (MATH SCIENCE LIBRARY)

The Math Science Library, operating on the CDC 6000, CYBER 70 (Models 71, 72, 73, and 74), and CYBER 170 series computer systems, is a collection of new existing mathematical routines – 75 percent of which are available from no other source. The library contains over 400 routines which are described in an 1800-page handbook to guide users to the routines most suitable for computational objectives. The handbook is written primarily for the user and contains a great deal of practical information.

The Math Science Library consists of eight major areas of computational mathematics as follows:

- Programmed arithmetic – normalized and unnormalized floating-point, fixed-point, rational arithmetic, and interval arithmetic
- Elementary functions – algorithms and subroutines for basic functions, $\sin x$, $\cos x$, etc.
- Polynomials and special functions – polynomials in one real or complex variable, and special functions and interrelationships
- Ordinary differential equations – initial-value, boundary-value, eigenvalue, linear, nonlinear, etc.
- Interpolation, approximation and quadrature – interpolant (approximate) a linear or nonlinear combination of basic discrete or continuous functions, various norms of approximation, and an adaptive quadrature algorithm.
- Linear algebra – equation-solving in the exactly and overdetermining cases matrix inversion, eigenvalue, and lambda-matrix problems
- Probability, statistics, and time series – random number generator, probability functions, descriptive statistics, and hypothesis tests
- Nonlinear equation – polynomials and other functions of a single real or complex variable

The library routines are capable of solving a wide variety of the most frequently occurring problems in these areas. The routines are designed to be building blocks in computer programs for engineering, scientific or management applications. They have been selected because of their favorable properties including:

- Speed – efficiency of the coded algorithm
- Accuracy – proper formulation to avoid numerical difficulties
- Reliability – safeguarding against breakdown from numerical difficulty or misuse
- Maintainability – ease of conversion to other machines and general clarity of coding
- Flexibility – ability to solve a class of related problems



NAM (NETWORK ACCESS METHOD)

The Network Access Method (NAM) provides a systems application program the ability to communicate with terminals and other application services in the network. NAM operates at a system control point, performing similar functions for communications equipment as other access methods provide for mass storage equipment. The NAM supports CDC-developed application programs which themselves support remote batch and interactive terminal activities and network control. NAM also supports special user applications programs which communicate with the network.

HIGHLIGHTS

- Guarantees network security by controlling access between application programs and terminals, and between different application programs.
- Establishes, maintains, and terminates connections between application programs and terminals, and between different application programs.
- Isolates the application program from hardware idiosyncracies by treating a variety of terminal devices within a terminal classification; i.e., interactive, as a single "virtual terminal" type. This relieves the application program from considering actual terminal characteristics.
- Isolates the application program from detailed network protocol.

DESCRIPTION

The Network Access Method (NAM) is a combination of software elements for managing communications data flow between the communication network and application services under either variant of the NOS operating system. Application programs utilizing the NAM can be written in COMPASS or higher level languages such as COBOL, FORTRAN, or SYMPL. NAM imposes no data structures on the application and encompasses a simplified user-oriented NAM/application protocol distinct from network protocols.

NAM provides a set of macros which allows an application program to communicate with NAM. Each application program may have a number of logical connections, where each logical connection is associated with a single terminal or with another application program. For each logical connection, NAM maintains a set of control tables and buffers. These allow NAM to queue data to and from the associated terminal or application program. NAM itself actually performs the physical input/output with the communications network.

NAM provides a "virtual terminal" mode which effectively relieves the application program from the necessity of knowing the detailed characteristics of the different terminal types. An optional "transparent" mode allows the application program to use all the functions of the specific terminal if it so desires.

A set of macros are provided permitting the application program to connect and disconnect to and from NAM. These macros perform similar functions for application programs as LOGIN and LOGOUT procedures provide for users at terminals. They allow the installation to control access to the communications network from programs executing in the host computer. An application program may continue to execute under the operating system even after it has disassociated itself from NAM.

As various events occur in the network, supervisory messages are passed to the application program. Such messages need not be solicited by the application program, but are transferred automatically each time the application program issues a call to NAM. They may, for example, inform the application program of a new logical connection for a terminal which desires service from it, or inform it of the fact that some failure has occurred. Similarly, the application program uses supervisory messages to communicate with NAM. For example, an application program may wish to disassociate itself from some terminal with which it has a logical connection. The use of supervisory messages between NAM and the application program obviates the necessity for defined table structure in the application program. NAM allows the application program to use whatever table structure is most efficient for itself. Additionally, NAM does not constrain the kind of buffering used by the application program. The program may provide a buffer for each logical connection, or, alternatively, it may perform all its input/output in a single buffer. This allows the application programmer maximum flexibility in the design of his program.



NETWORK PRODUCTS SOFTWARE

The CDC Network Products Software is based on an architecture that supports a wide range of configurations and applications. This software is supported by the CDC Network Operating System (NOS). NOS Network Products Software includes the following products:

- CCP — Communication Control Program
- NAM — Network Access Method
- RBF — Remote Batch Facility
- IAF — Interactive Facility
- TAF — Transaction Facility
- MCS — *Message Control System*

HIGHLIGHTS

- **Application Interface** — The Network Access Method (NAM) provides a centralized and defined interface between user applications and the communications network. Remote batch, interactive and transaction applications are provided. Additionally, users may prepare their own applications to interface to NAM.
- **Routing** — NAM controls both the logical and physical connections between terminals and applications. In the event of a communication line failure, NAM in conjunction with the Communication Control Program (CCP) can establish alternate routing.
- **Security** — Data paths are created only after proper access control procedures have been executed. Data paths are revalidated after a connection failure or when a connection is switched to a new application. Messages concerning all unauthorized access attempts are logged and made available to the computer center operations personnel.
- **Stimulator** — A Network Product Stimulator (NPS) software product is provided as a tool to test and evaluate system performance under varying load conditions, and to verify proper operation of applications interfacing to NAM. It also provides analysis of test results.
- **Network** — A network of local and remote communication processors (CDC 255X series), interfacing to a wide range of terminals, can be constructed with the Network Products Software. A bit-oriented protocol (HDLC/ADCCP based) is used to efficiently connect remote nodes to the central site communication processor. Additionally, terminals with various protocols (HASP Multileaving, Control Data Mode 4, Asynchronous, etc.) can connect to either remote nodes or central site communication processors.
- **Network Definition** — The network is defined by use of a Network Definition Language (NDL). When processed at a central site, the NDL statements build a Network Configuration File and a Local Configuration File. These files are used to define the initial state of the communication processors and terminals which can then be modified as required during network operations.

- **Network Control** – Network control is performed by the supervisory functions of NAM, which allow a network operator to enable, disable, reload, dump or status various network elements. Operator messages may be sent to all terminals or individual terminals.
- **Diagnostics** – Network products provide in-line, on-line and post-mortem diagnostics. Diagnostics are also provided to allow verification of the correct operation of terminals connected to the network. The on-line and in-line diagnostics provide appropriate alarm messages to the operator.
- **Cross Support** – The network processing units (CDC 255X series) within the network are supported by a cross system which provides for the installation, maintenance and enhancement of the CCP software which is then downline loaded into local and remote communication processors. The CCP software has been developed using PASCAL, a powerful and effective high-level implementation language.
- **Virtual Terminal** – Virtual terminal is a concept introduced with CDC CYBER Network Products to minimize the interdependencies between the terminal user's applications program and the physical characteristics of the terminal. This concept defines a universal interface to application programs and is implemented in CCP. The Terminal Interface Programs (TIPs) in CCP have been implemented to provide translation between the physical terminal characteristics and the universal terminal interface.
- **Terminal Support** – TTY and TTY-compatible, asynchronous at speeds to 9600 bits per second; synchronous to 19,200 bits per second. Compatible terminals are:
 - TTY M33, M35, M37, M40
 - CDC 751-10, 752-10, 756-10, 713-10
 - Memorex 1240
 - Tektronix 4010, 4014
 - Hazeltine 2000
 - IBM 2741 Correspondence
 - CDC Mode 4, synchronous at speeds to 19,200 bits per second
 - CDC 711, 714
 - CDC 200UT (BCD/ASCII)
 - CDC 731, 732, 734
 - CDC CYBER 18-5
 - HASP Multileaving Workstation at speeds to 19,200 bits per second.
 - IBM 360/25 HASP Workstation or equivalent
 - CDC 2551 remote node
 - Remote Node Link Interface at speeds to 19,200 bits per second



NPS (NETWORK PRODUCTS STIMULATOR)

The Network Products Stimulator (NPS) provides a method of testing and exercising network host product software; it is not an interactive test tool but rather a tool to test interactive systems. Through use of the Network Products Stimulator, the Network Access Method (NAM) and applications using it can be dynamically exercised as though by a live terminal network. Controlled terminal sessions are input to NAM from a previously prepared library of user written script that simulates actual terminal sessions.



PDS/MAGEN

PDS/MaGen is a complete, yet easy-to-learn, matrix generation and report writing system which operates under the Network Operating System (NOS). PDS/MaGen interfaces with and complements the APEX-III linear programming product and aids the user in generating models without regard to the syntax, formats and repetitious data entry requirements characteristic of linear programming programs. PDS/MaGen allows the user freedom to focus efforts on the structure of the problem and to significantly reduce model development time and cost.

Additional features and benefits of PDS/MaGen are:

- Generation of linear programming matrices in MPS format for access by APEX-III
- Data base generation and maintenance
- User-tailored report generation (by accessing the APEX-III solution)
- A data structure that is independent of generator programs making it easier to execute multiple runs of varying size and/or detail
- Low core requirements (only non-zero data elements are stored internally)
- In-core operation for fast execution



PERT/TIME

The PERT/TIME system developed for the CDC 6000-, the CYBER 70-, and the CYBER 170-series computer systems utilizes a time-oriented network structure. The network represents the flow of work activities and the events that mark their completion.

PERT/TIME is a fully integrated system of programs that are overlayed as needed in the processing of the network. Data entered into the system is stored on a master file until deleted or modified by the user.

The PERT/TIME system offers the following features:

- A network capacity of 8000 activities or 6000 events
- A network time span of up to a ten-year period from start date to completion date
- As many as seven activity or milestone event reports can be generated at each computer run
- Update capability using file maintenance feature
- Condensed report options
- Flexible calendar controlled by input



PL/I

Control Data PL/I is an implementation of the language described in ANSI document BSR X3.53-1976, with the exception of aggregate expressions, data-directed input/output, the DEFAULT statement and COMPLEX data type. The compiler interfaces to CCG, the Common Code Generator based on the FORTRAN Extended code generator.

PL/I was designed to be a language which would satisfy the needs of both scientific and commercial programming. It has many of the features of FORTRAN, COBOL, and ALGOL. Like ALGOL and PASCAL, PL/I is a block-structured, recursive language.

There are four types of storage defined in the PL/I language: static, automatic, controlled, and based. Static storage is defined at compile time, allocated at load time, and always occupies the same storage locations. Automatic storage is allocated dynamically at execution time, upon activation of a block. If a procedure is called recursively, generations of automatic storage are stacked. Controlled storage is explicitly allocated and freed at execution time by the program. Generations of controlled storage can also be stacked. Based storage is also allocated and freed explicitly by the program at execution time, but generations are referenced through pointers and are not stacked.

PL/I data can be either computational data or noncomputational data. Computational data is either arithmetic or string. Arithmetic data has the four attributes of base (decimal or binary), scale (fixed or float), mode (real or complex), and precision (number of digits and scale factor). String data can be of type character or binary, and of fixed or varying length. Noncomputational data includes label, format, pointer, offset, area, entry, and file data.

There are two categories of input/output in PL/I: stream and record. Stream input/output is character-oriented and deals with "human readable" data. Stream input/output can be edit-directed (an explicit format is given), list-directed (the format is determined from the data to be output), and data-directed (a feature similar to FORTRAN's Namelist). Record input/output is direct input/output with no conversion to character form. It is record-oriented and is meant as an efficient way to communicate data from one PL/I program to another.

An interesting feature of PL/I is the ability to set ON conditions which enable the program to execute out-of-line procedures upon the occurrence of an error or exception condition. The types of conditions which can be sensed are computational (such as divide by zero, overflow, etc.), area (attempt to allocate more storage than is available in an area), system-action (such as end-of-job), and programmer-named. Conditions can be explicitly SGINALed in the program.



QUERY/UPDATE

QUERY/UPDATE is a high-level, English-like, conversational language for use in querying and manipulating data files organized under CRM with multiple indexing. Since it is designed for use in a terminal-oriented environment, communications with a data base is in a conversational mode. This dramatically shrinks the feed-back loop between decision maker and critical information acquisition, and aids in achieving maximum exploitation of a data base. QUERY/UPDATE is especially useful for quick, one-time searches of a data file; an answer to a specific question can be provided to a QUERY/UPDATE user in a matter of seconds or minutes, rather than waiting for a programmer to write and debug a program to provide the desired information. QUERY/UPDATE provides all of the capabilities and advantages normally associated with a higher level language, including:

- Simple syntax
- Powerful verbs
- Use of arithmetic and boolean expressions
- Features for out-of-line procedures which may be recorded as cataloged sessions
- Use of the multiple indexing feature of CRM
- Extensive report writing facility
- Use of DDL-generated description of data base files for easy, standardized reference to the data
- Designed for interactive use
- Usable in batch mode as well as interactively

Experienced programmers can take advantage of such QUERY/UPDATE features as the ability to use arithmetic expressions and perform out-of-line procedures, organize data into arrays, and search interactively for individual elements. At the same time, employees with no programming knowledge can manipulate files through QUERY/UPDATE by following procedures established by experienced personnel for data base definition and maintenance.



RBF (REMOTE BATCH FACILITY)

The Remote Batch Facility (RBF) is an application service under the Network Operating System (NOS) which provides the capability to transfer data between files on a CDC CYBER 70/170 or 6000 series host computer and remote batch terminals in the network. RBF interfaces to the Network Access Method (NAM) in order to communicate with its terminals. It operates as a normal application program at a control point and requires no special privileges of NAM.

HIGHLIGHTS

- Concurrently services and assures approximately equal response and transmission times for all terminals active in batch mode in the network.
- Provides for interleaved input/output from the terminal's card reader and line printer.
- Accepts data from the terminal's card reader, creating from it files represented in the operating system's input queue.
- Transmits files from the system's print queue to the terminal's line printer.
- Accepts instructions conforming to the network command language from the user between, or during interruption of, batch input/output data flow.
- Allows the user to backspace or skip forward on a file that is being printed at his terminal, to repeat printing of a file one or more times, to terminate the reading or printing of a file, and to suppress format control and/or banner page for the printing of a file.
- Allows the user to delete his files, divert them to a different terminal, or to his host computer. The user may also change the priority of his print files in order to regulate the order in which they are transmitted to the terminal's line printer. A forms code capability is also provided to allow the user to control the set of files eligible for printing.
- Informs the user when one of his card files is transmitted and entered in the input queue. Informs the user when the normal flow of operation is disrupted due to system overflow, device conditions, unrecoverable hardware error, or invalid or inappropriate user commands so that the user may take appropriate action.
- Provides status information of the jobs, files, and devices belonging to the requesting terminal.
- Maintains accounting information which is posted in the system accounting files.



RPG II SYSTEM

DESCRIPTION

The CDC RPG II system is an easy to use language, well suited to common business applications. The functional aspects of this language provide the CDC CYBER 18 computer programmer with the following capabilities:

- Easily describe the formats of input/output (I/O) data
- Execute programs using both conditional and unconditional statements
- Easily describe access methods, matching, translation, update, and conditional processing of data files

HIGHLIGHTS

The RGP II system consists of a compiler, run-time routines, interpreter, and a set of utilities.

Compiler

The compiler executes only under the Mass Storage Operating System (MSOS) for the CDC CYBER 18-17, 20, and 30 computer systems. This compiler inputs source programs and generates interpretive object test in CDC CYBER 18 relocatable text format. In addition, it accepts source statements which are generally compatible with IBM System 3/RPG II. This compiler also consists of a control section which is resident in memory during the compiling process. Nine overlays are called in sequentially, to process the source program.

Run-Time Routines

These routines are linked with the compiler-generated code to form an executable RPG program. Each run-time program consists of a resident portion and three overlay segments.

Interpreter

The interpreter operates by sequentially examining each interpretive instruction and executing a corresponding routine. This interpreter executes under both MSOS and Real-Time Operating System (RTOS).

Data Manager

The data manager performs all input/output interface operations dealing with files. In addition to disk and magnetic tape, all devices that may be attached are considered to be recipients or suppliers of file data. Functions performed by the data manager include: open, read, write, update, add, set lower limit of key, and close.

Sort/Merge/Copy Program

This program allows the user to produce a sequenced file of data records from random input, and to merge two or more files.

Utilities Package

This package allows the user to exercise control over the file maintenance and assignment of input/output devices.

Disk Utilities

This packet allows the user to initialize the RGP file directory structure and to transfer files between an input/output device and a mass storage device.

Tape Utilities

These utilities provide the capabilities of a tape-handling system designed to process both EBCDIC and ANSI standard tape formats. The features include: blocking, conversion, record formats, initialization, labeling, printing, selection and verification.

Miscellaneous Utilities

The switch utility sets the external RPG indicators U1 to U8. The catalog utility enables the user to install the RPG object program in the MSOS program library for subsequent execution.



SANDERS GRAPHICS 7 POST-PROCESSOR UNDER TIGS

The SANDERS post-processor for the Terminal Independent Graphics System (TIGS) is a subroutine package which interfaces to a Sanders Graphics 7 intelligent refresh graphics terminal. The post-processor routines read graphics information from a pre-processor-generated Neutral Display File (NDF) and produce the appropriate commands which, when sent to the Sanders terminal, cause the graphics information to be displayed.

The minimum hardware configuration necessary to utilize this post-processor consists of a CRT display, a terminal controller unit, and an alphanumeric keyboard with function keys. The controller unit must be equipped with an asynchronous interface board and a 4K ROM board containing Sanders GSS4 firmware. Refresh memory consists of 8K words, expandable to 24K words. Hardware options supported are lightpen and trackball (or joystick) locators, and a hardcopy unit.

Features supported by the Sanders post-processor include selective erase, four hardware line styles, four hardware character sizes, eight intensity levels, highlighting, and 90⁰ character rotation. Communication is supported at baud rates of 300 or 1200 in asynchronous mode.



SIMSCRIPT II.5

SIMSCRIPT is a powerful, general purpose scientific programming language. Although it was developed primarily for discreet-event simulation programming, it is equally well-suited for non-simulation programming. SIMSCRIPT operates under the control of the operating system. Important features of SIMSCRIPT include:

- Dimension-free source and object programs
- Timing routine
- Report generator
- Random look-up tables
- Statements identifiable by either numbers or names
- Floating-point subscripts and DO loop control
- Integer truncation of floating-point numbers
- Octal input and output
- Hollerith output
- Optional text in many statements
- COMPASS and FORTRAN subroutines
- FORTRAN-like instruction repertoire
- Flexible, free-form syntax
- I/O statements similar to those of FORTRAN
- Partial block structure for structured programming
- Recursive subroutine capabilities
- Report generator facilities
- A sophisticated data structure concept with a world view composed of entities, attributes and sets.

SIMSCRIPT was designed primarily for simulation programming, and thus has many features which make analysis of simulated systems virtually automatic. The SIMSCRIPT system has two different methods for building simulation models. The one most familiar to simulation programmers uses special kinds of entities known as event notices which are scheduled at appropriate times during the simulation. SIMSCRIPT has a built-in timing routine to handle the scheduling. Alternately, the simulation model can be based on processes and resources. In either case, a wide variety of statistical functions, including random table look-up, enable the user to simulate most situations quite realistically. Many different types of reports can be generated using the report generator feature.

SIMSCRIPT is a powerful nonsimulation language, has access to the entire **FORTRAN** mathematical library, and allows the intermixing of assembly language and **SIMSCRIPT** code, and the calling of **FORTRAN** subroutines using **COMPASS** inserts.



SIMULA

SIMULA is a dynamic language which describes a sequence of events rather than permanent relationships. The SIMULA system for the CDC 6000, CYBER 70, and the CYBER 170 series is based on a general algorithmic language — ALGOL-60.

SIMULA illustrates similarities between systems and enables the programmer to consider all relevant aspects of a particular problem. Simulation is achieved by replacing a dynamic system with an abstract model. A SIMULA program provides a framework in which to build a model and analyze changes affecting the model. Continuous changes in the system are represented by a series of discrete changes in the model, and system time is held constant as each event in the changing system takes place. A set of instructions describes each component of the system separately and control transfers from one component to another to simulate interaction between them.

FEATURES

SIMULA provides the following features in addition to the ALGOL-60 capabilities:

- Text and character expressions are evaluated in the same way as arithmetic expressions
- Hierarchical, structured, compound objects called classes are described and dynamically generated
- Denotation and reference devices allow extensive list-processing of strings and compound objects
- Program blocks are extended by prefixing a class; all capabilities defined in the class become available within the block
- This extension capability enables the user to augment the language, orienting it to specialized applications
- System classes, which are part of the SIMULA 67 Common Base, provide specialized capabilities when used as a prefix to a user block
- Flexible input/output is easily adapted to specialized applications
- A quasiparallel system facilitates simulation of discrete event systems for processes interrelated by events, dynamically evaluated, rather than permanently related
- Extensive library capabilities provide tracing facilities for debugging, random drawing, and data analysis facilities, as well as the capabilities of the ALGOL library



SORT/MERGE

The SORT/MERGE system for CDC 6000, 7000, and CYBER 70/170 series computer systems is a high-speed processing program for manipulating and rearranging records into a prescribed order or collating sequence. The system is intended to satisfy the highest level of requirements specified by the ANSI COBOL standard (X3.23-1974). In addition, SORT/MERGE provides the capability to operate as a stand-alone product or as a subprogram called by a FORTRAN or COMPASS main program.

The modular construction of the SORT/MERGE program enhances the user's record processing options by providing the following capabilities:

- Sort only – sorts records from one or more input files (mass storage for Model 76, and mass storage or magnetic tape for other models of CDC CYBER 70/170 systems) into an order specified by the user
- Merge only – combines from 2 to 100 presorted input files into 1 output file
- Sort and Merge – sorts from 1 to 100 input files into an order or sequence specified by the user, and merges the output with from 1 to 100 presorted input files

At various stages during the SORT/MERGE, the system allows exits to modification routines (owncode routines). These exits (five exits for Model 76, and six exits for other models of CDC CYBER 70/170 systems) allow the user to perform various file maintenance operations, such as record editing, summarizing, deleting, or inserting.

The general features can be summarized as follows:

- Key fields
 - allows up to 100 key fields
 - supports several types of sort key codes:
 1. 8-bit ASCII (Model 76 only)
 2. Internal display
 3. 8-bit extended BCD interchange (Model 76 only)
 4. Floating point
 5. Internal BCD
 6. Fixed point integer
 7. Unsigned integer
 8. 6-bit COBOL
 - supports several types of collating sequences:
 1. 6-bit ASCII
 2. 8-bit ASCII (Model 76 only)
 3. 6-bit COBOL
 4. Internal display
 5. 8-bit extended BCD interchange code (Model 76 only)
 6. Internal BCD
- The collating sequence assigned to each character-coded sort key can be indicated in ascending or descending order

- All sort keys within a record are not restricted to a single type of code, collating sequence or order, but can be specified with variable codes, collating sequences or order
- Checkpoint/restart facility for capturing the total environment of a job which is abnormally terminated, thus permitting a restart from a checkpoint (all models except Model 76)



CONTROL DATA
CORPORATION

SPSS (STATISTICAL PACKAGE FOR THE SOCIAL SCIENCES)

The Statistical Package for the Social Sciences (SPSS) is an integrated system of computer programs for analyzing data, applicable in any area where statistical calculations are needed.

FEATURES

SPSS is a unified and comprehensive package, enabling the user to simply and conveniently perform many different types of data analysis. Although it was developed to meet the needs of social scientists, SPSS has general applicability in any discipline where statistical analysis is used. The system offers researchers a large number of statistical procedures commonly used in the social sciences. It allows a great deal of flexibility in the format of data, and provides the user with a comprehensive set of procedures for data transformation and file manipulation.

Rather than being merely a collection of individual programs, each requiring its own unique data definition and control statements, SPSS is an integrated system. Its wide range of statistical procedures all access the same standardized data file. Once the user describes a data file to SPSS, he can request any number of different statistical procedures to process that file. Thus, SPSS eliminates the time-consuming task of transferring data and results between noncompatible programs.

SPSS provides a comprehensive range of statistical techniques including:

- Descriptive statistics: mean, median, mode, standard error, standard deviation, variance, kurtosis, skewness, range, minimum, maximum
- Alphanumeric frequency counts
- Cross tabulations (contingency table analysis).
- Histogram and scattergram plots
- Student's t-test
- Correlation analysis: Pearson correlation, nonparametric correlation – Kendall and Spearman, partial correlation, tetrachoric correlation
- Regression analysis: multiple linear regression, stepwise regression, polynomial regression, nonlinear regression
- Analysis of variance and covariance (n-way)
- Multivariate analysis of variance
- Multiple classification analysis
- Factor analysis
- Principal component analysis

- Discriminant analysis
- Stepwise discriminant analysis
- Canonical correlation
- Guttman scale analysis
- Nonparametric statistics: Chi-square one-sample test, Kolmogorov-Smirnov one- and two-sample test, McNemar test, sign test, Wilcoxon signed-ranks test, Cochran Q test, Friedman two-way ANOVA, median test, Mann-Whitney U test, Wald-Wolfowitz runs test, Moses test of extreme reactions, Kruskal-Wallis one-way ANOVA
- Random number generation
- Generalized and 3-stage least squares
- Reliability (item and scale) analysis
- Repeated measurements analysis of variance
- Digital plotting: multiple line plots and scattergrams via UNIPLOT
- Item and scale analysis (test reliability)

The data management facilities of SPSS can be used to permanently modify a file of data, and can also be used in conjunction with any of the statistical procedures. These facilities enable the user to generate variable transformations; recode variables; sample, select or weight specified cases; and add to or alter the data or the file-defining information. Analysis is performed through the use of natural-language control statements and requires no programming experience.

SPSS is supported jointly by Northwestern University (Evanston, Illinois) and Control Data's Application Services and is fully documented.



TAF (TRANSACTION FACILITY)

To implement transaction processing, Control Data offers the Transaction Facility (TAF) which is a standard Network Operating System (NOS) product. TAF is a subsystem that interfaces to the Network Access Method (NAM) subsystem. NAM provides all terminal control and relieves the transaction programmer of any concern for the physical characteristics of the terminal network. A high-speed, single copy, data manager is included in TAF. TAF coexists with the other NOS products such as time-sharing and remote batch.

A single task may be used by many terminals, or multiple copies of the task may each be used by several terminals. Tasks may be chained together; that is, a task may initiate other tasks. These tasks may proceed asynchronously or, when complete, return control to the calling task. Tasks can be initiated by a terminal user or from a batch job. A batch job may be initiated from a task.

Tasks are easily written in COMPASS, FORTRAN, or COBOL with no special constraints other than the following:

- All terminal input/output is done through two simple CALL statements (a read and a write statement). Standard input/output is not allowed.
- An area of COMMON is set aside for communication with NAM.

TAF offers the following capabilities:

- Since TAF is a mini operating system that uses sub-control points (in a manner analogous to the control points of NOS) each task has its own RA and FL. This means that errors in tasks are trapped by TAF and cannot cause system failures. Each task has automatic memory protection. Multiprogramming between tasks is very efficient since each has its own exchange package.
- Initialization or recovery of TAF invokes standard procedure files that utilize any capability of the standard NOS system.
- Automatic queueing and scheduling of tasks. When the task library is built, the programmer designates the task as being either memory resident, serially re-usable or destructive. A memory resident task is a single copy task and all requests are queued for it. (Most tasks are not designated as memory resident.) A serially re-usable task has a queue limit associated with it and is kept on disk or in ECS. Once the number of requests exceed the queue limit, then another copy of the task is loaded. If its queue limit is exceeded, yet another copy is loaded. However, TAF checks to see if the first copy can handle another request thus avoiding extra loading time. By adjusting the queue limit, the programmer can guarantee at least one copy of the task to be resident in memory, plus retain the ability to use multiple copies. Most tasks are designated as serially re-usable. A destructive task is one that accepts a request and then releases all memory at completion. If another request comes in, another copy of the task is loaded from disk.
- Tasks are written so that they are available to TAF only as binaries. These binaries are in a preloaded format so as to incur none of the normal loader overhead. The library of tasks are kept under the originator's user number.
- Three data management choices are offered. The TAF Data Manager is a high performance data manager. A special version of Cincom's Total is also offered. A third choice is the CYBER Record Manager access methods Indexed Sequential and Direct Access.

- Performance tuning parameters are provided such as designating memory or disk residency for tasks.
- Tasks may be changed and the new binaries used while TAF is running.
- Dynamic memory management so that TAF uses only the memory required.
- Automatic recovery with notification to tasks that recovery has occurred.
- Both automatic journaling and optional journaling.
- Designation of a scheduling priority for each task.



TEKTRONIX 401X POST-PROCESSOR UNDER TIGS

Terminal Independent Graphics System (TIGS) 1 TEKTRONIX Post-processor is a subroutine package providing display generation and interaction with the Tektronix 4006 and 4010-4015. The display is produced from the neutral display file generated by the TIGS 1 Pre-processor. Locators supported are the cross-hairs and tablet.

Version 1.0 of TIGS has been implemented to run on Control Data 6000 series, CYBER 70 series and CYBER 170 series computers under the Network Operating System (NOS 1). The TIGS Pre-processor is required for installation and operation of TIGS.



TEXTJAB

TEXTJAB is an automated text processing language designed to promote efficient preparation and maintenance of evolving documents. The CDC TEXTJAB operates on CDC CYBER 70 or CYBER 170 series computers under control of the Network Operating System (NOS).

TEXTJAB outputs may be saved for later printout and/or printed in three different ways:

1. On CDC 580 series Line Printers with an extended print train containing both upper and lowercase characters.
2. Via interactive terminals supporting the full 128-ASCII character set. This may be performed locally or remotely as terminal communications permit.
3. An intermediate output listing can be printed on normal 64-character set printers in all uppercase. Overprinting emphasizes capitalized text and unavailable characters are mapped to the nearest approximation. (This form is normally used for draft working copies.)

TEXTJAB features extensive document formatting capabilities which include:

- Dynamic definition of page size, spacing margins, tabs, etc.
- Automatic left and right margin justification
- Skipping, indentation and tabs
- Underlining and capitalization
- Generation of tables and figures
- Footnote placement
- Automatic pagination
- Multi-level headings
- Page titles, subtitles and folio text
- Automatic section numbering linked to heading level, pagination and folio text
- Automatic tables of contents, tables and figures controlled by heading levels.

These features, along with the capabilities of the NOS operating system, provide the TEXTJAB user with a properly formatted document at any point during the evolution and maintenance cycles of a document. This allows the user to concentrate on content rather than format and to produce a quality document which when properly output is suitable for direct photo-offset printing.



TIGS (TERMINAL INDEPENDENT GRAPHICS SYSTEM)

TIGS (Terminal Independent Graphics System) is a graphics software system designed to support a variety of graphics display terminals while providing application program independence. It is not necessary to write programs tailored for a particular terminal's capabilities, although a programmer may do so.

TIGS consists of a single software preprocessor which provides display generation and interactive capabilities for a general class of terminals. There are also a number of software postprocessors which translate the generalized preprocessor output into specific display instructions. The preprocessor and postprocessors communicate through a neutral display file. Virtually all types of graphics terminals can be supported by adding the required postprocessors. Full time-sharing facilities are available at the terminal on an interactive basis.

Possible types of terminals, with varying levels of intelligence, for ultimate support might include:

- Alphanumeric/limited graphics
- Storage tube
- Raster scan
- Random stroke (refresh tube)
- Color

OPERATING ENVIRONMENT

TIGS is available under Control Data's Network Operating System (NOS) for CDC CYBER 170, CYBER 70 and 6000 series computers. TIGS software communicates with the host computer via the operating system's interactive facility - Network Access Method/Interactive Facility (NAM/IAF).

All TIGS functions are called from programs coded in Control Data's FORTRAN Extended, and are contained on special program libraries. An effort has been made to adhere to ANSI FORTRAN usage in the interest of clarity.

TIGS has been designed for ease of learning. This has been implemented not only in the system's composition but also by having a small number of parameters in any call to a graphics routine.

STRUCTURAL OVERVIEW

TIGS has the capability to construct, organize and display graphic material in either two- or three-dimensional mode and to interactively modify any display. The three-dimensional mode includes perspective viewing.

The TIGS preprocessor is an open-ended collection of routines which allows new features to be effected if necessary. The system is divided into the functions described below.

Primitives

The building blocks of TIGS are called primitives, which include lines, arcs, dots, plotting symbols and text. These are the basic drawing elements which graphics terminals are capable of producing and which in combination, make up even the most complex graphics displays.

The primitive's subroutines are used by the application to draw the required lines, dots, text, plot symbols and arcs on the terminal display surface.

Picture and Segment Manipulation

A collection of primitives is organized into a unit called a segment. Segments are defined by the application program to permit subsequent selection and/or identification by the terminal operator for functions such as translation, modification or deletion. No modification of the display can affect a level lower than a single segment. Segments are grouped to form the picture.

Feature Test, Mode Setting and Testing and Attribute Resetting and Testing

The feature test routines allow a programmer to inquire of the postprocessor to see if a particular feature is supported by the postprocessor or to allow the postprocessor to describe a particular feature. The features include such attributes as interaction support, number of intensity levels, types of line styles supported by the terminal and screen shape and range of coordinates.

The design of TIGS is centered around the concept of modality. When implementing a graphics application program, any of a large set of different modes and attributes, all of which have a default, can be set/tested and tested/reset.

Window and Viewport Manipulations

The TIGS user can specify that, at any given time, only a certain portion of a picture should be displayed on the terminal; this is called a window.

A viewport is that portion of the display screen on which a window is to be displayed. Windows and viewports control that portion of the picture displayed on the screen and where on the screen it is to be displayed. All or part of a picture can be selected to be included in a window.

The display screen may have the default viewport, which covers the entire screen, one viewport or many viewports dividing the screen into separate areas to suit the application program's needs. Further, viewports may be overlapped or superimposed to allow special viewing effects. Window and viewport operations include definition and deletion.

Transformation Matrix Operations

The transformation matrix operations include the facility to translate, rotate and scale a picture. Used in conjunction with stack manipulation functions, the user program is able to display the picture from any viewing point required by the application.

Interaction

The interaction facility of TIGS allows the application program to obtain information about events that occur when the terminal user responds to the display. Typical events are sensing a segment pick, sensing a function key, and accepting textual input from an alphanumeric keyboard.

Actions to be taken by the system when a specific event occurs can be preassigned by the application program. Specific actions can be assigned to the various functions keys. Functions such as echoing the vent at the terminal and reporting the event information are performed, depending on the action type.

Geometry Utilities

TIGS contains several geometric utilities which may be used to insure that an arc or line lies within a specific area.

Terminal Interface

Terminal control capability is provided by routines which allow the user programmer to initialize TIGS and set all terminal and mode conditions to their default values, activate the terminal alarm, clear the terminal screen, display all pictures and terminate the application in an orderly fashion.



TOTAL

The TOTAL system embodies a "network structure" philosophy. Indexes, directories, and overflow areas are eliminated. Relationships from one data set may be made on a direct basis to 2,500 other data sets within the data base. Up to 65,000 data sets may be managed on an integrated basis within one data base. It is possible to establish a virtually unlimited number of hierarchical levels or direct network structured relationships.

There is no limit to the number of data bases which may be developed, and any data set may be included in any number of different data bases.

The TOTAL system, is in effect, two systems in one:

- a. A system which provides for the initial generation of a data base module and all subsequent modifications and expansions to this data base in an English-like language. This phase is called Data Base Definition.
- b. A system which interacts with this data base, the operating system or supervisor and the application programmer. In this phase, TOTAL functions with the host language (COBOL, FORTRAN, etc.) for all types of communication with the data base. This phase is called Data Base Manipulation.

Major advances in such integrated data base problems as data integrity and security, concurrent updating, data independence and restart and recovery are facilitated by TOTAL.

The TOTAL Data Base Definition Language (DBDL) is the language used to describe and declare the data base system, and/or subsystem data bases within the overall data base. This is done in terms of names and types of data sets, data records, and data elements; and the data set and data record relationships required at this time. After definition, the data base is compiled and cataloged.

The TOTAL DBDL is an extremely high level, English-like free form coding.

In order to modify either an operational subsystem data base or the entire data base, it is only necessary to define the changes, recompile the data base, and recatalog.

After compilation and cataloging of this data base, the user is ready to begin application programming using his choice of host language processors such as COBOL, FORTRAN, Assembler, and the TOTAL Data Manipulation Language (TOTAL DML).

The TOTAL Data Manipulation Language (DML) is a language which the programmer uses to communicate between his program and the data base. TOTAL DML is not a complete language by itself. It relies on a host language to furnish a framework and to provide the procedural capabilities required to manipulate data in a primary storage.

TOTAL DML functions in conjunction with the host language, at the CALL or MACRO level. The user's application program then is a mixture of host language commands and DML functions. TOTAL DML interacts with the data base. It is the manipulative language for the data base.

All calls to and from the data base to retrieve data to add new data or data relationships, to delete data or data relationships to modify data or data relationships are defined in TOTAL DML.

TOTAL DML features comprehensive safeguards and analytical capabilities to assure proper processing. Diagnostics and messages are provided which indicate the successful execution of a function, or the status in case of an unsuccessful execution. For example, TOTAL DML will indicate that a duplicate record already exists if the user attempts to add such a duplicate record to the data base.

TOTAL functions at the element level. An element is defined as one or more of the "fields" that comprise a logical record. Upon the execution of a TOTAL DML command, one or more elements as specified by an element list are passed to or from the user host program in the stated sequence of that element list. The user is not required to do any further manipulation as to sequencing, positioning, inclusions or omission of elements. TOTAL features an extremely powerful and comprehensive repertoire of data base manipulation commands.

Since the TOTAL system is capable of manipulating data at the element level, subsequent expansions of the record for additional elements or relationships have no adverse effect on programs which use the originally defined record. Old programs do not require recompilation when new elements are added to records, unless the new elements will be used in these programs.

After the host language program has received the data, the application programmer uses the host language for whatever logical arithmetic, or manipulative processing he wishes.

The host language, then, is the language of specific data "policy" manipulation. This manipulation is application-oriented and very specific. TOTAL DML functions within the framework of the host language to provide specific information to and from the data base.

TOTAL DML is the "language of the data base." TOTAL DML function and host language statements are intimately mixed in the user application program.

TOTAL, utilizing the facilities of the DBDL and the DML, provides an integrated data base which is available to any application programmer, any of whom may be using different host language processors. TOTAL provides data elements to these programs in such fashion that new elements, new data sets and new data relationships may be added to the data base, in order to respond to changing requirements, without adversely affecting the current operational programs.

In order to effectively handle the many varieties and categories of types of data, TOTAL provides two types of files:

- Master data sets

These data sets are organized and managed according to any user-selected primary control field. These control fields are unrestricted as to length or content.

Within each record in the data set the number of "data elements" and their individual sizes are limited only by the size of the logical record itself. Master data sets are relatively stable and predictable as to content and number of records. Each record within the data set may be directly related to up to 2500 other data sets.

In normal maintenance TOTAL Master Data Sets are self-optimizing and rarely require reorganization. As a record is deleted, the space is immediately available for reuse by the system.

TOTAL Master Data Sets are reloaded only if the user wishes to reformat the basic record or if the physical space extents are exceeded. In both instances, only the affected data set is mounted on the disk storage drives.

All other data sets in the data base are unaffected and do not require reorganization, processing, or updating of any type. This is true irrespective of the number of data sets that may be directly related to the Master Data Set being reorganized.

- Variable entry data sets

Variable Entry Data Sets are data sets which may be entered by a variable number of control fields. They contain a variable (and highly volatile) number of data records. The data records may have variable formats and variable numbers of record types within each format. The data sets and data records may have unique relationships based on user-specified attributes. They reflect the business functions as they occur and are interactive and interrelated between Master Data Sets.

There may be up to 2,500 different record types within one variable entry data set. Each type may have any number of unique relationships to other data sets.

The design philosophy and criteria of the TOTAL system provides certain very handsome benefits. Certain of these criteria are also felt to be essential requirements of a data base management system by such groups as CODASYL, GUIDE, and SHARE.

- Modular and evolutionary in design and use – Significantly reduces the cost involved with change; major or trivial.
- Data independence – Application programs are sensitive only to data elements (fields), not records.
- Data nonredundancy – As a benefit of this, less DASD space is required to retain the data base. Significantly reduces the cost of maintenance.
- Data integrity and security – Vital to the data base.
- Equipment, language and environment independent – Supplies to you a major level of independence in such vital areas as operating system and computer, DASD devices, application programming language used, and your processing approach: batch or on-line.

The TOTAL system does encompass each of the aforementioned design criteria. It also offers a number of other facilities and advantages which have been discussed.



TOTAL/ATHENA

ATHENA is an on-line interactive retrieval/update language which interfaces to the TOTAL data base management system. ATHENA permits the retrieval from single or multiple TOTAL files in a single query. The combination could be either one master file and multiple variable files or multiple master files and one variable file. ATHENA also provides an optional interface to CDC CYBER Record Manager Indexed Sequential files.

The user communicates with ATHENA through statements which originate from remote interactive terminals. These statements are in a free-field format. The vocabulary consists of keywords and user-supplied values. The system possesses features that are management information-oriented, such as on-line histogram plots, single- and multi-file capabilities, and report generation.

ATHENA provides the capability to:

- Maintain and process TOTAL data files
 - Data entry from terminals or cards
 - Numeric field editing
 - Data maintenance to add, change or delete records
- Retrieve, analyze, report, and display data
 - multiple on-line users
 - Direct access and/or serial search and inquiry
 - Predefined cataloged queries
 - Text scanning searches
 - Boolean and relational operation comparison
 - Arithmetic and statistical functions
 - Encode/decode function

OPTIONAL FEATURES

- Subsetted sequential files
- Report generator
- Plot capabilities
- Indexed sequential file interface

GENERAL CHARACTERISTICS

- Easy-to-use, English-oriented command language
- Quick response
- Complete modular
- New modules easily added

- Supports COBOL 5
- Can be tuned or tailored to user specifications
- Processes multiple TOTAL files
- On-line interactive and batch versions
- ALIAS capability - a secondary name may be used to further describe a field name on the DEFINE command.
- Reverse order processing on variable records allows reading the last record first, etc.
- Last record processing on a variable file allows accessing only the last record in the chain.
- REPEAT suppresses repetitious master record output when both master and variable record data exist on output.
- Enhanced bridging capabilities.
- COBOL signed numbers feature allows specifying when TOTAL/ATHENA expects COBOL signed values in numeric field.



TOTAL EXTENDED

TOTAL Extended provides exactly the same features as TOTAL Universal plus some significant extensions. The primary extension is that multiple user tasks being processed in a transaction mode can concurrently access and update the same data base. All of the transaction tasks make input/output requests to the same copy of the TOTAL object code running under the NOS 1 TAF transaction subsystem.

TOTAL Extended controls concurrent update attempts made by multiple transaction tasks; it provides lockout at the record level. When one task performs a read with the intent to update, that record is locked out from other tasks which may also want to read with the intent to update until the first update is completed or the record released. TOTAL Extended contains provisions to solve the "deadly embrace" situation.

A single copy of TOTAL Extended can process multiple data bases under TAF, as well as control the concurrent update attempts by multiple transaction tasks against the same data base. When TAF is initialized, the data base descriptions of the data bases to be processed by transaction tasks are loaded with TOTAL Extended under TAF. These data bases are, thus, opened and attached to TAF; transaction tasks then SINON to TOTAL Extended without each having to attach the data base.

Both TOTAL Extended and TOTAL Universal can process the same data base. It is normally recommended that a user start in batch mode with TOTAL Universal to get his data organized and under control. After a data base is organized and loaded, it can then be processed by TOTAL Extended in a transaction mode.

Extensions of the TOTAL facilities are contained within the Data Base Definition phase of the system (DBGEN), as well as in the Data Manipulation phase (DML) and the DATBAS interface routines. These extensions include:

- A single copy of TOTAL Extended, residing in the TAF control point, can be shared by any number of user tasks running under TAF control. The number of user tasks is subject to the number of TAF subcontrol points in use (32 subcontrol points maximum) and the memory size available for queues.
- Any number of TAF tasks may access and update a common file or set of files at the same time. The default limit is ten tasks; all others are placed in a rollout queue. TOTAL Extended provides the necessary logic to ensure the integrity of the data base in this environment.
- All data may be defined within one Data Base Description Module (DBMOD), or there may be concurrent use of many separate DBMODs.
- Extended status codes are provided to the tasks due to the multi-tasking environment.
- Compatibility is maintained in all areas with TOTAL Universal. Data base files can be created and processed by TOTAL Extended without modification, and vice versa. Programs developed under TOTAL Universal can be run under TOTAL Extended with no changes where data base processing is concerned.

- **TOTAL Extended, DBMODs, and all user tasks run under the control of the TAF Executive.**
- **TOTAL Extended may operate concurrently with one or more batch jobs using TOTAL Universal and running at separate control points, with the protection that no two separate control points may modify the same file concurrently. However, any file may be used in read-only mode by any number of concurrently operating jobs.**



TOTAL UNIVERSAL

Control Data proposes the TOTAL Universal data base management system to fulfill the requirements for a comprehensive management information system.

The TOTAL Universal system embodies a network data structure philosophy. There is no hierarchial index overhead. Indexes, directories, and overflow areas are eliminated. Relationships from one data set may be made on a direct basis to 2,500 other data sets within the data base. Up to 65,000 data sets may be managed on an integrated basis within one data base. It is possible to establish a virtually unlimited number of hierarchial levels or direct network structured relationships.

There is no limit to the number of data bases which may be developed, and any data set may be included in any number of different data bases. The TOTAL Universal system features the highest possible performance in the areas of disk and core utilization and in retrieval speeds. Further, the system is self-optimizing, which prevents the usual performance degradation.

The TOTAL Universal system is, in effect, two systems in one.

- A system which provides for the initial generation of a data base module and all subsequent modifications and expansions to this data base in an English-like language. This phase is called Data Base Definition.
- A system which interacts with this data base, the operating system or supervisor and the application programmer. In this phase, TOTAL Universal functions with the host language (COBOL, FORTRAN, etc.) for all types of communication with the data base. This phase is called Data Base Manipulation. At all times in both phases, you, the user, are in complete control with TOTAL Universal.

Major advances in such integrated data base problems as data integrity and security, concurrent updating, data independence and restart and recovery are facilitated by TOTAL Universal.

DATA BASE DEFINITION (DBDL)

TOTAL Universal DBDL is the language used to describe and declare the data bases system and/or sub-system data bases within the overall data base. This is done in terms of names and types of data sets, data records, and data elements; and the data set and data record relationships required at this time. After definition, the data base is compiled and cataloged.

The TOTAL Universal DBDL is a high level, English-like, free form language. A user can be fully trained in a matter of hours and then be capable of defining and developing data bases of great complexity.

DATA MANIPULATION LANGUAGE (DML)

TOTAL Universal DML is a language which the programmer uses to communicate between the program and the data base. TOTAL Universal DML is not a complete language by itself. It relies on a host language to furnish a framework and to provide the procedural capabilities required to manipulate data in primary storage.

TOTAL Universal DML functions in conjunction with the host language, such as COBOL, FORTRAN, Assembler, etc., at the CALL or MACRO level. The user's application program then is a mixture of host language commands and DML functions. TOTAL Universal DML interacts with the data base. It is the manipulative language for the data base.

All calls to and from the data base to retrieve data, to add new data or data relationships, to delete data or data relationships, to modify data or data relationships are defined in TOTAL Universal DML.

TOTAL Universal DML features comprehensive safeguards and analytical capabilities to assure proper processing. Diagnostics and messages are provided which indicate the successful execution of a function, or the status in case of an unsuccessful execution. For example, TOTAL Universal DML will indicate that a duplicate record already exists if the user attempts to add such a duplicate record to the data base.

TOTAL Universal functions at the element level. An element is defined as one or more of the fields that comprise a logical record. Upon the execution of a TOTAL Universal DML command, one or more elements as specified by an element list are passed to or from the user host program in the stated sequence of that element list. The user is not required to do any further manipulation as to sequencing, positioning, inclusions or omission of elements. TOTAL Universal features an extremely powerful and comprehensive repertoire of data base manipulation commands.

Since the TOTAL Universal system is capable of manipulating data at the element level, subsequent expansions of the record for additional elements or relationships have no adverse effect on programs which use the originally defined record. Old programs do not require recompilation when new elements are added to records. After the host language program has received the data, the application programmer uses the host language for whatever logical arithmetic or manipulative processing desired. The host language, then, is the language of specific data policy manipulation. This manipulation is application oriented and very specific.

TYPES OF DATA SETS

In order to effectively handle the many varieties and categories of types of data, TOTAL Universal provides two types of files or data sets: master and variable entry.

Master Data Sets

These data sets are organized and managed according to any user-selected primary control field. These control fields are unrestricted as to length or content.

Within each record in the data set, the number of data elements and their individual sizes are limited only by the size of the logical record itself. Master data sets are relatively stable and predictable as to record content and number of records. They may be thought of as existing because the company is in business. Each record within the data set may be directly related to up to 2,500 other data sets. In normal maintenance TOTAL Universal master data sets are self-optimizing and never require reorganization. As a record is deleted the space is immediately available for reuse by the system. TOTAL Universal master data sets are reloaded only if the user wishes to reformat the basic record or if the preallocated physical space extents are exceeded. In both instances, only the affected data set is mounted on the disk storage drives.

All other data sets in the data base are unaffected and do not require reorganization, processing or updating of any type. This is true irrespective of the number of data sets that may be directly related to the master data set being reorganized.

Data retrieval performance is extremely high and does not degrade as the number of relationships to other data sets increase.

Variable Entry Data Sets

Variable entry data sets are data sets which may be entered by a variable number of control fields. Further, they contain a variable (and highly volatile) number of data records. The data records may have variable formats and variable numbers of record types within each format. The data sets and data records may have unique relationships based on user-specified attributes.

It can be said that variable entry data sets exist because the company is doing business. They reflect the business functions as they occur and are interactive and interrelated between master data sets.

There may be up to 2,500 different record types within one variable entry data set. Each type may have any number of unique relationships to other data sets. There are no restrictions to the number, size, content, or relationships or records or data elements other than those imposed by hardware and operating systems considerations.

Just as in master data sets, variable entry data sets do not require reorganization due to performance degradation. The system is self-organizing and always functions at peak performance. If the read arm is already on the proper cylinder, the seek will not be performed. If the record is already in core storage, the disk read will not be performed. There are no reserved cylinders for indexes or directories. All disk space is available to the user for storing his prime data.

DATA INDEPENDENCE

In order to achieve a level of modularity and an evolutionary approach toward growth and change, the data independence feature is of vital importance. As discussed earlier, it is that ability to remove any and all physical dependency by an application program on data. By communicating at the data element level, this independence is achieved.

TOTAL UNIVERSAL I/O POOL SHARING FEATURE

Optimum performance and efficiency is one of the TOTAL Universal system's strongest advantages or benefits to the user. Traditionally these two vital attributes of any system tend to be mutually exclusive of each other. Often the move to data base means significant increases to computer requirements. This is not so with TOTAL Universal.

The TOTAL Universal system helps bridge this gap with the I/O POOL SHARING feature. An example that is common in describing this topic is that of having large buffers for each file so that higher density recording of data and a better performance curve is achieved.

Under the traditional approach, this would mean large core requirements for such an approach. What is accomplished here is a high level of performance, but efficient use of core is the trade-off.

However, by invoking TOTAL Universal's concept of POOL SHARING, the example depicts the same physical file formats but by utilizing more than one file for the same buffer, a significant gain in efficient use of core is seen while maintaining the same disk utilization, and in most instances with the little trade-off in through-put.

DATA BASE INTEGRITY AND SECURITY

Due to the widely diversified environments and specialized requirements of different TOTAL Universal users, no simple generalized solution is possible for the problem of data base integrity and security. Therefore, TOTAL Universal provides a set of facilities to be used as the need arises.

Task Access Mode Facility

When a task signs on to TOTAL Universal, it must declare the type of processing it intends to perform; read-only, update, or recovery. TOTAL Universal will monitor the task's activity to ensure that it does not go beyond the bounds of its stated purpose. Any attempt to do so will be stopped by TOTAL Universal, and an error status will be returned to the program.

Read-only tasks may perform any logical retrieval function that does not modify the contents of the data base. The task may not perform any update function which changes the contents of the data base (i.e., add, delete, or write) or any of the special recovery functions.

Update tasks may perform any retrieval function or any logical update function. The task may not perform the special recovery functions.

Recovery tasks may perform any function including the special recovery functions.

LOGGING FACILITIES

Logging is an optional facility and will operate only if it is included in the system.

The logging facility must be activated in the SINON command of the user program by specifying a particular choice of logging options (LOGOPTS).

The logging options which may be coded into the SINON command are as follows:

- LG – Before image logging (logging a record before it is updated)
- NL – No logging

Logging options and logging functions provide the user with various methods of recording particular information at particular times. A discussion of the logging options and functions follows.

Before Image Logging

This option is used to automatically log records as they exist before being updated in the data base. Using this option, a data base may be restored to some previous form. That is, the contents of the data base after restoration will resemble the contents of that data base at some past point in time when an image of the contents was logged. This restoration of a data base to some prior status is termed "backout".

Log Mark

This function is used to log any data specifically requested by the user, including transaction data. This function will allow the user to reprocess transactions from the last backout point to the point of failure, without resubmitting the transactions.

Log Quiet

This function is used to force all pending updated input/output buffers to the data base. An indication that a "quiet point" has been established will be placed on the log data set at the time the function is executed. The quiet log record will also contain any data requested by the user. This synchronization of logical and physical processing will create a "restart point" for the purpose of data base backout. Forcing all pending buffers to be written to the data base ensures that the log data set will be a complete image of the data base at that point in time.

Sign-On and Sign-Off Logging

This facility is used to log the SINON and SINOF commands after they are executed. In this way, TOTAL Universal can determine which tasks are active and inactive if restarting is necessary.

Special Considerations

If the LOGOPTS parameter is invalid or missing, the default will be no logging.

The SINON and SINOF commands themselves will be logged if the option codes indicate that any logging is to be performed.

The use of the log data set for recovery is the user's responsibility. However, a utility program (DBRCV) is provided to more easily implement a recovery system.

The log data set must be a multivolume data set, containing standard labels and variable length, unblocked records. The file name specified must be TLOG.

RECOVERY

DBRCV is an executable program that operates with TOTAL Universal. It is used to recover "before images" for TOTAL Universal data sets. The TOTAL Universal log file is the input to the DBRCV program. The program uses control cards to specify the mode of operation, logging device, data base descriptor module, and options required. The program reads and validates the control cards, prints a run description and a listing of any options in effect on the system printer, and processes the log file from the beginning to a point specified on the control cards.

A separate execution of DBRCV is required for each log file to be processed. The standard label checking provided by the operating system is the only means of validating the sequence of multiple volumes of a log file.



TRS (TAPE RESERVATION SYSTEM)

The Tape Reservation System (TRS) operates under the control of the NOS operating system on the Control Data 6000, CYBER 70, or CYBER 170 series computer systems. The purpose of the system is to allow an interactive or batch user to easily access and manage magnetic tape files.

The present system consists of two absolute overlay programs, one to access and update the TRS data base files, (i.e., CALLTRS), while the other module (i.e., AUDITRS) allows the user to report on the status of his tape library under various sort controls.

TRS usage has the following advantages over the classical tape usage processing:

- 1) TRS standardizes user tape processing by enforcing usage of ANSI-labeled tapes.
- 2) TRS provides a high degree of tape files security through the following methods:
 - a. Comparison of labels for read and write processing.
 - b. Usage of the File Access parameter (FA) to allow only users of the TRS system to access TRS tapes.
 - c. Usage of both write and access passwords to use tapes and update the data base.
- 3) TRS frees the user from using/recalling basic tape parameters. In writing the tape, TRS defaults to reasonable parameters that may be overridden by the user. At read time, TRS will request the tape with the parameters used for the last write operation.
- 4) TRS utilizes symbolic tape "names", allowing the user to avoid the actual Volume Serial Number (VSN). This is particularly useful when multiple users access a master tape that is updated by another user. The VSN may change but to the user the symbolic name remains the same.



UNIPILOT (UNIVERSAL PLOTTING)

UNIPILOT (Universal Plotting) is a general purpose device independent plotting system which produces graphic output for a variety of graphics devices. These devices include pen plotters, display tubes, refresh scopes, and microfilm plotters. This new software provides a link between an application program that produces CalComp compatible output and any supported graphics device.

UNIPILOT uses CalComp call compatible neutral plot routines to write a file which can then be read by a postprocessor to produce the output. UNIPILOT is designed to be used by all application programs that produce graphic output by making calls to the CalComp basic plotting software, except those programs which require real-time or interactive graphic displays.

SYSTEM DESIGN

The UNIPILOT system is composed of two software components: a set of CDC neutral plot routines and a stand-alone postprocessor. These components communicate through standard format files known as neutral plot files. The neutral plot routines are called by application programs in the same manner as standard CalComp routines. Instead of producing a file that directly drives a CalComp plotter, the routines produce a neutral plot file. The UNIPILOT postprocessor can then be instructed to read the neutral plot file, modify the graphics data in various ways, and produce a file that will be displayed or plotted on any user-selected device.

PLOT REVIEW

The most unique feature of UNIPILOT is the Plot Preview facility. It provides the capability to view plots on a CRT screen before committing the time and resources to generate a hardcopy output. In short, a user will benefit from obtaining a plot first from a terminal such as a CDC 777 CYBERGRAPHICS or a Tektronix 4000 series and avoid rerunning a large application which might generate many unnecessary pen plot displays as was the case before UNIPILOT. It provides a single set of plotting routines to interface with a variety of applications and devices, a standard plot file format and a standard set of user options to control plotting regardless of the device selected.

It permits the user to preview the plot on the display, to delay selection of the plotting device until plot time, or to have a number of plot devices for the same plot. Preprocessing can be performed on one system (e.g., the Network Operating System, NOS), and postprocessing on another system (NOS/BE). The UNIPILOT utility is designed for future expansion to include new devices, systems, and/or applications.

Plots from the neutral plot file may be previewed on the Tektronix 4010 or 4014 low-cost graphics terminal or on the 777 terminal. After previewing, the user may then select plots for permanent hardcopy on any of the following currently supported devices:

- Houston Instrument DP1 (on-line) and DP7 (off-line) plotters
- CalComp 563, 736, 936 and 1136 (off-line) plotters
- Tektronix 401X-series (on-line display tubes)
- Zeta 230, 1240 and 3640 (on-line) plotters
- SC4020 computer output microfilm

UNIPLLOT provides the user with great flexibility in the selection of a device for displaying graphic output. Even after running an application, the user has complete control over:

- Drawing scale
- Drawing layout on a display surface
- Mirroring of the drawing
- Windowing (displaying a portion of a drawing)

The neutral plot routines include a wide variety of routines which perform such functions as:

- Placing vectors, character strings, arcs, and curves in the neutral plot file
- Structuring the plot file hierarchy into drawings and display groups
- Controlling the placement of drawings on a display surface
- Permitting mirroring, offsetting, and scaling of display elements before they are placed in the plot file

The postprocessor is a stand-alone program which can be instructed to:

- Read a neutral plot file
- Modify the graphic data according to the postprocessor directives
- Create a file that produces the plot on a display device

WRITING GRAPHIC DATA

UNIPLLOT contains many subroutines used to write graphic data and to simplify plotting graphs. These subroutines allow the user to:

- Employ such vectors as solid-line, invisible, dashed, or dotted line
- Write character strings and a variety of centered symbols
- Plot floating point numbers as a string of digits, punctuated
- Draw smooth curves through a set of points
- Draw arcs
- Simplify the task of plotting graphs by inspecting an array of values and computing a scale factor that will cause the values to fall within a specified dimension

- Draw an annotated axis which indicates the unscaled unoffset values of numbers in an array once they have been scaled, offset and plotted beside the axis.
- Perform the task of offsetting, scaling, and plotting points given in a pair of arrays, one for x-coordinates and the other for y-coordinates.
- Modify graphic data in many ways: reduced or enlarged, moved, or mirrored.

Optional plot routine extensions are also available:

- Error code return
- Dashed and dotted lines
- Invoked use of scale and offset
- Picture rotation
- Curve generation (arcs and cubic interpolation)
- Plot size and orientation control
- Drawing and figure definition
- Mirroring
- Optional metric units



UNISTRUC

UNISTRUC (Unified Structural Design System) is an interactive graphics interface which aids in structural design and analysis of engineering problems. Using UNISTRUC, a user can generate finite element models and analyze the output of the programs, reducing the total problem analysis cost and time by as much as 90 percent. UNISTRUC complements preparation of input data to finite element application programs, and allows a rapid visual review of the model geometry before processing. The system supports both refresh and storage-type graphic devices at high communication rates, with both on- and off-line hard copy availability. Any finite element application program can be interfaced to UNISTRUC, including customer proprietary codes.

With UNISTRUC, a user can generate data for large numbers of nodes and elements in a fraction of the time required to manually perform these operations. By generating a visual display of the structural component, rotating it, using hidden line, perspective, or sectioning features, and viewing it from different angles, model errors can be rapidly identified and corrected.

A user converses with UNISTRUC in specially designated areas on the screen of the interactive graphics terminal. During the dialog, UNISTRUC speaks the engineer's language in requiring simple, fixed length input, keyword input or menu selection. The user selects features, responds to questions or inputs keywords containing commands and/or data.

Results can be displayed on the interactive graphics screen, giving the user quick access to the data needed to interpret the solution. The decision to modify the model and resubmit it to the finite element processor can be made without delay. Changes, modifications and input errors may be corrected via interactive sessions. The time consumed by resubmitting batch corrections is eliminated.

FEATURES

Model Generation

Model generation features include:

- Modeling ease by natural subdivision of the structure
- Interactive surface definition and drawing subsystem
- Three coordinate systems (cartesian/rectangular, cylindrical and spherical)
- User input of surface definition from external sources
- Macros for geometry creation/storage/retrieval/manipulation
- Automatic three-dimensional curve generation (spatial splines)
- Automatic point generation along lines and curves (equal and weighted spacing)

- Automatic two, three, and four-node element mesh generators
- Manual mesh generator/editor
- Neutral element library containing truss and beam elements, translational and torsional linear spring elements, triangular and quadrilateral elements with membrane, plate bending-membrane and composite
- Model geometry verification and correction; mesh boundaries and holes, normals and planarity of quadrilateral elements
- Material property generator/editor with display and material orientation angle definition/display
- Physical property generator/editor and display and stress reference angle definition/display
- Node number optimizer with quick and full optimization options (**Giggs-Polle-Stockmeyer method**)
- Node and element loading (concentrated, beam, pressure and gravitational)
- Nodal restraint and enforced displacement definitions with visual location verification

Analysis Processing

The Control Data-supplied interfaces to finite element analysis programs allow:

- Input/output translation interface via the neutral file scheme (format available)
- On-line editing of input data
- Interfaces with all finite element analysis programs

Analysis Review

In addition to standard finite element code output, UNISTRUC offers the following functions:

- Display of displacement locations for critical limit selections
- Deformation display/plots of structural mesh or boundaries
- Stress distribution plot options, and
- Selective force and stress summary listing options

Display and Utility

UNISTRUC's usefulness in model creation and visual review is enhanced by the following graphics functions:

- Windowing, scrolling, zooming and scissoring of view
- Surface display with optional point and/or line numbering
- Three-dimensional transformations such as rotation, perspective and hidden line removal
- Selective sectioning of model
- Mesh display with optional node and/or element numbers with surface display superimposed
- Advanced algorithms for rapid mesh display

System Design

Overall system features include:

- Designed for use with low-cost graphics terminals
- Integrated model generation and post-analysis system under control of an application executive
- Exclusive capabilities for high-speed (up to 4800 bps) interactive synchronous communications (minimizing display times)
- Analysis of very large problems (as many as 20,000 nodes and elements)
- Free format menu/command type dialog
- Command listing capability
- Structural data base (SDB) integrity
 - Recovery system to preserve SDB in case of abnormal termination
 - Internal SDB file maintenance (CATALOG/ATTACH/EXTEND) including checkpoint capability
 - Error recoverable dialog
 - External diagnostic system
 - Full SDB file security
- COPYSDB utility to generate sequential files for tape storage of SDB off line
- Variety of plotter output options and on-line hard copy

USER BENEFITS

The overall objective of UNISTRUC is to facilitate the analysis process by providing the user with a single interface to finite element analysis programs. Among the benefits of having one system for pre- and post-processing structures problems are:

- Individual engineering productivity is enhanced by reducing the time and cost required to perform the major activity (model layout, generation, error correction and data interpretation)
- The user is required to learn only one set of easy-to-use procedures to interface with analysis programs
- A variety of solution techniques and packages can be sampled to determine which is most cost effective or accurate
- Results obtained from different methods can be compared for verification.



UPDATE

The UPDATE program enables the user to organize a collection of programs into a file and operate on them with updating facilities using decks from the central or a remote site, as well as keyboard input from terminals. The primary feature in UPDATE that differentiates it from other symbolic updating programs is the invariance of card identification. A card receives a sequence identification that never changes and is used as a reference when making modifications. Unless there are overlapping corrections, symbolic corrections to the program library can be keyed to any version of the file.

The UPDATE (control) card contains information which directs UPDATE to specific files and modifies program operation. UPDATE control and data cards (processed by UPDATE from the input file unless otherwise specified) are grouped into the following categories.

File Manipulation cards:

- REWIND
- SKIP
- READ
- LABEL

Creation directive control cards:

- DECK
- COMDECK
- END

Correction directive control cards:

- IDENT
- PURGE
- DELETE
- RESTORE
- INSERT
- YANK
- /
- ADDFILE

Assembly directive control card:

- COMPILE

Output directive control cards:

- DECK
- COMDECK

- WEOR
- CALL

Text cards:

Any card not included above.

The file manipulation cards allow text and other control cards to be read from files other than the main input file. Other files could be COMMON files, etc. Creation directive control cards are used to create new program libraries. Correction control cards are used for correcting and updating program libraries. The assembly card, COMPILE, causes specified decks (files) to be assembled. Output cards control the compile file output.



XEDIT

XEDIT is an extended interactive text editor available to users of the CDC CYBER 170 computer system under the NOS operating system. While XEDIT performs the same editing functions as most conventional text editors, the following features and benefits illustrate why XEDIT is considered to be an enhanced editing system:

<u>Feature</u>	<u>Benefit</u>
Simple command formats	XEDIT requires fewer command delimiters than most conventional editors. As a result, users have fewer rules to remember and are less likely to commit syntax errors.
Multiple commands in a single line	XEDIT users can submit up to 39 commands in a single line. Consequently, XEDIT users have greater flexibility and can speed up their interactive editing sessions.
Verification of user entries	XEDIT automatically verifies that key user commands have been executed by listing those portions of the file that the user modified. Accordingly, XEDIT users save time since otherwise they would have to issue separate PRINT commands if they wanted to obtain this result.
Internal interrupt processing	Under XEDIT, users stay under the control of the editor even though they interrupt processing (e.g., interrupting printer listings). They are not transferred back to operating system control.
Editing on the basis of line numbers	While XEDIT can edit on the basis of searching for key alphanumeric phrases, it can also unambiguously search on the basis of just a line number.
Availability of permanent file commands	XEDIT users can issue certain permanent file commands while remaining under control of the editor. This feature saves time since XEDIT users do not have to terminate editor control, issue a NOS file command, and then re-call the editor to perform more editing.
Easier line modification	XEDIT users can issue a MODIFY command to alter a line by entering only a minimal number of characters.
Editing multi-record files	By employing XEDIT, users can edit multi-record files. This is particularly significant since the absence of this feature would mean that only the first record in the multi-record file would be retained (the remaining records would be lost).

SECTION 3

COMPUTER MAINFRAMES



CYBER 18-5M BATCH TERMINAL CONTROLLER

The CDC CYBER 18-5M Batch Terminal Controller is a processor-based complement of elements featuring flexible configuration plus a variety of terminal emulation programs. The CDC CYBER 18-5M, when configured with input and output peripheral devices, operates as an effective remote batch terminal. The batch terminal controller is upward expandable by field-installable options which include memory expansion to 65K bytes, and a maximum of three magnetic tape units, supported by terminal emulation programs. Additionally, a memory based operating system is available for local processing purposes. The CDC CYBER 18-5M Terminal Controller allows the input/output capabilities of a central computer system to be extended to locations far removed from the central site. This distribution of computing power provides an efficient computer network which meets the demands of scientific and commercial data processing. Thus, the central site resources are only employed as required to satisfy the particular data processing needs.

The CDC CYBER 18-5M Batch Terminal Controller consists of the following components:

- **Controller** – A microprogrammable processor including 16K bytes of main memory and input/output channel, providing capacity for control of other elements in the terminal configuration.
- **Keyboard/CRT Display** – Functions as the terminal operator's control and information interface.
- **Communication Adapter** – With RS232-C and CCITT V.24 interface compatibility, this adapter provides for attachment to a synchronous modem (not included) and associated circuit. Communications take place in half-duplex, two-wire or four-wire synchronous mode, at transmission rates which may range from 1200 to 9600 bits per second.
- **Controlware Programs** (specified at time of order) – Provides specific terminal control and emulation of various types of terminals. One of which is integral to the CDC CYBER 18-5M controller package. These controlware programs also provide a common set of operator interface, error monitoring, error recovery and diagnostic routines.
- **Peripheral Adapter** – Interface/controller logic to provide data input/output and control for the terminal card reader and line printer.

To complement the CDC CYBER 18-5M and complete the configuration for operator's as a batch terminal the following peripherals are available:

- **Card Readers** - Card readers are self-contained desktop units with operator's control/indicator panel, 1000-card hopper/stacker output capacity and 80-column card media. Reflective fiber optics read station, with light/dark checks and a mechanical edge-pick feed mechanism. Card read speeds are 300 or 600 cards per minute.
- **Line Printers** - A 300 or 600-line-per-minute printer, equipped with a sound-suppressed cabinet. The printer allows vertical spacing selection of either six or eight lines per inch, and up to 136 or 132 print positions per line respectively. Six-part forms can be handled, ranging in width from 4 to 20 5/8 inches (10.16 to 52.4 cm) and up to 22 inches (55.88 cm) in length. These printers are equipped with a standard 64-character ASCII symbol set.

The 300 lpm printer features a drum printing technique; the 600 lpm version utilizes a horizontal band-print mechanism.

- **Hardware Options** – Up to three magnetic tape transports can be added for alternate input/output versatility and are supported by the emulation programs.

Available as 7- or 9-track units, these transports have 800 bpi recording densities, 25 ips tape speeds, and use NRZI recording technique. Single capstan drive provides smooth starts and stops; velocity and tape position are digitally controlled.

The associated magnetic tape controller handles 800 bpi, NRZI mode, 7- or 9-track tape units.

Memory also can be expanded to 65K bytes as a user option.

- **Cables** – All necessary cables used to connect the terminal controller to the display, card reader and line printer are included. Cable length allows some flexibility in physical placement of devices. In addition, a standard 20-foot (6.1 m) cable is provided for controller-to-modem connection.

CONTROLWARE PROGRAMS

Each controlware program performs individual emulation of a particular terminal and is loaded into the controller processor, via the card reader, as required by the user. One controlware program, specified when ordering, is included as part of the basic CDC CYBER 18-5M configuration.

Universal Controlware Features

Many operating features of the CDC terminal controlware are available, regardless of the type of terminal being emulated.

- **Punch Code Alternatives** (operator selectable)
 - Hollerith 026
 - Hollerith 029
- Continuous display of selected operating parameters and basic operator instructions.
- **Automatic Error Notification** – Transmission line and terminal device error condition are presented on display screen.
- **Local Operation** – In addition to on-line operations, the terminal controlware also performs useful local operations. These include the display-to-print operation, where data appearing on the display screen can be printed on the line printer; and card-to-print operation, which permits punch-coded information of a card deck to be listed on the line printer.

The following local media transfer operations can be performed by those systems using the magnetic tape options:

- **Tape-to-print processing** takes magnetic tape data and prints it in a specified format on the line printer.
- **Card-to-tape processing** takes punched-card data and stores this information on magnetic tape for future use. For example, transmission to the central computer system or creation of hard copy by the line printer.

- Tape-to-tape processing involves reading data from one magnetic tape unit and transferring this information to another tape unit.
- Batch terminal tape motion commands allow the operator to control motions of magnetic tape from the keyboard.

Local operations may be used at any time, although on-line functions of the terminal have priority. Additional standard features include:

- Audible/visible alert and alarms
- Local diagnostics simplify terminal maintenance and reduce downtime:
 - Terminal controller memory dump to printer or display
 - Communications line trace function enables monitoring of transmit line activity on display or printer
- Failure of one peripheral unit in the terminal configuration does not degrade performance of any others

CDC 200 U.T. EMULATION CONTROLWARE PROGRAM (1890-1)

A unique feature of this program permits terminal compatibility with CDC central computer systems and associated operating systems.

Computer System	Operating System
CDC CYBER 70 Model 76 with CDC 6000 Series or CDC CYBER 70 Series, or CDC CYBER 170 Series station	SCOPE 2 with NOS/BE or SCOPE 3
CDC 6400, 6500 and 6600 CDC CYBER 70 Models 71, 72, 73 and 74, CDC CYBER 170 Models 171, 172, 173, 174 and 175	NOS

- The keyboard/CRT display also may be used in a standalone mode, operating interactively with INTERCOM under SCOPE or NOS/BE Operating Systems
- Transmission code alternatives (operator selectable):
 - External binary-coded decimal
 - Internal binary-coded decimal
 - American Standard Code for Information Interchange (ASCII)
- Operator Message Mode Alternatives (operator-selectable method of transmission):
 - Line-by-line mode
 - Block mode
- Multidrop terminal operation on a dedicated communications line
- Half-duplex CDC Mode 4A communications protocol
- Automatic retransmission

- 1040-character block size
- Decompression of received data blocks increases transmission efficiency of data destined for output to printer

IBM 2780 EMULATION CONTROLWARE PROGRAM (1890-2)

- Plug compatible for user of IBM 2780 Model 1
- Half-duplex IBM binary synchronous communication protocol as standard
- Transmission codes, ASCII or EBCDIC (normal or transparent)
- Multiple record transmission
- Variable record length for card reader by encoding end-of-media character
- Transmission checking (cyclic redundancy character for EBCDIC, vertical redundancy/horizontal redundancy checks for ASCII)
- Automatic retransmission of any message received which contains transmission errors
- Terminal identification transmitted when used with switched, dial-up circuit
- Extended inquiry retry transmission
- Multidrop operation on a dedicated line
- Contention of line timeouts
- Automatic answer and disconnect
- Keyboard display simulates switches, keys, and status indicator of IBM 2780 operator panel
- Horizontal forms (tabbing) control
- Vertical forms control, including operator selectable use of ASA forms control characters
- The batch terminal interfaces with those systems which have communication hardware and software to support IBM 2780 binary-synchronous communications protocol, such as:
 - IBM 360/370 Operating Systems (DOS/POWER/RJE, OS or DOS/BTAM OS/MVT)
 - MFT/HASPII/RJE OS/RJE
 - IBM 2780 Data Transmission Terminal (Models 1, 2, 3 and 4)

IBM 3780 EMULATION CONTROLWARE PROGRAM (1890-3)

Because the IBM 3780 is a more sophisticated terminal than the IBM 2780, the 1890-3 Option has a number of additional features, over and above those of the 1890-2 (IBM 2780 emulation) option:

- 512-character buffer size
- Automatic end-of-media indication for reading partially punched cards or for printing short lines (optional on 2780)
- Unlimited blocking of records into buffer

- **Space compression for transmitting imbedded spaces more efficiently**
- **Test mode which allows transmission of all special characters normally reserved; useful in a test or restart condition**

OPTIONAL SOFTWARE

The CDC CYBER 18-5M Real-Time Operating System (RTOS) is a real-time multiprogramming operating system for CDC CYBER 18-5M and resides in memory with no need for a mass storage device. Systems are tailored to the user's hardware and job requirements.

RTOS is a compatible subset of MSOS. Up to 16 program-priority levels are provided, and input/output requests are also processed on a priority basis.



CYBER 18-10M COMPUTER SYSTEM

The CDC CYBER 18-10M is a general-purpose, 16-bit processor. Execution of macro programs stored in MOS main memory is controlled by micro-level programs stored in micro memory. ROM micro memory is provided for execution of the basic CDC 1700 instruction set and the additional enhanced instructions, including character and field manipulation indexing, micro memory referencing, autodata transfer, and main memory paging control. Arithmetic is one's complement, signed, fixed-point hardware and/subtract/multiply/divide.

FEATURES

- General-purpose digital processor, using microprogrammed architecture
- Accommodate 32K through 128K bytes macro main memory
- Main memory effective read/write cycle time of 750 nanoseconds
- Powerful instruction repertoire
- Eight addressing modes for accessing main memory
- Main memory parity detection, optional error correction
- Direct memory access
- Integral flexible disk drive for diagnostic loading (optional use as system peripheral).
- Automatic program load (deadstart) facility for loader type peripherals
- Integral real-time clock
- Modular design, CPU and controllers on 11 by 14-inch PC boards for ease of handling
- High reliability and ease of maintenance through state-of-the-art technology and advanced diagnostic capability
- RS232-C compatible input/output interface for console display or TTY
- Priority-oriented interrupt system with sixteen levels of interrupts
- Optional breakpoint controller
- Wide range of peripherals supported
- Cabinet, operator's panel, power distribution, and power supplies included

ADDRESSING MODES

The following eight addressing modes are provided for maximum flexibility:

- Absolute
- Indirect
- Relative
- Relative Indirect
- Constant
- Storage
- Storage Indirect
- Field

INSTRUCTION REPERTOIRE

CDC CYBER 18-10M incorporates the basic CDC 1700 instruction set and additional enhanced instructions not previously available. This repertoire includes one, two and three word (two 8-bit bytes per word) instructions and is flexible for increased programming efficiency. Instruction groups include the following:

- Transfer
- Logical
- Stop
- Shift
- Interrupt
- Generate Parity
- Character/Field Manipulation
- Execute Micro Code Sequence
- Arithmetic
- Jump
- Decision
- Input/Output
- Memory Paging Control

Some instructions are immediate (literal), resulting in a saving of operand storage and execution time. Multi-word instructions, such as indirect addressing, are a means of addressing locations which cannot be accessed directly.

REGISTERS

CDC CYBER 18-10M provides 15 registers, including four general-purpose registers to support the enhanced instruction set, and four special-purpose registers used exclusively for machine control.

REGISTER FUNCTIONS

A(16 bit) — Principal arithmetic register; data register during input/output operations

Q(16 bit)	—	Auxiliary arithmetic register; peripheral address register during input/output operations
P(15 or 16 bits)	—	Program address register
X(16 bits)	—	Storage data register
Y(16 bits)	—	Address register; hold temporary results during address computation
M(16 bits)	—	Interrupt mask register
B(16 bits)	—	Breakpoint address register
I(16 bits)	—	Indexing, accumulation, and loop control register
1,2,3,4 (16 bits)	—	Indexing, accumulation, and loop control registers
LB, UB (16 bits)	—	Lower and upper bound registers for unprotected area
MFP (64 x 9 bits)	—	Memory page file

PROGRAM PROTECTION

CDC CYBER 18-10M offers two modes of protection from damage which may be caused by programs accessing memory outside their own region. Traditional word level protection of the CDC 1700 series allows individual words to be declared protected by setting a bit in memory associated with that word. A second means of protection uses upper and lower bounds to define an unprotected region. This has the same effect as word protection, except that a large unprotected area can be defined more quickly.

MAIN MEMORY SYSTEM

CDC CYBER 18-10M features high-speed dynamic MOS LSI storage elements. Each word in memory consists of two data bytes, one protect, and one parity bit. Memory is organized as a single bank with two ports — CPU and DMA.

Storage capacity is expandable from 32K to 128K bytes by the simple insertion of individual PC boards. CDC CYBER 18-10M includes no main memory; however, up to two card slots are provided to accept any mixture of 32K and 64K byte MOS memory array boards (Options 1882-16 and 1882-32). The effective memory cycle time at either port is 750 nanoseconds; however, the memory processes simultaneous requests from both ports with an average effective cycle time of 600 nanoseconds.

INTERRUPT SYSTEM

CDC CYBER 18-10M firmware emulates 16 levels of vectored interrupt. This system consists of 15 levels of external interrupt and one internal interrupt.

Certain conditions such as an illegal instruction, a memory parity error, or a power failure generate an internal interrupt. External interrupts occur when a computer peripheral device has finished an input/output operation or requires attention. The interrupt system will handle up to 16 interrupts in a flexible and efficient manner.

REAL-TIME CLOCK

The real-time clock is an integral part of the CDC CYBER 18-10M, and provides a macro-level interrupt at a programmable interval. The real-time clock appears as a CDC CYBER 18 peripheral to the macro program.

INPUT/OUTPUT CAPABILITY

CDC CYBER 18-10M contains nine card slots for peripheral controllers. Three levels of interface are provided for the peripherals: Direct Memory Access (DMA), Auto Data Transfer (ADT), and AQ.

The DMA channel permits direct transfer of data between the peripherals and main memory. The DMA channel supports four devices and permits data transfer rates up to 2,800,000 bytes per second.

ADT provides pseudo DMA transfers of data blocks between main memory and those peripherals designed to accommodate ADT.

At the macro level each transfer appears as DMA; however, each transfer is controlled at the micro level by the emulator in micro memory. Data transfer rates up to 160,000 bytes per second are possible. Three ADT devices are supported.

The AQ channel provides data transfers between CPU registers and peripherals. The transfers are macro-program controlled. CDC CYBER 18-10M supports a maximum of four AQ devices. AQ data transfer rates are software dependent.

One additional input/output interface is included for the operator console device. This interface is both KSR 33/35 TTY compatible and RS232-C compatible.

PROGRAM DEADSTART

Loading programs into main memory is provided by this feature. Data is input bit-serially from the dead-start program loading device.

OPERATOR'S PANEL

An operator's panel is also included, and is used to initiate operation of the processor and deadstart device.

CONFIGURATION

Basic configuration includes a cabinet with operator's panel, a basic processor, a flexible disk drive and controller, an input/output controller to support the operator console, and power supply (no main memory is included).

Minimum system configuration consists of 32K bytes main memory, a load device such as a card reader, and a comment device such as a conversational display terminal.

SOFTWARE

Supporting software includes Mass Storage Operating System (MSOS), Real-Time Operating System (RTOS), and Interactive Terminal Operating System (ITOS). Both MSOS and RTOS are real-time, multiprogramming operating systems, with 16 program priority levels.

- RTOS – resides within the CPU memory and has no mass storage requirements. It includes a monitor (subset of MSOS) which occupies less than 1500 words of main memory, exclusive of drivers and optional features.
- MSOS – supports applications requiring dedicated system utilization, batch processing, and program checkout features in a real-time environment. Its modular design provides flexibility in system updating or modification.
- ITOS – provides an environment in which a terminal user operates with an on-line data base, using interactive application programs. ITOS Release 1 operates in conjunction with MSOS 5.0.

MAINTENANCE FEATURES

Self-test and echo mode tests are included for troubleshooting the basic processor and optional controllers.

The system is also supported by the Operational Diagnostic System (ODS). This maintenance system includes diagnostic software with fault isolation capability, Diagnostic Decision Logic Tables (DDLTs) and detailed repair procedures. These tools produce a highly effective and efficient maintenance system.



CYBER 18/MODEL 17 COMPUTER SYSTEM

The CDC CYBER 18/Model 17 is a family of compact, powerful, high-performance computer systems. It uses MSI TTL technology for the logic circuitry and LSI MOS for the main memory. The basic enclosure, which includes up to 32,768 words of memory, is rack-mountable. The central processor features arithmetic and control operations, interrupt processing, and program protection. Options include CDC 1700 plug-compatible channels, allowing the use of 1700 peripherals.

HIGHLIGHTS

- 18-bit storage word (16 data or instruction bits, program protect bit, and parity bit)
- 4,096-word basic storage, expandable to 65,536 words in 4,096-word increments
- 600 or 900 nanosecond storage cycle time depending on model
- Program protection system
- Two arithmetic registers
- Two index registers
- 16-level interrupt system
- Software compatibility with the CDC 1700 series systems

DESCRIPTION

The Model 17 provides high-speed, random, access, dynamic LSI MOS storage. The basic system has a memory of 4,096 words. This may be expanded in 4,096 word increments up to 32,768 words in the same enclosure. An auxiliary enclosure allows expansion up to a total of 65,536 words. Memory cycle time of either 900 or 600 nanoseconds is available. A number of peripheral controllers can also be included in the mainframe.

The central processor, besides arithmetic and control operations, includes both a program protection and an interrupt capability. The program protection is built around a program protect bit contained in each word of storage. If the bit is set, that word is an operand or instruction of the protected program.

The interrupt system has one internal and 15 external interrupt levels. A test is made for interrupt conditions at each storage cycle. When a condition occurs that can cause an interrupt, the program may either process the interrupt or ignore the conditions. If desired, a program need not use the interrupt system at all.

In addition to its own new set of low-to-medium-speed peripherals and controllers, the Model 17 has optional CDC 1700 plug-compatible channels. This means that any of the standard CDC 1700 peripherals can be plug-compatible with the Model 17.

SOFTWARE

The Model 17 has a full complement of software. This includes a Mass Storage Operating System (MSOS), a Real-Time Operating System (RTOS), and assemblers, compilers and utility subsystems. These are all available system components that support various applications, from paper tape input to real-time mass storage capabilities. In addition, there are a number of standard terminal software products that enable the Model 17 to operate as a terminal to a CDC CYBER 70/170 or CDC 6000 system.

MODELS

The Model 17 central processor is available in two models:

- Model 17A cycle time of 900 nanoseconds
- Model 17B memory cycle time of 600 nanoseconds.



CYBER 18/MODEL 20 PROCESSOR

The CDC CYBER 18/Model 20 is a highly versatile, general-purpose, micro-programmed, 16-bit processor that emulates the CDC 1700 instruction set (basic set plus a new enhanced instruction set). Its micro-programmed architecture provides a high degree of flexibility which allows the Model 20 to fill the needs of a wide range of application areas.

Execution of systems programs stored in macro main memory is performed under the control of a micro-level program stored in ROM micro memory. The micro level programs also operate programmed input/output channels, service the computer interrupt and program protect systems, and control the operating mode of the processor. They provide additional facility for character and field manipulation, indexing, and other system-oriented processes. Arithmetic is one's complement, signed, fixed-point hardware add/subtract/multiply/divide.

The arithmetic section consists primarily of several operation registers that are interconnected by selectors. The primary data registers that interface to the arithmetic and logical unit are the I, P, X, A, M, Y, and Q. The Model 20 has register files available at the microprogram level.

The Model 20 has 10 card positions available to support AQ/ADT/DMA type peripherals. The AQ channel provides data transfers between Central Processor Unit (CPU) registers and the internal peripheral controllers. All data transfers are performed under macroprogram control; transfer rates are software dependent.

An additional input/output interface is available for the optional operator comment device. This interface is both ASR/KSR 33/35 TTY compatible and RS232-C compatible.

The Model 20 has an integral real-time clock which provides a macro level interrupt at a programmable interval. The real-time clock appears as a CDC 1700 type peripheral to the macro program.

FEATURES

The Model 20 features include:

- General-purpose digital processor using micro-programmable architecture
- Accommodates 32K through 262K bytes macro main memory
- Cabinet, operator's panel, power distribution, and power supplies included
- High reliability and easy maintainability through state-of-the-art technology and advanced diagnostic capability
- Main memory effective read/write cycle time of 750 nsec
- Eight addressing modes for accessing main memory
- Main memory word and region protection
- Main memory parity detection

- Priority-oriented interrupt system with 16 levels each of micro and macro interrupts
- Powerful macro instruction repertoire
- Integral real-time clock
- Basic processor supporting a wide range of peripherals
- Modular design CPU and controllers on 11-inch x 14-inch printed circuit boards for ease of handling
- Automatic program load (deadstart) facility for loader-type peripherals
- High-speed input/output data transfer for integral peripheral controllers
- Input/output communications interface for teletypewriter or RS232-C compatible display terminal
- Optional micro/macro breakpoint controller
- Self test and echo mode tests are included as an aid in trouble-shooting the basic processor and optional controllers. The system is also supported by controlware diagnostics which are included in the Operational Diagnostic System (ODS). Tests are performed while using Diagnostic Decision Logic Tables (DDLs) and special maintenance procedures to isolate and correct the fault. These features provide for maximum efficiency in system maintenance.

CONFIGURATION

The configuration includes basic processor, cabinet with operator's panel and power supply, and input/output controller to support required communications console.

No main memory is included.

For operation, minimum system requirements are: CDC CYBER 18/Model 20, 32K bytes main memory, an input device such as a card reader, and a comment device such as a Conversational Display Terminal.

Macro Instruction Repertoire

The CDC CYBER 18/Model 20 incorporates the basic CDC 1700 instructions and new enhanced instructions. The repertoire includes one, two and three word instructions and is flexible enough to increase programming efficiency. Instruction groups include the following:

- Transfer
- Logical
- Stop
- Shift
- Interrupt
- Parity Generation

- Character/Field Manipulation
- Micro Code Sequence Execution
- Arithmetic
- Jump
- Decision
- Input/Output
- Program Protect
- Loop Control
- Memory Paging Control

Some instructions are immediate (literal), resulting in a saving of operand storage space and execution time. Multi-word instructions, like indirect addressing, provide the means for addressing locations which cannot be accessed directly.

Program Protection

The CDC CYBER 18/Model 20 offers two modes of protection from the damage which may be done by programs accessing memory outside their own region. The traditional word level protection of the CDC 1700 series is featured. This allows individual bytes to be declared protected by setting a bit in memory associated with that byte. A second means of protection is also employed using upper and lower bounds to define an unprotected region. This has the same effect as word protection, except that a large unprotected area can be defined more quickly.

Interrupt System

The CDC CYBER 18/Model 20 firmware emulates the 16 levels of vectored interrupt featured on the CDC 1700 series computers. This system consists of 15 levels of external interrupt and one internal interrupt.

Certain conditions such as an incorrect instruction, a memory parity error, or a power failure will generate an internal interrupt. External interrupts occur when a computer peripheral device has finished an input/output operation or requires attention.

The strength of the interrupt scheme is its ability to handle a significant number of interrupts in a flexible and efficient manner.



CYBER 18/MODEL 30 TIME-SHARE SYSTEM

The CDC CYBER 18-30 Time-share System consists of two general-purpose, microprogrammable, 16-bit processors sharing a common main memory. One processor functions as the timeshare processor, the other as a communications processor. Execution of macro programs stored in MSOS main memory is controlled by microlevel programs stored in each processor's micro memory. ROM micro memory is provided for execution of the basic CDC 1700 instruction set, and additional enhanced instructions including character and field manipulation, indexing, micro memory referencing, auto-data transfer, and main memory paging control. Read/write micro memory is available for user microprogramming requirements. Arithmetic is one's complement, signed, fixed-point hardware add/subtract/multiply/divide.

HIGHLIGHTS

- Two general purpose digital processors using microprogrammable architecture
- Accommodates 65K through 524K bytes shared macro main memory
- Main memory effective read/write cycle time of 750 nanoseconds, local bank
- Micro instruction cycle time of 168 nanoseconds
- Powerful instruction repertoire
- Eight addressing modes for accessing main memory
- Main memory word and region protection
- Main memory parity detection, with optional automatic single-error correction and double-error detection
- Direct memory access
- High-speed input/output data transfer for integral peripheral controllers
- Automatic program load (deadstart) facility for loader-type peripherals
- Integral real-time clock
- Modular-design CPU and controllers on 11 by 14-inch PC board for ease of handling
- High reliability and easy maintainability through state-of-the-art technology and advanced diagnostic capability
- Input/output communications interface for teletypewriter or RS232-C compatible display terminal
- Priority-oriented interrupt system with sixteen levels each of micro and macro interrupts
- Optional breakpoint controller

- Basic processor supports wide range of peripherals
- Cabinet, operator's panel, power distribution, and power supplies included
- Cassette tape subsystem included with each processor
- Optional read/write micro memory

SOFTWARE

The CDC CYBER 18-30 Time-Share System includes a dual processor with a software package designed to support up to 64 concurrent users at remote interactive terminals. It also provides access to a host CDC CYBER processor for remote job entry. Interactive programming from the terminals is performed via a powerful text editor, extended basic compiler and interpreter. Programs and data are stored on-line in a file management system, supporting a sequential file structure.

Program Deadstart

Loading programs into main memory and read/write micro memory is provided by this feature. Data is input bit-serially from the deadstart program loading device.

Addressing Modes

Each processor provides the following eight addressing modes for maximum flexibility:

- Absolute
- Indirect
- Relative
- Relative Indirect
- Constant
- Storage
- Storage Indirect
- Field

Macro Instruction Repertoire

Each processor incorporates the basic CDC 1700 instruction set and additional enhanced instructions not available before. The repertoire includes one, two, and three word instructions, and is flexible for increased programming efficiency. Instruction groups include the following:

- Transfer
- Logical
- Stop
- Shift
- Interrupt
- Generate Parity
- Character/Field Manipulation

- Execute Micro Code Sequence
- Arithmetic
- Jump
- Decision
- Input/output
- Memory Paging Control

Some instructions are immediate (literal), resulting in a saving of operand storage space and execution time. Multi-word instructions, like indirect addressing, are a means of addressing locations which cannot be accessed directly.

Registers

Each processor provides fourteen registers. The seven traditional registers are used in execution of the normal CDC 1700 instruction set, and four general purpose registers have been added to support the enhanced instruction set. Three special-purpose registers are used exclusively for machine control.

Program Protection

Each processor offers two modes of protection from the damage which may be caused by programs accessing memory outside of their own region. The traditional word level protection of the 1700 Series allows individual words to be declared protected by setting a bit in memory associated with that word. A second means of protection uses upper and lower bounds to define an unprotected region. This has the same effect as word protection, except that a large unprotected area can be defined more quickly.

INTERRUPT SYSTEM

Each processor's firmware emulates the 16 levels of vectored interrupt featured on the CDC 1700 series computers. This system consists of 15 levels of external interrupt and one internal interrupt.

Certain conditions such as an incorrect instruction, a memory parity error, or a power failure will generate an internal interrupt. External interrupts occur when a computer peripheral device has finished an input/output operation or requires attention. The strength of the interrupt scheme is the ability to handle a significant number of interrupts in a flexible and efficient manner.

MAIN MEMORY SYSTEM

The system features high-speed dynamic MOS LSI storage elements. Each word in memory consists of 2 data bytes, one protect bit, and one parity bit. Memory is organized as two banks with three ports per bank — local CPU, local DMA, and external CPU/DMA. Both processors have access to total memory.

Storage capacity is expandable from 65K to 524K bytes by the simple insertion of individual PC boards. The time-share system includes 196K bytes of main memory. In each processor, four card slots are provided to accept any mixture of 32K and 65K byte MOS memory array boards.

Average memory cycle time for the CPU is 700/960 nanoseconds for local/remote bank accesses. DMA average memory cycle time is 765/1020 nanoseconds. Each memory bank processes simultaneous requests

from all ports, with an effective cycle time of 600 nanoseconds.

Double-error detection and automatic single-error correction for up to 393K bytes is available.

INPUT/OUTPUT CAPABILITY

The processor contains 10 card slots for peripheral controllers. Three levels of interface are provided for the peripherals; Direct Memory Access (DMA), Auto Data Transfer (ADT), and AQ.

The DMA channel permits direct transfer of data between the peripherals and main memory, by-passing the CPU entirely. The DMA channel supports four devices and permits data transfer rates up to 1,400,000 words per second.

ADT provides pseudo DMA transfers of data blocks between main memory and those peripherals designed to accommodate ADT. At the macro level, each transfer appears as DMA; however, each transfer is controlled at the micro level by the 1700 emulator in micro memory. Data transfer rates up to 80,000 words per second are possible. Ten ADT devices are supported.

The AQ channel provides data transfers between CPU registers and peripherals. Transfers are macro-program controlled. The processor supports a maximum of nine AQ devices. AQ data transfer rates are software dependent.

Each processor includes an additional input/output interface for the shared operator input device. This interface is both ASR/KSR 33/35 TTY compatible and RS232-C compatible.

REAL-TIME CLOCK

The real-time clock is an integral part of each processor, and provides a macro-level interrupt at a programmable interval. This clock appears as a CDC CYBER 18 peripheral to the macro program.

OPERATOR'S PANEL

The processor includes a basic operator's panel to initiate the operation of each processor and each dead-start device.

MAINTENANCE FEATURES

Self-test and echo mode tests are included as an aid in trouble-shooting the basic processor and optional controllers. The system is also supported by controlware diagnostics included in the Operational Diagnostic System (ODS). Tests are performed using Diagnostic Decision Logic Tables (DDLTs) and special maintenance procedures isolate and correct the fault. These features provide maximum efficiency in system maintenance.



CYBER 70/MODEL 71 COMPUTER SYSTEM

The CDC CYBER 70/Model 71 computer system is a general-purpose digital computing system which can be used as a central computer for batch operations and as the nucleus for networks of interactive remote terminals. It is suitable for data management, scientific computation, commercial data processing, and data communications. Because of hardware and software compatibility among all CDC CYBER 70 systems, the Model 71 serves as an expandable base for long-range growth plans.

The Model 71 computing system has one or two large central processing units accessible through a central memory and a group of 10, 14, 17, or 20 peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor(s). Through the exchange jump feature and central memory communication, these peripheral processors control the central processor(s). They communicate with themselves through central memory and input/output channels.

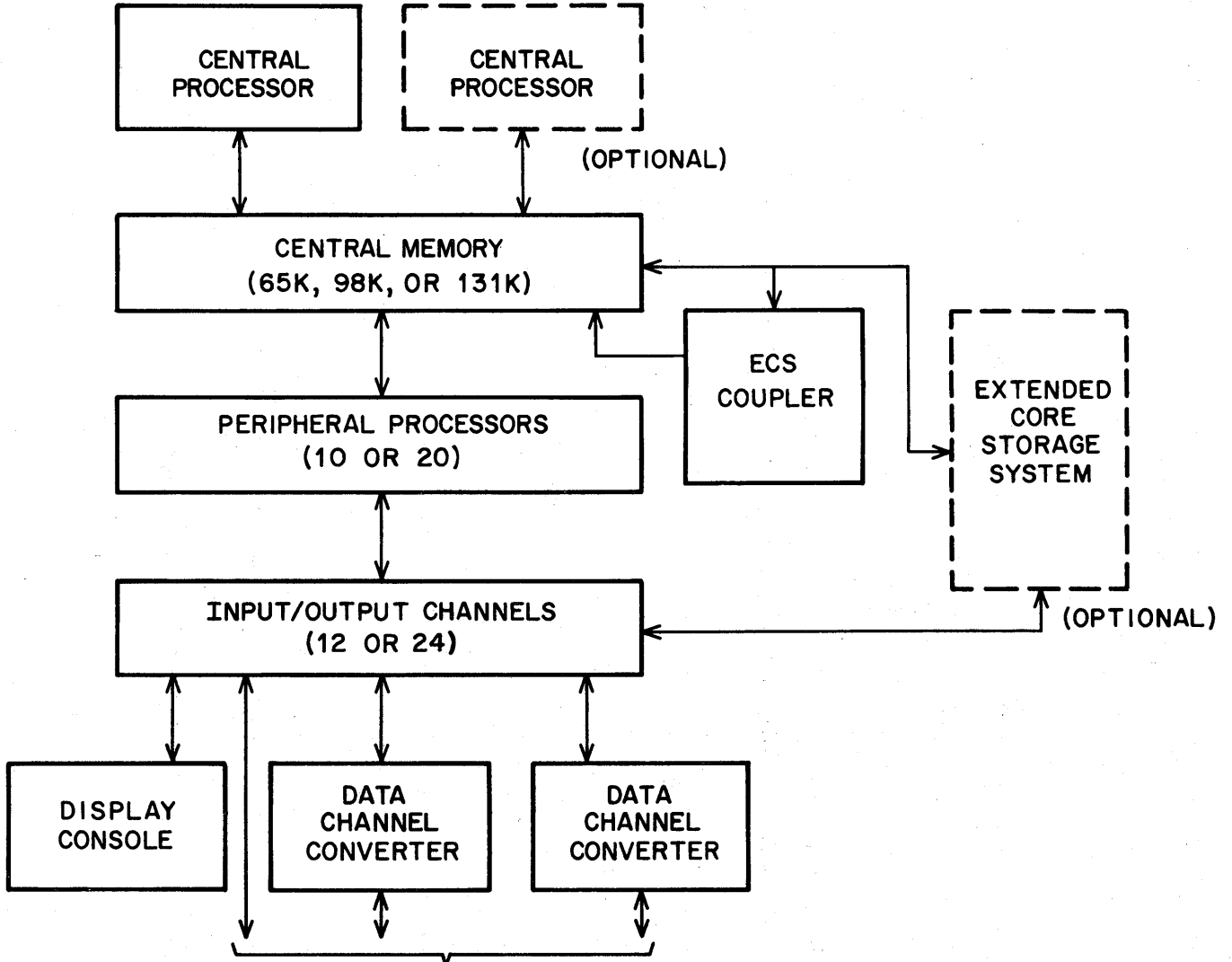
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor(s). Further concurrency is obtained with the central processor by parallel action of various functional segments.

Central memory is organized into logically independent banks of 4096 words. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Dual central processor configuration option
- 24 operating registers per central processor
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction issue rate — 1 processor, 1.2 million instructions per second
 - 2 processors, 2.0 million instructions per second
- Computation in floating-point and fixed point, single and double precision
- Integer multiply
- Central exchange jump



Magnetic Tape Transports
 Line Printers
 Mass Storage
 Card Readers
 Terminals (Batch or Interactive)

Peripheral Processor Characteristics

- 10, 14, 17, or 20 peripheral processor configurations available (characteristics as listed are per processor)
- 4096-word magnetic core memory (12-bit)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel of one word per microsecond
 - all channels can be active simultaneously
 - all channels 12-bit bidirectional
- Real-time clock period of 4.096 milliseconds
- Computation in fixed point
- Time-shared access to central memory

Central Memory Characteristics

- Memory listed in the amount associated with the model options given (60-bit words)

Model 71-14	65,536 words
Model 71-16	98,304 words
Model 71-18	131,072 words
Model 71-24†	65,536 words
Model 71-26†	98,304 words
Model 71-28†	131,072 words
- Memory organized in logically independent banks of 4096 words with corresponding multiphasing of banks (maximum memory size of 32 banks)
- Transfer rate of up to one word each 100 nanoseconds.

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates only with central memory. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of an arithmetic, manipulative, and logical operations. The control unit directs the execution of operations and provides the interface between the arithmetic unit and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing.

†Models which contain two central processors.

The following are the general categories of central processor instructions:

- Branch
- Central exchange jump
- Fixed point arithmetic
- Floating-point arithmetic
- Extended core storage communication
- Increment
- Logical
- Monitor, stop
- Pass
- Shift

PERIPHERAL PROCESSORS

The peripheral processors are identical, and operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor(s).

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit, 4096 word random access memory (independent of central memory) with a cycle time of 1000 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- Arithmetic
- Branch
- Central processor and central memory communications
- Data transmission
- Input/output
- Logical

- No operation
- Replace
- Shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the status of the program (registers, and so forth), and initiates execution of another program. The Model 71 provides four types of exchange jump instructions; one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus control), and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of one word per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controllers and card read/punch controllers.

Operator Console

The operator console consists of two cathode-ray tube displays and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 71 input/output channels.

MODEL 71 MEMORY

The CDC CYBER 70/Model 71 computer offers a hierarchical memory concept:

- Central core memory
- Extended Core Storage (ECS) (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 71 central memory is composed of banks of 4096 60-bit words of core storage. The complete cycle time for one bank is one microsecond. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 100 nanoseconds.

Central memory is available in sizes ranging from 65,536 words (16 banks) to 131,037 words (32 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory.

- One or two central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processor/central memory

Central memory includes a control section that provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts.

Extended Core Storage (ECS)

The optional extended core storage subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Each 60-bit word is contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This **phasing**, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 6 billion bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; and for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between ECS and the peripheral processors. It allows fast peripheral processor access to data in ECS using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of one-to-four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to ECS. This arrangement allows up to four peripheral processors to transfer data simultaneously to ECS at the maximum rate of the channel. A 480-bit ECS word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 70/MODEL 72 COMPUTER SYSTEM

The CDC CYBER 70/Model 72 computer system is a general-purpose digital computing system which can be used as a central computer for batch operations and as the nucleus for networks of interactive remote terminals. It is suitable in data management, scientific computation, commercial data processing and data communications. Because of hardware and software compatibility among the CDC CYBER 70 systems, the Model 72 serves as an expandable base for long-range growth plans.

The Model 72 computing system has one or two large central processing units. These are accessible through a very fast central memory and a group of 10, 14, 17, or 20 peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor(s). Through the exchange jump feature and central memory communication, these peripheral processors control the central processor(s). They communicate with themselves through central memory and input/output channels.

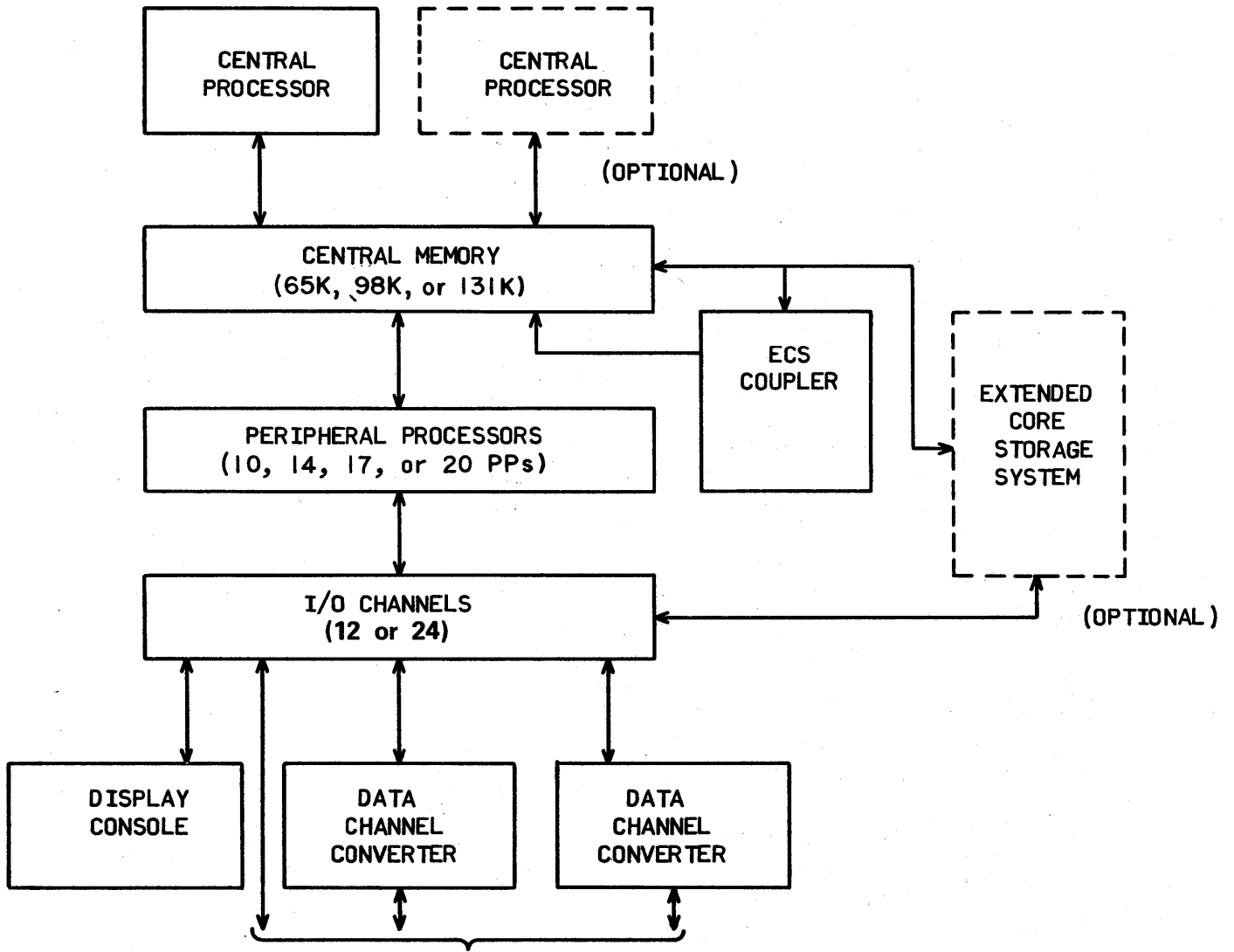
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor(s). Further concurrency is obtained with the central processor by parallel action of various functional segments.

Central memory is organized into logically independent banks of 4096 words. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Dual central processor configuration option
- 24 operating registers per central processor
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction issue rate — one processor, 0.9 million instructions per second
 - two processors, 1.5 million instructions per second
- Computation in floating point and fixed point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions



MAGNETIC TAPE TRANSPORTS
 LINE PRINTERS
 MASS STORAGE
 CARD READERS
 TERMINALS (BATCH OR INTERACTIVE)

CDC CYBER 70/Model 72 Computing System

Peripheral Processor Characteristics

- 10, 14, 17 or 20 peripheral processor configuration available (characteristics as listed are per processor)
- 4096-word magnetic core memory (12-bit)
- 12 or 24 input/output channels
 - All channels common to all processors
 - Maximum transfer rate per channel of one word per microsecond
 - All channels can be active simultaneously
 - All channels 12-bit bidirectional
- Real-time clock period of 4.096 milliseconds
- Computation in fixed-point
- Time-shared access to central memory

Central Memory Characteristics

- Memory listed in the amount associated with the model options given (60-bit words)

Model 72-14	65,536 words
Model 72-16	98,304 words
Model 72-18	131,072 words
Model 72-24†	65,536 words
Model 72-26†	98,304 words
Model 72-28†	131,072 words

- Memory organized in logically independent banks of 4096 words with corresponding multiphasing of banks (maximum memory size of 32 banks)
- Transfer rate of up to one word each 100 nanoseconds.

CENTRAL PROCESSOR

The central processor is an extremely high-speed arithmetic unit which communicates with central memory. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of an arithmetic and control unit. The arithmetic unit contains all logic necessary to execute the arithmetic, manipulative, and logical operations. The control unit directs the execution of operations and provides the interface between the arithmetic unit and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing.

The following are the general categories of central processor instructions:

- Branch
- Central Exchange Jump

†Models which contain two central processors.

- Fixed Point Arithmetic
- Floating-Point Arithmetic
- Extended Core Storage Communication
- Increment
- Logical
- Monitor, Stop
- Move, Compare Data Handling
- Pass
- Shift

PERIPHERAL PROCESSORS

The peripheral processors are identical and operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor(s).

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit, 4096 word random-access memory (independent of central memory) with a cycle time of 1000 ns. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- Arithmetic
- Branch
- Central Processor and Central Memory Communications
- Data Transmission
- Input/Output
- Logical
- No Operation
- Replace
- Shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, etc.), and initiates execution of another program. The Model 72 provides four types of exchange jump instructions; one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other via the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus control) and each can be connected to one or more external devices. Only one external equipment can utilize a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of one word per microsecond. As many as eight different types of external equipment can be connected to an input/output channel (magnetic tape controllers, card read/punch controllers, etc.).

Operator Console

The operator console consists of two cathode-ray tube displays and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 72 input/output channels.

MODEL 72 MEMORY

The CYBER 70/Model 72 computer offers a hierarchical memory concept:

- Central memory
- Extended Core Storage (ECS) (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 72 central memory is composed of banks of 4096 60-bit words or core storage. The complete cycle time for one bank is one microsecond. The banks are phased so that successive addresses are in different banks, to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 100 nanoseconds.

Central memory is available in sizes ranging from 65,536 words (16 banks) to 131,037 words (32 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory.

- One or two central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory includes a control section that provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts.

Extended Core Storage

The optional extended core storage subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive eight-word records come from different banks. This phasing, combined with the wide (eight-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; for storage of large data arrays; for storage of frequently used programs and system routines; and for multimainframe communication devices.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage via the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage via an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 70/MODEL 73 COMPUTER SYSTEM

The CDC CYBER 70/Model 73 computer system is a general-purpose digital computing system which can be used as a central computer for batch operations and as the nucleus for networks of interactive remote terminals. It is suitable for data management, scientific computation, commercial data processing, and data communications. Because of hardware and software compatibility among all CDC CYBER 70 systems, the Model 73 serves as an expandable base for long-range growth plans.

The Model 73 computing system has one or two large central processing units accessible through a central memory and a group of 10, 14, 17, or 20 peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor(s). Through the exchange jump feature and central memory communication, these peripheral processors control the central processor(s). They communicate with themselves through central memory and input/output channels.

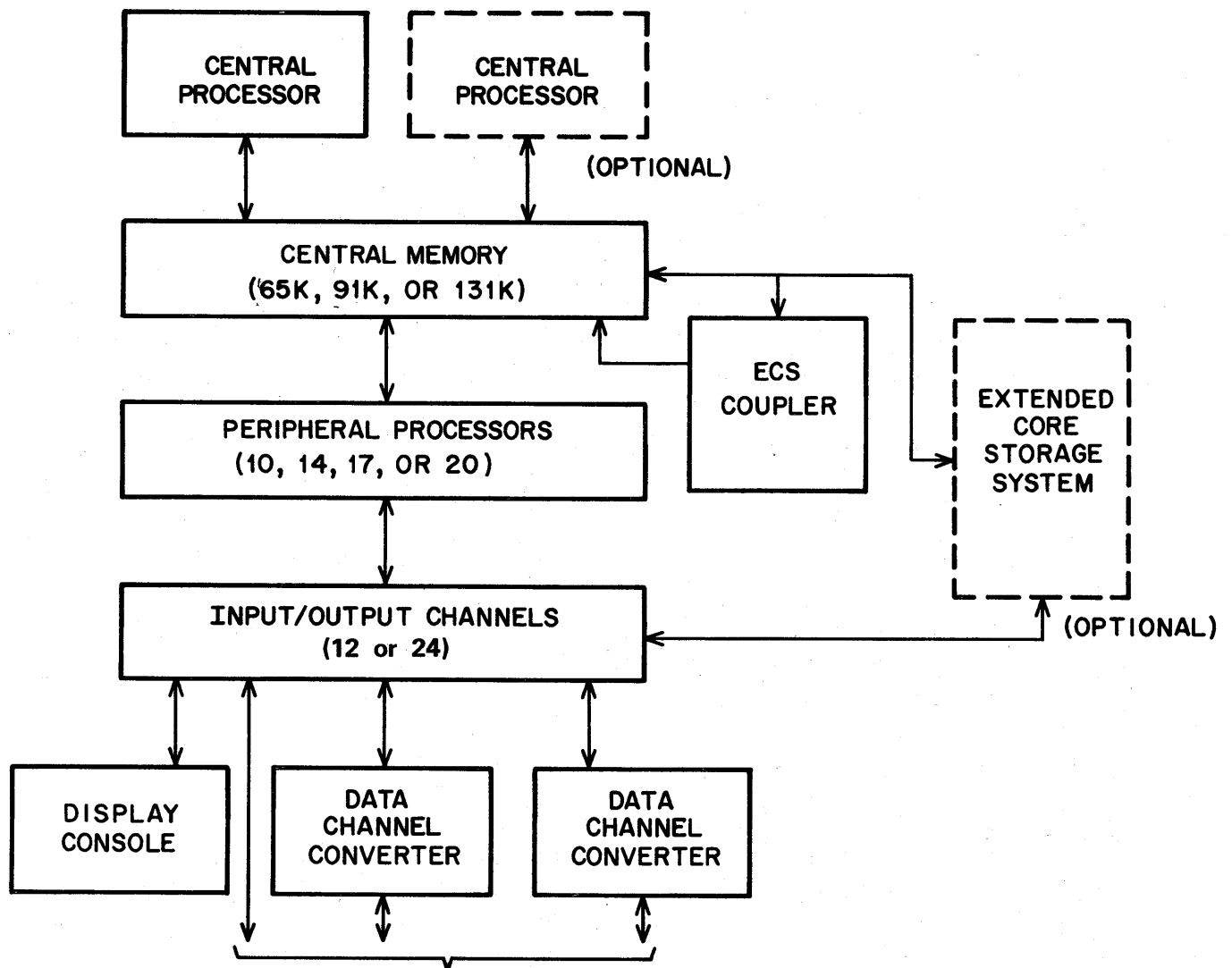
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor(s). Further concurrency is obtained with the central processor by parallel action of various functional segments.

Central memory is organized into logically independent banks of 4096 words. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Dual central processor configuration option
- 24 operating registers per central processor
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction issue rate — 1 processor, 1.2 million instructions per second
2 processors, 2.0 million instructions per second
- Computation in floating-point and fixed point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions



Magnetic Tape Transports
 Line Printers
 Mass Storage
 Card Readers
 Terminals (Batch or Interactive)

CDC CYBER 70/Model 73 Computing System

Peripheral Processor Characteristics

- 10, 14, 17, or 20 peripheral processor configurations available (characteristics as listed are per processor)
- 4096-word magnetic core memory (12-bit)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel of one word per microsecond
 - all channels can be active simultaneously
 - all channels 12-bit bidirectional
- Real-time clock period of 4.096 milliseconds
- Computation in fixed point
- Time-shared access to central memory

Central Memory Characteristics

- Memory listed in the amount associated with the model options given (60-bit words)

Model 73-14	65,536 words
Model 73-16	98,304 words
Model 73-18	131,072 words
Model 73-24†	65,536 words
Model 73-16†	98,304 words
Model 73-28†	131,072 words

- Memory organized in logically independent banks of 4096 words with corresponding multiphasing of banks (maximum memory size of 32 banks)
- Transfer rate of up to one word each 100 nanoseconds

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates only with central memory. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of an arithmetic and control unit. The arithmetic unit contains all logic necessary to execute the arithmetic, manipulative, and logical operations. The control unit directs the execution of operations and provides the interface between the arithmetic unit and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing.

The following are the general categories of central processor instructions:

- Branch
- Central exchange jump

†Models which contain two central processors.

- Fixed-point arithmetic
- Floating-point arithmetic
- Extended core storage communication
- Increment
- Logical
- Monitor, stop
- Move, compare data handling
- Pass
- Shift

PERIPHERAL PROCESSORS

The peripheral processors are identical, and operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor(s).

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit, 4096 word random access memory (independent of central memory) with a cycle time of 1000 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- Arithmetic
- Branch
- Central processor and central memory communications
- Data transmission
- Input/output
- Logical
- No operation
- Replace
- Shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 73 provides four types of exchange jump instructions; one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus control), and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of one word per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controllers and card read/punch controllers.

Operator Console

The operator console consists of two cathode-ray tube displays and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 73 input/output channels.

MODEL 73 MEMORY

The CDC CYBER 70/Model 73 computer offers a hierarchical memory concept:

- Central core memory
- Extended Core Storage (ECS) (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 73 central memory is composed of banks of 4096 60-bit words of core storage. The complete cycle time for one bank is one microsecond. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 100 nanoseconds.

Central memory is available in sizes ranging from 65,536 words (16 banks) to 131,037 words (32 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory.

- One or two central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory includes a control section that provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access requests as necessary, and resolves any access conflicts.

Extended Core Storage

The optional extended core storage subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 125,952 60-bit words of core storage. Each 60-bit word is contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 6000 million bits per second.

Extended core storage is available in sizes ranging from 125,952 words (one bank) to 2,015,232 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; and for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of one-to-four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 70/MODEL 74 COMPUTER SYSTEM

The CDC CYBER 70/Model 74 computer system is a large-scale, general-purpose digital computing system. Its unusual multiple-computer architecture permits data processing and scientific computing applications with large data bases or communications. In addition, this system is capable of multiprocessing, time-sharing multiprogramming, real-time, and hybrid applications. Because of hardware and software compatibility among all CDC CYBER 70 systems, the Model 74 serves as an expandable base for long-range growth plans.

The Model 74 computing system has one or two large central processing units accessible through a very fast central memory and a group of 10, 14, 17, or 20 peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor(s). Through the exchange jump feature and central memory communication, these peripheral processors control the central processor(s). They communicate with themselves through central memory, input/output channels.

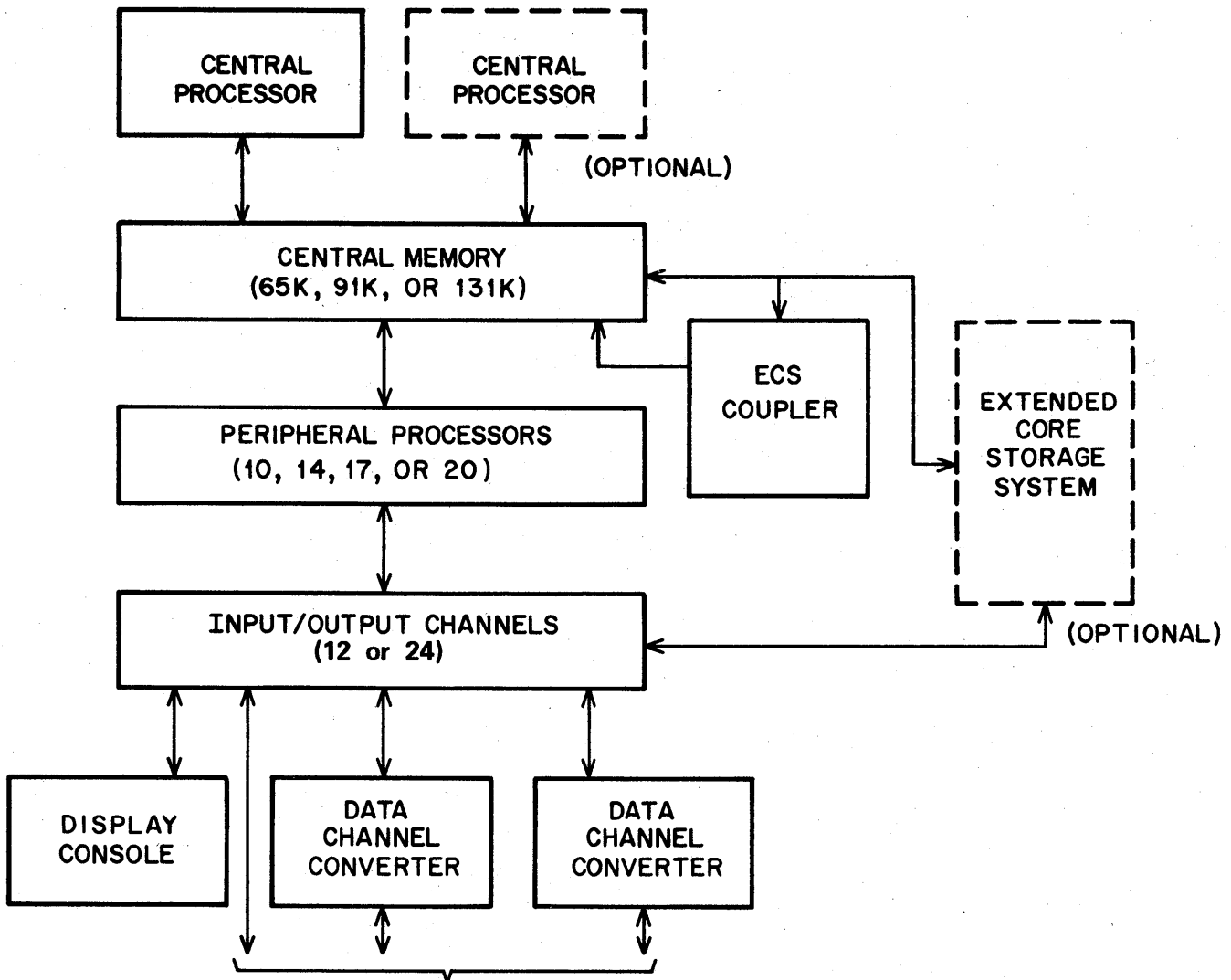
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor(s). Further concurrency is obtained with the central processor by parallel action of various functional segments.

Central memory is organized into logically independent banks of 4096 words. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- 10 independent arithmetic functional units for concurrent operations
- Dual central processor configuration option (second processor is a unified arithmetic unit)
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack of 8 60-bit words
- Instruction issue rate
 - 1 processor, 3.0 million instructions per second
 - 2 processors, 3.7 million instructions per second
- Computation in floating-point and fixed point, single and double precision
- Integer multiply
- Central exchange jump



Magnetic Tape Transports
 Line Printers
 Mass Storage
 Card Readers
 Terminals (Batch or Interactive)

CDC CYBER 70/Model 74 Computing System

Peripheral Processor Characteristics

- 10, 14, 17, or 20 peripheral processor configurations available (characteristics as listed are per processor)
- 4096-word magnetic core memory (12-bit)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel of one word per microsecond
 - all channels can be active simultaneously
 - all channels 12-bit bidirectional
- Real-time clock period of 4.096 milliseconds
- Computation in fixed point
- Time-shared access to central memory

Central Memory Characteristics

- Memory listed in the amount associated with the model options given (60-bit words)

Model 74-14	65,536 words
Model 74-16	98,304 words
Model 74-18	131,072 words
Model 74-24†	65,536 words
Model 74-26†	98,304 words
Model 74-28†	131,072 words

- Memory organized in logically independent banks of 4096 words with corresponding multiphasing of banks (maximum memory size of 32 banks)
- Transfer rate of up to one word each 100 nanoseconds in phased operation.

CENTRAL PROCESSOR

The central processor, composed of 10 separate functional processors which operate in parallel, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The 10 arithmetic and logical units (add, long add, increment-2, shift, multiply-2, divide, boolean, and branch) execute the arithmetic, manipulative, and logical operations. A control unit directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing.

The following are the general categories of central processor instructions:

- Branch
- Central exchange jump

†Models which contain two central processors.

- Fixed point arithmetic
- Floating-point arithmetic
- Extended core storage communication
- Increment
- Logical
- Monitor, stop
- Pass
- Shift

PERIPHERAL PROCESSORS

The peripheral processors are identical and operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor(s).

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit, 4096 word random-access memory (independent of central memory) with a cycle time of 1000 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- Arithmetic
- Branch
- Central processor and central memory communications
- Data transmission
- Input/output
- Logical
- No operation
- Replace
- Shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 74 provides

four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus control) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of one word per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Operator Console

The operator console consists of two cathode-ray tube displays and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 74 input/output channels.

MODEL 74 MEMORY

The CDC CYBER 70/Model 74 computer offers a hierarchical memory concept:

- Central core memory
- Extended core storage (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Extended Core Storage (ECS) is optional and fully supported to increase system performance.

Central Memory

The Model 74 central memory is composed of banks of 4096 60-bit words of core storage. The complete cycle time for one bank is one microsecond. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 100 nanoseconds.

Central memory is available in sizes ranging from 65,536 words (16 banks) to 131,037 words (32 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory.

- One or two central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory includes a control section that provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts.

Extended Core Storage

The optional extended core storage subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Each 60-bit word is contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 6000 million bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a high-speed program swapping device; and for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of one-to-four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 70/MODEL 76 COMPUTER SYSTEM

The CDC CYBER 70/Model 76 computer system is a multipurpose computing system that provides commercial, data management, scientific, and real-time/time-critical capabilities. The Model 76 provides computing capacity substantially beyond that of the Model 74.

The Model 76 contains a computation section, semiconductor memory, a large core memory, input/output multiplexers, peripheral processors, and a maintenance control unit. The computation section contains the arithmetic units, registers, and instruction stack used to execute a program stored in the semiconductor memory. Besides storing the executing system or user program, small semiconductor memory also contains buffer areas through which data is passed to or from the input/output channels. The **large** core memory provides the communication channels between the computation section and the peripheral processors using the small semiconductor memory. There are a total of 15 channels in the input/output multiplexer. The system is initially started through the maintenance control unit. This unit is used almost exclusively for maintenance functions.

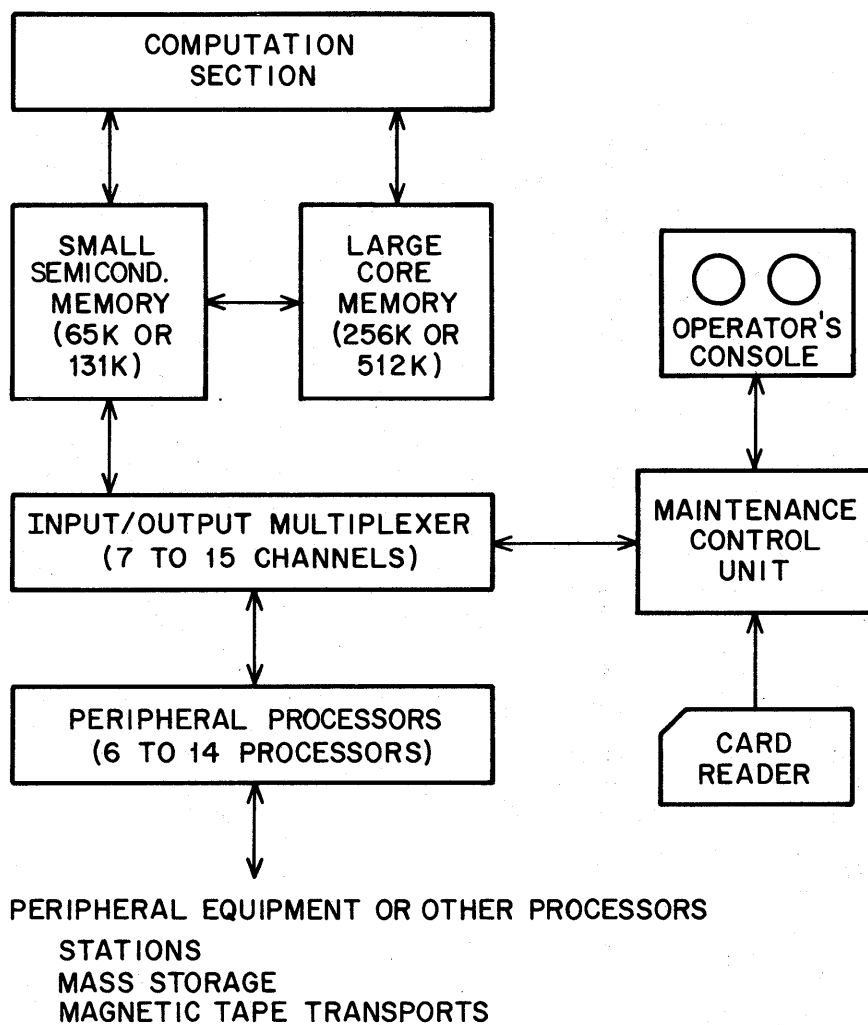
SYSTEM CHARACTERISTICS

Computation Section

- 9 arithmetic units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to 48 instructions
- Instruction issue rate of 15-million instructions per second
- Computation in floating-point and fixed point, single and double precision
- Interrupt structure

Large and Small Core Memories (LCM/SSM)

- Memory listed in the amount associated with the model options given (60-bit words)
 - Model 76-121 65,536 SSM 256,000 LCM
 - Model 76-122 65,536 SSM 512,000 LCM
 - Model 76-142 131,072 SSM 512,000 LCM
- Small Semiconductor Memory
 - Memory organized in logically independent banks of 4096 words with corresponding multiphasing of banks (maximum memory size of 32 banks)
 - Transfer rate of up to one word each 27.5 nanoseconds in phased operations



CDC CYBER 70/Model 76 Computing System

- Large Core Memory
 - Memory organized in logically independent banks of 64,000 words with corresponding multiphasing of banks (maximum memory size of 8 banks)
 - Transfer rate of up to one word each 27.5 nanoseconds in phased operations

Input/Output Multiplexer

- 7 independent 12-bit channels (expandable to 15)
- Each channel bidirectional
- Fixed small semiconductor memory buffer areas for each channel, 128 or 256 (60-bit words)
- Provides both normal-speed and high-speed channels

Peripheral Processors

- 6 independent processors (expandable to 14)
- 4096 words of coincident current memory (12-bit) per processor
- Eight input/output channels per processor
 - Maximum transfer rate of one 12-bit word each 247.5 nanoseconds
 - All input/output channels 12-bit bidirectional
- Computation in fixed-point

COMPUTATION SECTION

The computation section consists of nine arithmetic and logic function units, 24 operating registers, and a 12-word instruction stack. The functional units (long add, floating add, floating multiply, floating divide, boolean, shift, normalize, population count, and increment) operate in parallel and each will handle multiple instructions simultaneously, in a segmented fashion. (Segmentation time of each unit is 27.5 nanoseconds.) Operating concurrently, the functional units receive operands from, and deliver results back to the 24 operating registers. The 12-word instruction stack can hold 24 to 48 of the most current instructions. Up to 39 of these noncontiguous instructions can be executed without further references to small memory.

The following are the general categories of computation section instructions:

- Monitor, small core memory, large core memory, and input/output
- Branch
- Boolean
- Shift
- Fixed point arithmetic
- Floating-point arithmetic
- Increment
- Pass

MEMORY

The Model 76 contains two types of memories. These are the small semiconductor memory and a slower large core memory. This combination provides an efficient hierarchy of storage space and access time to support the central processor. Data and instructions can be transferred between the two memories or between either memory and the computation section at a maximum rate of one 60-bit word every 27.5 nanoseconds, or 36 million words per second.

INPUT/OUTPUT MULTIPLEXER

The input/output multiplexer transfers data between the peripheral processors and the small semiconductor memory. The multiplexer can have up to 15 bidirectional channels (one of which is reserved by the maintenance control unit), which can transfer a total of 18 million words per second. Each channel has its own fixed input buffer and output buffer in small memory (normally 128 or 256 words). Some of the channels are high-speed channels. High-speed channels transfer data at approximately twice the speed of normal-speed channels.

PERIPHERAL PROCESSORS

The peripheral processors are separate and independent computers which can be connected to small semiconductor memory via the input/output multiplexer, another processor, or a peripheral device. Each processor has its own 12-bit 4096 word memory, and each has eight input/output channels as well as individual operating registers and an arithmetic section. The primary function of the processor is to control, direct, and disseminate input/output data between the multiplexer and computer stations or peripheral equipment.

The following are the general categories of peripheral processor instructions:

- Input/Output
- Arithmetic
- Branch
- Logical
- Replace
- Shift

MAINTENANCE CONTROL UNIT

The maintenance control unit is a specially designed processor with the ability to control/scan all activities within the processors and memory sections of the Model 76. The maintenance control unit also includes a card reader and a cathode ray tube visual display, with entry keyboard. Facilities are provided for on-line maintenance and diagnostic programs used in testing all sections of the system.

Tests are performed in parallel with normal system operations. The maintenance control unit detects, diagnoses, and logs all system errors — both recoverable and nonrecoverable — for later statistical analysis.



CYBER 170/MODEL 171 COMPUTER SYSTEM

The CDC CYBER 170/Model 171 computer is a member of the CDC CYBER 170 family of compatible computers which includes Models 171, 172, 173, 174, 175 and 176. This system can be used as a central computer for batch operations, as the nucleus for a network of interactive and/or batch terminals, or function efficiently in mixed-mode environments. The Model 171 is a multi-purpose system suitable for data management applications, scientific computation, commercial data processing and data communications. Capitalizing on hardware and software compatibility characteristics of the CDC CYBER 170 computer line, the Model 171 serves as a base to support long-range growth plans. The Model 171 is field-upgradable to a CDC CYBER 170/Model 172, 173, 174, or 175 system.

A combination of current circuit technologies including Medium-Scale Integration (MSI) and Emitter-Coupled Logic (ECL) unit with Metallic-Oxide Semiconductor (MOS) memories to provide high levels of performance and reliability.

The basic mainframe includes:

- Central Processor Unit
- Central Memory Control
- Central Memory
- Peripheral Processor Subsystem

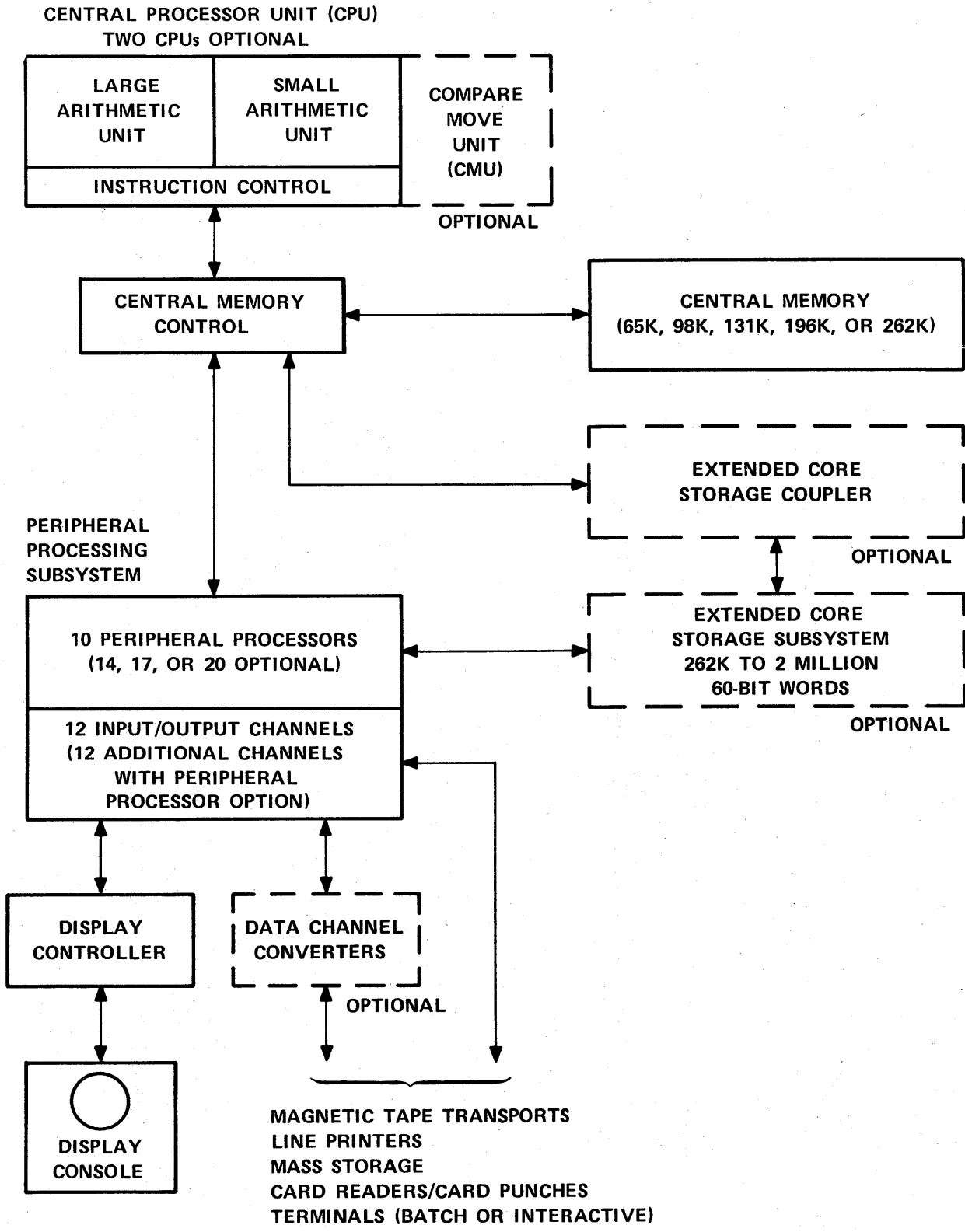
CENTRAL PROCESSOR UNIT

The Model 171 is configurable with one, or as an option, two Central Processor Units (CPUs), each consisting of a unified arithmetic unit with 24 operating registers. The instruction control section of the CPU directs arithmetic operations, manages the character-manipulative functions of an optional Compare/Move Unit (CMU)†, and interfaces the Central Memory Control (CMC) with the arithmetic sections of the CPU.

COMPARE/MOVE UNIT (CMU) (Optional)

The compare/move arithmetic provides multiple character manipulative functions. The characters are six bits in length. Characters can be moved from one central memory location to another, and fields of characters can be compared either directly or through a collation table.

†If CMU is desired in a dual CPU system, each CPU must have its own CMU component.



CDC CYBER 170/Model 171 Computing System

CENTRAL MEMORY CONTROL

The CMC provides for the orderly flow of data between central memory and requesting elements of the system. The CMC assigns and regulates request priorities, controls read/write operations, increments addresses, and processes parity generation and checking for addresses and data. Single-Error Correction/Double-Error Detection (SECEDED) circuitry is located in the CMC. The SECEDED feature permits computer operation to proceed after single-bit failures, which are detected and then logged by software system during a read operation, are corrected. Multiple-bit failures are detected and logged by the software system to facilitate remedial maintenance; further execution of the program or operating system is halted.

CENTRAL MEMORY

A selection of optimal central memory sizes is available to an installation in the process of configuring a Model 171. Five increments, ranging from 65K to 262K 60-bit words, provide excellent growth capability. Address parity and data SECEDED insure user security, while a central memory reconfiguration feature permits maximum system availability even after solid, uncorrectable, memory failures are detected. Central memory is organized in 8 or 16 banks of interleaving storage. Memory sizes to 131K are accommodated in a single Central Storage Unit (CSU), while a second CSU is configured for the larger sizes.

PERIPHERAL PROCESSOR SUBSYSTEMS

The Model 171 can be configured to include one or two Peripheral Processor Subsystems (PPS). The basic PPS consists of 10 Peripheral Processors (PPs) and 12 data channels, while the second PPS (optional) can be selected to expand to 14, 17, or 20 PPs overall. The optional PPS includes an additional 12 data channels. PPs are each a functionally independent computer comprising a 4K memory and arithmetic section supporting a repertoire of 64 arithmetic and input/output instructions. The PPs share multiplexed access to central memory and to the 12-bit bidirectional input/output channels.

EXCHANGE JUMP

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, etc.), and initiates execution of another program. The Model 171 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case, contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with new exchange package information from central memory. This permits a new program to be started by the central processor while retaining the information needed to resume the program that was previously executing.

INPUT/OUTPUT DATA CHANNELS

All peripheral processors can communicate with external equipment and each other, using the independent, bidirectional input/output channels. The number of channels (12 or 24) depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at a time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. Different types of external equipment can be connected to an input/output channel; for example, magnetic tape controllers and card read/punch controllers, etc.

DATA CHANNEL CONVERTER (Optional)

The Data Channel Converter allows CDC 3000-series type peripheral equipment to be attached to the Model 171 data channel. In a single PPS configuration, two converters may be added. If a second PPS is configured, then up to four converters may be added. The Model 171 computer transmits codes to the converter prior to starting any operation on an external equipment. These codes establish conditions in the converter so that the proper signals accompany the Model 171 input/output operations. Up to eight peripheral devices may be connected to the output of the converter.

DISPLAY CONTROLLER/CONSOLE

The display controller provides digital and analog information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- Displays registers
- Dynamic display of any word or block of words in central memory
- Provides an ability to test sense switches under program control
- The capability to manually start execution from any given location
- Displays current and past status (dayfile) of activity within the system relative to individual jobs
- Displays system messages
- Permits the operator to modify computer programs and view the result on the screen

A second display console can be configured with the Model 171 upon request.

EXTENDED CORE STORAGE (ECS) (Optional)

ECS is optionally available for attachment to the Model 171 as an on-line, random access, magnetic core memory system augmenting central memory. ECS can be accessed through a coupler facilitating direct block transfers to/from central memory, and through a device called a Distributive Data Path (DDP) providing one to four discrete paths through PPS data channels to the PPs. High transfer rates (up to 10 central memory words per microsecond over the ECS Coupler) and short access time make ECS an

ideal high-speed program swapping device as well as a direct access storage device for large arrays addressable by Model 171 users. ECS is also useful as an input/output buffer between central memory and rotating mass storage, and as a link, or communication device, in a multiframe environment since ECS can be accessed through four logically independent ports. ECS is available in several increments ranging from 262K to 2 million 60-bit words.

SOFTWARE

The CDC CYBER 170/Model 171 is supported by the CDC CYBER 170 Network Operating System (NOS). NOS exists in two variants: NOS, optimized for large interactive terminal networks, and NOS/BE, providing a feature-rich operating system designed for heavy batch-load environments. Since a great deal of feature overlap exists between the two variants, most Model 171 users could install either to advantage. Common to both operating systems is an extensive set of software products including: COMPASS (assembler), FORTRAN Extended, SORT/MERGE, BASIC, COBOL, simulation languages, data management subsystems, and various communications access methods.

PERIPHERAL EQUIPMENT SUPPORT

A large variety of peripheral equipment can be configured with the Model 171:

- Line printers
- Card readers
- Card punches
- Rotating mass storage
- Magnetic tape units
- Second console display
- Interactive terminals
- Remote batch terminals
- Graphics terminals
- Communications subsystem interfaces

SYSTEM CHARACTERISTICS

Central Processor

- 18 and 60-bit operands
- Floating-point arithmetic
- 24 operating registers
 - 8 operand (60-bit)
 - 8 operand (18-bit)
 - 8 increment (18-bit)

- Unified arithmetic unit
- 15 and 30-bit instructions
- Second CPU optional
- Compare/move unit optional
- Compatible with CDC 6000/CYBER 70

Central Memory

- 65K, 98K, 131K, 196K, and 262K 60-bit words available
- SECEDED logic
- 8 or 16-bank phased memory
- 400 nanosecond cycle time/50 nanosecond effective (phased)
- 20 words per microsecond transfer rate
- MOS 1K chip storage
- Reconfigurable for availability

Peripheral Processor Subsystem

- 10, 14, 17, or 20 PPs configurable
- MOS memory 4K with parity, each PP
- 12 and 24-bit instructions
- 12 or 24 data channels, 12-bit bidirectional with parity
- PPs each functionally an independently programmable computer
- Data channel converters optional

Extended Core Storage (Optional)

- 262K to 2 million 60-bit words of auxiliary storage
- 10 million words per second maximum transfer rate
- CMC and DDP to peripheral processors

MODELS

Model 171-4	65,536 central memory words
Model 171-6	98,304 central memory words
Model 171-8	131,072 central memory words
Model 171-12	196,608 central memory words
Model 171-16	262,144 central memory words



CYBER 170/MODEL 172 COMPUTER SYSTEM

The CDC CYBER 170/Model 172 computer system is a general-purpose digital computing system which can be used as the nucleus for networks of interactive and remote batch terminals and as a central computer for batch operations. It is suitable in data management, scientific computation, commercial data processing and data communications. Because of hardware and software compatibility among the CDC CYBER 70 and CDC CYBER 170 systems, the Model 172 serves as an expandable base for long-range growth plans.

The Model 172 computing system has a high-speed central processor unit, accessible through central memory and a group of 10 (or as options) 14, 17, or 20 peripheral processors known as the Peripheral Processor Subsystem (PPS). A second central processor is available as an option. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

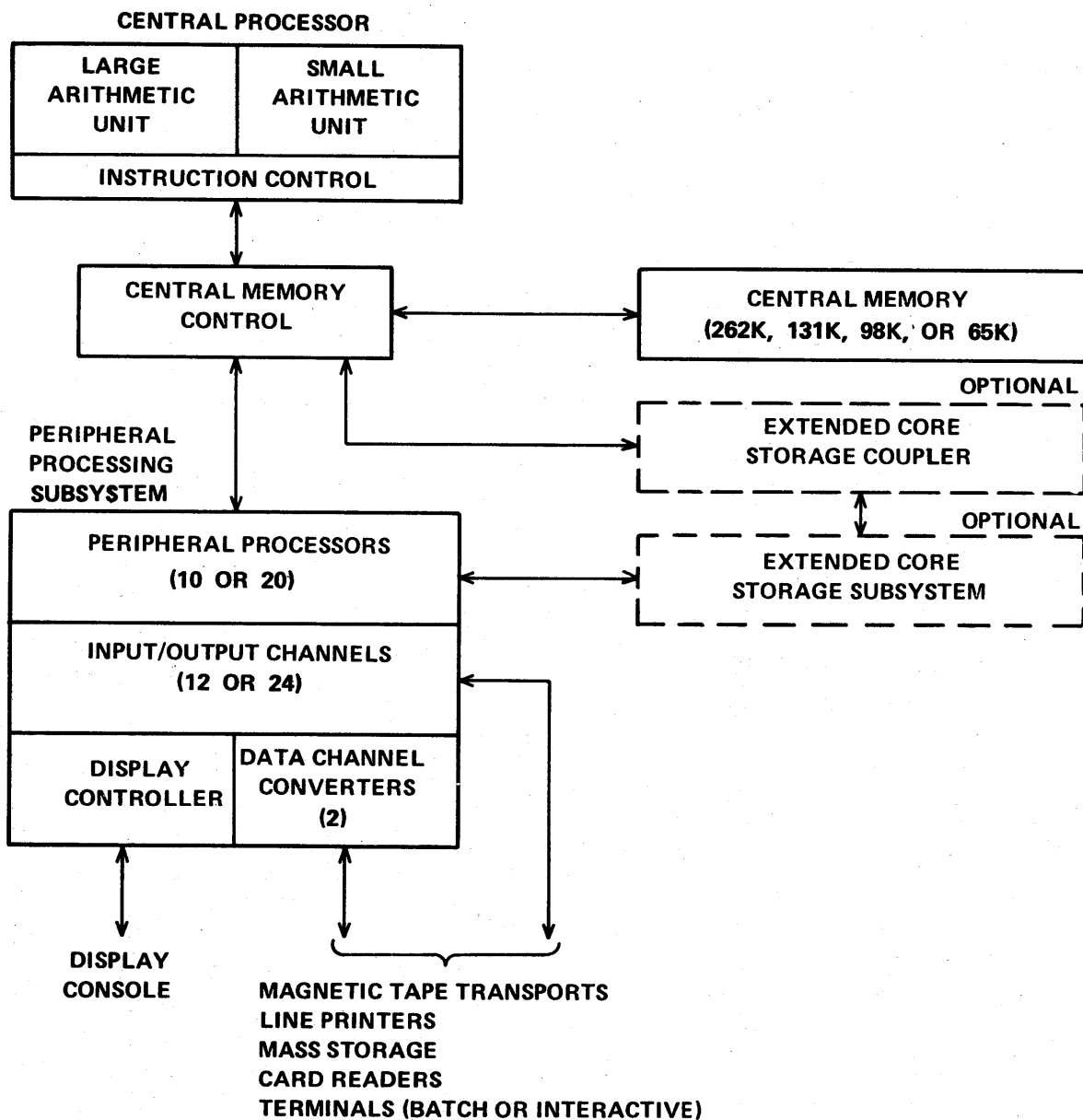
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor.

Central memory is organized into logically independent banks. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Uses Emitter-Coupled Logic (ECL) integrated circuits
- 24 operating registers
 - 8 operand (60-bit)
 - 8 operand (18-bit)
 - 8 increment (18-bit)
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions



CDC CYBER 170/Model 172 Computing System

Peripheral Processor Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 peripheral processor configuration (characteristics as listed are per processor)
- Semiconductor memory (MOS) of 4096 12-bit words plus one parity bit word (odd parity)
- 12 (or 24 with option) input/output channels
 - All channels common to all processors of the PPS
 - Maximum transfer rate per channel is 2 MHz
 - All channels can be active simultaneously
 - All channels 12-bit plus parity (odd) bidirectional
- Cycle time of 500 nanoseconds
- Computation in fixed-point
- Time-shared access to central memory
- Status and control register monitors error conditions (maintenance aid)
- 64-instruction repertoire

Central Memory Characteristics

- MOS semiconductor memory model options listed with associated capacity of 60-bit words plus eight error correction bits per word

Model 172-4	65,536 words
Model 172-6	98,304 words
Model 172-8	131,072 words
Model 172-16	262,144 words
- Memory organized in logically independent banks of words with corresponding multi-phasing of 8 banks.
- Transfer rate of up to one word each 50 nanoseconds.

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates with central memory via central memory control. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of a large and a small arithmetic unit and an instruction control unit. The arithmetic units contain all logic necessary to execute the arithmetic, manipulative, and logical operations. The instruction control unit directs the arithmetic operations and provides the interface between the arithmetic units and central memory control. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. A second central processor is available as an option.

The following are the general categories of central processor instructions:

- Branch
- Central exchange jump

- **Fixed-point arithmetic**
- **Floating-point arithmetic**
- **Extended core storage communication**
- **Increment**
- **Logical**
- **Monitor, stop**
- **Move, compare data handling**
- **Pass**
- **Shift**
- **Normalize**

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of **10, 14, 17, or 20 identical** units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), 4096-word, random-access memory (independent of central memory) with a cycle time of 500 ns.

The following are the general categories of peripheral processor instructions:

- **Arithmetic**
- **Central processor and central memory communications**
- **Data transmission**
- **Input/output**
- **Logical**
- **No operation**
- **Replace**
- **Shift**
- **Direct and indirect addressing**

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, etc.), and initiates execution of another program. The Model 173 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other via the independent, bidirectional input/output channels. All 12 channels (24 with option) are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can utilize a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel (magnetic tape controllers, card read/punch controller, etc.).

Display Controller/Console

The display controller provides digital and analog information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard (second display optionally available). The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 172 input/output channels.

MODEL 172 MEMORY

The CDC CYBER 170/Model 172 computer offers a hierarchical memory concept:

- Central MOS memory
- Extended core storage (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 172 central memory is composed of banks of 60-bit words of Metallic-Oxide Semiconductor storage. There are also eight error correction bits per 60-bit word. The complete cycle time for one bank is 400 nanoseconds (major cycle). The banks are phased so that successive addresses are in different banks, to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds (minor cycle).

Central memory is available in sizes ranging from 65K words (eight banks) to 262K words (eight banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are three access paths to central memory:

- Central processor/central memory
- Extended core storage/central memory
- Group of peripheral processors/central memory

Central memory control provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, the error correction bits will correct all single-bit errors. Multiple-bit errors are detected and their occurrence is noted in the status and control register.

Extended Core Storage

The Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of from 2 to 16 banks of 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262K words (two banks) to 2M words (16 banks). The very fast transfer rates and short access time of extended core storage make it ideal for use as a buffer between central memory and rotating mass storage devices as a high-speed program swapping device for storage of large data arrays and for storage of frequently used programs and system routines. ECS is also used as a linking medium in NOS multimainframe configurations.

<u>Extended Core Storage Options</u>	
<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262K	2
524K	4
1048K	8
2097K	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage via the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage via an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170/MODEL 173 COMPUTER SYSTEM

1037

The Control Data CYBER 170/Model 173 computer system is a general-purpose digital computing system which can be used as the nucleus for networks of interactive and remote batch terminals and as a central computer for batch operations. It is suitable for data management, scientific computation, commercial data processing, and data communications. Because of hardware and software compatibility among CDC CYBER 70 and CDC CYBER 170 systems, the Model 173 serves as an expandable base for long-range growth plans.

The Model 173 computing system has a high-speed central processor unit, accessible through a central memory control, and a group of 10 (20 optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor.

Central memory is organized into logically independent banks. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

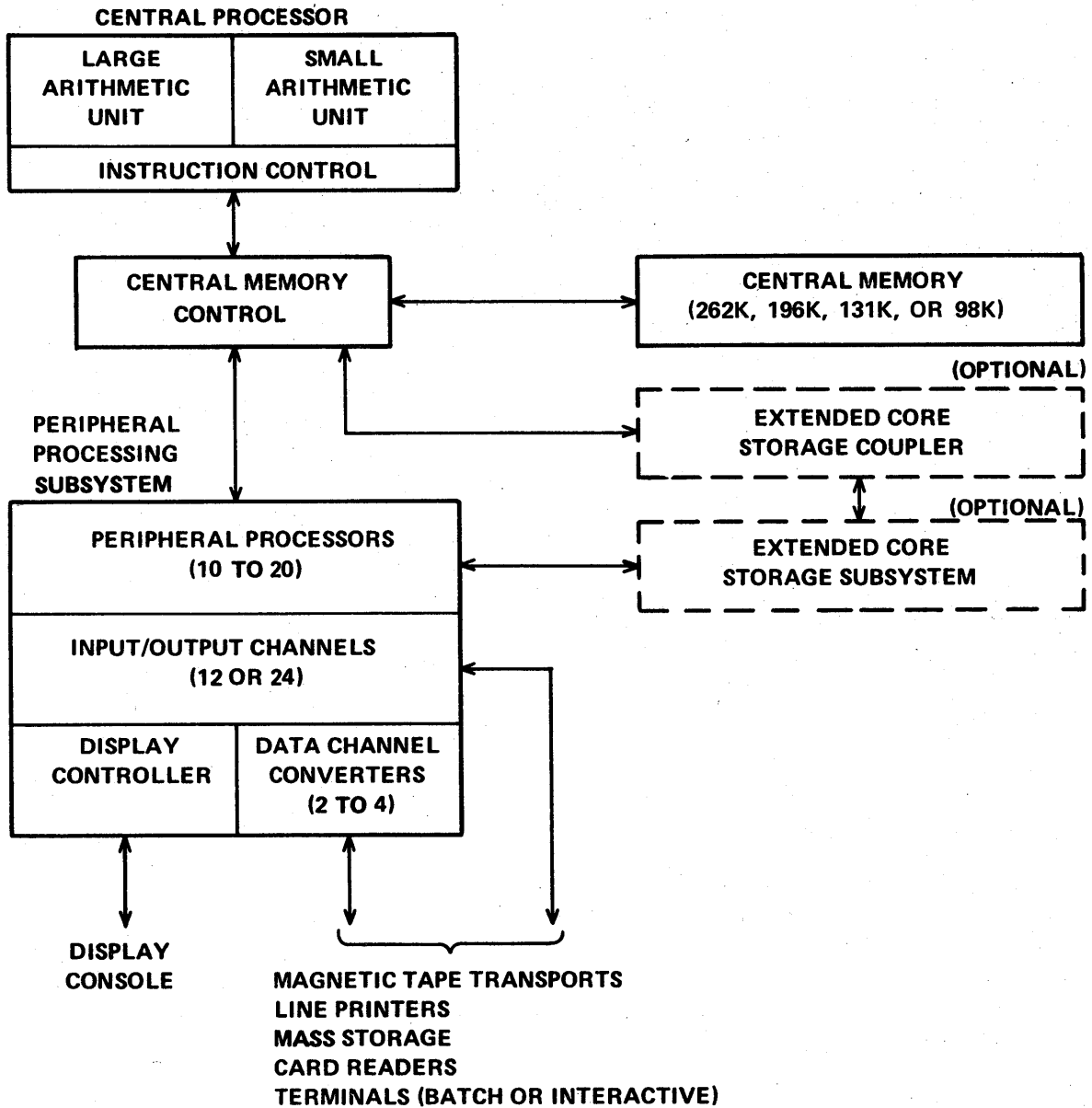
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits



CDC CYBER 170/Model 173 Computing System

- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory (MOS) of 4096 12-bit words plus one parity bit per word (odd parity)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel 2 MHz
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- MOS semiconductor memory model options listed with associated capacity of 60-bit words plus eight error corrections bits per word

Model 173-6	98,304 words
Model 173-8	131,072 words
Model 173-12	196,608 words
Model 173-16	262,144 words
- Memory organized in logically independent banks of words with corresponding multiphasing of at least 8 banks (maximum 16 banks).
- Transfer rate of up to one word each 50 nanoseconds in phase operation.

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates only with central memory via central memory control. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of a large and a small arithmetic unit and an instruction control unit. The arithmetic units contain all logic necessary to execute the arithmetic, manipulative, and logical operations. The instruction control unit directs the arithmetic operations and provides the interface between the arithmetic units and central memory control. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing.

The following are the general categories of central processor instructions:

- branch
- central exchange jump
- fixed point arithmetic

- floating-point arithmetic
- extended core storage communication
- increment
- logical
- monitor, stop
- move, compare data handling
- pass
- shift

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 or 20 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slow input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), 4096-word, random-access memory (independent of central memory) with a cycle time of 500 nanoseconds.

The following are the general categories of peripheral processor instructions:

- arithmetic
- central processor and central memory communications
- data transmission
- input/output
- logical
- no operation
- replace
- shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 173 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels, 12 or 24, depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity), and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controllers and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard (second display optionally available). The console performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

The data channel converters allow Control Data 3000 series peripheral equipment to be attached to the Model 173 input/output channels.

MODEL 173 MEMORY

The CDC CYBER 170/Model 173 computer offers a hierarchical memory concept:

- Central MOS memory
- Extended core storage (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering containing large data arrays, and to support job swapping and other operating system service functions. Extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 173 central memory is composed of banks of 60-bit words of metallic-oxide semiconductor storage. There are 8 error correction bits per 60-bit word. The complete cycle time for one bank is 400 nanoseconds (major cycle). The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds (minor cycle).

Central memory is available in sizes ranging from 65,536 words (eight banks) to 262,144 words (16 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory of which four are used for:

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, the error correction bits will correct all single-bit errors. Multiple-bit errors are detected, and their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of from 2 to 16 banks of 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262K words (two banks) to 2M words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices as a high-speed program swapping device for storage of large data arrays and for storage of frequently used programs and system routines. ECS is also used as a linking medium in multimainframe configurations (required for NOS MMF, as an alternative device for NOS/BE MMF).

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262K	2
524K	4
1048K	8
2097K	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data paths consist of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170/MODEL 174 COMPUTER SYSTEM

The Control Data CYBER 170/Model 174 computer system is a large-scale, general purpose digital computing system. Its dual central processor architecture permits network application or data processing and scientific computing applications with large data bases. In addition, this system is capable of multiprocessing, time-sharing, multiprogramming, real-time, and hybrid applications. Because of hardware and software compatibility among CDC CYBER 70 and CDC CYBER 170 systems, the Model 174 serves as an expandable base for long-range growth plans.

The Model 174 computing system has high-speed central processor units, accessible through a central memory control, and a group of 10 (20 optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processors. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor(s). They communicate with themselves through central memory and the input/output channels.

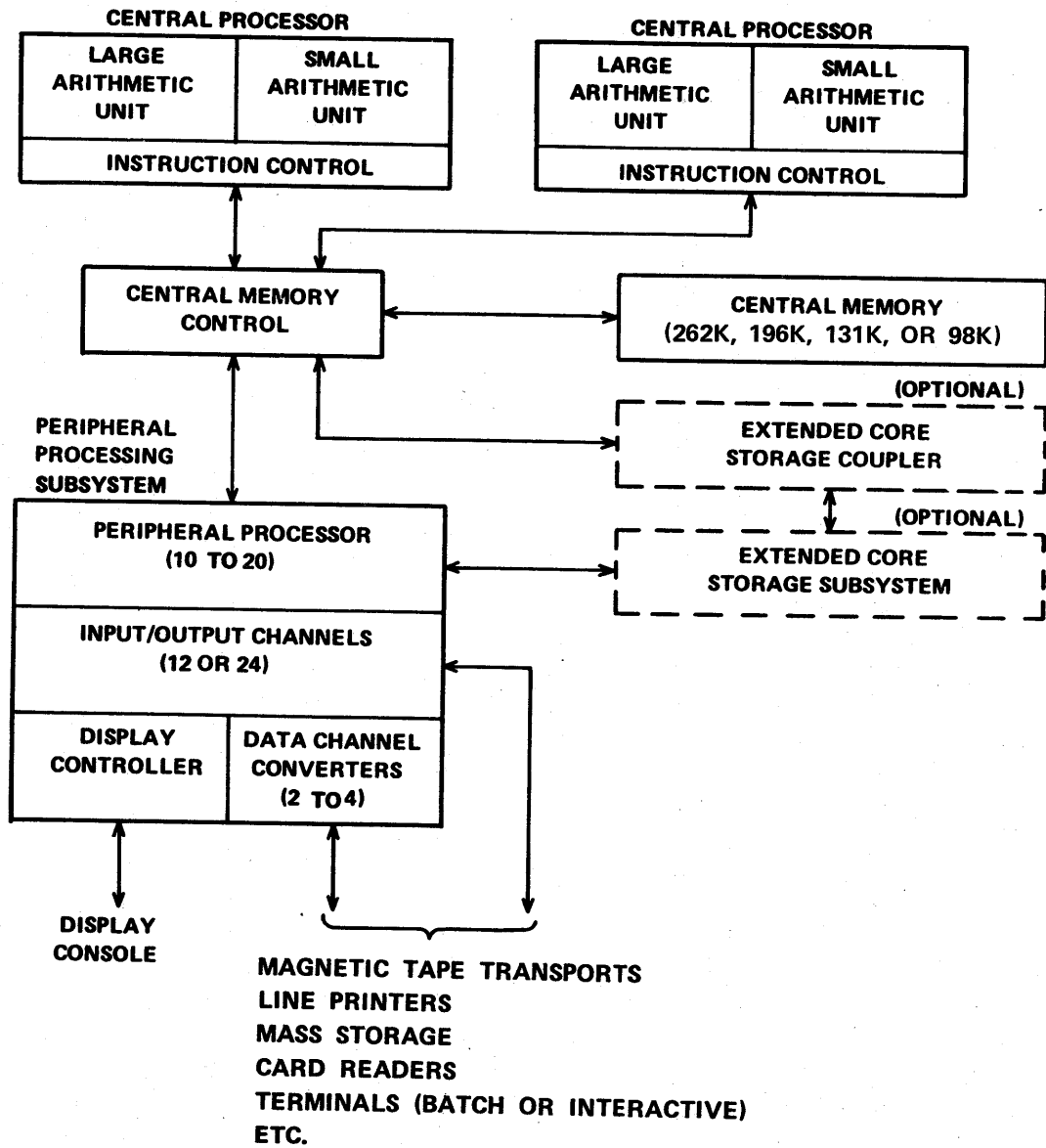
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems are normally in operation concurrently by time-sharing the central processors.

Central memory is organized into logically independent banks. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- Two independent central processors
- 24 operating registers (each CPU)
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Large and small arithmetic section for each CPU
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump



CDC CYBER 170/Model 174 Computing System

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory of 4096 12-bit words plus one parity bit per word (odd parity)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel 2 MHz
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitor error conditions (maintenance aid)

Central Memory Characteristics

- MOS semiconductor memory model options listed with associated capacity of 60-bit words plus eight error correction bits per word.

Model 174-6	98,344 words
Model 174-8	131,072 words
Model 174-12	196,608 words
Model 174-16	262,144 words
- Memory organized in logically independent banks of words with corresponding multi-phasing of at least 8 banks (maximum of 16 banks)
- Transfer rate of up to one word each 50 nanoseconds in phase operation

CENTRAL PROCESSORS

Each central processor is an independent high-speed arithmetic unit which communicates only with central memory via central memory control. They operate independent from the PPUs and are free to carry on computation unencumbered by input/output requirements. Each processor consists (functionally) of a large and a small arithmetic unit and an instruction control. They also perform instruction retrieving, address preparation, memory protection, and data retrieving and storing.

The following are the general categories of central processor instructions:

- branch
- central exchange jump
- fixed-point arithmetic

- floating-point arithmetic
- extended core storage communication
- increment
- logical
- monitor, stop
- move, compare data handling
- pass
- shift

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 or 20 units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processors.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), 4096 word random-access memory (independent of central memory) with a cycle time of 500 nanoseconds.

The following are the general categories of peripheral processor instructions:

- arithmetic
- central processor and central memory communications
- data transmission
- input/output
- logical
- no operation
- replace
- shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 174 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then located with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels (12 or 24) depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard (second display optionally available). The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

MODEL 174 MEMORY

The CDC CYBER 170/Model 174 computer offers a hierarchical memory concept:

- Central MOS memory
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Extended core storage is optional, and fully supported to increase system performance.

Central Memory

The Model 174 central memory is composed of banks of 60-bit words of metallic-oxide storage. There are also eight correction bits per 60-bit word. The complete cycle time for one bank is 400 nanoseconds (major cycle). The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds (minor cycle).

Central memory is available in sizes ranging from 65,536 words (eight banks) to 262,144 words (16 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory.

- Two central processor/central memory
- Extended core storage/central memory
- One or two groups of 10 peripheral processors/central memory

Central memory control provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, the error correction code corrects all single-bit errors. Multiple-bit errors are detected, and their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channel.

The extended core storage is composed of from 2 to 16 banks of 60-bit words of core storage. Each 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262K words (two banks) to 2M words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices as a high-speed program swapping device for storage of large data arrays and for storage of frequently used programs and system routines.



CYBER 170/MODEL 175-100 COMPUTER SYSTEM

The CDC CYBER 170/Model 175-100 computer system is a multipurpose computing system that provides real-time/time-critical network, commercial, data management, and scientific capabilities. This processor can serve as an entry level processor to the Model 175 family of computers that includes two other models. The Model 175-100 is field-upgradable to a Model 175-200 or Model 175-300 system.

The Model 175-100 computing system has one large central processor, accessible through central memory, and a group of 10 to 20 (optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor. Further concurrency is obtained with the central processor by parallel action of various functional units.

Central memory is organized into logically independent banks. Many banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

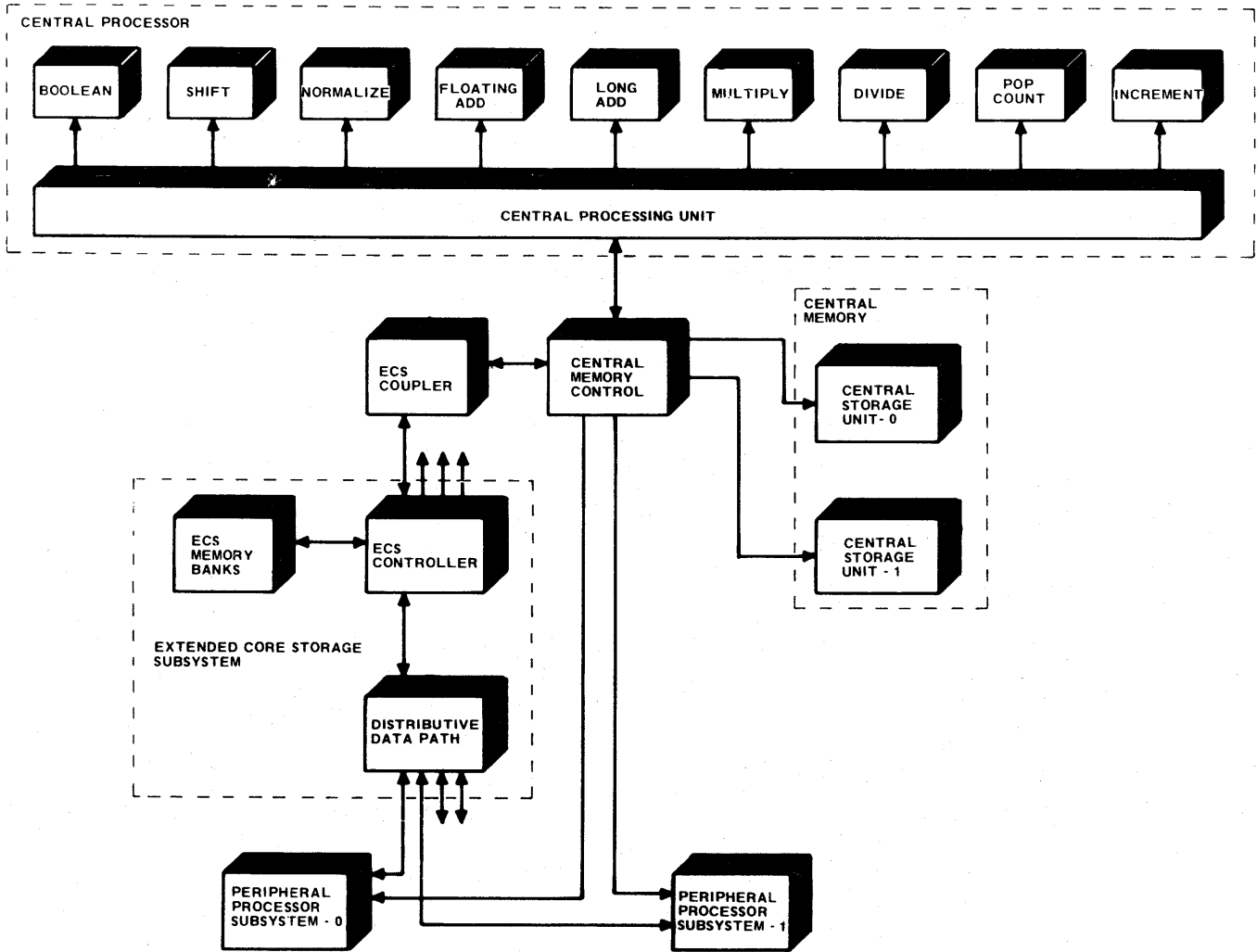
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Nine arithmetic functional units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to twelve 60-bit instruction words with 2-word look-ahead capability
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)



CDC CYBER 170/Model 175-100 Computing System

- Semiconductor memory of 4096 12-bit words plus one parity bit per word (odd parity)
- 12 or 24 input/output channels
 - all channels common to all processors
 - transfer rate per channel 2 MHz (optionally 1 MHz) in 12-bit words
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- Semiconductor memory model options listed with associated capacity of 60-bit words plus eight error correction bits per word.

Model 175-108	131,072 words
Model 175-112	196,608 words
Model 175-116	262,144 words
- Memory organized in logically independent banks of words with corresponding multiphasing of banks (maximum memory size of 16 banks)
- Transfer rate of up to one word each 50 nanoseconds in phased operation.

CENTRAL PROCESSOR

The central processor, composed of a central processing unit, nine separate functional units which operate in parallel, and the central memory control, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The nine arithmetic and logical units (floating add, normalize, long add, increment, shift, multiply, divide, boolean, and population count) execute the arithmetic, manipulative, and logical operations. The central memory control directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. The central memory control provides for orderly flow of data between central memory and the requesting elements of the system.

The following are the general categories of central processor instructions:

- branch
- central exchange jump
- fixed-point arithmetic
- floating-point arithmetic

- extended core storage communication
- increment
- logical
- monitor, stop
- pass
- shift

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computations while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), random-access memory (independent or central memory) with a cycle time of 500 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- arithmetic
- central processor and central memory communications
- data transmission
- input/output
- logical
- no operation
- replace
- shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 175-100 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jump are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

Two data channel converters, included with the system, allow Control Data 3000 series peripheral equipment to be attached to the Model 175-100 input/output channels.

MODEL 175 MEMORY

The CDC CYBER 170/Model 175-100 computer offers a hierarchical memory concept:

- Central core memory
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional, it is fully supported and recommended for optimum system performance.

Central Memory

The Model 175-100 central memory is composed of banks of 60-bit words of metallic-oxide storage. There are also eight error correction bits per 60-bit word. The complete cycle time for one bank is 400 nanoseconds. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds.

Central memory is available in sizes ranging from 131,072 words (8 banks) to 262,144 words (16 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are four access paths to central memory.

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control (part of central processor) provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, manipulation of the error correction bits allows a 60-bit word containing a single-bit error to be corrected. Multiple-bit errors are not corrected, but their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170/MODEL 175-200 COMPUTER SYSTEM

The CDC CYBER 170/Model 175-200 computer system is a multipurpose computing system that provides real-time/time-critical network, commercial, data management, and scientific capabilities. This model is field-upgradable to a Model 175-300 system.

The Model 175-200 computing system has one large central processor, accessible through central memory, and a group of 10 to 20 (optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor. Further concurrency is obtained with the central processor by parallel action of various functional units.

Central memory is organized into logically independent banks. Many banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

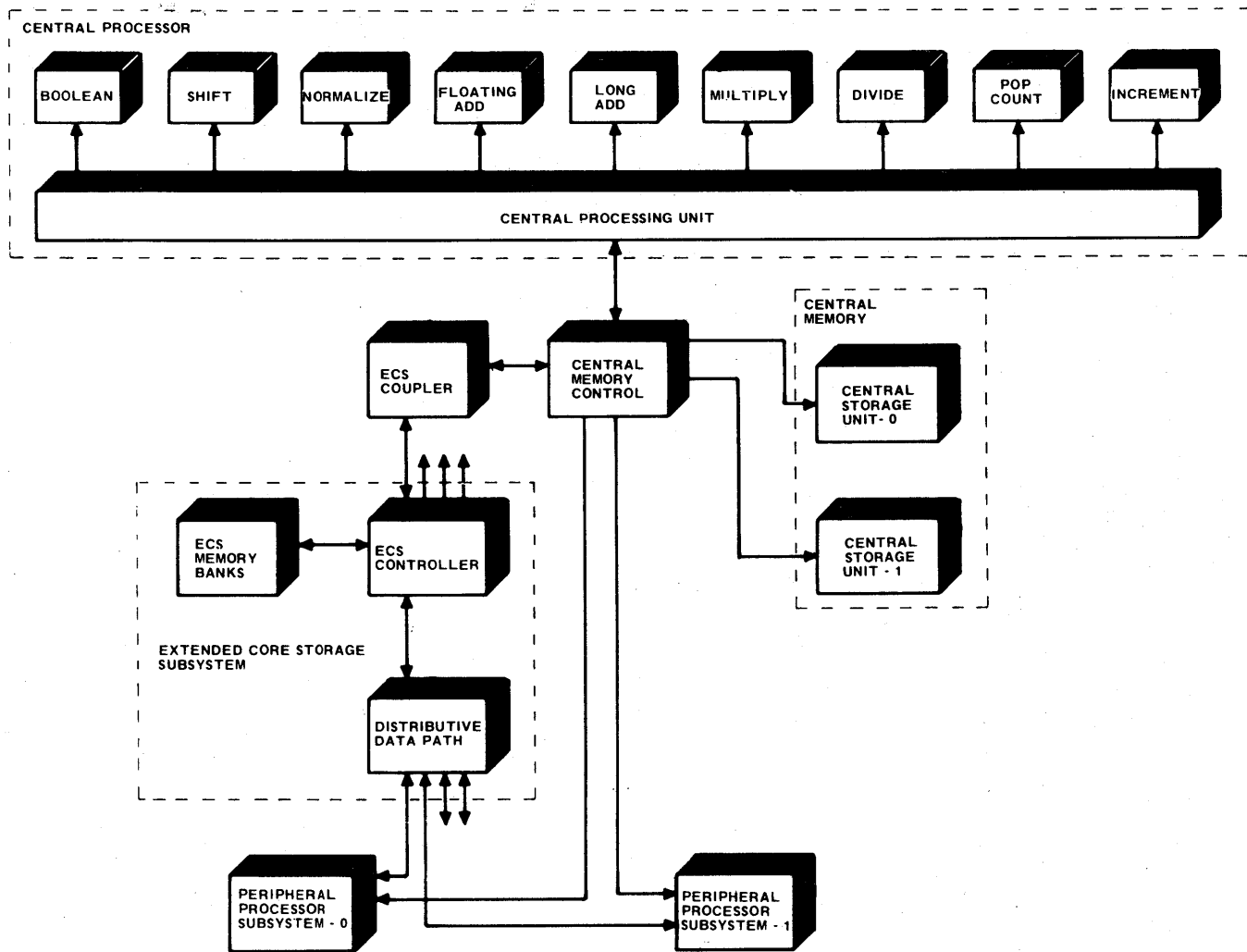
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Nine arithmetic functional units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to twelve 60-bit instruction words with 2-word look-ahead capability
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)



CDC CYBER 170/Model 175-200 Computing System

- Semiconductor memory of 4096 12-bit words plus one parity bit per word (odd parity)
- 12 or 24 input/output channels
 - all channels common to all processors
 - transfer rate per channel 2 MHz (optionally 1 MHz) in 12-bit words
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- Semiconductor memory model options listed with associated capacity of 60-bit words plus eight error correction bits per word.

Model 175-208	131,072 words
Model 175-212	196,608 words
Model 175-216	262,144 words
- Memory organized in logically independent banks of words with corresponding multiphasing of banks (maximum memory size of 16 banks)
- Transfer rate of up to one word each 50 nanoseconds in phased operation.

CENTRAL PROCESSOR

The central processor, composed of a central processing unit, nine separate functional units which operate in parallel, and the central memory control, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The nine arithmetic and logical units (floating add, normalize, long add, increment, shift, multiply, divide, boolean, and population count) execute the arithmetic, manipulative, and logical operations. The central memory control directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. The central memory control provides for orderly flow of data between central memory and the requesting elements of the system.

The following are the general categories of central processor instructions:

- branch
- central exchange jump
- fixed-point arithmetic
- floating-point arithmetic

- extended core storage communication
- increment
- logical
- monitor, stop
- pass
- shift

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computations while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), random-access memory (independent or central memory) with a cycle time of 500 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- arithmetic
- central processor and central memory communications
- data transmission
- input/output
- logical
- no operation
- replace
- shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 175-200 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

Two data channel converters, included with the system, allow Control Data 3000 series peripheral equipment to be attached to the Model 175-200 input/output channels.

MODEL 175 MEMORY

The CDC CYBER 170/Model 175-200 computer offers a hierarchical memory concept:

- Central core memory
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional, it is fully supported and recommended for optimum system performance.

Central Memory

The Model 175-200 central memory is composed of banks of 60-bit words of metallic-oxide storage. There are also eight error correction bits per 60-bit word. The complete cycle time for one bank is 400 nanoseconds. The cycle time is field-upgradable to 300 nanoseconds. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds.

Central memory is available in sizes ranging from 131,072 words (8 banks) to 262,144 words (16 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are four access paths to central memory.

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control (part of central processor) provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, manipulation of the error correction bits allows a 60-bit word containing a single-bit error to be corrected. Multiple-bit errors are not corrected, but their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170/MODEL 175-300 COMPUTER SYSTEM

The CDC CYBER 170/Model 175-300 computer system is a multipurpose computing system that provides real-time/time-critical network, commercial, data management, and scientific capabilities. The Model 175-300 is the upper member of the Model 175 computer family.

The Model 175-300 computing system has one large central processor, accessible through central memory, and a group of 10 to 20 (optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor. Further concurrency is obtained with the central processor by parallel action of various functional units.

Central memory is organized into logically independent banks. Many banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

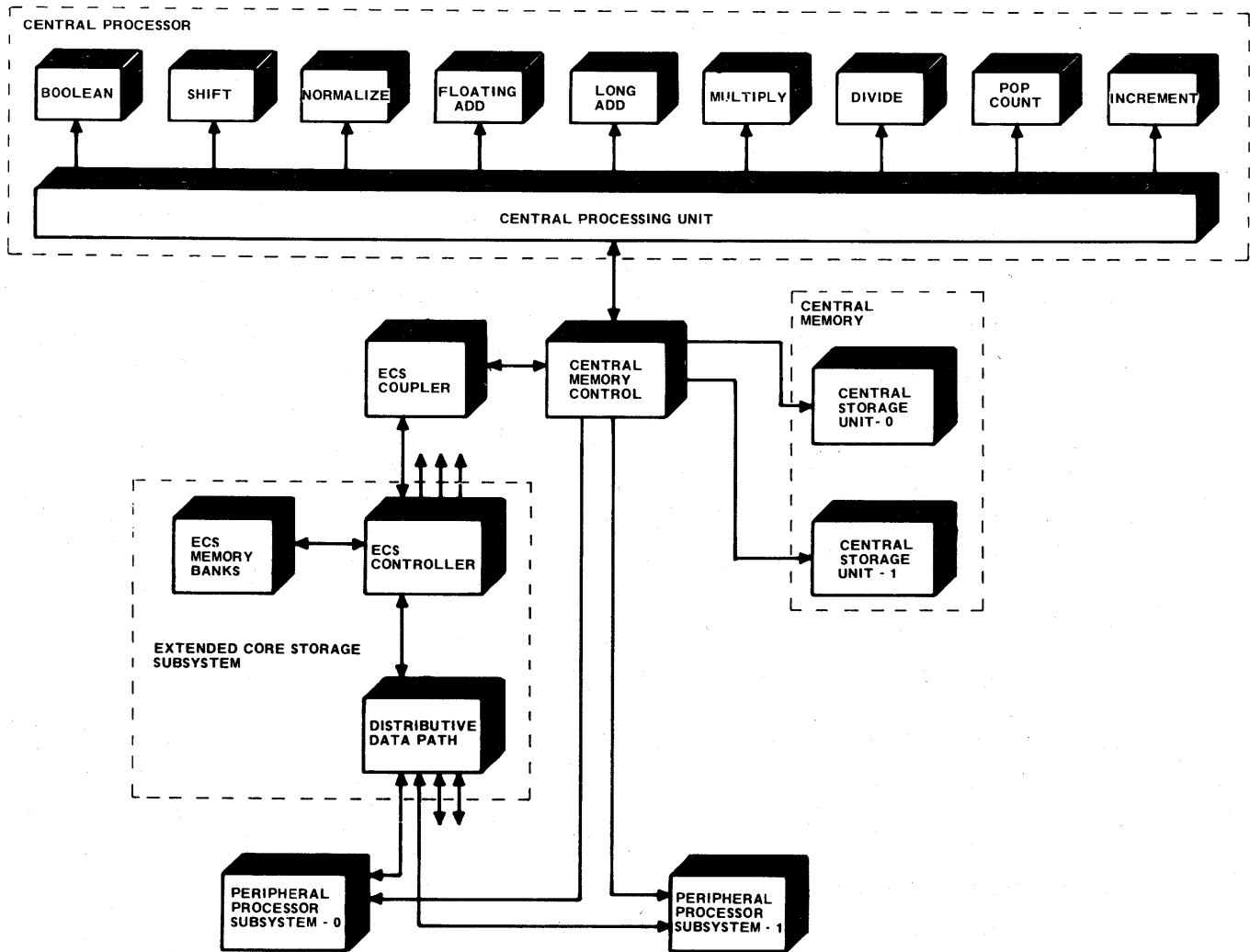
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Nine arithmetic functional units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to twelve 60-bit instruction words with 2-word look-ahead capability
- Computation in floating-point and fixed point, single and double precision
- Integer multiply
- Central exchange jump

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory of 4096 12-bit words plus one parity bit per word (odd parity)



CDC CYBER 170/Model 175-300 Computing System

- 12 or 24 input/output channels
 - all channels common to all processors
 - transfer rate per channel 2MHz (optionally 1 MHz) in 12-bit words
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- Semiconductor memory model options listed with associated capacity of 60-bit words plus eight error correction bits per word.

Model 175-308	131,072 words
Model 175-312	196,608 words
Model 175-316	262,144 words
- Memory organized in logically independent banks of words with corresponding multiphasing of banks (maximum memory size of 16 banks)
- Transfer rate of up to one word each 50 nanoseconds in phased operation.

CENTRAL PROCESSOR

The central processor, composed of a central processing unit, nine separate functional units which operate in parallel, and the central memory control, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The nine arithmetic and logical units (floating add, normalize, long add, increment, shift, multiply, divide, boolean, and population count) execute the arithmetic, manipulative, and logical operations. The central memory control directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. The central memory control provides for orderly flow of data between central memory and the requesting elements of the system.

The following are the general categories of central processor instructions:

- branch
- central exchange jump
- fixed-point arithmetic
- floating-point arithmetic
- extended core storage communication
- increment
- logical
- monitor, stop
- pass
- shift

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computations while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), random-access memory (independent or central memory) with a cycle time of 500 nanoseconds. Execution time of processor instructions is dependent on memory cycle time.

The following are the general categories of peripheral processor instructions:

- arithmetic
- central processor and central memory communications
- data transmission
- input/output
- logical
- no operation
- replace
- shift

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 175-300 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use

a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converters

Two data channel converters, included with the system, allow Control Data 3000 series peripheral equipment to be attached to the Model 175-300 input/output channels.

MODEL 175 MEMORY

The CDC CYBER 170/Model 175-300 computer offers a hierarchical memory concept:

- Central core memory
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for input/output buffering, containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional, it is fully supported and recommended for optimum system performance.

Central Memory

The Model 175-300 memory is composed of banks of 60-bit words of metallic-oxide storage. There are also eight error correction bits per 60-bit word. The complete cycle time for one bank is 300 nanoseconds. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is one 60-bit word per 50 nanoseconds.

Central memory is available in sizes ranging from 131,072 words (eight banks) to 262,144 words (16 banks). The large number of banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are four access paths to central memory.

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control (part of central processor) provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 60-bit word placed in central memory. When a word is read from central memory, manipulation of the error correction bits allows a 60-bit word containing a single-bit error to be corrected. Multiple-bit errors are not corrected, but their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 131,072 60-bit words of core storage. Eight 60-bit words are contained in a 488-bit physical extended core storage word. Each 60-bit word has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 8-word records come from different banks. This phasing, combined with the wide (8-word) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of one 60-bit word per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 262,144 words (two banks) to 2,097,152 words (16 banks). The very fast transfer rates and short access time of extended core storage makes it ideal for use as a buffer between central memory and rotating mass storage devices, such as a high-speed program swapping device; for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (60-bit words)</u>	<u>Number of Banks</u>
262,144	2
524,288	4
1,048,576	8
2,097,152	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CDC CYBER 170/MODEL 176 COMPUTER SYSTEM

The CONTROL DATA CYBER 170/Model 176 Computer System is a multipurpose system which provides real-time/time-critical network, commercial, data management, and scientific capabilities.

The CDC CYBER 176 provides computational capacity generally ascribed to super-scale computer systems. It also provides the diversity, and comprehensiveness of software associated with CONTROL DATA'S CYBER 170 product line.

The Model 176 Computing System has a single high-performance central processor, plus a supporting hierarchy of memory and peripheral processors. The CDC CYBER 176 central memory consists of high-performance, bipolar semiconductor memory, ranging from 131,072 to 262,144 60-bit words. It can be complemented by an optional extended memory subsystem of 524 thousand, one or two million words. The CDC CYBER 176 has a set of 11, 14, 17 or 20 CDC CYBER 170 peripheral processors, each with a separate memory which can operate independent of each other and of the central processor. Used to drive peripheral equipments, they also perform many functions of the system software.

The CDC CYBER 176 can also include, as optional equipment, additional high-performance peripheral processors, unique to that model whose function is solely to support high-speed rotating mass storage.

Concurrency of operation among the various components described above, as well as within the central processor itself, complement the basic high-speed circuitry of the system to achieve high levels of processing power.

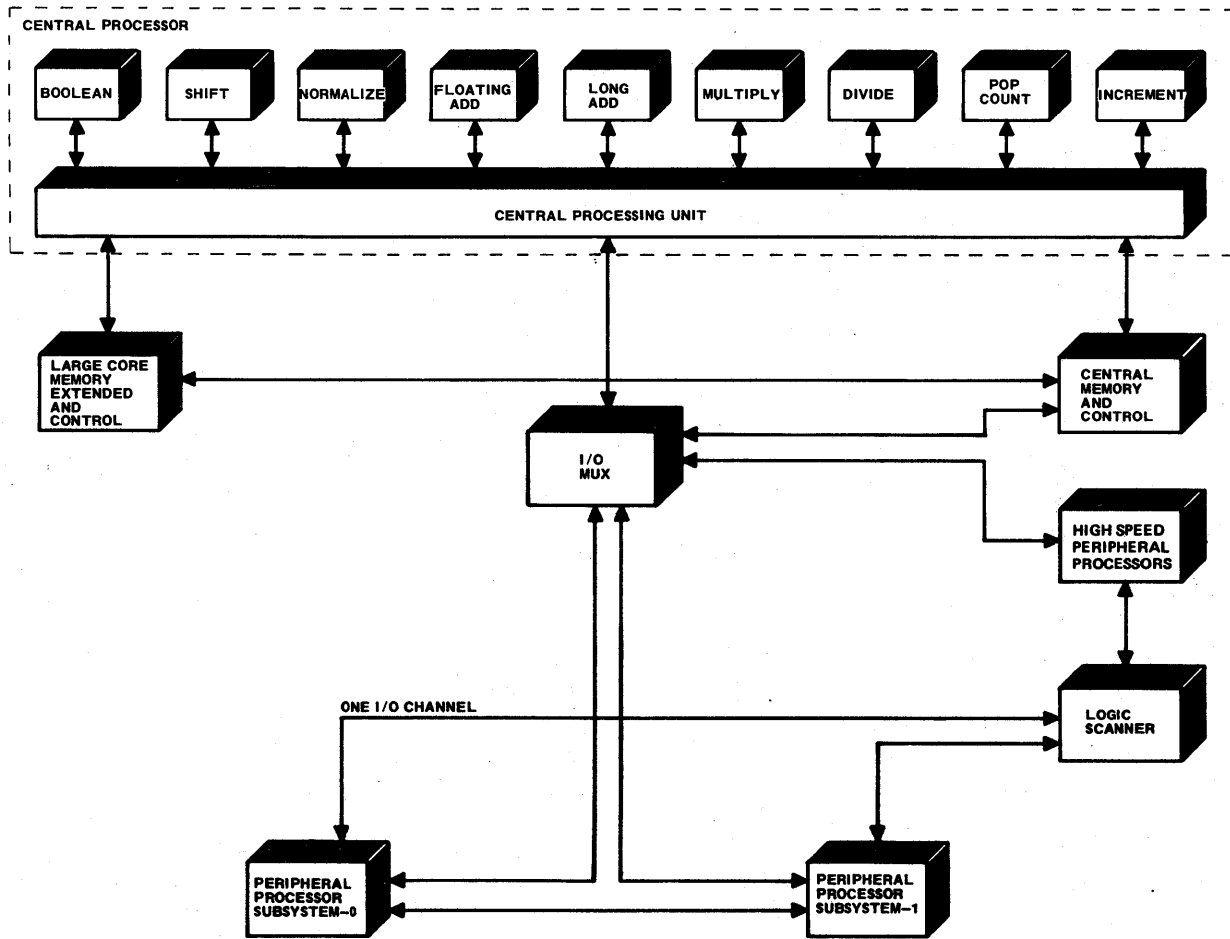
SYSTEM CHARACTERISTICS

Central Processing Characteristics

- Nine independent functional units
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack holding up to 48 instructions
- Computation in floating-point and fixed point, single and double precisions
- Real time clock
- Exchange jump/interrupt control mechanism

Central Memory Characteristics

- Bipolar semiconductor memory in options of 131K, 196K, or 262K 60-bit words
- Transfer rate of one 60-bit word per 27.5 nanoseconds
- Single-Error Correction/Double-Error Detection (SEC/DED) logic



CDC CYBER 170/MODEL 176 Computer System

Extended Memory Characteristics

- 0.5M, 1M, or 2M 60-bit word options
- Single-Error Correction/Double-Error Detection (SECCDED) logic
- 18 or 36 million word-per-second bandwidth to/from central memory (one or two million word configuration required for 36 million rate)
- Direct access capability to/from CPU operand registers

CDC CYBER 170 Peripheral Processor Subsystem Characteristics

- Emitter Coupled Logic (ECL) integrated circuits
- 10, 14, 17, 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory of 4096 12-bit words plus parity
- 12 or 24 input/output channels
 - All channels common to all CYBER 170 Peripheral Processors
 - Maximum transfer rate of 2MHz per channel
 - All channels can be active simultaneously
 - All channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- 64 instruction repertoire
- System status and control register

CDC CYBER 176 Input/Output Subsystem Characteristics

- Input/Output multiplexer with 180 million characters-per-second total bandwidth
 - One or two 60-bit ports for CDC CYBER 170 peripheral processor subsystems
 - 4-14 12-bit data channels for high speed peripheral devices
 - Central processor interrupt system

CDC CYBER 176 High-Speed Peripheral Processor Characteristics

- 4096 12-bit words of coincident current memory per processor (plus one parity bit word)
- Eight full-duplex I/O channels per processor
- 27.5 nanosecond cycle time

CENTRAL PROCESSOR

The two primary architectural elements contributing to the speed of the Model 176 central processor are the twelve-word instruction stack and its nine independent functional units. Instruction words are read from central memory into the stack on a two word look-ahead basis. After the 15 or 30-bit instructions are issued for execution, they flow through the remaining ten words of the instruction stack. If the program sequence is such that any of these instructions still in the stack are to be re-executed, they are issued from the stack; thereby, reducing access time and minimizing memory references. Each word in the stack has an associated address register. This allows non-contiguous sequences to operate totally from the instruction stack.

Instructions are issued from the stack at a maximum rate of one per system clock period (27.5 ns) to any of the nine functional units. Each functional unit is independent of the others, permitting significant parallelism of operation to occur. Additionally, most functional units are segmented (or pipelined) so that while it may require several clock periods for a given operation to complete, several operations can be at various stages of completion within a single functional unit.

Operating totally within the environment of central and the optional extended memory, and with the peripheral processors supporting much of the system control as well as I/O functions, the central processor's application to productive computation is maximized.

Effective and efficient scheduling of the central processor for multiprogramming is accomplished through the exchange jump mechanism. Through a simple and rapid sequence, this mechanism temporarily stores the operating registers and conditions of one program, while loading and activating comparable elements for a different program sequence.

CDC CYBER 176 MEMORY

The CDC CYBER 176 offers a hierarchical memory subsystem to efficiently support the computational power of the central processor.

The CDC CYBER 176 central memory is a bipolar, semiconductor memory which provides fast, random storage for executing programs and data. Built in individual memory banks of 4096 words, it is organized in a 16-level, interleaved manner so that references to successive addresses do not reference the same bank. This interleaving, plus the short bank cycle time of 82.5 nanoseconds for read operations and 165 nanoseconds for write operations, minimize any degradation from memory access conflict. The memory transfer rate is capable of supplying the central processor a 60-bit word every minor cycle (27.5 ns).

In addition to having high basic reliability characteristics, the memory has eight error-correction bits associated with each word, which permit the subsystem to correct any single-bit errors.

Central memory can be supplemented by the CDC CYBER 176 extended memory described below. A critical feature of the relationship between central and extended memory is the transfer rate of up to 363-million characters per second between memories for block-transfer operations.

The peripheral processors of the system also have access to central memory through the I/O multiplexer. The CDC CYBER 170 peripheral processors can reference any location in memory, while the optional CDC CYBER 176 peripheral processor, through the CDC CYBER 176 channels of the I/O multiplexer, transfer I/O data through pre-assigned circular buffer areas in central memory.

THE OPTIONALLY AVAILABLE CDC CYBER 176 EXTENDED MEMORY

Extended memory provides additional storage at memory-level speeds, supporting the central processor and central memory subsystem.

It is composed of ferrite core memory with integrated-circuit control logic. Extended memory also features SEC/DED logic to augment subsystem reliability. The basic CDC CYBER 176 extended memory subsystem option is 0.5M 60-bit

words. Field upgrades are available to provide 1M or 2M word configurations.

Structurally, the memory is composed of individual banks of 262,144 60-bit words. Each bank has a 16-word holding register which contains a 960-bit extended memory word from the latest bank reference. The banks are interleaved to maximize transfer rates and minimize bank conflicts. Bank cycle time for the extended memory is 1.76 microseconds. Block transfer operations between central and extended memories can achieve a data rate of one 60-bit word per 27.5 nanoseconds, on one- and two-million word configurations. On half a million word extended memory system the maximum data rate is on 60-bit word per 55 nanoseconds.

The CDC CYBER 176 central processor can also directly reference elements in extended memory on a single word basis.

CDC CYBER 176 INPUT/OUTPUT MULTIPLEXER

The I/O multiplexer is a critical element in the CDC CYBER 176 architecture which provides for the high transfer rate and wide bandwidth required to complement the system's computational power. Total bandwidth of the I/O multiplexer permits it to transfer a 60-bit word to or from central memory every 55 nanoseconds at a rate of approximately 180-million characters per second. The I/O multiplexer's basic structure permits data to be received from, or sent to, peripheral processors over 12-bit channels. The multiplexer forms the 12-bit words from the data channels into 60-bit words and transfers them into pre-assigned central memory circular buffer areas, which are managed by the central processor through an interrupt system. Up to 14 data channels can be incorporated into the I/O multiplexer. The basic CDC CYBER 176 option includes four channels, structured as channel pairs to support the 7639/819 Mass Storage Subsystem. Additional channels can be added on an individual basis.

CDC CYBER 176 HIGH-SPEED PERIPHERAL PROCESSOR UNITS

These peripheral processors provide support for peripheral devices which have high streaming rates. Dedicated solely to that function, their role is to support the 7639/819 Mass Storage Subsystem. The CDC CYBER 176 configuration optionally includes four CDC CYBER 176 high speed peripheral processors -- enough to sustain two parallel data streams from a mass storage subsystem. Each peripheral processor unit is an independent entity with a processing unit, 4096 12-bit words of memory, and its own set of data channels. Providing the link between the 7639 Controller and the high-speed channels of the I/O multiplexer, two CDC CYBER 176 peripheral processors work together in a master-slave arrangement to sustain data transfers to and from the 7639/819 Subsystem. Additional processors can be added on an individual basis.

CDC CYBER 176 PERIPHERAL PROCESSOR SUBSYSTEM

The CDC CYBER 176 includes the CDC CYBER 170 Peripheral Processor Subsystem (PPS) - an element common to all CDC CYBER 170 systems - consisting of 10 functionally independent processors and 12 bidirectional I/O channels. The PPS provides linkage to all peripheral equipments except the 7639/819 Subsystem, and has the ability to perform many of the operating system tasks. Having its own 50-nanosecond clock, the PPS is organized in a multiplexing system which allows the peripheral processors to share common circuitry for arithmetic, logical, and I/O operations. The individual peripheral processor can operate with a basic cycle time of either 500 or 1000 nanoseconds, and each has a 4,096 12-bit word MOS memory. CDC CYBER 170 peripheral processors can access any location in central memory.

The Peripheral Processor Subsystem interfaces central memory through port zero of the I/O multiplexer. Twelve data channels, accessible by any of the CDC CYBER 170 peripheral processors, are included in the subsystem. Up to eight different peripheral equipments can be connected to each of these data channels, each of which can transmit data at a rate up to one 12-bit word every 500 nanoseconds. An optional peripheral processor subsystem can be added to utilize port one of the I/O multiplexer. This option adds 12 data channels and 4, 7, or 10 additional peripheral processors.

CDC CYBER 170 peripheral processors perform a major portion of the operating system functions in addition to their role in supporting I/O operations, when operating under control of a single CDC CYBER 170 operating system.

DATA CHANNEL CONVERTERS

One data channel converter included with the system, allows Control Data 3000 series peripheral equipment to be attached to the Model 176 input/output channels.

MODELS

<u>Model</u>	<u>Central Memory/Core Memory</u>
176-8	131,072
176-12	196,608
176-16	262,144
176-21†	131,072/524,288
176-22†	131,072/1,048,576
176-24†	131,072/2,097,152
176-31†	196,608/524,288
176-32†	196,608/1,048,576
176-34†	196,608/2,097,152
176-41†	262,144/524,288
176-42†	262,144/1,048,576
176-44†	262,144/2,097,152

† Also includes four bidirectional input/output channels with assembly and disassembly logic, and four high-speed peripheral processors.



CYBER 170 SERIES 700 COMPUTER SYSTEM MODEL 720

The CDC CYBER 170 Model 720 computer system is a general-purpose digital computing system which can be used as the nucleus for networks of interactive and remote batch terminals and as a central computer for batch operations. It is suitable in data management, scientific computation, commercial data processing and data communications. Because of hardware and software compatibility among the CDC CYBER 170 systems, the CDC CYBER 170 Model 720 serves as an expandable base for long-range growth plans. This model is field-upgradable to a CDC CYBER 170 Model 730, 750, or 760.

The Model 720 computing system has a high-speed Central Processor Unit, with an option of adding a second CPU, accessible through central memory and a group (or as options) of 10, 14, 17, or 20 peripheral processors known as the Peripheral Processor Subsystem (PPS). A second central processor is available as an option. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

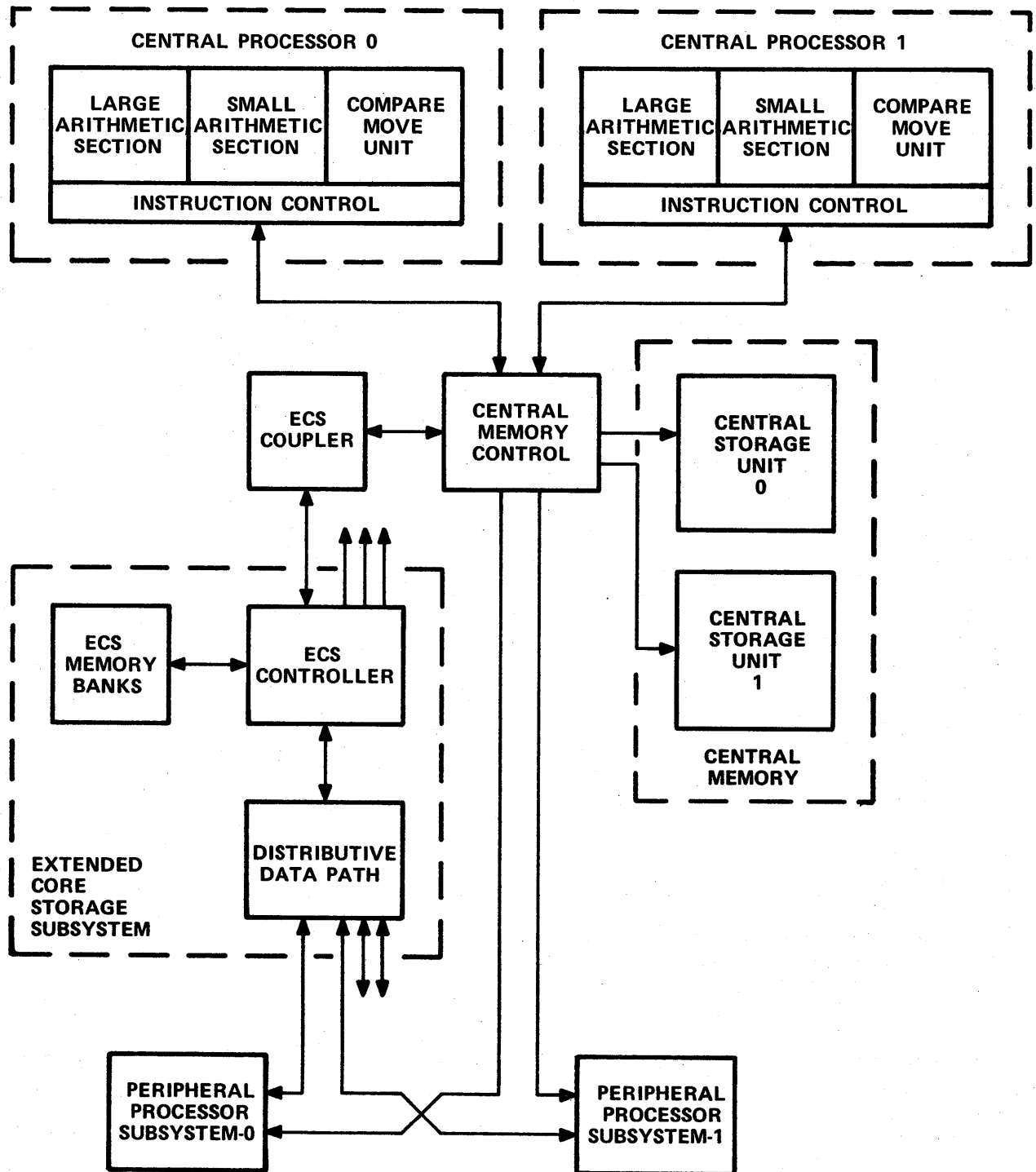
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor.

Central memory is organized into logically independent banks. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Uses Emitter-Coupled Logic (ECL) integrated circuits
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions



CDC CYBER 170/Model 720 Computing System

Peripheral Processor Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10, 14, 17, 20 peripheral processor configuration (characteristics as listed are per processor)
- Semiconductor memory (MOS) of 4,096 12-bit words plus one parity bit per word (odd parity)
- 12 (or 24 with option) input/output channels
 - All channels common to all processors of the PPS
 - Maximum transfer rate per channel is 2 MHz
 - All channels can be active simultaneously
 - All channels 12-bit plus parity (odd) bidirectional
- Cycle time of 500 nanoseconds
- Computation in fixed-point
- Time-shared access to central memory
- Status and control register monitors error conditions (maintenance aid)
- 64-instruction repertoire

Central Memory Characteristics

- Metal oxide semiconductor memory (4K chips) arranged in 60-bit (10-character) words plus eight error correction bits per memory word
- Memory organized in logically independent banks of words with corresponding multiphasing of 8 banks
- Transfer rate of up to one word each 50 nanoseconds
- Minimum size of .98M characters, optionally expandable to 1.3M, 1.9M, or 2.6M characters

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates with central memory via central memory control. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of a large and a small arithmetic unit and an instruction control unit. The arithmetic units contain all logic necessary to execute the arithmetic, manipulative, and logical operations. The instruction control unit directs the arithmetic operations and provides the interface between the arithmetic units and central memory control. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. A second central processor is available as an option.

PERIPHERAL PROCESSOR SUBSYSTEM

The peripheral processor subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), 4,096-word, random-access memory (independent of central memory) with a cycle time of 500 ns.

A second peripheral processor subsystem may be added to the system with incremental options of 4, 7, or 10 processors.

Exchange Jump

The exchange jump interrupts an executing central processor, saves a "snapshot" status of the program (registers, etc.), and initiates execution of another program. The Model 720 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other via the independent, bidirectional input/output channels. All 12 channels (24 with option) are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can utilize a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel (magnetic tape controllers, card read/punch controller, etc.).

Display Controller/Console

The display controller provides digital and analog information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- Display of all registers
- Dynamic display of any word or block of words in central memory
- Execute single steps of a program

Data Channel Converter

The data channel converter allows Control Data 3000 series peripheral equipment to be attached to the Model 720 input/output channels.

MODEL 720 MEMORY

The CDC CYBER 170 Model 720 computer offers a hierarchical memory concept:

- Central MOS memory 4K chip
- Extended core storage (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for containing large data arrays, and to support job swapping and other operating system service functions. Extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 720 central memory is composed of banks of 60-bit (10-character) words of Metallic-Oxide Semiconductor storage. There are also eight error correction bits per 10 characters. The complete cycle time for one bank is 400 nanoseconds (major cycle). The banks are phased so that successive addresses are in different banks, to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is 10 6-bit characters per 50 nanoseconds (minor cycle).

Central memory is available in sizes ranging from .98M characters to 2.6M characters. The eight banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are three access paths to central memory:

- Central processor/central memory
- Extended core storage/central memory
- Group of peripheral processors/central memory

Central memory control provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the error correction bits for each 10 characters placed in central memory. When 10 characters are read from central memory, the error correction bits will correct all single-bit errors. Multiple-bit errors are detected and their occurrence is noted in the status and control register.

Extended Core Storage (Optional)

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of from 2 to 16 banks of 60-bit (10-character) words of core storage. Eight 60-bit (10-character) words are contained in a 488-bit physical extended core storage word. Each 10 characters has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 80-character records come from different banks. This phasing, combined with the wide (80-character) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of 10 6-bit characters per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 2.6M characters (two banks) to 20M characters (16 banks). The very fast transfer rates and short access time of extended core storage make it ideal for use as a high-speed program swapping device for storage of large data arrays and for storage of frequently used programs and system routines. ECS is also used as a linking medium in multimainframe configurations for NOS MMF.

Extended Core Storage Options

<u>ECS Size (6-bit characters)</u>	<u>Number of Banks</u>
2.62M	2
5.24M	4
10.48M	8
20.97M	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage via the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage via an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170 SERIES 700 COMPUTER SYSTEM MODEL 730

The CDC CYBER 170 Model 730 computer system is a general-purpose digital computing system which can be used as the nucleus for networks of interactive and remote batch terminals and as a central computer for batch operations. It is suitable for data management, scientific computation, commercial data processing, and data communications. Because of hardware and software compatibility among CDC CYBER 70 and CDC CYBER 170 systems, the CDC CYBER 170 Model 730 serves as an expandable base for long-range growth plans. This model is field-upgradable to a CDC CYBER 170 Model 750 or 760 computer system.

The Model 730 computing system has a high-speed Central Processor Unit (two CPUs are optional), accessible through a central memory control, and a group of 10 (up to 20 optional) peripheral processors known as the Peripheral Processor Subsystem. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

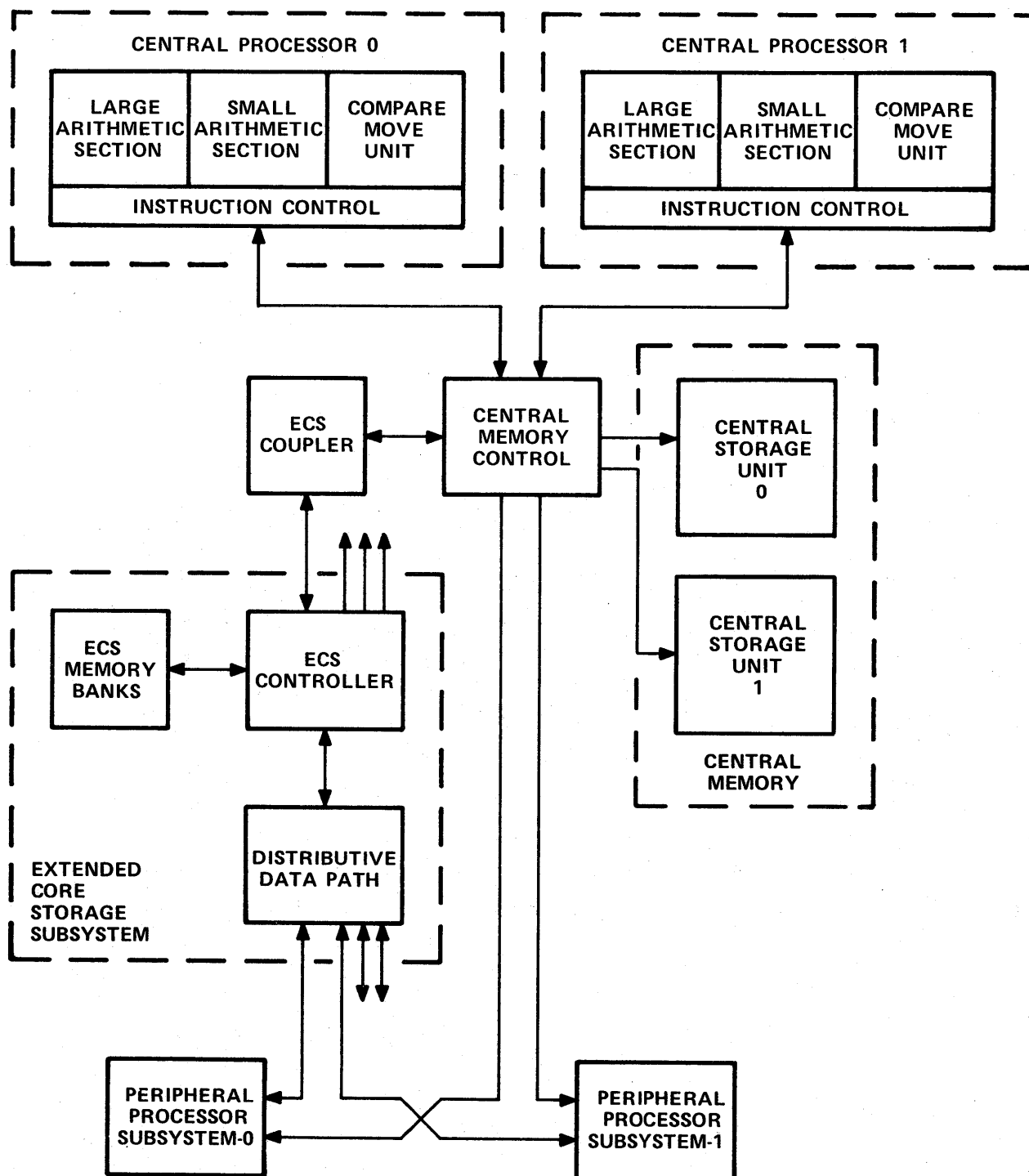
In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor.

Central memory is organized into logically independent banks. Banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump
- Move and compare instructions



CDC CYBER 170/Model 730 Computing System

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory (MOS) of 4,096 12-bit words plus one parity bit per word (odd parity)
- 12 or 24 input/output channels
 - all channels common to all processors
 - maximum transfer rate per channel 2 MHz
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- MOS semiconductor memory (4K chips) arranged in options listed with associated capacity of 60-bit (10-character) words plus eight error correction bits per 10 characters.
- Memory organized in logically independent banks of words with corresponding multiphasing of eight banks.
- Transfer rate of up to 10 characters each 50 nanoseconds in phase operation.
- Minimum size of 1.3M characters, optionally expandable to 1.9M or 2.6M characters.

CENTRAL PROCESSOR

The central processor is a high-speed arithmetic unit which communicates only with central memory via central memory control. It operates independent from the peripheral processors and is free to carry on computation unencumbered by input/output requirements. This processor consists (functionally) of a large and a small arithmetic unit and an instruction control unit. The arithmetic units contain all logic necessary to execute the arithmetic, manipulative, and logical operations. The instruction control unit directs the arithmetic operations and provides the interface between the arithmetic units and central memory control. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. An optional second processor is available on the Model 730.

PERIPHERAL PROCESSING SUBSYSTEM

The peripheral processing subsystem consists of 10 or 20 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slow input/output and supervisory operations.

Each processor has a 12-bit (plus one parity bit), 4,096-word, random-access memory (independent of central memory) with a cycle time of 500 nanoseconds.

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 730 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels, 12 or 24, depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity), and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controllers and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converter

The data channel converter allows Control Data 3000 series peripheral equipment to be attached to the Model 730 input/output channels.

MODEL 730 MEMORY

The CDC CYBER 170 Model 730 computer offers a hierarchical memory concept:

- Central MOS memory 4K chip
- Extended core storage (optional)

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for containing large data arrays, and to support job swapping and other operating system service functions. Extended core storage is optional and fully supported to increase system performance.

Central Memory

The Model 730 central memory is composed of banks of 60-bit (10-character) words of metallic-oxide semiconductor storage. There are 8 error correction bits per 10 characters. The complete cycle time for one bank is 400 nanoseconds (major cycle). The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is 10 6-bit characters per 50 nanoseconds (minor cycle).

Central memory is available in sizes ranging from 1.3M characters to 2.6M characters. The eight banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are five access paths to central memory of which four are used for:

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 10 characters placed in central memory. When 10 characters are read from central memory, the error correction bits will correct all single-bit errors. Multiple-bit errors are detected, and their occurrence is noted in the status and control register.

Extended Core Storage (Optional)

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of from 2 to 16 banks of 60-bit (10-character) words of core storage. Eight 60-bit (10 character) words are contained in a 488-bit physical extended core storage word. Each 10 characters has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 80-character records come from different banks. This phasing, combined with the wide (80-character) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of 10 6-bit characters per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 2.6M characters (two banks) to 20M characters (16 banks). The very fast transfer rates and short access time of extended core storage make it ideal for use as a high-speed program swapping device for storage of large data arrays and for storage of frequently used programs and system routines. ECS is also used as a linking medium in multimainframe configurations for NOS MMF.

Extended Core Storage Options

<u>ECS Size (6-bit characters)</u>	<u>Number of Banks</u>
2.62M	2
5.24M	4
10.48M	8
20.97M	16

There are two access paths to extended core storage

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data paths consist of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170 SERIES 700 COMPUTER SYSTEM MODEL 750

The CDC CYBER 170 Model 750 computer system is a multipurpose computing system that provides real-time/time-critical network, commercial, data management, and scientific capabilities. The Model 750 is field-upgradable to a Model 760 computer system.

The Model 750 computing system has one large central processor, accessible through central memory, and a group of 10 to 20 (optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor. Further concurrency is obtained with the central processor by parallel action of various functional units.

Central memory is organized into logically independent banks. Many banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

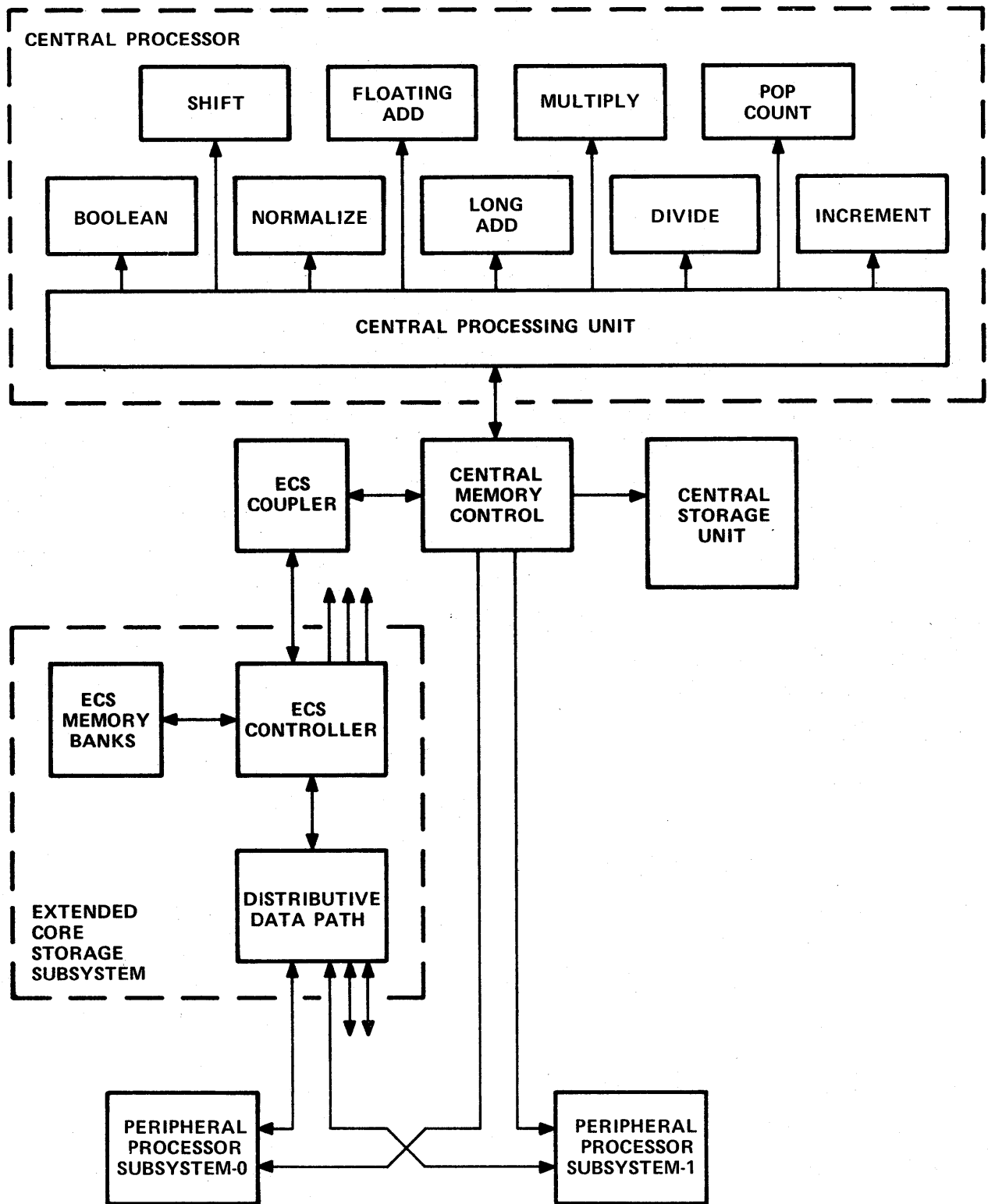
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Nine arithmetic functional units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to twelve 60-bit instruction words with 2-word look-ahead capability
- Computation in floating-point and fixed point, single and double precision
- Integer multiply
- Central exchange jump

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory of 4,096 12-bit words plus one parity bit per word (odd parity)



CDC CYBER 170/Model 750 Computing System

- 12 or 24 input/output channels
 - all channels common to all processors
 - transfer rate per channel 2 MHz (optionally 1 MHz) in 12-bit words
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error conditions (maintenance aid)

Central Memory Characteristics

- Metal oxide semiconductor memory (4K chips) arranged in 60-bit (10-characters) words plus eight error correction bits per 10 characters.
- Memory organized in logically independent banks of words with corresponding multiphasing of banks.
- Transfer rate of up to one word each 50 nanoseconds in phased operation.
- Minimum size of 1.3M characters, optionally expandable to 1.9M or 2.6M characters.

CENTRAL PROCESSOR

The central processor, composed of a central processing unit, nine separate functional units which operate in parallel, and the central memory control, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The nine arithmetic and logical units (floating add, normalize, long add, increment, shift, multiply, divide, boolean, and population count) execute the arithmetic, manipulative, and logical operations. The central memory control directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. The central memory control provides for orderly flow of data between central memory and the requesting elements of the system.

PERIPHERAL PROCESSOR SUBSYSTEM

The peripheral processor subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has 12-bit (plus one parity bit), 4,096-word, random-access memory (independent of central memory) with a cycle time of 500 ns.

A second peripheral processor subsystem may be added to the system with incremental options of 4, 7, or 10 processors.

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 750 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converter

One data channel converter, included with the system, allows Control Data 3000 series peripheral equipment to be attached to the Model 750 input/output channels.

MODEL 750 MEMORY

The CDC CYBER 170 Model 750 computer offers a hierarchical memory concept:

- Central MOS memory 4K chips
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional, it is fully supported and recommended for optimum system performance.

Central Memory

The Model 750 central memory is composed of banks of 60-bit (10-character) words of metallic-oxide storage. There are also eight error correction bits per 10 characters. The complete cycle time for one bank is 400 nanoseconds. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is 10 6-bit characters per 50 nanoseconds.

Central memory is available in sizes ranging from 1.3M characters to 2.6M characters. The eight banks in central memory minimizes memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are four access paths to central memory.

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control (part of central processor) provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 10 characters placed in central memory. When 10 characters are read from central memory, manipulation of the error correction bits allows the 10 characters containing a single-bit error to be corrected. Multiple-bit errors are not corrected, but their occurrence is noted in the status and control register.

Extended Core Storage (Optional)

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 1.3M characters of core storage. Eight 60-bit (10-character) words are contained in a 488-bit physical extended core storage word. Each 10 characters has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 80-character records come from different banks. This phasing, combined with the wide (80-character) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of 10 characters per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 2.6M characters (two banks) to 20M characters (16 banks). The very fast transfer rates and short access time of extended core storage make it ideal for use as a high-speed program swapping device; for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (6-bit characters)</u>	<u>Number of Banks</u>
2.62M	2
5.24M	4
10.48M	8
20.97M	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 170 SERIES 700 COMPUTER SYSTEM MODEL 760

The CDC CYBER 170 Model 760 computer system is a multipurpose computing system that provides real-time/time-critical network, commercial, data management, and scientific capabilities. The Model 760 is the upper member of the Series 700 computer family.

The Model 760 computing system has one large central processor, accessible through central memory, and a group of 10 to 20 (optional) peripheral processors. Each of these peripheral processors has a separate memory and can execute programs independently of each other or of the central processor. Through the exchange jump feature and central memory communication, these peripheral processors control the central processor. They communicate with themselves through central memory and the input/output channels.

In solving a problem, one or more peripheral processors are used for high-speed information transfer in and out of the system, and to provide operator control. A number of problems can be in operation concurrently by time-sharing the central processor. Further concurrency is obtained with the central processor by parallel action of various functional units.

Central memory is organized into logically independent banks. Many banks can be in operation simultaneously, thereby minimizing execution time. The multiple operation modes of all segments of the computer, in combination with high-speed circuits, produce a very high overall computing speed.

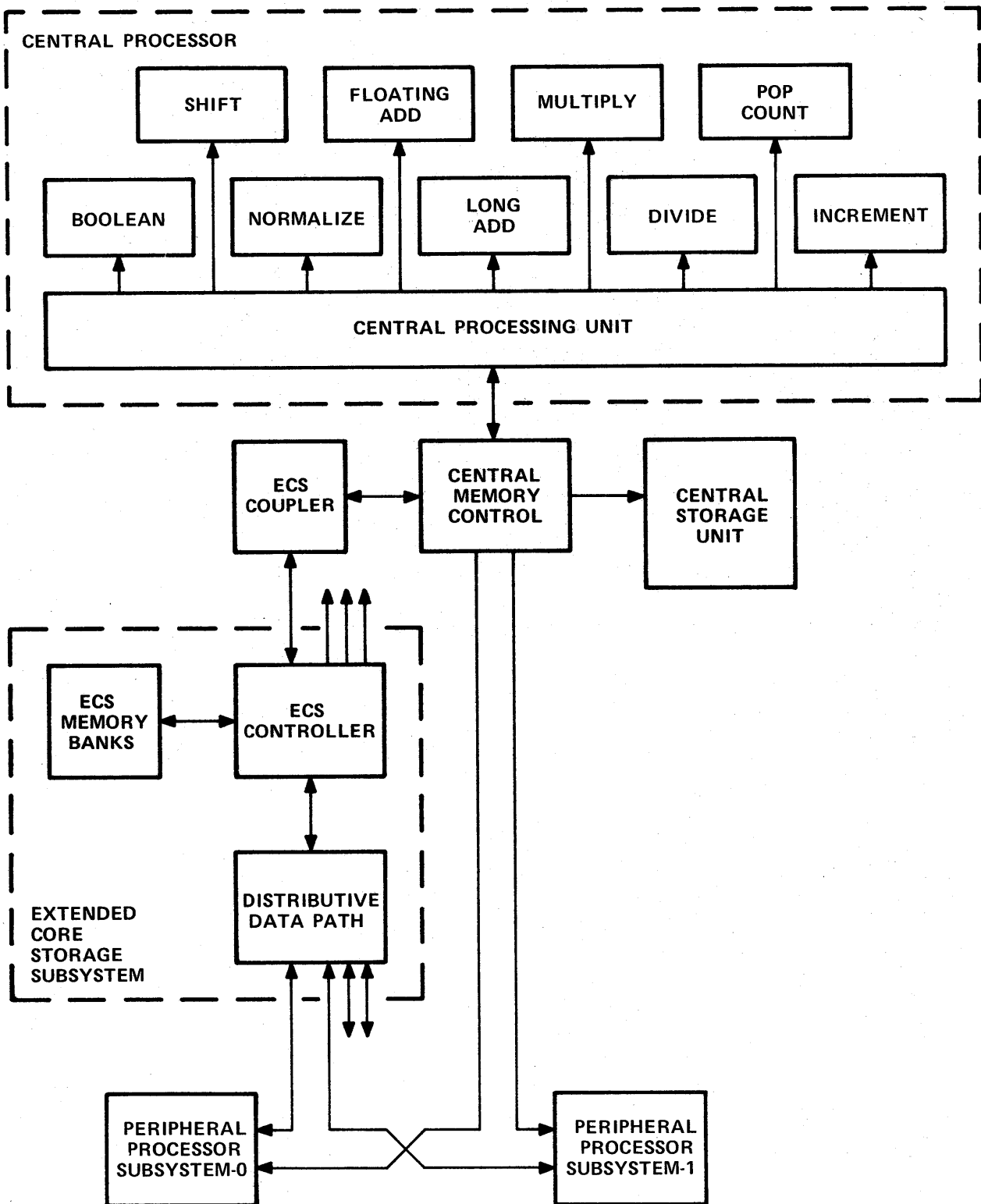
SYSTEM CHARACTERISTICS

Central Processor Characteristics

- Nine arithmetic functional units for concurrent operations
- 24 operating registers
 - 8 operand (60-bit)
 - 8 address (18-bit)
 - 8 increment (18-bit)
- Instruction stack which holds up to twelve 60-bit instruction words with 2-word look-ahead capability
- Computation in floating-point and fixed-point, single and double precision
- Integer multiply
- Central exchange jump

Peripheral Processing Subsystem Characteristics

- Emitter-Coupled Logic (ECL) integrated circuits
- 10 to 20 peripheral processor configurations available (characteristics as listed are per processor)
- Semiconductor memory of 4,096 12-bit words plus one parity bit per word (odd parity)



CDC CYBER 170/Model 760 Computing System

- 12 or 24 input/output channels
 - all channels common to all processors
 - transfer rate per channel 2 MHz (optionally 1 MHz) in 12-bit words
 - all channels can be active simultaneously
 - all channels 12-bit plus parity (odd) bidirectional
- Major cycle time of 500 nanoseconds
- Computation in fixed point
- Time-shared access to central memory
- 64-instruction repertoire
- Status and control register monitors error condition (maintenance aid)

Central Memory Characteristics

- Metal oxide semiconductor memory (4K chips) arranged in 60-bit (10 character) words plus eight error correction bits per 10 characters.
- Memory organized in logically independent banks of words with corresponding multiphasing of banks.
- Transfer rate of up to one word each 50 nanoseconds in phased operation.
- Minimum size of 1.3M characters, optionally expandable to 1.9M or 2.6M characters.

CENTRAL PROCESSOR

The central processor, composed of a central processing unit, nine separate functional units which operate in parallel, and the central memory control, communicates only with central memory. It is isolated from the peripheral processors and is thus free to carry on computation unencumbered by input/output requirements. The nine arithmetic and logical units (floating add, normalize, long add, increment, shift, multiply, divide, boolean, and population count) execute the arithmetic, manipulative, and logical operations. The central memory control directs the arithmetic operations and provides the interface between the functional units and central memory. It also performs instruction retrieving, address preparation, memory protection, and data retrieving and storing. The central memory control provides for orderly flow of data between central memory and the requesting elements of the system.

PERIPHERAL PROCESSOR SUBSYSTEM

The peripheral processor subsystem consists of 10 identical units that operate independently and simultaneously as stored-program computers. Many programs can be running at one time or a combination of processors can be involved in one problem which may require a variety of input/output tasks as well as use of the central memory and the central processor.

The peripheral processors act as system control computers and input/output processors. This permits the central processor to continue computation while the peripheral processors do the slower input/output and supervisory operations.

Each processor has 12-bit (plus one parity bit), 4,096-word, random-access memory (independent of central memory) with a cycle time of 500 ns.

A second peripheral processor subsystem may be added to the system with incremental options of 4, 7, or 10 processors.

Exchange Jump

The exchange jump interrupts an executing central processor program, saves a "snapshot" status of the program (registers, and so forth), and initiates execution of another program. The Model 760 provides four types of exchange jump instructions: one exchange jump initiated by the central processor, and three types of peripheral processor initiated exchange jumps.

One of the peripheral processor exchange jumps is an unconditional jump. The other types of exchange jumps are similar, except that each is dependent on a different condition of a hardware flag. However, in each case the contents of the operating and control registers are moved into storage, and the vacated registers are then loaded with the exchange package information from central memory. This permits a new program to be started by the central processor, and it maintains the information needed to resume the program that was executing.

Input/Output Channels

All processors communicate with external equipment and each other using the independent, bidirectional input/output channels. The number of channels depends on the number of peripheral processors in the system. All channels are 12-bit (plus parity) and each can be connected to one or more external devices. Only one external equipment can use a channel at one time, but all channels can be simultaneously active. Data is transferred into or out of the system in 12-bit words at a maximum rate of two words per microsecond. As many as eight different types of external equipment can be connected to an input/output channel; for example, magnetic tape controls and card read/punch controllers.

Display Controller/Console

The display controller provides digital and analog input information to control the various presentations on the display console. The display console consists of a cathode-ray tube display and a keyboard. The console keyboard performs the functions of a typewriter and permits manual entry of data. Some of the operator actions at the console include:

- display of all registers
- dynamic display of any word or block of words in central memory
- execute single steps of a program

Data Channel Converter

One data channel converter, included with the system, allows Control Data 3000 series peripheral equipment to be attached to the Model 760 input/output channels.

MODEL 760 MEMORY

The CDC CYBER 170 Model 760 computer offers a hierarchical memory concept:

- Central MOS memory 4K chips
- Extended core storage

Central memory provides a fast, random storage for executing programs and data, while the larger extended core storage is used for containing large data arrays, and to support job swapping and other operating system service functions. Although extended core storage is optional, it is fully supported and recommended for optimum system performance.

Central Memory

The Model 760 memory is composed of banks of 60-bit (10-character) words of metallic-oxide storage. There are also eight error correction bits per 10 characters. The complete cycle time for one bank is 200 nanoseconds. The banks are phased so that successive addresses are in different banks to permit operation of central memory at much higher rates than the basic cycle time. The maximum transfer rate is 10 6-bit characters per 50 nanoseconds.

Central memory is available in sizes ranging from 1.3M characters to 2.6M characters. The eight banks in central memory minimize memory access conflicts; therefore, central memory is highly effective for fetching instructions and accessing random data items at very high rates.

There are four access paths to central memory.

- Central processor/central memory
- Extended core storage/central memory
- One or two groups of peripheral processors/central memory

Central memory control (part of central processor) provides service to each of these access paths on a priority basis, queues access requests as necessary, and resolves any access conflicts. The control section also generates the eight error correction bits for each 10 characters placed in central memory. When 10 characters are read from central memory, manipulation of the error correction bits allows the 10 characters containing a single-bit error to be corrected. Multiple-bit errors are not corrected, but their occurrence is noted in the status and control register.

Extended Core Storage

The optional Extended Core Storage (ECS) subsystem is comprised of the extended core storage, its controller, and one or more distributive data paths which attach to input/output channels.

The extended core storage is composed of banks of 1.3M characters of core storage. Eight 60-bit (10-character) words are contained in a 488-bit physical extended core storage word. Each 10 characters has an associated parity bit in the extended core storage word. The complete cycle time for one bank of extended core storage is 3.2 microseconds per 488-bit extended core storage word.

In multiple bank extended core storage subsystems, banks are phased such that consecutive 80-character records come from different banks. This phasing, combined with the wide (80-character) access span enables very fast transfers to or from extended core storage. After initial access, extended core storage can transfer at a rate of 10 6-bit characters per 100 nanoseconds. This gives a maximum rate of 600 million bits per second.

Extended core storage is available in sizes ranging from 2.6M characters (two banks) to 20M characters (16 banks). The very fast transfer rates and short access time of extended core storage make it ideal for use as a high-speed program swapping device; for storage of large data arrays; and for storage of frequently used programs and system routines.

Extended Core Storage Options

<u>ECS Size (6-bit characters)</u>	<u>Number of Banks</u>
2.62M	2
5.24M	4
10.48M	8
20.97M	16

There are two access paths to extended core storage:

- Central memory to extended core storage
- Input/output channel(s) to extended core storage using the distributive data paths

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of from one to four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



CYBER 1000 HARDWARE

The CDC CYBER 1000 Data Communication System can be used either as a stand-alone message switch or as a front-end to a host processor. The CDC CYBER 1000 System is available either as a Protected Message Exchange (PMX) or in a Distributed Network System (DNS). The PMX uses an additional hardware component called the Fast Disk System (FDS) which provides message protection. The PMX uses hardware/software redundancy to prevent any loss of messages in the system. As a front-end system, the CDC CYBER 1000 is provided a special Computer Interface Package (CIP) integrated into the CDC CYBER 1000 System, and it is necessary that the host processor have an access method that can communicate with the CIP.

The DNS achieves very high throughput and short response times by performing the message switch in main memory. The fast disk system is not required.

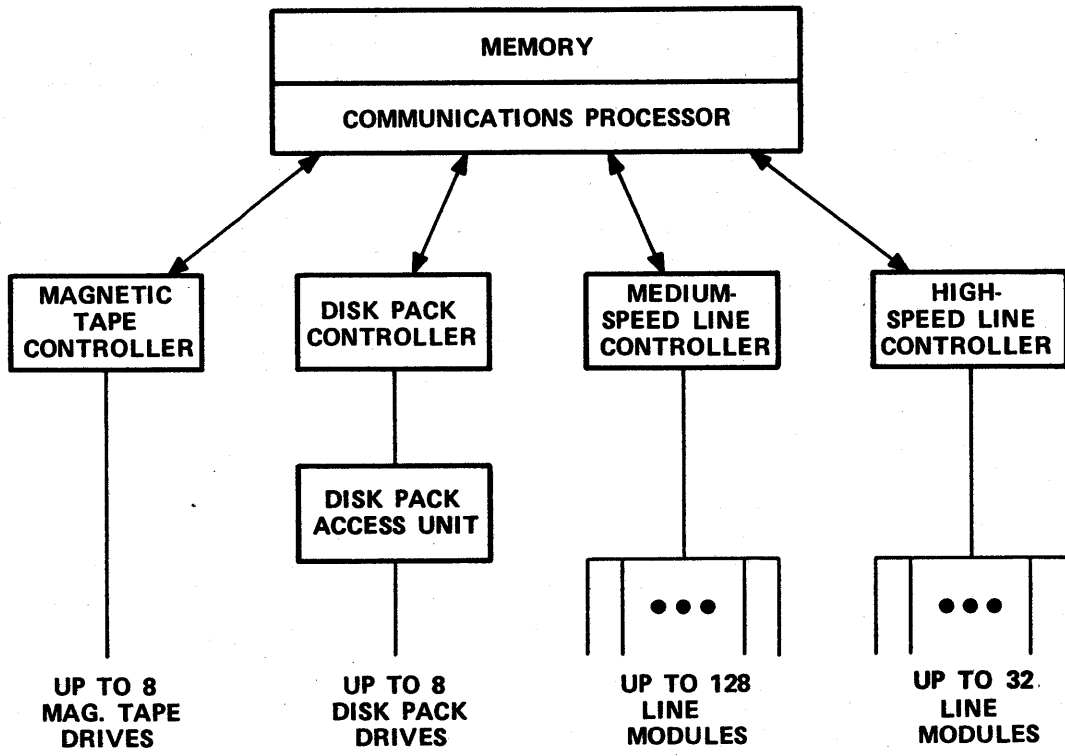
The CDC CYBER 1000 is available in configurations ranging from one to eight communications processors. The basic one processor system (called the Exchange Unit or EU) consists of the central processor unit and its memory, a magnetic tape drive, line termination hardware, and a local supervisory console (M40 or equivalent). A remote switched console can be optionally selected. The magnetic tape drive is primarily used as a deadstart device for initial or intermediate program loading. Each processor can service up to 12 device controllers, each operating at transfer rates up to 50 4.8 megabits/second simultaneously with processing. All of the functional parts of the processor are modular in design as shown in the following figure.

PROCESSOR CHARACTERISTICS

- MOS memory, 1.2 microseconds cycle time
- Recommended memory size, 192K bytes
- Memory expansion to 768K bytes, using optional memory extension kit
- Word size = 27 bits – 3 data bytes, 3 parity bits
- 27 communication-oriented instructions
- Indirect, indexed and immediate addressing
- Four-level priority interrupt system
- Four devices allowed to cause interrupts at each level

LINE TERMINATION

The line termination system is composed of a controller, line modules (one for each line), and level converters. Each circuit enters the communications processor through a level converter. Level converters and associated power supplies provide isolation and compatible termination of the line facility's circuits, i.e.,



CDC CYBER 1000 Hardware

RS-232 or loop current, to the CDC CYBER 1000 line modules. The line module contains hardware specialized to the terminal device and circuit type and matches a standardized hardware interface in the communications processor. The line controller scans the line modules and provides data transfer between the line modules and main memory. The line scan speed per line module is under program control which allows complete flexibility in terminating lines of different speeds (up to 50K baud). The scan program for each application is configured to optimize the line sampling rate for each group of lines operating at different speeds to minimize the number of memory accesses.

Transfer to and from main memory is one-character-at-a-time for the Medium-Speed Line Controller (MSLC), and three characters (one 24-bit word) at a time for the High-Speed Line Controller (HSLC). Transfer rates are 9600 baud for MSLC and 50,000 baud for HSLC. The MSLC can control up to 128 primary (and 128 secondary) lines. The HSLC can control up to 32 primary (and 32 secondary) lines. The HSLC can operate simultaneously with the MSLC increasing the line termination capability per processor to 160 lines.

Processor Status and Control Panel provides displays for:

- Line groups currently being serviced by the processor
- Status of individual lines
- Status of Fast Disk System

And controls for:

- Line group selection
- Transfer of line group responsibility
- Emergency halt
- Use of Supervisory Console as an outstation (ORDER WIRE)
- Audible alarm (allowing processor to be left unattended)

Supervisory Messages entered from the console are provided for:

- Report generation
- Retrieval/journaling
- Traffic control
- System parameter adjustment
- On-line configuration of lines/terminals

System Responses are placed in one of the following five queues, each (except the last) having an associated light indicator and a pushbutton control on the panel:

- Normal report
- Priority report

- Retrieval backlog
- Order wire
- Fast response

CDC CYBER 1000 PERIPHERALS

The CDC CYBER 1000 Data Communications System can support magnetic tape drives, disk pack drives, card reader, and line printer. Magnetic tape drive is primarily used as a deadstart device for program loading. It is also used for system dumps and initial line/terminal configuration. Magnetic tape can be used to journal (record) previously delivered messages and also to retrieve these messages selectively at a later time. It can also be used as an outstation like the other terminals connected to the CDC CYBER 1000 exchange.

Disk pack drives can be used to store the temporarily undeliverable traffic, overflow traffic, and also the delivered traffic that needs to be retrieved at a later time. The support software needed for system generation resides on the disk packs. Card reader and line printer are also required peripherals for system generation.

For additional information, please contact Marketing Support, Control Data Corporation, Communications Systems Division, 3285 East Carpenter Avenue, Anaheim, California, 92806, (714) 630-2022.

SECTION 4

HARDWARE PRODUCTS



405 CARD READER

The Control Data 405 Card Reader is an input device for use with Control Data computer systems. The 405 reader communicates with the computer through a variety of controllers.

HIGHLIGHTS

- Reads 80-column cards at 1200 cpm, 51-column cards at 1600 cpm
- Dual photo-electric read station permits binary, Hollerith, and ASCII codes
- Pneumatic card handling
- Large (4000-card) hopper and stacker trays

DESCRIPTION

The 405 reader photo-electrically senses data punched in standard 80-column or less cards. This reading is performed in a column-by-column fashion, thus eliminating the need for large storage and disassembly registers. The movement of punched cards is done pneumatically. Cards proceed from the input tray, where they are picked one at a time, to the read station.

The photo-electric read station then reads each card twice, compares the two information blocks to assure accuracy, and accepts or rejects the card. As directed by the computer, the reader delivers the card to the primary or secondary receiving tray. Cards can be removed from the feed and receiving trays while the reader is in operation. The trays vibrate to provide a constant force at the picker and the stacker regardless of card deck size.

The character set consists of 64 symbols including the English alphabet in upper case, arabic numerals, punctuation marks, and special symbols. Various controllers allow the reader to read binary, Hollerith, or ASCII card codes.



415 CARD PUNCH

The CDC 415 Card Punch is an output device used with the Control Data computer systems. The 415 punch requires the CDC 3446 Card Punch Controller.

HIGHLIGHTS

- Punches 80-column card at 250 cpm using binary, Hollerith or ASCII card codes
- Input hopper capacity of 1200 cards; output hopper capacity of 1500 cards with programmable off-set stacking
- Automatic read after punch

DESCRIPTION

The card punch perforates standard 80-column cards at a rate of 250 cards per minute. Cards are punched row-by-row, using a bank of 80 punch elements. Following the 12-row punch operation, the card is sent to a post-punch read station. The card is read and the number of holes actually punched is verified against the number specified in the punch operation. If in error, the card is identified via the selectable offset feature of the punch.

Input hopper capacity is 1200 cards and the output stacker accommodates up to 1500 cards. Cards can be added to the hopper or removed from the stacker while the punch is in operation.

The character set consists of 64 symbols including the English alphabet in upper case, arabic numerals, punctuation marks and special symbols.



580 TRAIN PRINTER

The Control Data 580 Train Printer subsystem is an electromechanical device that converts data in the computer to characters on a printed form.

The 580 printer receives data from a 3000-type data channel or equivalent.

HIGHLIGHTS

- Integrated circuit electronics provide high electrical reliability
- A blower provides constant air circulation to prolong component life
- Pluggable modules simplify removal and replacement of electrical components
- Operates at 1200, 1600, and 2000 lines per minute (model option) with a line width of 136 columns
- Program selectable 6- or 8-line-per-inch spacing
- Built-in maintenance features

DESCRIPTION

The 580 printer is an on-line device that can be used for data recording. The horizontally moving type train of the 580 eliminates wavy lines and can be interchanged with a variety of 596 Train Cartridges containing 48-, 63-, or 94-character type fonts.

The 580 uses any standard edge-punched, fan-folded paper (up to six copies from a single impression). An internal controller permits the selection of many vertical printing formats using a loop of punched tape. Paper condition and movement are continuously monitored and detection of a paper-out, paper-torn, or abnormal paper-whip condition stops the printing operation before any damage can occur.

The forms alignment controls (individual tractor adjustment, right-tractor positioner, vertical forms positioners), line indicator, and direct-reading form alignment scale permit the operator to load forms so that the first printout is exactly positioned.

A powered stacker automatically maintains the top of the output listing stack at a constant position: freeing the operator for other duties.

MODELS

The 580 printer is available in the following models:

Model

Features

580-12
580-16
580-20

†1200 lpm (using a 48-character train cartridge)
†1600 lpm (using a 48-character train cartridge)
†2000 lpm (using a 48-character train cartridge)

†Higher speeds attainable when used with a train cartridge containing fewer characters.

CONTROL DATA® 580 SERIES LINE PRINTERS

CONTROL DATA
CORPORATION



The CONTROL DATA 580 Series of Line Printers is a complete printer family featuring 1200, 1600, and 2000 line-per-minute operation, an acoustically-dampened cabinet, enclosed stacking, and extended-font print capability.

The horizontally moving character array of the CDC® 580 family assures printed lines that are always straight. Forms are advanced under a feedback control system which assures even spacing of lines. High throughput is achieved by rapid acceleration to and deceleration from maximum skip speed. The type-array cartridge is easily changed by the operator to match the font to each application.

CHARACTERISTICS SUMMARY

MODEL	LPM	SKIP - IPS	COLUMNS	STACKING
580-20	2000	90	136	Power
580-16	1600	70	136	Power
580-12	1200	70	136	Power

TRAIN CARTRIDGES

596-01	63 printing characters + space (same characters as 595-1)
--------	--

596-02	48 printing characters + space (same characters as 595-2)
596-03	48 printing characters + space (same characters as 595-3)
596-04	6 sets, 63 characters (same characters as 595-4)
596-05	63 characters + space (same characters as 595-5)
596-06	94 characters + space (same characters as 595-6)

SPECIFICATIONS

	580-12	580-16	580-20
Printing Speed			
Nominal (48 characters):	1200 lpm	1600 lpm	2000 lpm
Sustained (24 characters):	1500 lpm	2250 lpm	3000 lpm
Burst (16 characters):	2500 lpm	3000 lpm	3500 lpm
Paper-Advance Speed			
Single Space			
6 lpi:	11.0 ms	11.0 ms	8.3 ms
8 lpi:	10.0 ms	10.0 ms	7.0 ms
Skip Rate			
6 lpi:	70±5 ips	70±5 ips	90±5 ips
8 lpi:	50±5 ips	50±5 ips	70±5 ips
Hammer Recycle:			
	12.5 ms	12.5 ms	8 ms
Type Cartridge Velocity:			
	145 ips	216 ips	250 ips
Type Array Motor:			
	1/3 HP	1/2 HP	1/2 HP
Power:			
	5.7 KVA	6.8 KVA	9.0 KVA

Current: 16 amps 19 amps 25 amps

Heat Dissipation
(max. BTU/HR): 12,000 19,000 24,000

Forms Capacity
Internal Stacking: 4 to 20 inches wide, 4 to 14 inches long
External Stacking: 4 to 20 inches wide, 4 to 22 inches long
Printing Width: 16¾ inches maximum

Electrical: 208 VAC, 50 or 60 (± 1) Hz, 3-phase, 5-wire, wye-connected

Physical
Height: 52 inches (69 inches with hood opened)
Width: 62 inches, max
Depth: 32 inches less stacker, 46 inches with stacker
Weight: 1500 pounds (approximately)

Temperature

Operating: 60 to 100°F
Storage: -30 to +150°F

Relative Humidity

Operating: 10 to 90 percent
Storage: 5 to 95 percent
Line Width: 136 columns
Line Spacing: 6 or 8 lines per inch
Character Array: 48, 64 and 94
Print Cartridge: 384 characters

Specifications are subject to change without notice.



580 TRAIN PRINTER WITH PFC

The CDC 580 Series Train Printers, with Programmable Format Control (PFC), is a complete printer family. This series features 1200, 1600, and 2000 line-per-minute operation, an acoustically-dampened cabinet; enclosed, powered paper stacking; extended-font print capability; and a micro processor-based paper motion control system which replaces the electro-mechanical vertical format unit and associated format tapes.

A printer subsystem consists of a train printer and a single-access controller, packaged together in the same cabinet. A CDC 596 Train Cartridge is required to complete this subsystem.

The controller may be interfaced to CDC CYBER 170, CYBER 70, 6000 or 3000 computer systems. It includes the electronics required to interface to the data channel, control print and paper motion, and through use of memory, stores one line of print and the train image currently being used. This controller also contains a PFC memory which stores the format channel locations as received from the data channel and is operated via external function control.

The character array used by the 580 printer family moves on a horizontal plane and produces uniform, straight-line copy. The type-array cartridge is easily changed by the operator containing 48-, 63-, or 94-character type fonts.

The 580 uses any standard edge-punched, fan-folded paper (up to six copies from a single impression). Forms are advanced under a feedback control system which ensures even line spacing. High throughput is achieved by rapid acceleration to, and deceleration from, maximum skip speed.

Printer design provides easy access to the print-head area, and minimum adjustments are required in loading forms. As an additional operator convenience, easy-to-read status indicators are provided on the control panel which alert the operator to any condition requiring attention.

HIGHLIGHTS

- Micro processor-based paper motion control system
- Integrated circuit electronics provide high electrical reliability
- A blower provides constant air circulation to prolong component life
- Paper condition is monitored to preclude possible loss of information
- Pluggable modules simplify removal and replacement of electrical components
- Operates at 1200, 1600, and 2000 lines per minute (model option) with a line width of 136 columns
- Program selectable 6 or 8 line per inch spacing
- Built-in maintenance features

MODELS

The 580 Train Printer with PFC is available in the following models:

<u>Model</u>	<u>LPM</u>	<u>Skip-IPS</u>	<u>Columns</u>
580-120	1200	70	136
580-160	1600	70	136
580-200	2000	90	136

SPECIFICATIONS

Printing Speed

	<u>580-120</u>	<u>580-160</u>	<u>580-200</u>
Nominal (48 Characters):	1200 lpm	1600 lpm	2000 lpm
Sustained (24 Characters):	1500 lpm	2250 lpm	3000 lpm
Burst (16 Characters):	2400 lpm	2850 lpm	3500 lpm

Paper Advanced Speed

Single Space, 6 lpi:	12.5 mSec	12.5 mSec	9.4 mSec
Single Space, 8 lpi:	12.5 mSec	12.5 mSec	9.4 mSec
Skip Rate, 6 lpi:	70 ips	70 ips	90 ips
Skip Rate, 8 lpi:	50 ips	50 ips	70 ips

Forms Capacity

Internal Stacking:	4 to 20 inches wide, 4 to 14 inches long
External Stacking:	4 to 20 inches wide, 4 to 22 inches long
Line Width:	136 columns
Character Pitch, Horizontal:	10 characters per inch
Character Pitch, Vertical:	6 or 8 lines per inch
Print Cartridge Capacity:	384 characters



595 TRAIN CARTRIDGE

The Control Data 595 Train Cartridge is an operator-interchangeable type train that allows the printer font to be matched to the user's application. Printing speeds vary inversely with the size of the character set.

HIGHLIGHTS

- Permits high document throughput
- Eliminates wavy print line often seen in printouts from drum printers

MODELS

The 595 Train Cartridge is available in the following models:

<u>Model</u>	<u>Features</u>
1	Sixty-three printing characters plus a blank. Same as 501 printer character set, modified standard gothic font.
2	Forty-eight printing characters plus a blank. Same as IBM-AN-arrangement, modified standard gothic font.
3	Forty-eight printing characters plus a blank. Same as IBM-AH-arrangement, modified standard gothic font.
4	Array cartridge contains six sets of characters which in combination comprise the 63 different characters of the CDC 64 character subset of USASCII 3.4, Revision I. Interfaces 512-1 or 517-1 line printer.
5	Sixty-three printing characters plus a blank. Array cartridge contains six sets of characters which are comprised of the center four columns of the ASCII code table. (ASCII 3.4 Revision I subset.) Interfaces 512-1 or 517-1 line printers.
6	Ninety-four printing characters plus a blank and a delete. Array cartridge comprised of the six rightmost columns of the ASCII code table. This array cartridge includes upper and lower case characters. Interfaces 512-1 or 517-1 line printers.



596 TRAIN CARTRIDGE

The 596 Train Cartridge is an operator-interchangeable type train that allows the printer font to be matched to the user's application. Printing speeds vary inversely with the size of the character set.

HIGHLIGHTS

- Permits high document throughput
- Eliminates wavy print line often seen in printouts from data printers

MODELS

The 596 Train Cartridge is available in the following models:

<u>Model</u>	<u>Features</u>
1	Sixty-three printing characters plus a blank. Same as 501 and 595-1 printer character sets, modified standard gothic font.
2	Forty-eight printing characters plus a blank. Same as IBM-AN-arrangement and 595-2 character set, modified standard gothic font.
3	Forty-eight printing characters plus a blank. Same as IBM-HN-arrangement and 595-3, modified standard gothic font.
4	Array cartridge contains six sets of characters which in combination comprise the 63 different characters of the CDC 64 character subset of USASCII 3.4, Revision I. Same as 595-4 character set.
5	Sixty-three printing characters plus a blank. Array cartridge contains six sets of characters which are comprised of the center four columns of the ASCII code table. (ASCII 3.4 Revision I subset.) Same as 595-5 character set.
6	Ninety-four printing characters plus a blank. Array cartridge comprised of the six right-most columns of the ASCII code table. This array cartridge includes upper and lower case characters. Same as 595-6 character set.



667 MAGNETIC TAPE TRANSPORT

The CDC 667 Magnetic Tape Transport is a storage device which serves as either on-line or off-line auxiliary data storage.

HIGHLIGHTS

- Transports feature single-capstan, automatic reel-hub latching, cartridge load (or non-cartridge manual load) automatic threading and power-operated window.
- Seven-track recording using NRZI recording method.
- Industry, USASI and ASCII compatible recording densities of 556 and 800 characters per inch.

DESCRIPTION

The 667 tape transport records seven tracks of information on magnetic tape. Tape motion is accomplished by a single capstan controlled by a digital tachometer.

Vacuum columns act as buffers between the synchronous tape speed, as produced by the capstan, and asynchronous motion, as produced by the photoelectric cell steered reel-drive servo system. Air bearing and ceramic guides in the tape path allow only the read/write heads and tape cleaner to make contact with the tape. Automatic reel hub latching, cartridge load, automatic threading, and a power-operated window minimize tape handling. Standard features include: integrated circuits (MSI and LSI) used throughout; MDI (Marginal Detection Indicators) monitor operating conditions; self-diagnostics fault-isolation; automatic write-head degaussing; advanced data-error correction.

MODELS

The 667 tape transport is available in three models with the following characteristics:

<u>Characteristics</u>	<u>667-2</u>	<u>667-3</u>	<u>667-4</u>
Tape speed-read/write	100 ips	150 ips	200 ips
Rewind and unload	60 sec	45 sec	45 sec
Transfer rate (6-bit char)	55.6K-80K cps	83.4K-120K cps	111.2K-160K cps

All models will function with either model of the 3521, 7021, or 7622 Tape Controllers.



669 MAGNETIC TAPE TRANSPORT

The CDC 669 Magnetic Tape Transport is a storage device which serves as either on-line or off-line auxiliary data storage.

HIGHLIGHTS

- Transports feature single-capstan, automatic reel-hub latching, cartridge load (or non-cartridge manual load) automatic threading and power-operated window.
- Nine-track recording using NRZI or Phase-Encoded (PE) recording method.
- Industry, USASI and ASCII compatible recording densities of 800 bpi using NRZI or 1600 bpi using PE recording method.

DESCRIPTION

The 669 tape transport records nine tracks of information on magnetic tape. Tape motion is accomplished by a single capstan controlled by a digital tachometer.

Vacuum columns act as buffers between the synchronous tape speed, as produced by the capstan, and asynchronous motion, as produced by the photoelectric cell steered reel-drive servo system. Air bearing and ceramic guides in the tape path allow only the read/write heads and tape cleaner to make contact with the tape. Automatic reel hub latching, cartridge load, automatic threading, and a power-operated window minimize tape handling. Standard features include: integrated circuits (MSI and LSI) used throughout; MDI (Marginal Detection Indicators) monitor operating conditions; self-diagnostics fault-isolation; automatic write-head degaussing; advanced data-error correction.

MODELS

The 669 tape transport is available in three models with the following characteristics:

<u>Characteristics</u>	<u>669-2</u>	<u>669-3</u>	<u>669-4</u>
Tape speed-read/write	100 ips	150 ips	200 ips
Rewind and unload	60 sec	45 sec	45 sec
Transfer rate (8 bit char)	80K-160K	120K-240K	160K-320K

All models will function with either model of the 3521, 7021, or 7622 Tape Controllers.



677 MAGNETIC TAPE TRANSPORT

The CDC 677 series Magnetic Tape Transports provide data storage with very high performance and heavy duty usage.

HIGHLIGHTS

- Transports feature single capstan, automatic reel hub latching, cartridge load, automatic threading, and power operated window.
- Seven-track recording at 556 or 800 characters per inch using NRZI recording mode.
- Vacuum powered tape scraper and a special tape cleaner that operates during rewind, load, and unload operations.

DESCRIPTION

The 677 tape transports record seven tracks of information on magnetic tape. Tape motion is accomplished by a single capstan.

Vacuum columns act as buffers between the synchronous tape speed produced by the capstan and the asynchronous motion produced by the reel-drive servo system.

The 677 units have been specially developed to provide high levels of performance, reliability, and maintainability based on the latest technology. Integrated circuits and modular logic are used throughout. This modular design makes it easy to maintain without sacrifice to reliability. Internal diagnostic data loops and extensive unit checking status bits are stored and available for fast diagnosis and corrective action, if required.

The 677 provides 7-track industry compatibility, read or write, with computer selected densities of 556 or 800 bpi in NRZI recording mode.

MODELS

<u>Characteristics</u>	<u>677-2</u>	<u>677-3</u>	<u>677-4</u>
Tape Speed (ips)	100	150	200
Rewind speed in seconds	60	50	45



679 MAGNETIC TAPE TRANSPORT

The CDC 679 series Magnetic Tape Transports provide data storage with very high performance and heavy duty usage.

HIGHLIGHTS

- Transports feature single capstan, automatic reel hub latching, cartridge load, automatic threading, and power operated window.
- Nine-track recording at 800 bpi NRZI and 1600 bpi phase-encoded or at 1600 bpi phase-encoded and 6250 bpi group-coded recording.
- Improved throughput via automatic gain control on 6250 bpi models to dynamically adjust for a wide variety of tape qualities.
- Vacuum powered tape scraper and a special tape cleaner that operates during rewind, load, and unload operations.

DESCRIPTIONS

The 679 tape transports record nine tracks of information on magnetic tape. Tape motion is accomplished by a single capstan.

Vacuum columns act as buffers between the synchronous tape speed produced by the capstan and the asynchronous motion produced by the reel-drive servo system.

The 679 units have been specially developed to provide high levels of performance, reliability, and maintainability based on the latest technology. Integrated circuits and modular logic are used throughout. This modular design makes it easy to maintain without sacrifice to reliability. Internal diagnostic data loops and extensive unit checking status bits are stored and available for fast diagnosis and corrective action, if required.

Advanced recording techniques provide models with 6250 bpi density, which boosts system throughput, reduces the frequency of input/output errors, and significantly improves error recovery. Additionally, increased density decreases human handling time, reduces physical storage space requirements and reduces tape procurement costs.

MODELS

<u>Characteristics</u>	<u>679-2</u>	<u>679-3</u>	<u>679-4</u>	<u>679-5</u>	<u>679-6</u>	<u>679-7</u>
Recording densities	800/1600	800/1600	800/1600	1600/6250	1600/6250	1600/6250
Tape speed (ips)	100	150	200	100	150	200
Rewind speed in seconds	60	50	45	60	50	45



751 DISPLAY TERMINAL

The CDC 751 Display Terminal is a microprocessor-based, asynchronous remote terminal, which supports a wide range of configuration options. The 751 provides extensive capability; features ease of installation, ease of use, reliability, maintainability, configuration expansion, and safety. The 751 display terminal communicates with a computer system by a telecommunications line.

HIGHLIGHTS

- A 1920-character display
- A 95-key, detachable keyboard
- Preprogrammed microprocessor controller
- Displayed data editing and highlighting
- Operator choice of character, line or page transmission modes
- RS232-C/CCITT V.24 compatible telecommunication interface
- Available options to add multidrop, automatic answerback, current loop and peripherals for hard copy and tape cassette storage

DESCRIPTION

The 751 display terminal consists of a Cathode-Ray Tube (CRT) display, a detachable keyboard, a communications interface and a controller. The CRT has a 30 cm (12 inches) diagonal which will display 1920 characters (24 lines at 80 characters). Each of the characters are formed within a 7 x 9 dot matrix. Character repertoire consists of 95 ASCII (upper and lower case) plus the ASCII representation of 33 control codes. The keyboard is detachable up to 60 cm (24 inches) from the 751 cabinet. It features a typewriter layout plus an 11 key numeric pad, cursor controls, and special function keys. Key "roll-over" circuitry prevents multikey false character generation. The communication interface is designed to connect to a modem as defined by RS232-C or CCITT V24 at rates up to 9600 bps. The controller generates the 64 or 96, switch selectable characters. Selected areas of displayed data may be highlighted by reduced brightness or blinking. Displayed data editing includes tabulation, character and line insert and delete, automatic justification, character "wraparound", protected format, and cursor addressing. The cursor consists of a blinking underline at the position to be entered. Switch selectable data transmission may be character-by-character, line-at-a-time (up to 80 characters), or block mode (up to 1920 characters).

Additional features include: received character parity error indication; scroll enable/disable; display/non-display of control codes; partial screen transmit; space or null code selection for increased transmission efficiency; data transmission mode/status indicators; and front-mounted display intensity control. The operator also has a choice of delimiter codes, including carriage return, when operating in character mode half-duplex. Options are simple, plug-in printed-circuit cards. Self test facilities also permit the operator to initiate six discrete off-line tests which verify terminal operation.



752 DISPLAY TERMINAL

The CDC 752 Display Terminal is an easy-to-operate, asynchronous remote terminal with optional hard-copy printers available. This character-mode terminal is applicable to all CDC computer systems as well as non-CDC systems which support TTY-compatible asynchronous terminals.

The 752 operates in either scroll or page mode, using X-Y coordinate addressing and reading with a blinking underline cursor that can be moved to any position on the screen. All display logic is packaged on a single printed circuit board which is easily removed. The 752 provides extensive capability; features ease of use, ease of installation, reliability, maintainability, totally silent operation and safety.

FEATURES

- A 1920-character display
- Switch-selectable transmission speeds from 110-9600 bps
- An 80-key, detachable keyboard including numeric pad and "Enter +, Enter -" keys
- N-Key rollover circuitry to prevent loss of input data
- Displayed data highlighting
- Front panel selection of communication parameters
- RS232-C/CCITT V.24 compatible telecommunications interface standard
- Optional version with 20 ma loop current interface
- Circuit assurance (channel monitor) feature
- Self-test mode
- Interface provided to add optional impact or nonimpact printer

DISPLAY

A 12-inch (30 cm) diagonal CRT displays 1920 white characters (24 lines of 80 characters per line) on a dark background. Each of the 94 displayable ASCII characters are formed within a 7 by 9 dot matrix with 22 mm by 47 mm dimensions. ASCII representation of 33 control codes can also be displayed under test conditions. The bonded, non-glare, glass faceplate resists scratching and enhances viewing in high ambient light.

KEYBOARD

The detachable keyboard may be located up to 24 inches (60 cm) from the display terminal cabinet and features a typewriter layout. This arrangement includes: a 13-key numeric pad, trilevel keyboard operation,

and cursor controls. Keycaps have a non-slip matte finish, a lifetime injection-molded symbol marking, and four deep-dished keys at home-position. Circuitry prevents illegal character generation and guards against instantaneous typing speeds greater than the slowest communication rate.

COMMUNICATION INTERFACE

The 752 Display Terminal is designed to connect to an external, asynchronous modem having an interface characteristic as defined by Specifications RS232-C or CCITT V.24. Switch-selectable data transmission may be half-duplex or full-duplex at the following communication rates: 110, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud. Word size is 11 bits for the 110 rate, 10 bits for all other communication rates, with switch-selectable parity, including odd, even and none. For the "none" condition the user may select mark or space in parity bit positions.

An optional version of the 752 is configured with a 20 ma loop current interface instead of the RS232-C interface.

DISPLAY TERMINAL

The terminal displays up to 1920 characters on the CRT. A switch setting allows choice of either the 64-character or 96-character ASCII set. Selected areas of displayed data may be highlighted by reduced brightness or blinking. An audible alarm sounds as the cursor moves to the 73rd character position or in page mode enters the 24th (bottom) line. The cursor consists of a blinking underline at the position to be entered.

Additional features include: display intensity control, carrier on indicator, on-line or local operation, display of parity error symbol in bad character position, and switches to select communication speed, full or half duplex, and parity convention.

CDC STANDARD SOFTWARE/SYSTEM SUPPORT

CDC software systems support various levels of 752 features; the most recent and planned future releases offer increasing levels of features supported.

CONFIGURATIONS

Four models of 752 Display Terminal are available, depending upon the combination of primary power source and communication line interface desired.

PERIPHERAL OPTIONS (One printer allowed)

753-11 Nonimpact Printer

A desktop, thermal character printer, with 5 x 7 dot matrix print head. Produces hard copy on 8½ inch wide, heat sensitive, continuous roll paper, and prints up to 80 columns at 30 characters per second. Prints 94 ASCII symbols. It also includes a 10.5 ft. input/output cable.

755-11 Impact Printer

A desktop, line printer with 7 x 7 dot matrix print head. Produces up to four copies plus one original, and prints up to 132 columns at speeds of up to 55 lines per minute (173 characters per second). Prints 94 ASCII symbols. Includes 1000-character buffer, self-test, format tape and a 10.5 ft. input/output cable.

SPECIFICATIONS

Dimensions (keyboard collocated with display cabinet)

Height: 15.7 in (40 cm)

Width: 22 in (55 cm)

Depth: 20 in (52 cm)

Weight: 55 lbs (25 kg)

Color: Harvest gold and black accent

Power Requirements: 120 V ac, 60 Hz, 3.5 A maximum
220/240 V ac, 50 Hz, 2.0 A maximum

Operating Temperature: +50°F to 104°F
(+10°C to 40°C)

Humidity: 20% to 80% RH

Cooling: Convection cooled

Heat: 332 BTU/hr

Cables:

RS232-C Data Set Cable, 10.5 ft (3 m) length

120 V, 60 Hz Power, 8 ft (2.4 m) length, 3-conductor with plug

220/240 V, 50 Hz Power, 8 ft (2.4 m) length, 3-conductor less plug



753 NON-IMPACT PRINTER

The CDC 753 Non-Impact Printer is a desk-top, thermal, serial-character printer which uses a single 5 by 7 dot matrix print head. This unit produces copies of messages directed to it by the micro-processor controller with the CDC 75X series terminal. The 753 quietly prints up to 30 characters per second in 80 columns, on thermal-sensitive paper.

PRINTER FUNCTIONS

- Line Feed
- Carriage Return
- Space
- Back Space
- Character Print
- Automatic Carriage Return/Line Feed
- No Action on null and undefined codes

SPECIFICATIONS

- Horizontal Character Spacing (10 columns per inch): 2.54 mm (0.100 in), nominal
- Vertical Character Spacing Operator selectable: 6 lines per 2.54 cm (1 in); 3 lines per 2.54 cm (1 in.)
- Paper
 - Type: Heat-sensitive, thermal paper which changes color when contacted by the heated elements in the print head
 - Size: 6.35 cm (2.5 in) diameter roll
22.23 cm (8.75 in) wide by 30.5 m (100 ft) long
- Operator Controls
 - Power On/Off
 - Paper Advance
 - Carriage Return
 - Single or Double Line Spacing
- Throughput
 - Up to 30 cps; operation on 300 bps line
- Interface: RS232-C



755 IMPACT SLAVE PRINTER

DESCRIPTION

The CDC 755 Impact Slave Printer is a desktop line printer which uses a 7 x 7 dot matrix print head. This printer produces copies of messages directed to it by the controller of the CDC 751 Display Terminal or the CDC 752 Display Terminal. The 755 is capable of printing multiple copies at speeds of 173 characters per second (60 Hz version), and 180 characters per second (50 Hz version).

The 755 follows a continuous communication line data stream of 1200 baud for a 132-character line. A 1000-character printer buffer is provided to handle shorter lines and retain the 1200 baud communication rate.

The 755 handles standard, continuous form paper (one original plus four copies) having $6.35 \pm .76$ mm (0.25 ± 0.03 inch) feed holes on each edge, with or without marginal perforations. Forms may be from 10.16 to 41.9 cm (4 to 16.75 in) in width, including margin and 8.89 to 45.7 cm (3.5 to 18 in) long from fold to fold.

HIGHLIGHTS

- Accepts forms from 4 to 16 3/4 inches (101.6 to 425 mm) wide
- Three-channel format tape reader for forms control
- Front-of-form impacting elements
- Self-contained print test
- RS232-C interface
- 132-character line buffer, plus 1K buffer memory
- Plug-in card rack swings out 180 degrees for easy maintenance
- Immediate visibility of last line printed
- Quiet operation
- Low mass print head results in low vibration
- Meets U.S., Canadian, and European safety standards
- Easily replaced ribbon (20 m x 13 mm) with life span of 2 million characters
- Prints original plus four copies
- External forms-thickness control, easily accessible
- 96-character set contained in single ROM

MODELS

The 755 is available in the following models:

<u>Model</u>	<u>Description</u>
10	Used when interfacing the 751-10 Display Terminal
11	Used when interfacing the 752 Display Terminal



756 DISPLAY TERMINAL

The CDC 756 Display Terminal is a microprocessor-controlled, asynchronous remote unit that provides local and remote command control. These terminals are designed for applications where low-cost, distributive processing is a major concern. They can be used on all Control Data computer systems as well as any non-CDC systems which support TTY-compatible, asynchronous terminals.

The 756s operate in either character mode or block mode, using X-Y coordinate addressing. A blinking underline cursor, which can be moved to any position on the screen, permits quick and easy editing of displayed data. All display logic is packaged on a single printed circuit board which is easily removed for maintenance. These terminals provide extensive capability in a unit easy to install, use and maintain. They also feature a high degree of safety and reliability and are totally silent in operation.

HIGHLIGHTS

- A 1920-character display
- Switch-selectable transmission speeds from 110-9600 bps
- Choice of two detachable keyboard layouts including numeric pad and "Enter +, Enter -" keys
- N-Key rollover circuitry to prevent loss of input data
- Front panel selection of communications parameters
- RS232-C/CCITT V.24 compatible telecommunications interface standard
- Optional version with 60 ma loop current interface
- Self-test mode
- Interface provided to add optional impact or nonimpact printer
- Character or block transmission
- Local/remote command control
- Dual-intensity and blink highlighting
- Character and line insert/delete
- Tabs and protected data fields

Two different keyboard layouts are available on the 756 Terminals. One is similar to the ISO Standard 3243 layout; the other is patterned after the common U.S. typewriter layout. Either keyboard can be located up to 24 inches (60 centimeters) away from the display terminal cabinet. Both keyboard arrangements include: numeric pad, cursor controls and special function keys (including a HERE IS key to be used with the answer-back function). Special internal circuitry automatically prevents illegal character generation.

The 756 Display Terminals are designed to connect to an external, asynchronous modem having an interface characteristic as defined by Specifications RS232-C or CCITT V.24. Switch-selectable data transmission may be half-duplex or full-duplex at the following communications rates: 110, 150, 200, 300, 600, 1200, 1800, 2400, 4800, or 9600 baud. Word size is 11 bits for the 110 rate, 10 bits for all other communications rates. Switch-selectable parity, including odd, even, mark, and space, is standard.

Display Terminal – A 12-inch (30 cm) diagonal CRT displays 1920 white characters (24 lines of 80 characters per line) on a dark background. A switch setting allows choice of either the 64-character or 96-character ASCII set. In addition, 33 control code characters are provided. Each of the 128 displayable characters is formed by a 7 x 9 dot matrix. Selected areas of displayed data may be highlighted by reduced brightness or blinking. An audible alarm sounds as the cursor moves to the 73rd character position or, in the page mode, enters the 24th (bottom) line. The cursor consists of a blinking underline at the position to be entered.

Additional editing features include forms protection, split-screen, partial page transmission, character and line insert and delete, forward/backward tab, and 12 special function keys providing 24 unique two-character sequences. An additional feature allows the operator to clear data from the cursor to the end of the line or end of page.

SOFTWARE/SYSTEM SUPPORT

The 756 Display Terminals will operate on all systems which currently support the CDC 713-10 asynchronous TTY-type terminal. These include systems under:

SCOPE 3.4
KRONOS 2.1
NOS
NOS/BE

In addition, the 756s are supported by CDC 2550 software interfacing to INTERCOM 4.5 under SCOPE 3 and NOS/BE.

MODELS

<u>Display Terminal</u>	<u>Power Source</u>	<u>Comm. Line Interface</u>
756-10	120 Vac, 60 Hz	RS232-C/CCITT V.24
756-11	120 Vac, 60 Hz	60 ma current loop
756-20	220/240 Vac, 50 Hz	RS232-C/CCITT V.24
756-21	220/240 Vac, 50 Hz	60 ma current loop

Keyboard Options

756-101	Standard U.S. Typewriter Layout
756-102	International Keyboard Layout (ISO 3243)

Peripheral Options

753-11	Nonimpact Printer
755-21	Impact Printer



777 CYBER GRAPHICS TERMINAL 774 DIGIGRAPHICS IV CONSOLE

The CDC 777/774 is an interactive entry/display subsystem designed to interface to CDC CYBER 70, CYBER 170, or 6000 series computers. The subsystem provides access to stored data which can be scanned visually, selected, processed, modified and redisplayed in alphanumeric and graphic representation. The CDC 777 CYBER Graphics Terminal consists of a stored program controller and one to three 774 Digigraphics IV Consoles.

HIGHLIGHTS

- Stored program
- Memory cycle 0.6 μ s.
- Word size 16 bits
- Memory 24K (minimum) to 65K
- Capable of supporting up to three 774 consoles

774 Digigraphics IV Console

- Viewing area (20-inch round CRT) 314 sq. inch
- Characters/12-inch – maximum of 136/line
- Lines/12-inch – maximum of 68
- Symbols – 96 symbol USASCII, plus 31 arithmetic and Greek symbols
- Symbols per line – 45 to 136 along a 12-inch line
- Light pen provides for operator interaction

DESCRIPTION

The 777X Terminal Controller contains stored programs and is equivalent to a computer regarding capability. The controller contains terminal controlware, handles the communications with the host computer, maintains the display file and processes graphics output as directed by the host computer.

The 777X Terminal Console (774) consists of a keyboard, CRT, light pen, and a Display Code Interpreter (DCI). Computer-programmed data is displayed at high-speed on the console CRT. Data is entered and modified using the keyboard and/or light pen. Graphic functions are performed by the DCI, a part of the console, as directed by commands initiated by the controller.

MODELS

The 777 terminal is available in the following models:

<u>Model</u>	<u>Feature</u>
777-2 CDC CYBER Graphics Terminal	Includes terminal controller with 24K words of memory, necessary input/output, software load capability, one 774-2 Digigraphics IV console and a synchronous Communications Line Adapter (CLA) for 40.8K bps.
777-3 CDC CYBER Graphics Terminal	Includes terminal controller with 24K words of memory, necessary input/output, software load capability, one 774-1 Digigraphics IV console and a synchronous communication adapter for line speeds of up to 9.6K bps.
774-1 Digigraphics IV Console	Add-on console for use in multiple console configurations. Maximum of three consoles per controller.
774-2 Digigraphics IV Station	Console for adaptation to those computer systems already installed/owned by customers that may be used as the CDC CYBER Graphics terminal controller (1704, 1774, CDC CYBER 18-17). Includes 40.8K bps synchronous communications adapter.



819 DISK STORAGE UNIT

The CDC 819 Disk Storage Unit is an electromechanical access storage unit that records and retrieves data on disk surfaces.

HIGHLIGHTS

- Phase-lock read recovery
- Absolute addressing
- Carriage offset
- Maintenance aids
- Data strobe offset
- Data transfer rate: 38.7 megabits per second (4 heads parallel) nominal
- Spindle speed: 3600 rpm
- Positioning time 15-80 milliseconds
- Average positioning time 50 milliseconds
- Average rotational delay 8.33 milliseconds
- Average access time: 58.33 milliseconds

DESCRIPTION

The CDC 819 unit is a peripheral mass memory device. It consists of a cabinet containing a disk pack spindle and associated drive motor, voice coil positioning mechanism, disk pack, power supply, and logic chassis.

The recording medium consists of 22 magnetic-oxide coated disks (nonremovable pack), with 40 data surfaces and one servo surface containing head positioning information.

MODELS

The 819 is available in the following models:

<u>Model</u>	<u>Bits/Drive (billions)</u>	<u>Bits/Inch (Nominal)</u>	<u>Cylinders Per Unit</u>	<u>Access Time</u>	<u>Capacity Unit (6-bit characters)</u>	<u>Transfer Rate</u>
819-1	2.4	6000	404	58.33 ms	412	38.7M bps
819-11	2.4	6000	404	58.33 ms	412	38.7M bps
819-21	4.8	6000	808	58.33 ms	824	38.7M bps

The Model 819-11 drive is field-upgradable to a Model 819-21.



844 DISK STORAGE UNIT

The CDC 844 Disk Storage Unit is an electromechanical access storage unit that records and retrieves data on a removable disk pack.

HIGHLIGHTS

- Disk pack rotational speed of 3600 rpm with an average latency time of 8.3 milliseconds
- Average positioning time of 30 milliseconds; 10 ms minimum, 55 ms maximum
- Up to eight units can be connected to create a subsystem

DESCRIPTION

The 844 unit provides access to data organized in tracks and stored on a rotating disk pack. The disk pack consists of a stack of eleven 14-inch diameter, oxide-coated disks. Data is accessed by 20 read/write heads mounted on a carriage that is positioned to the required track by a voice-coil linear actuator operating in a closed loop, proportional-servo system.

MODELS

The 844 is available in the following models:

<u>Model</u>	<u>Disk Pack</u>	<u>Tracks</u>	<u>Capacity Unit (6-bit characters)</u>	<u>Bits per Drive (millions)</u>	<u>Transfer Rate</u>
844-41†	883	808 plus spares	237 million	1,423 sectored 1,650 unsectored	6.45 MHz
844-44†	883	808 plus spares	237 million	1,423 sectored 1,650 unsectored	6.45 MHz

†Models 844-41 and 844-44 are functionally interchangeable.



881/883 DISK PACKS

CDC 881/883 Disk Packs are removable, interchangeable memory devices that are used on the disk storage units of mass storage facilities. This disk pack and associated disk storage drive provide random access data storage for computer systems.

HIGHLIGHTS

- Twenty magnetic oxide coated, recording surfaces: 19 data storage, 1 servo loop and data organization (883 only) control.
- Compatibility: 881 Disk Pack – CDC 844-2 and 844-21
883 Disk Pack – CDC 844-41 and 844-44
- Designed and manufactured under strict quality control to meet highly critical performance standards.

DESCRIPTION

The 881 and 883 Disk Packs contain 10 recording surfaces. The 10 disk surfaces are used to record data organized in tracks. The twentieth disk surface is pre-recorded with servo loop track identifier information: 404 plus tracks on the 881, and 808 plus tracks on the 883.

The 881 pack includes a sector disk (data organization) adjacent to and below the bottom recording disk surface. The sector disk contains slots that are detected to signal the start of a revolution (index) and the beginning of each of the 24 sectors that comprise a revolution. The 883 pack uses an index mark that is pre-recorded on the surface containing the servo loop information. Sector signals are derived from a counter.



885 FIXED MODULE DRIVE

The CDC 885 Fixed Module Drive (FMD) is an electromechanical access storage unit that records and retrieves data on two fixed disk modules. A CDC 7155 Fixed Module Drive Controller is required to interface between the 885 unit and a CDC CYBER 170 series computer system.

HIGHLIGHTS

- Two spindles are included in each unit.
- Large block nominal transfer rate of 1.2 million 6-bit characters per second for each spindle.
- Disk pack rotational speed of 3600 rpm with an average latency time of 8.3 milliseconds, 16.7 milliseconds maximum.
- Average positioning time of 25 milliseconds; 10 ms minimum, 50 ms maximum.
- Up to four units (eight spindles) can be connected in a basic subsystem.

DESCRIPTION

The CDC 885 provides access to data organized on two fixed disk modules. Each module has a user storage capacity of 692 million 6-bit characters and uses a sectored addressing method of 644 characters per sector and 32 sectors per track.

The 885 unit provides access to data organized in tracks and stored on two rotating disk modules. The disk module consists of a stack of twelve 14-inch diameter, oxide-coated disks. For each module data is accessed by 40 read/write heads mounted on a carriage that is positioned to the required track by a voice-coil linear actuator operating in a closed loop, proportional-servo system.

MODELS

<u>Model</u>	<u>Description</u>
885-11	Single Controller Access Unit, Bit Serial FMD
885-12	Dual Controller Access Unit, Bit Serial FMD

OPTIONS

10396-1	Provides a second controller connection to both spindles of an 885-11 Disk Storage Unit making it equivalent to the 885-12.
---------	---



1711 TELETYPEWRITER

The CDC 1711 Teletypewriter includes a keyboard and page printer which permits the manual insertion of both data and instructions into computer memory and the retrieval and printout of data from computer memory.

HIGHLIGHTS

- The pin-feed platen provides positive action
- A pneumatic shock absorber provides smooth, quiet carriage return that prolongs the life of its mechanical components
- Nylon gears and pulleys at critical points ensure smooth, dependable operation
- All steel clutches provide slip-free operation
- Four row keyboard enables easy operation for the typist
- Uses a standard nylon typewriter ribbon

DESCRIPTION

The 1711 Teletypewriter functions as a rugged send/receive unit that operates at 100 words per minute. Each character typed on the keyboard is translated into a standard eight-level code. Data from the 1711 Teletypewriter is transmitted to the computer character by character. Similarly, as each eight-level code is received from the computer, the 1711 Teletypewriter translates the code and prints the corresponding characters.

MODELS

The 1711 Teletypewriter is available in the following models:

<u>Model</u>	<u>Features</u>
1711-4	33 KSR, 100 wpm, keyboard and printer. Uses 64-character subset of ASCII.
1711-5	35 KSR, 100 wpm, keyboard and printer. Uses 64-character subset of ASCII.



1713 TELETYPEWRITER

The CDC 1713 Teletypewriter includes a keyboard, a character printer, a paper tape punch, a paper tape reader, and a controller. The 1713 Teletypewriter permits the manual insertion of both data and instructions into computer memory and the retrieval and printout of data from computer memory.

HIGHLIGHTS

- All parts of the unit have been designed for long life with minimum maintenance
- Four row keyboard enables easy operation for the typist
- Uses a standard nylon typewriter ribbon

DESCRIPTION

The 1713 Teletypewriter functions as a rugged send/receive unit that operates at 100 words per minute. Each character that is typed on the keyboard is translated into a standard eight-level code. Data from the 1713 Teletypewriter is transmitted to the computer character by character. Similarly, as each eight-level code is received from the computer, the 1713 Teletypewriter translates the code and prints the corresponding character.

MODELS

The 1713 Teletypewriter is available in the following models:

<u>Model</u>	<u>Features</u>
1713-4	33 KSR, 100 wpm, keyboard, printer, paper tape reader and punch. Uses 64-character subset of ASCII.
1713-5	35 KSR, 100 wpm, keyboard, printer, paper tape reader and punch. Uses 64-character subset of ASCII.



1811-1 CONVERSATIONAL DISPLAY TERMINAL

The CDC 1811-1 Conversational Display Terminal is a self-contained, single station CRT keyboard display terminal. It provides a 1920-character (24 lines of 80 characters) display unit with character-by-character, line-at-a-time, or page-at-a-time data transmission in either half- or full-duplex modes. The data rate is selectable from 110 to 9600 bits per second. The interface meets RS232-C, CCITT recommendations V24 as applied to asynchronous data communication. The character repertoire includes 128 symbols displayed within a 9 by 7 dot matrix.



1827-30/31 LINE PRINTER

The 1827-30/31 is a 300 line-per-minute drum printer in a quietized cabinet. It provides full line buffer facility, a 136-column print line, 12 VFU channels, a 64-character print drum, and six or eight line-per-inch print line spacing. The 1827-30/31 includes 20 feet of interface cable. The 1827-30 operates from 120 vac, 60 Hz source power. The 1827-31 operates from 220 vac, 50 Hz source power.



1827-60 LINE PRINTER

The CDC 1827-60 Line Printer is a high-speed band printer designed for use with the CDC CYBER 18/Models 5, 10, 20, and 30 computer systems. The internal control logic primarily consists of TTL-integrated circuits plus an MOS static shift register memory which stores one full line of print data. The heavy-duty hammers are driven by LSI logic. Sound-dampened cabinetry and automatic motor turn-off after 30 seconds without printing contribute to the quiet operation of the 1827-60.

The print mechanism consists of one hammer per two print columns. Sharing is accomplished by shifting the hammer bank between positions using a servo-controlled voice coil. The print band is easily changed.

The printing ribbon is 2 inches wide by 48 yards long. Ribbon velocity is constant in either direction regardless of spool load.

FEATURES

- 600 lines per minute
- Character set size – 64 ASCII
- Quietized cabinet
- 132 columns, 10 characters per inch
- 6 or 8 lines per inch
- Designed for low-grade or recycled paper
- Prints up to six-part forms
- Test print maintenance feature with fault indicators
- 50/60 Hz option

OPERATION

Control and data signals from the controller permit printing and/or paper motion using up to six-part forms. The 1827-60 prints 600 lines per minute, achieved with the standard character set size (64 ASCII) at a single-line forms advance rate. Paper motion without printing may be controlled to space between lines or to initialize to top-of-form.



1828-1 CARD READER/ LINE PRINTER CONTROLLER

The CDC 1828-1 Controller provides two independent controllers for connection of one card reader and/or one line printer to the processor unit; it occupies one A/Q card position within the processor unit. Card reader controller features are data/control interface between the processor and one card reader; it accepts card reader input data in form of Hollerith code, binary code, or any other desired format; it provides facility for deadstart operation of the processor unit; it performs Hollerith-to-ASCII code conversion during deadstart operation; and it performs normal data transfers under program control. The line printer controller features are data/control interface between the processor and one line printer; it performs data transfers under program control; and it has data buffer facility. An additional feature is the test mode capability of closed-loop operation under software control for diagnostic purposes. Cables are included as part of the card reader or line printer.



1829-30 CARD READER

The CDC 1829-30 Card Reader is a self-contained desk top unit provided with interface control logic and an operator's control/indicator panel. Functional characteristics of the unit are 300 card-per-minute read speed, 1000 card hopper/stacker capacity, an 80 column punch card input medium, and a photoelectric read station with light/dark read checking facility. One 7-foot interface cable is supplied. The 1829-30 operates from 120 Vac, 60 or 50 Hz source power. Option 1881-1 is available for 220 Vac operation.



1832-4 MAGNETIC TAPE CONTROLLER

The 1832-4 provides interface and control for up to four drives in any mix of seven- or nine-track. Both seven- and nine-track operation is 800 bits per inch NRZI mode at 25 inches per second. The controller features functional capabilities of read record, write record, write filemark, backspace, erase, rewind to load-point, rewind unload, recovery read, controlled backspace, and single track error correction for nine-track units. It occupies one position within the processor unit. It includes a 30-foot interface cable and a tape translate board.



1833-1 STORAGE MODULE DRIVE INTERFACE

The CDC 1833-1 Storage Module Drive Interface provides a single CPU A/Q-DMA channel interface to the 1833-3 Storage Module Control Unit. The interface handles all control and status operations via the A/Q channel and all data transfers via the DMA channel. The interface supports the control unit connection of up to eight CDC 1867-xx Storage Module Drives in any mix. Connection to the 1833-3 control unit is via two 25-foot (7.62m) cable assemblies. The interface occupies one A/Q-DMA position within the processor unit.



1833-3 STORAGE MODULE DRIVE CONTROL UNIT

The CDC 1833-3 is the control unit for the CDC 1867-10/11/20/21 Storage Module Drive. It provides control for up to eight drives in any mix of 25 million 8-bit bytes and 50 million 8-bit bytes of formatted data capacity. The control unit handles all SMD data transfers, formatting, and error recovery. It provides for either single or dual CPU connection via the CDC 1833-1 and 1833-2 SMD interface. The control unit is physically housed in the base cabinet of the first SMD in the subsystem. Input power may be either 50 or 60 Hz, 120 V or 220 Vac, single phase.



1843-1 COMMUNICATIONS LINE ADAPTER

The CDC 1843-1 Communications Line Adapter provides multiplexed dual channel interfaces for connection of two synchronous or asynchronous modems that conform to CCITT recommendations V24 or EIA RS232-C interface standards. A selectable baud rate of 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19,200 in asynchronous mode or 1200, 2400, 4800, and 9600 in the synchronous mode may be selected. The 1843-1 features software selection and control of half-duplex or full-duplex operation, character code lengths of 5, 6, 7, or 8 bits, stop bit length of 1x, 1.5x, or 2x, odd, even, or no parity bit generation and checking, and variable input/output speeds. An additional feature is an internal cyclic encoder for cyclic checkword generation. The 1843-1 adapter occupies one A/Q position within the processor unit. One 20-foot modem cable is supplied.



1860-3 MAGNETIC TAPE TRANSPORT

The CDC 1860-3 Magnetic Tape Transport is a nine-track unit that operates in 800 bit-per-inch NRZI mode at 20K eight-bit characters per second. It rewinds at 150 inches per second.



1860-92 MAGNETIC TRANSPORT

The CDC 1860-92 is a nine-track magnetic tape transport that operates in 800-bit-per-inch NRZI mode at 20K eight-bit characters per second. It rewinds at 150 inches per second. The transport does not include skins and must be housed in an 1887-4 cabinet. It requires an installation kit (1860-200 for upper cabinet installation or 1860-201 for lower cabinet installation). The 1860-92 operates from 120 Vac, 50 to 60 Hz source power. Option 1888-1 is available for 200 Vac operation.



1865 FLEXIBLE DISK DRIVE

The CDC 1865 Flexible Disk Drive is a random-access device using removable diskettes for the storage medium. It has a formatted data capacity of 256K bytes when using the IBM format (128 bytes/sector) or 280K bytes using the CDC 1700 format (192 bytes/sector). The data transfer rate is 31.2K bytes per second. The average access time is 343 milliseconds. The 1865-1 is the first drive (unit 0) within an FDD subsystem and connects to the CDC 1833-5 via a 10-foot unit 0 cable. The CDC 1865-2 Flexible Disk Drive is the second unit in a dual flexible disk drive subsystem. The CDC 1865-3 and 1865-4 Flexible Disk Drives operate from 220 Vac, 50 Hz source power and are unit 0 and unit 1, respectively, in a FDD subsystem.



1867-10 STORAGE MODULE DRIVE

The 1867-10 Storage Module Drive (SMD) is a random-access device, using removable disk packs as the storage medium. It has a formatted data capacity of 25 million bytes. The maximum data transfer rate is 1.2 million bytes per second. The average access time is 30 milliseconds. The drive includes a base cabinet, one 10-foot A cable (daisy chain) and one 20-foot B cable. The 876 disk pack is not included. The 1867-10 operates from 120 vac, 60 Hz source power. The 1867-11 is available, which operates from 220 vac, 50 Hz source power.



1867-20/21 STORAGE MODULE DRIVE

The CDC 1867-20 Storage Module Drive is a random-access device using removable disk packs for the storage medium. It has a formatted data capacity of 50 million bytes. The maximum data transfer rate is 1.2 million bytes per second. The average access time is 30 milliseconds. The drive includes the base cabinet, one 10-foot A cable (daisy chain) and one 20-foot B cable (star). The CDC 877 Disk Pack is not included. The 1867-20 operates from 120V ac, 60 Hz source power. The 1867-21 is available, which operates from 220 Vac, 50 Hz source power.



1875-1 BREAKPOINT CONTROLLER

The 1875-1 Breakpoint Controller provides the breakpoint halt register for both micro memory and main memory for program debug. It also provides a hardware interface to the micro processor. This allows the operator to load and display all registers and interface to the function control register, which allows the setting of all control bits. When the 1875-1 is not installed, all functions except breakpoint are emulated by the panel simulator of the 1700 emulator. Operator interface is via the console display or the breakpoint panel. The controller occupies one dedicated card position within the processor.



1875-2 BREAKPOINT PANEL

The CDC 1875-2 Breakpoint Panel provides the operator with an input medium to the breakpoint controller in the absence of the console display. It contains a 16-bit LED display and a limited keyboard. It receives parallel input from the breakpoint controller. The panel does not require a processor card position.



255X NETWORK PROCESSING UNIT

The CDC 255X Network Processing Unit (NPU) relieves the host computer (CDC 6000, CYBER 70, or CYBER 170) of data communication handling responsibilities, thereby freeing it for additional application functions. The NPU is comprised of modular hardware and combined with specifically tailored software for the efficient handling and processing of data communications in a network environment. This modularity affords a user the ability to expand the system to keep pace with increasing network communication requirements. Some of the features of the NPU include:

- Allows use of a wide range (in terms of characteristics and vendors) of terminal devices
- Communication Control Program Software provides intercommunication with NOS
- Provides significant throughput capability
- Assumes data buffering requirements from the host computer

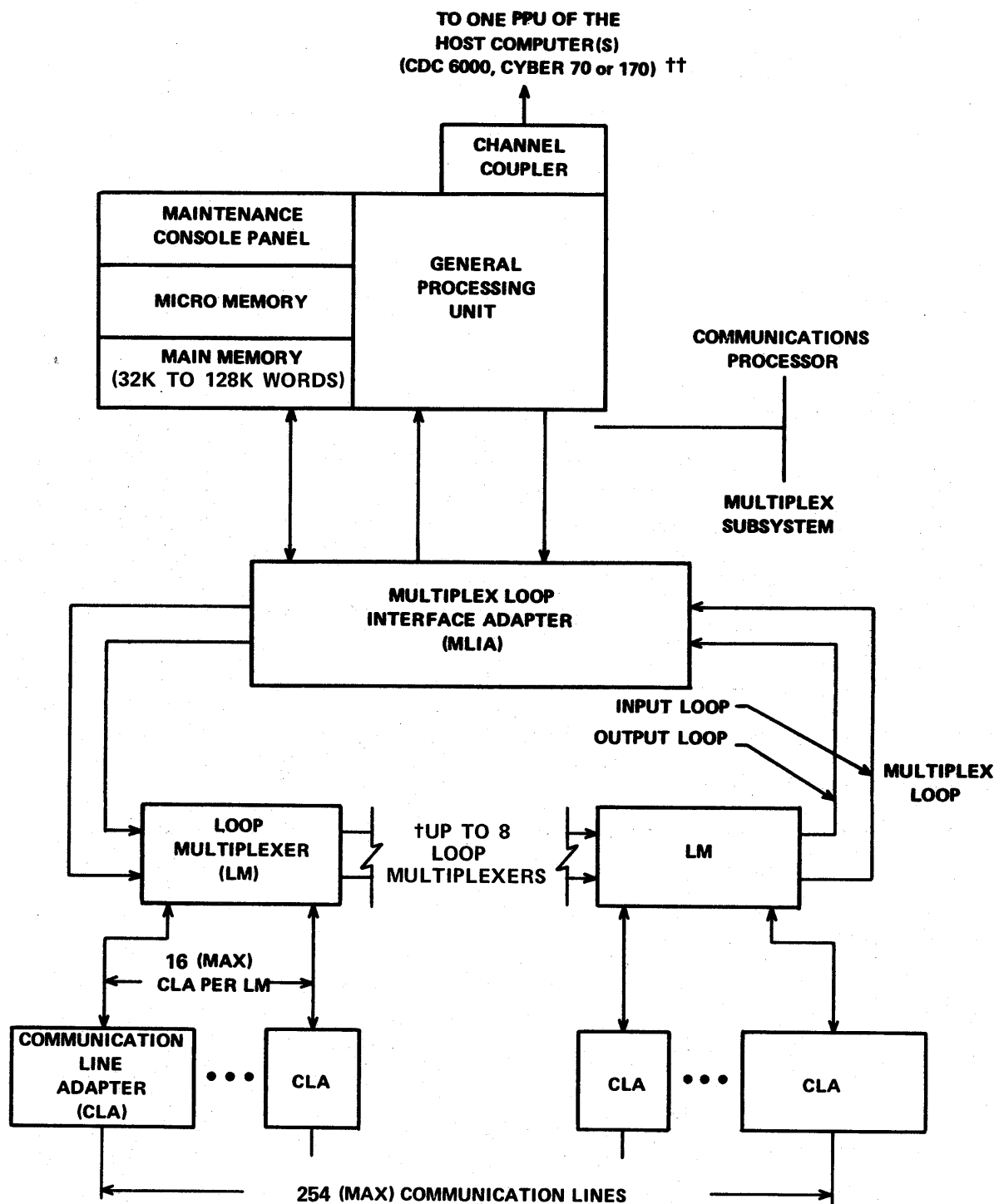
SYSTEM CHARACTERISTICS

The 255X NPU products consist of the following elements:

- Communications processor
- Multiplex subsystem
- Channel coupler

Communications Processor

- 32K to 128K of 16-bit (plus 1 parity and 1 protect bit) words of MOS main memory
- Main memory cycle time of 475 nanoseconds
- Microcode instruction (program control) cycle time of 168 nanoseconds
- Eight memory addressing modes
- Instruction repertoire containing 14 instruction groups
- Memory word and region protection
- Main memory parity detection
- 15 levels of external interrupt and one internal interrupt
- Interrupt data channel transfer rate of 160,000 bytes/second



255X NETWORK PROCESSING UNIT

† Depending on model.

†† Option available to add second CDC CYBER channel coupler for 255X.

Channel Coupler

- Direct memory access transfer rate of 1,600,000 words/second

Multiplex Subsystem

- Input and output loop rates of 20,000,000 bps
- Cyclic check character generated on output loop and verified on input loop
- Demand driven communication line adapters
- Synchronous and asynchronous transmissions handled

COMMUNICATIONS PROCESSOR

The heart of the communications processor is the central processing unit. It is an integrated circuit, fully parallel unit which supplies the communications processor with basic stored program control and computational capabilities. It contains the facilities for fetching and storing data, provides microcode control for sequencing and executing instruction, has capabilities for arithmetic and logical processing data, and initiates and emulates the interrupts that constitute communication between input/output devices. The communications processor has a special set of communications-oriented instructions.

The communications processor monitors connected communication lines and takes appropriate alarm and safety action in the event of line or terminal faults.

The communications processor contains a maintenance console panel to implement the following:

- Display of register contents
- Display of memory locations
- Location of operational switches and indicators

CHANNEL COUPLER

The channel coupler controls the direct exchange of data between a peripheral processor unit in each of up to two host computers (with optional coupler) and the communication processor memory. Multiple couplers may share a host computer data channel or two couplers may be connected to a communication processor, thereby allowing multiple NPU or multiple host computer configurations for the purpose of redundancy.

The primary function of the channel coupler is to pass data blocks between the main computer and the communication processor with minimum software supervision. The coupler contains provisions for data protection and error detection.

MULTIPLEX SUBSYSTEM

The multiplexing subsystem uses a demand driven technique wherein data movement between a line and the communications processor's memory occurs at the instigation of a communication line adapter; not limited by scanning rate used in other techniques. The elements comprising the subsystem are described in the following paragraphs.

Multiplex Loop

The multiplex loop is the demand driven mechanism which gathers input data and status information from communications lines and distributes output data and control information to the lines. The loop is composed of two high-speed serial-transmission links: one for input, one for output. Data cycles around the links are at a speed of 20 million bits per second. The loop is controlled by the communication processor via the multiplex loop interface adapter.

Multiplex Loop Interface Adapter (MLIA)

The MLIA interfaces between the loop multiplexer and either the communication processor or the multiplex loop controller. It converts bit-serial loop data from the input loop to bit-parallel for storage in the communication processor's memory, and bit-parallel data from storage to bit-serial for placement on the output loop. The MLIA ensures data integrity on the loops and controls data exchange with the loop multiplexer(s).

Loop Multiplexer

The loop multiplexer manages data flow between the multiplex loop and associated circuits. Up to eight loop multiplexers may be connected to a single multiplex loop, allowing a maximum of 254 communications lines to be terminated per NPU.

Communication Line Adapter (CLA)

CLAs interface communication lines to the loop multiplexer and provide in-transit single byte buffering, two classes of CLAs are available:

- Asynchronous interfacing circuits operating at all standard speeds within 9600 bps.
- Synchronous character-oriented interfacing circuits operating at speeds to 56,000 bps.
- Synchronous bit-oriented interfacing circuits operating at speeds to 19,200 bps.

The CLAs themselves demand data input and output. The asynchronous CLA supports autospeed detection, therefore a user need not dedicate a CLA for a particular line speed and/or terminal type. When data is being received, a programmed detection feature tests the line speed and code (ASCII, BCD, etc.), facilitating proper handling of the data by the NPU.

MODELS

The 255X Network Processing Unit is available in the following models and options:

<u>Model</u>	<u>Feature</u>
2551-1	<p>Memory: includes 32K words of 475 nanosecond, 16-bit main MOS memory.</p> <p>Throughput Capacity: nominally rated at 10,000 characters per second at the CDC CYBER channel coupler. Actual throughput is dependent on a number of factors. These include: NPU and software, number of circuits connected, block length, circuit activity, circuit speed, and available memory.</p> <p>Multiplexer Capacity: includes loop multiplexers for interfacing 16 communication line adapters (up to 32 communication lines).</p> <p>Expansion Capacity –</p> <p>Maximum Connectability: sixteen communication line adapters (CLAs) which define up to 32 communication lines (max), depending on the types of CLAs used.</p> <p>Maximum Main Memory: 128K words (with additional 2554-X Memory Expansion Units).</p> <p>Options –</p> <p>Channel Coupler: CDC 2558-3, 2558-4 or 10344-1 Channel Couplers provide host computer input/output channel interface for the 2551-1 NPU. The 2551-1 can control up to two channel couplers. At least one 2558-3, 2558-4 or 10334-1 Channel Coupler is required when the 2551-1 is used as a front-end processor.</p> <p>Upgrade: The 2580-3 Option is field-installable and upgrades the 2551-1 NPU to a 2551-2.</p> <p>Remote Loading: The 2580-4 Option provides the hardware necessary for automatic loading and restarting a remote NPU. This option is field-installable, and is required when the 2551-1 is used as a remote NPU.</p>
2551-2	<p>Memory: includes 32K words of 475 nanosecond, 16-bit MOS memory</p> <p>Throughput Capacity: nominally rated at 10,000 characters per second, measured at the channel coupler. Actual throughput is dependent on a number of factors. These include: NPU and host software, number of circuits connected, block length, circuit activity, circuit speed and available memory.</p> <p>Multiplexer Capacity: includes loop multiplexer for interfacing 32 communication line adapters (up to 64 communication lines, depending on the type of CLA).</p>

Model

Feature

2551-2 (Continued)

Maximum Connectability: 127 Communication Line Adapters (CLAs) which define a maximum connectability of up to 254 communication lines. Expansion is attained by adding 2556-10 Expansion Cabinets and 2556-11 Loop Multiplexer Units.

Maximum Main Memory: 128K words (with additional 2554-16, -32 Memory Expansion Units).

Channel Couplers: CDC 2558-3 or 2558-4 Channel Couplers provide host computer input/output channel interface for the 2551-2 NPU. The 2551-2 can control up to two couplers. At least one 2558-3 or 2558-4 Channel Coupler is required when the 2551-2 is used as a front-end NPU.

Multiplexer Expansion: The 2556-10 Expansion Cabinet and 2556-11 Loop Multiplexer options are used for expansion beyond the basic 32 CLA capacity of the 2551-2 NPU. Each 2556-11 Loop Multiplexer will interface an additional 16 CLAs. A 2556-10 Expansion Cabinet is used to house the additional loop multiplexers. One expansion cabinet is required for every two 2556-11 Loop Multiplexers.

Remote Loading: The 2580-4 option provides necessary hardware for the automatic loading and restoring of a remote NPU. It is required when the 2551-2 is used as a remote NPU. This option is field-installable.



2550-101 6671/6676 EMULATION CONTROLWARE

The CDC 2550-101 6671/6676 Emulation controlware enables the 2551 NPU to emulate up to four 6671 or 6676 multiplexers when operating with standard host software products. Limited to a maximum of 128 communication lines regardless of the number of multiplexers emulated. Terminal support is dependent on host software.



2554-16/32 MEMORY EXPANSION

The CDC 2554-16 and 2554-32 Memory Expansion units provide an additional 16,384 or 32,768 words, respectively, of read/write MOS memory for the CDC 255X Network Processing Unit (NPU). Each word in the expansion unit consists of 16 data bits, one protect bit, and one parity bit. Cycle time is 475 nanoseconds.



2556-11 LOOP MULTIPLEXER EXPANSION

The CDC 2556-11 Loop Multiplexer Expansion provides an additional Loop Multiplexer which supports 16 CLAs or up to an additional 32 asynchronous or synchronous communication lines. The 255X can be expanded to a total of 127 CLAs or up to 254 lines by addition of up to six 2556-11s. The additional 2556-11s are housed in the 2556-10 expansion cabinet as required. There are two 2556-11s per expansion cabinet.



2558-3 COMMUNICATIONS COUPLER

The CDC 2558-3 Communications Coupler provides an interface between a CDC 255X Network Processing Unit (NPU) and a second CDC CYBER 70/170 series mainframe. The 2558-3 plugs directly into the NPU card cage and connects to the mainframe data channel via coaxial cable.



2558-4 EMULATION COUPLER

The CDC 2558-4 Emulation Coupler is used with the CDC 2550-101 Emulation Controlware to interface a CDC 2551 Network Processing Unit (NPU) to a second peripheral processor on the same or second CDC CYBER 170 series mainframe. The coupler plugs directly into the NPU card cage occupying card positions established for the CDC CYBER data channel coupler.



2560 SERIES COMMUNICATION LINE ADAPTER

The 2560 Series Communication Line Adapter (CLA) interfaces the CDC 255X Network Processing Unit (NPU) to various types of communication lines at the modem or data set.

Data is received from remote terminals in serial form via communication lines and placed in the CLAs single-character buffers. When a character is received, the loop multiplexer is informed and the character is retained until permitted to enter the central processor.

2560 SYNCHRONOUS CLAs

The 2560 series provides connection facilities for two synchronous communication lines at speeds up to 56,000 bits per second. They feature software selection of full duplex/half duplex, 5- to 8-bit code length (plus parity), frame synchronization and testing facilities.

Model 2560-1 accommodates lines conforming to the EIA RS232-C or CCITT V.24 Interface Standards at speeds up to 19,200 bits per second. This adapter is compatible with AT&T 201/208 Data Sets and provides interface for two lines. The adaptation of strapping and cable options permit local terminal connection without modem.

Model 2560-2 is similar to Model 2560-1, except that interconnection is provided for two lines which are compatible with AT&T Data Sets 301/303 and at speeds up to 50,000 bits per second.

Model 2560-3 is also similar to the other two models of this series, except that it allows connection of two lines which conform to CCITT Rec. V35 (AT&T Digital Data Network) at speeds up to 56,000 bits per second.

FEATURES

- Half-duplex operation†
- Full-duplex operation†
- Variable code length (5 to 8 bits plus parity per character)†
- Variable bit rate to 56K bps (determined by modem)†
- Transparent data handling
- Variable synchronization character†
- Loopback test mode†
- Provision for external modem clock source

†Program Controlled

- **Variable 8-bit CLA address**
- **EIA RS232/CCITT V.24 interface (Model 2560-1)**
- **AT&T 301/303 interface (Model 2560-2)**
- **CCITT V.35 interface (Model 2560-3)**
- **Two synchronous CLAs per card**
- **Modem interface connector on card handle**
- **Circuit sophistication (MSI and LSI)**
- **Modem indicator lamps on card handle**
- **Card size: 11 x 14 inches**
- **Local terminal connection optional**



2561 SERIES COMMUNICATION LINE ADAPTER

The CDC 2561 Series Communication Line Adapter (CLA) provides connection facilities for two asynchronous communication lines at all standard speeds up to 9,600 bits per second. This series features software selection and control of half-duplex/full-duplex/echoplex operation; code lengths of 5, 6, 7 and 8 bits plus parity; variable speed; even, odd, or no parity operation; stop bit duration of 1.0, 1.5 or 2.0 units; and self-testing capability.

The Model 2561-1 provides interconnection facilities for two lines conforming to EIA RS232C or CCITT Rec. V24 Standards. This adapter is compatible with AT&T 103/113/202 Data Sets or their equivalents. Local connection without modem is also offered.

FEATURES

- Half-duplex operation†
- Full-duplex operation†
- Echoplex operation†
- Variable code length (5 to 8 bits plus parity per character)†
- Variable baud rate (45 to 9600 baud – all standard speeds)†
- Input and output speeds may vary†
- Stop bits: 1, 1.5 or 2†
- Even, odd, or no parity check on input†
- Even, odd, or no parity generation on output†
- Self test mode†
- Break detection and generation†
- Reverse channel detection and control†
- Data transfer over-run detection†
- Variable 8-bit CLA address
- Full EIA RS232-C/CCITT V.24 interface
- Two lines per CLA card
- Modem interface connector on card handle
- Modem indicator lamps on card handle
- Circuit sophistication (MSI and LSI)
- Card size: 11 x 14 inches

†Program controlled



2563 SERIES COMMUNICATIONS LINE ADAPTER

The CDC 2563 Series Communications Line Adapter (CLA) provides for the connection of one communication line utilizing an HDLC type protocol at speeds up to 19,200 bits per second. The 2563-1 provides for connection via modems conforming to the EIA RS232C or CCITT V.24 interface standards.

FEATURES

- Half/full duplex operation
- Variable bit rate (determined by modem)
- Transparent data
- Loopback self-test mode
- Provision for external modem clock source
- Variable 8-bit CLA address
- EIA RS232/CCITT V.24 interface
- Modem signal indicator lamps
- Local terminal connection



3270/8271 TRANSFER SWITCH SUBSYSTEM

The CDC 3270/8271 Transfer Switch Subsystem provides for switching of peripheral controllers between channels of the same computer system, or between channels of co-located computer systems. The switching operation is controlled manually by an operator.

The 3270 Transfer Switch Controller provides cabinetry for mounting 8271 Transfer Switches and includes power supply necessary to activate switching operation. The 8271 Transfer Switch is a set of relay-operated switches controlled by an externally mounted alternate-action momentary contact switch.

MODELS

The 3270/8271 Transfer Switch Subsystem consists of the following models:

<u>Model</u>	<u>Features</u>
3270-A	Controls up to 4 transfer switches
3270-B	Controls up to 8 transfer switches
8271-2	Manually operated transfer switch



3446 CARD PUNCH CONTROLLER

The Control Data 3446 Card Punch Controller is a single channel device, with memory, which provides an interface between the 415 Card Punch and a computer data channel.

HIGHLIGHTS

- Provides buffer memory for one 80-column card
- Permits translating binary, Hollerith, or ASCII (64-character subset) codes
- Logic to perform most of the controlling, checking, conversion, and formatting functions that would otherwise be performed by the computer

DESCRIPTION

The 3446 controller acts as an interface between the computer and the 415 punch in the transmission of all data, external function codes, and status codes. The 3446 controller receives data from the computer in 12-bit bytes and is capable of translating binary, Hollerith, or a 64-character subset of ASCII Hollerith codes.

The controller contains a 12 by 80 turnaround core memory for buffering. The controller forms a card image in a buffer memory from a series of 12-bit bytes received from the data channel. The card image is formed on a column-by-column basis, beginning with the lower-order column. With the card image in memory, the controller then transfers the data to the card punch. The punch then punches the card on a row-by-row basis.

MODELS

The 3446 controller is available in the following models:

<u>Model</u>	<u>Features</u>
3446	BCD to binary, or Hollerith codes
3446-2	BCD to binary, Hollerith or ASCII (64-characters) codes



3447 CARD READER CONTROLLER

The CDC 3447 Card Reader Controller is a single channel device, with a full card buffer memory, which provides an interface between the 405 Card Reader and a computer input/output channel. It is contained within the 405 Card Reader cabinet.

HIGHLIGHTS

- Provides buffer memory for one 80-column card
- Logic to perform most of the controlling and checking functions that would otherwise be performed by the computer
- Performs translation of binary, Hollerith, or ASCII (64-character subset) codes

DESCRIPTION

The 3447 controller acts as an interface between the computer and the 405 reader in the transmission of all data, external function codes, and status codes. The controller transfers data to the computer in 12-bit bytes. The logic of the controller also checks the parity of each code and all data bytes. If an error is detected in a code, the controller sets an error signal and does not act on the code. If an error is detected in a byte, an error signal is generated. The controller is capable of translating binary, Hollerith, or a 64-character subset of ASCII Hollerith card codes to 6-bit internal BCD-codes.

The controller contains a buffer memory that holds 80 bytes (one card image). The reader enters data into this memory at the relatively slow rate of 50 milliseconds per card. After a complete card has been read, the 3447 transfers the data (byte-by-byte) at a rapid rate (maximum of 390 microseconds per card). The 405 reader automatically reads another card after the first card has been transferred.

MODELS

The 3447 controller is available in the following models:

<u>Model</u>	<u>Features</u>
3447	Binary or Hollerith to BCD codes
3447-2	Binary, Hollerith, or ASCII (64-characters) to BCD codes



6671-3 DATA SET CONTROLLER

The CDC 6671 Data Set Controller interfaces the CDC CYBER 70-, the CYBER 170-, or the 6000-series data channels to as many as 16 telecommunication lines.

HIGHLIGHTS

- Operates in synchronous or asynchronous modes
- Operates in both half- or full-duplex
- Speeds of either 110, 134.5, 150, 300, 600 or 1200 baud asynchronous and 2000, 2400, 4800 or 9600 bps synchronous

DESCRIPTION

The 6671 controller permits communications between a remote terminal and a central computing facility. The remote terminal can be located adjacent to the computer, or several thousand miles away, providing communication channels can be serviced. Serial data rates of 110, 134.5, 150, 300, 600, 1200, 2000, 2400, 4800, or 9600 bps are permitted. The 6671 is capable of transferring 4800 bps on each of sixteen data channels simultaneously.

The 6671 controller can operate in both half- and full-duplex modes. In half-duplex mode, serial data transfer is in one direction at a time (either receive or transmit). In full-duplex mode, serial data transfer is in both directions at the same time.



6673 DATA SET CONTROLLER

The CDC 6673 Data Set Controller interfaces the CDC High-Speed Remote Job Entry Station or High-Speed Graphics stations to either a CDC CYBER 70/170 or 6000 series computer system. The controller can accommodate two remote systems and is designed to permit fast, efficient remote batch processing over broad-band communications lines, coaxial cable, or microwave facility.

HIGHLIGHTS

- Transmits data in half-duplex mode with control signals and response indications utilizing the full-duplex capability of the data set
- Transmission rates up to 56,000 bits per second
- Twelve-bit cyclic code is used for encoding and decoding blocks of data in error detection

DESCRIPTION

The 6673 controller is used mostly in conjunction with a Bell System wide-band lines. In this case, the controller connects the remote computer system to a CYBER 70/170 or 6000 series data channel through Bell 301B or 303 Data Sets. A data set is required both at the remote and at the central station. The controller converts 12-bit parallel words from the central computer to serial bits, which are transmitted through the central Bell data sets to the remote system over the wide-band line. A remote controller then receives the data from the remote data set and reconverts the serial data into 12-bit parallel words for terminal input.

In error detection, the 6673 controller automatically checks for errors when receiving data blocks using a 12-bit cyclic code.



6676 DATA SET CONTROLLER

The CDC 6676 Data Set Controller interfaces with up to 64 communication terminals to a CDC CYBER 70, CYBER 170, or 6000 series computer system.

HIGHLIGHTS

- Operates in both half-duplex mode or full-duplex mode.
 - in half-duplex mode, data transfer between the controller and the terminals is in one direction at a time (either receive or transmit)
 - in full-duplex mode, data transfer is in both directions at the same time
- Services a maximum of 64 communications channels at serial data rate of 110 bits per second. Additional bit rates can be accommodated with special option 10294-1.

DESCRIPTION

The 6676 Data Set Controller is designed for remote terminal communications with a large central computing facility. With its software package, it allows an immediate contact between the terminal and the computer, permitting maximum on-line access to the central facility. The remote terminal can be located adjacent to the computer or several thousand miles away, wherever a telephone connection is available. At remote sites, compatible data sets are used to connect Model 33 or 35 teletypewriters and CDC TTY-compatible terminals.



6681 DATA CHANNEL CONVERTER

The CDC 6681 Data Channel Converter allows CDC 3000 series peripheral equipment to be attached to a CYBER 70, CYBER 170, or 6000 series data channel. The CDC CYBER 70, CYBER 170, and 6000 series computers can perform Connect, Function, Read, Write, and Status operations on these equipments via the 6681 converter.

DESCRIPTION

The CDC CYBER 70, CYBER 170, or 6000 series computer transmits codes to the 6681 converter prior to starting any operation on an external equipment. These codes establish conditions in the converter so that the proper 3000 type signals accompany the CDC CYBER 70, CYBER 170, or 6000 series computer input/output operations. Up to eight peripheral equipments may be connected to the output of the 6681 converter.



6683 SATELLITE COUPLER

The CDC 6683 Satellite Coupler allows direct 12-bit data channel communication between two CDC CYBER 70, CYBER 170, or 6000 series computer systems for a fast and effective intercommunication of data. It is commonly used in conjunction with multmainframe software. Two 6683s interface the computers (up to a distance of 1000 feet). One 6683 can also be connected to a 7683 to permit communication between a 6000 data channel and a 7000 data channel. The 6683 is also used to connect a CDC CYBER or 6000 series computer to a CDC STAR coupling station.

MODELS

The 6683 Satellite Coupler is available in two models: the 6683-1 and the 6683-2. The 6683-2 includes channel parity logic useful in CDC CYBER 170 interconnection.



7012 DISPLAY CONSOLE

The CDC 7012 Display Console consists of a single cathode-ray tube display and a keyboard for manual entry of data. The console keyboard performs the functions of a typewriter. The keyboard contains the alphabet, the numerals 0 through 9, period, comma, asterisk, slash, plus and minus signs, space, carriage return, left and right parentheses, and equal sign.

Multiple consoles may be used on a single CDC CYBER 170 Series Computer System in order to control independent programs simultaneously.

The console may be selected to display in either the character or dot mode.

Variable character sizes: 64, 32, or 16 characters per line.

The 7012 Display Console permits the display of data from a CDC CYBER 170 Series Computer System. The display console consists of one 21 inch diagonal cathode ray tube display unit and manual keyboard containing 50 alphanumeric and special characters. The consoles may be selected to display in either single or split screen by a toggle switch. The split screen display simulates the dual screen display of the 6612 console. The display system must be reselected each time an output is intended for a different CRT.

Under the standard operating system, the operator's console 7012 includes:

- The display of all registers.
- The dynamic display of any word or block of words in memory.
- Six sense switches which can be tested under program control.
- The capability to manually start execution from any given location.
- The capability to manually execute single steps of a program.
- The DAYFILE may be displayed. The DAYFILE is a log of the current and past status of activity within the system relative to individual jobs.
- The status of all central processor and peripheral processor activity is continuously displayed.
- The display console may be further programmed by the user to provide specialized displays.
- Monitor input/output queues, change program priorities and review all aspects of system status.



7021-2X MAGNETIC TAPE CONTROLLER

The Control Data 7021 Magnetic Tape Controller is an interface between a Control Data CYBER 70/ Models 71, 72, 73, and 74, CYBER 170/Models 172, 173, 174, and 175, or a 6000-series computer systems and as many as eight Control Data Models 667 and 669 Magnetic Tape Transports (intermixed).

HIGHLIGHTS

- Programmable processors (controlware-non-alterable software) enhance normal operation and provides diagnostic and recovery capabilities.
- Handles both 7- and 9-track tape transports recorded either NRZI or phase-encoded at 556 bpi and 800 bpi NRZI or 800 bpi and 1600 bpi PE.
- Tape transports are connected to the controller in parallel allowing individual transports to be switched off-line for running diagnostic routines.

DESCRIPTION

The 7021 reads or writes on both 667 and 669 tape transports. Reading or writing of tape is at 556 bpi or 800 bpi NRZI or 800 bpi or 1600 bpi PE. Single or dual programmable processors (dependent on model) provides diagnostic and recovery capabilities ensuring a high level of throughput and recovery. The controller can perform character-code translation during data transfer with no degradation of the character transfer rates. Various assembly/disassembly modes of operation for data are also provided. Dual coupler access accommodates two simultaneous data transfers (read and/or write) to any two tape transports.

MODELS

The 7021 controller is available in the following models:

<u>Model</u>	<u>Feature</u>
7021-1, 21	Single Processor 1 x 8 Tape Transports Single Coupler
7021-2, 22	Dual Processors Dual Coupler Access 2 x 8 Tape Transports



7021-3X MAGNETIC TAPE CONTROLLER

The CDC 7021-3X Magnetic Tape Controller provides an interface between the CDC 677 and 679 Magnetic Tape Transports and the CDC CYBER 170, CYBER 70, or 6000 series computer system input/output channels. Up to eight 67X transports of any model type can be intermixed on a 7021-3X controller and will automatically be recognized by model type and addressed accordingly.

HIGHLIGHTS

- Transports are connected to the controller in parallel, making it possible to switch individual drives off-line for maintenance purposes.
- State-of-the-art modular design along with the latest in integrated circuitry results in a subsystem that is highly reliable and easy to maintain.
- Comprehensive on-line/in-line diagnostic software.
- Built in maintenance panels and internal data loop checking capabilities.

DESCRIPTION

The 7021-3X reads or writes tapes on both the 677 and 679 series of tape transports. Such a subsystem can read or write tapes using industry standard NRZI, phase-encoded, or group coded recording (GCR) methods of recording, depending on tape transport selection. All subsystems can perform character-code translation during data transfer with no degradation of the character transfer rates specified for the various tape transports. Various assembly/disassembly modes of operation for data are also provided. Advanced technology in the 6250 BPI GCR tape systems helps boost system throughput, reduces the frequency of input/output errors, and significantly improves error recovery. Error recovery includes two-track, "on-the-fly" error correction in GCR mode and single track "on-the-fly" error correction in phase-encoded mode, assuring a high level of subsystem performance.

MODELS

The 7021-3X is available in the following models:

<u>Model</u>	<u>Feature</u>
7021-31	Single microprogrammed control unit. 1 channel to up to 8 tape transports.
7021-32	Dual microprogrammed control units, 2 channels to up to 8 tape transports, two simultaneous data transfers.



7030 CYBER EXTENDED CORE STORAGE

The CDC 7030 CYBER Extended Core Storage (ECS) is a random-access word-organized mass storage device. ECS, through its controller, communicates with one or more CDC CYBER 70 and CYBER 170 series computer systems using both central memory and input/output channels. Two model series are available, ECS I and ECS II. The ECS II series uses integrated circuit technology.

HIGHLIGHTS

- Storage capacity from 125,952 to 2,097,152 (60-bit) words, depending on the model
- Transfer rate up to 10,000,000 (60-bit) words per second
- Data paths between ECS and either central memory or as many as four input/output channels or both

DESCRIPTION

Extended core storage consists of a magnetic core storage unit plus controllers. ECS serves as an extension to the computer central memory, and provides for instruction and information storage as well as serving as a high-speed input/output buffer. The ECS I storage unit is available in 1-, 2-, 4-, 8-, or 16-bank configurations with each logically independent memory bank containing 125,952 (60-bit) words. The ECS II storage unit is available in 2-, 4-, 8-, or 16-bank configurations, each bank containing 131,070 words. Each ECS bank has 5K of reserve memory location to minimize down time. The storage unit provides up to two million directly addressable 60-bit words. Eight 60-bit words are organized into a 488-bit data word in ECS (a parity bit is attached to each 60-bit word).

The ECS controller provides the data paths between central memory (60-bit) and the input/output channels (480-bit). The distributive data path (DDP) consists of 480-bit buffer register that connects ECS to an input/output channel. Three additional buffer registers may be added as options (Control Data DDP Register). Thus, four peripheral processors can access ECS simultaneously.

MODELS

The 7030 storage system is available in the following models:

<u>ECS I</u>		<u>ECS II</u>	
<u>Model</u>	<u>Feature</u>	<u>Model</u>	<u>Feature</u>
1	125K	102	262K
2	250K	104	524K
4	500K	108	1048K
8	1000K	116	2097K
16	2000K		

Distributive Data Path

The distributive data path provides a path of data flow between extended core storage and the peripheral processors. It allows fast peripheral processor access to data in extended core storage using an input/output channel, and greatly reduces the data traffic through the central memory. This reduces central memory conflicts and also reduces the overhead of the operating system. The distributive data path consists of one-to-four 480-bit buffer registers. Each register connects to a standard input/output channel for maximum overlap of data transfer. All four registers share a single access to extended core storage. This arrangement allows up to four peripheral processors to transfer data simultaneously to extended core storage at the maximum rate of the channel. A 480-bit extended core storage word is assembled from a 40-word peripheral processor block in about 43 microseconds.



7054 MASS STORAGE CONTROLLER

The CDC 7054 Mass Storage Controller serves as an interface between the CDC 844 Disk Storage Unit and the CDC CYBER 70, CYBER 170, or 6000 series computer systems.

HIGHLIGHTS

- Performs multiple-overlapped seek operations concurrently with one read or write operation
- Includes a small programmable processor
- Dual controllers in a subsystem provide two simultaneous data transfers (read or write or both)

DESCRIPTION

The 7054 controller can control up to eight 844 units. The basic 7054/844 mass storage subsystem has a capacity of 118 million 6-bit characters and uses a sectored-addressing method of 644 characters per sector and 24 sectors per track. The 7054-4X/844 Mass Storage Subsystem supports a mix of single-density drives as described and double-density drives each having a capacity of 237 million characters (per spindle). The controller includes a small, programmable processor which enhances the normal operation of the subsystem and provides for more diagnostics and recovery capabilities.

MODELS

The 7054 controller is available in six models:

<u>Model</u>	<u>Feature</u>
7054-1 and 21	Connects to one standard input/output channel. Drives up to eight 844-2 or 844-21 disks.
7054-2 and 22	Connects to one or two standard input/output channels. Drives up to eight 844-2 or 844-21 disks.
7054-41	Connects to one standard input/output channel. Drives up to eight 844-2 through 21 or 844-41 through 44 disks in a mix-or-match configuration.
7054-42	Connects to one or two standard input/output channels. Drives up to eight 844-2 through 21 or 844-41 through 44 disks in a mix-or-match configuration.



7152 MASS STORAGE/MAGNETIC TAPE CONTROLLER

The CDC 7152 Mass Storage/Magnetic Tape Controller cabinet houses the functions of two distinct controllers. The 7152 controller handles the interface between CDC 844 Mass Storage Devices/CDC 66X Magnetic Tape Units and the CDC CYBER 170 family of computers.

MASS STORAGE CONTROLLER

The mass storage controller connects to one standard input/output channel, and can control up to four 844-2 through -21 or 844-41 through -44 disks in a mix-or-match configuration. The mass storage subsystem supports a mix of single-density drives with a capacity of 118 million 6-bit characters and uses a sectored addressing method of 644 characters per sector and 24 sectors per track, and double-density drives each having a capacity of 237 million characters (per spindle). The controller includes a small, programmable processor which enhances the normal operation of the subsystem and provides for more diagnostics and recovery capabilities.

HIGHLIGHTS

- Performs multiple-overlapped seek operations concurrently with one read or write operation
- Includes a small programmable processor with 1K 12-bit data buffer
- Allows full device rated transfer speeds

MAGNETIC TAPE CONTROLLER

The magnetic tape controller connects to one standard input/output channel, and is a single microprogrammable control unit. The magnetic tape controller reads or writes on up to four 667 and/or 669 tape transports, 7-track and 9-track, respectively. Reading or writing of tape is at 556 cpi and 800 cpi NRZI for 7-track or 800 cpi NRZI and 1600 cpi PE for 9-track. The programmable processor provides diagnostic and recovery capabilities ensuring a high level of throughput and recovery. The controller can perform character-code translation during data transfer with no degradation of the character transfer rates. Various assembly/disassembly modes of operation for data are also provided.

HIGHLIGHTS

- Programmable processor enhance normal operation and provides diagnostic and recovery capabilities.
- Handles both 7- and 9-track tape transports recorded at 556 cpi and 800 cpi NRZI for 7-track or 800 cpi NRZI and 1600 cpi PE for 9-track.
- Tape transports are connected to the controller in parallel allowing individual transports to be disconnected for off-line diagnostic purposes.



7154 MASS STORAGE CONTROLLER

The CDC 7154 Mass Storage Controller serves as an interface between the CDC 844 Disk Storage Unit and the CDC CYBER 70, CYBER 170 or 6000 series computer systems.

HIGHLIGHTS

- Performs multiple-overlapped seek operations concurrently with one read or write operation
- Includes a small programmable processor with 1K 12-bit data buffer
- Dual controllers in a subsystem provide two simultaneous data transfers (read or write or both)

DESCRIPTION

The 7154 controller can control up to eight 844 units. The 7154/844 Mass Storage Subsystem supports a mix of single-density drives with a capacity of 118 million 6-bit characters and uses a sectored addressing method of 644 characters per sector and 24 sectors per track, and double-density drives each having a capacity of 237 million characters (per spindle). The controller includes a small, programmable processor which enhances the normal operation of the subsystem and provides for more diagnostics and recovery capabilities. On the CDC CYBER 170 the 7154 controller allows full device rated transfer speeds.

MODELS

The 7154 controller is available in four models:

<u>Model</u>	<u>Feature</u>
7154-1	Connects to one standard I/O channel. Drives up to eight 844-2 through -21 or 844-41 through -44 disks in a mix-or-match configuration.
7154-2	Connects one or two standard I/O channels. Drives up to eight 844-2 through -21 or 844-41 through -44 disks in a mix-or-match configuration.
7154-3	Connects up to 3 standard I/O channels. Drives up to eight 844-2 through -21 or 844-41 through -44 disks in a mix-or-match configuration.
7154-4	Connects up to 4 standard I/O channels. Drives up to eight 844-2 through -21 or 844-41 through -44 disks in a mix-or-match configuration.

OPTIONS

10365-1	Upgrades a 7154-1 to 2 standard I/O channel operation.
10367-1	Adds a third channel operation to 7154-2 or 7154-1 with a 10365-1 option. It also adds a fourth channel operation to a 7154-3, to a 7154-2 with a 10367-1, or a 7154-1 with 10365-1 and 10367-1.



7155 FIXED MODULE DRIVE CONTROLLER

The CDC 7155 Fixed Module Drive Controller serves as an interface between the CDC 885 Fixed Module Drive (FMD) and the CDC CYBER 70, CYBER 170, or 6000 series computer systems. The 885 FMD is a dual spindle device.

HIGHLIGHTS

- Performs multiple-overlapped seek operations concurrently with one read or write operation to maximize the data throughput rate of the subsystem.
- Includes a small programmable processor with a 2K 16-bit data buffer. The programmable capability of the subsystem enables data to be transferred at channel speed for one sector bursts.
- Dual controllers in a subsystem provide simultaneous data transfers (read or write or both).
- Provides three types of error recovery: head positioning, address field, and data field.
- Controlware supports subsystem function set, including maintenance function set.

DESCRIPTION

The basic 7155 controller can control up to four 885 units (eight spindles). The hardware allows for expansion of up to eight 885 drives (16 spindles), eight CDC 844-4X Disk Storage Units, and four channel accesses. The subsystem supports a mix of single-density drives with a capacity of 692 million 6-bit characters and uses a sectored addressing method of 644 characters per sector and 32 sectors per track, and double-density 844-4X drives each having a capacity of 237 million characters (per spindle). The controller includes a small, programmable processor which enhances the normal operation of the subsystem and provides for more diagnostics and recovery capabilities. On CYBER 170 mainframes, the 7155 controller allows full device rated transfer speeds.

MODELS

The 7155 controller is available in a basic model. Options may be added to enhance the capacity of the 7155.

<u>Model</u>	<u>Feature</u>
7155-1	Connects to one standard input/output channel. Drives up to four 885-xx FMDs (eight spindles) in a mix or match configuration.

OPTIONS

<u>Model</u>	<u>Feature</u>
10397-1	Provides an additional channel connection to a 7155 Mass Storage Controller. Up to three additional channel options may be added to the basic 7155-1 to provide a total of four channel accesses.
10398-1	This interface option connects up to eight 844-4x DSUs. Resulting controller drives up to four 885-xx FMDs (eight with 10399-1 option) and up to eight 844-4x DSUs in a mix or match configuration.
10399-1	This interface option connects up to four additional 885-xx FMDs. Resulting controller drives up to eight 885-xx FMDs (and eight 844-4x with 10398-1 option) in a mix or match configuration.



7622 MAGNETIC TAPE CONTROLLER

The CDC 7622 Magnetic Tape Controller is an interface between a CDC 7000 series or CDC CYBER 70, Model 76 Computer System and as many as eight CDC Models 667 or 669 Magnetic Tape Transports (intermixed) from one to four input/output channels.

HIGHLIGHTS

- Programmable processors (controlware-nonalterable software) enhance normal operation and provide diagnostic and recovery capabilities.
- Handles both 7- and 9-track tape transports recorded either NRZI or phase-encoded (PE) recording method, at 556 and 800 bits per inch NRZI or 800 and 1600 bits per inch PE.
- Tape transports are connected to the controller in parallel allowing individual transports to be switched off-line for running diagnostic routines.

DESCRIPTION

The 7622 reads or writes on both 667 and 669 tape transports. Reading or writing of tape is at 556 or 800 bits per inch NRZI or 800 or 1600 bits per inch PE. Single or dual programmable processors (dependent on model) provide diagnostic and recovery capabilities ensuring a high level of throughput and recovery. The controller can perform character-code translation during data transfer with no degradation of the character transfer rates. Various assembly/disassembly modes of operation for data are also provided. Dual coupler access accommodates two simultaneous data transfers (read and/or write) to any two tape transports.

MODELS

The 7622 controller is available in two models.

<u>Model</u>	<u>Feature</u>
1	Single Processor 1 x 8 Tape Transports Single Coupler
2	Dual Processors Dual Coupler Access 2 x 8 Tape Transports



7639 MASS STORAGE CONTROLLER

The CDC 7639 Mass Storage Controller provides a data and control path between the CDC 819 Disk Storage Unit and a High Capacity Disk (HCD) Station of a CDC CYBER 203 computer system. The CDC 7639 is available in a single or double controller interface (7639-21 and 7639-22, respectively).

HIGHLIGHTS

- Connects the disk storage unit
- Addresses the disk storage unit
- Assembles and disassembles data transferred between the 12-bit peripheral processor channel or HCD channel and the 4-bit disk storage unit
- Deskews read data from the disk storage unit
- Generates a preamble and sync byte that precedes each data sector
- Generates a 32-bit checksum for each channel

DESCRIPTION

The CDC 7639 controller can control up to eight CDC 819-1, 819-11, 819-21 disk storage units in any combination. The basic capacity of a CDC 819/7639 subsystem is 2.4 billion bits; however, it can be increased to 38.4 billion bits with a configuration that includes two CDC 7639-22 controllers and 16 single density disk storage units. Using dual density Model CDC 819 disk storage units, the minimum capacity is 4.8 billion bits, which can be increased to 76.8 billion bits with a configuration of two CDC 7639-22 controllers and 16 storage units.

The data transfer rate is nominally 3.1 million 12-bit words per second. The maximum possible burst transfer rate is 3.33 million 12-bit words per second.

MODELS

The CDC 7639 controller is available in two models each of which can handle the single or dual density CDC Model 819 disk drives:

- | | |
|---------|--|
| 7639-21 | Interfaces one to four disk storage units and permits multiple overlapped seek operations concurrently with one read or write operation. |
| 7639-22 | Interfaces to a maximum of eight drives and provides two simultaneous data transfers (read and/or write) while retaining the multiple-seek feature. Using the dual access feature of the CDC 819 disks, four drives can be interfaced providing two simultaneous data transfers to any two drives. |



7681 DATA CHANNEL CONVERTER

The CDC 7681 Data Channel Converter allows CDC 3000 series peripheral equipment to be attached to a 7602-1 Peripheral Processor. The 7600 computer can perform Connect, Function, Read, Write and Status operations on 3000 series peripherals via the 7681 converter.

DESCRIPTION

The 7600 computer transmits codes to the 7681 converter prior to starting any operation on an external equipment. These codes establish conditions in the converter so that the proper 3000 type signals accompany the 7000 series computer input/output operations. As many as eight peripheral devices can be connected to the output of the 7681 converter.



7654 MASS STORAGE CONTROLLER

The CDC 7654 Mass Storage Controller serves as an interface between CDC 844-2 or 844-21 Disk Storage Units and standard CDC 7000 and CDC CYBER 70/Model 76 series peripheral processors.

HIGHLIGHTS

- Performs multiple-overlapped seek operations concurrently with one read or write operation
- Includes a small programmable processor
- Dual controllers in a subsystem provide two simultaneous data transfers (read or write or both)

DESCRIPTION

The 7654 controller can control up to eight 844-2 or 844-21 disk units. The 7654/844 mass storage subsystem has a capacity of 108 million 6-bit characters (per spindle) and uses a sectored-addressing method of 4800 characters per sector and 3 sectors per track, 19 tracks per cylinder, and 404 cylinders per device. The controller includes a small, programmable processor which enhances the normal operation of the subsystem and provides for more diagnostics and recovery capabilities.

MODELS

The 7654 controller is available as a 7654-1 or 7654-21 (dual channel).



7683 SATELLITE COUPLER

The CDC 7683 Satellite Coupler allows direct communications between one 7602 Peripheral Processor unit and one 6000 Series data channel for a fast and effective inter-communication of data and programs between computers at computer speeds. One 7683 coupler is used with one 6683 coupler; each coupler interfaces its respective computer and the two satellite couplers interface each other (up to a distance of 1000 feet). This design allows a switching function to facilitate data communications between computers without requiring additional equipment.



7880 MASS STORAGE SUBSYSTEM

The CDC 7880 Mass Storage System (MSS) is a fully automated device for storing up to 19.6 billion characters of on-line data in a single unit within a CDC CYBER 170, CYBER 70, or 6000 Series system. It is designed to complement existing tape and disk operations and to provide added performance and economies obtained from more rapid and accurate storage/retrieval, tighter operating security, and greater storage capacity.

MSS hardware consists of a Mass Storage Controller (MSC), a Cartridge Storage Unit (CSU), and up to four Mass Storage Transports (MST). A minimum of two MSTs are required for each CSU.

HIGHLIGHTS

- System can be tailored to the capacity and performance requirements of the central processing unit.
- All resources under control of the operating system.
- Minimum disk space used; only those tracks required by the staged file set are assigned.
- Active files not restricted to, nor concentrated on, segregated disks.
- Easy installation and upgrading procedures.
- Cartridge size, realistically related to data sets, results in easier and more efficient management.
- Field expansion possible through add-on units.
- Each CSU includes an initial supply of 500 cartridges.

7880-1 MASS STORAGE CONTROLLER

The 7880-1 Mass Storage Controllers provide control and attachment facilities to link the Mass Storage subsystem to the central processor. It has two CDC CYBER peripheral processor channel connections as standard offering.

7881-1 CARTRIDGE STORAGE UNIT

The 7881-1 Cartridge Storage Unit (CSU) accommodates up to 2000 user-cartridges in its vertical file ports. The average access time of the X-Y selector is 3.5 seconds with a maximum of 5.5 seconds required to select a cartridge and deliver it to the mass storage transport. Since the CSU operates continuously, overlapped operations can be performed between the read, write, retrieval, and storage functions.

7882-1 MASS STORAGE TRANSPORT

The 7882-1 Mass Storage Transport (MST) contains two vacuum columns, a single tape-drive capstan, and two sets of read/write heads. Tape movement in the components can be continuous or incremental. The single-capstan motor provides fast start/stop operation, with a data rate of 806,000 bytes per second.

Each of the two read/write heads is capable of reading nine tracks in either direction. Write operations are performed with a read-after-write capability while tape is moved in alternate directions. Since 18 tracks are covered by one position of the read/write head assembly, the heads move eight times to cover fully the total 144 tracks. A voice-coil actuator moves the head assembly to various track positions and requires a positioning time of approximately 20 milliseconds.

7883-XX MAGNETIC TAPE CARTRIDGE

One-hundred inches (2540 millimeters) of usable magnetic tape are enclosed in each tape cartridge. The tape is 2.7 inches (69 millimeters) wide and coated on one side with an oxide material capable of high-density recording. When the Group-Coded Recording (GCR) mode is used, recording density is 9040 flux changes per inch (9040 flux changes per 25.4 millimeters). Effective recording density is 6250 bits per inch (6250 bits per 25.4 millimeters). Each cartridge holds up to 9.8 million characters.

MODELS

<u>Model</u>	<u>Description</u>
7880-1	Mass Storage Controller
7881-1	Cartridge Storage Unit
7882-1	Mass Storage Transport
7883-20	Mass Storage Tape Cartridge Pack

OPTIONS

10390-1	MSC 16 Device Address Option Increases the number of allowable device connections to a Mass Storage Controller from eight to a total of sixteen.
10391-1	MSC Channel Access Option Provides two additional CDC CYBER PPU channel connections to a Mass Storage Controller.
10392-1	MST Alternate Path Option Allows the connection of one 7882-1 MST to two MSCs (7880-1).
10393-1	CSU Alternate Path Option Allows the connection of one CSU (7881-1) to two MSCs (7880-1).



10312 CDC CYBER 170 MEMORY INCREMENT

The CDC 10312 Memory Increment adds a specified number of 60-bit words plus an 8-bit error correction code of semiconductor memory to CDC CYBER 170/Models 171 through 174 computer systems.

The increment options include:

<u>Increment Model</u>	<u>CDC CYBER 170 Model</u>	<u>Description</u>
10312-3	172-2	Adds 16K words of memory, increasing memory from 32K to 49K words.
10312-4	172-3	Adds 16K words of memory, increasing memory from 49K to 65K words.
10312-6	171-4, 172-4, 173-4, 174-4	Adds 32K words of memory, increasing memory from 65K to 98K words.
10312-8	171-6, 172-6, 173-6, 174-6	Adds 32K words of memory, increasing memory from 98K to 131K words.
10312-12	171-8, 172-8, 173-8, 174-8	Adds 65K words of memory, increasing memory from 131K to 196K words.
10312-16	171-12, 172-12, 173-12, 174-12	Adds 65K words of memory, increasing memory from 196K to 262K words.



10313 CDC CYBER 170/MODEL 175 MEMORY INCREMENT

The CDC 10313 Memory Increment adds a specified number of 60-bit words plus an 8-bit correction code of semiconductor memory to a CDC CYBER 170/Model 175 computer system.

The increment options include:

<u>Increment Model</u>	<u>CDC CYBER 170 Model</u>	<u>Description</u>
10313-6	175-4	Adds 32K words of memory, increasing memory from 65K to 98K words.
10313-8	175-6	Adds 32K words of memory, increasing memory from 98K to 131K words.
10313-12	175-8, 175-108, 175-208	Adds 65K words of memory, increasing memory from 131K to 196K words.
10313-16	175-12, 175-112, 175-212	Adds 65K words of memory, increasing memory from 196K to 262K words.
10313-112	175-308	Adds 65K words of memory, increasing memory from 131K to 196K words.
10313-116	175-312	Adds 65K words of memory, increasing memory from 196K to 262K words.



10314 CDC CYBER 170 PERIPHERAL
PROCESSOR UNIT INCREMENT

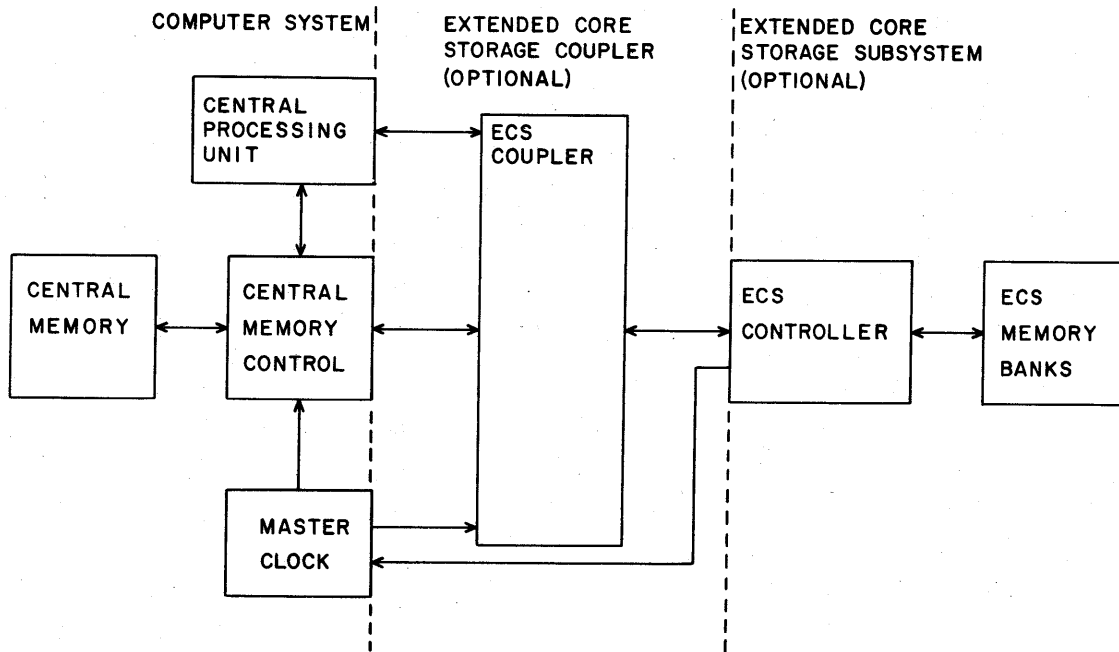
The CDC 10314 Peripheral Processor Unit (PPU) Increment adds additional peripheral processors and associated input/output channels to existing CDC CYBER 170 computer systems.

The different PPU increments available include:

<u>Increment Model</u>	<u>CDC CYBER 170 Model</u>	<u>Description</u>
10314-1	171, 172, 173, 174	This model adds four peripheral processors and 12 input/output channels to existing system. The resulting system has 14 PPU's and 24 input/output channels. This increment requires prior installation of an expansion cabinet (Model number 10317-1) on Models 172 and 173 computer systems.
10314-2	171, 172, 173, 174	This model adds three peripheral processors. The resulting system has 17 PPU's and 24 input/output channels. This increment is usually added with increment Model 10314-1.
10314-3	171, 172, 173, 174	This model adds three peripheral processors. The resulting system has 20 PPU's and 24 input/output channels. This increment is usually added with increment Model 10314-2.
10314-51	175	This model adds four peripheral processors and 12 input/output channels. The resulting system has 14 PPU's and 24 input/output channels.
10314-52	175	This model adds three peripheral processors to the computer system. The resulting system has 17 PPU's and 24 input/output channels. This increment can only be added after addition of increment Model 10314-51.
10314-53	175	This model adds three peripheral processors to the computer system. The resulting system has 20 PPU's and 24 input/output channels. This increment can only be added after addition of increment Model 10314-52.
10314-151	175-1XX, 175-2XX, 175-3XX	This model adds four peripheral processors and 12 input/output channels. The resulting system has 14 PPU's and 24 input/output channels.
10314-152	175-1XX, 175-2XX, 175-3XX	This model adds three peripheral processors to the computer system. The resulting system has 17 PPU's and 24 input/output channels. This increment can only be added after addition of increment Model 10314-51.
10314-153	175-1XX, 175-2XX, 175-3XX	This model adds three peripheral processors to the computer system. The resulting system has 20 PPU's and 24 input/output channels. This increment can only be added after addition of increment Model 10314-52.

10318-1/2 EXTENDED CORE STORAGE COUPLER

A computer system that contains the optional Extended Core Storage (ECS) subsystem to augment central memory (CM) requires an ECS coupler connected as the interface between the computer system and the ECS subsystem. The 10318-1/2 ECS Coupler controls the transfer of data, as noted in the following figure, from CM to the ECS subsystem memory banks (write operation) or from the ECS memory banks to CM (read operation).



ECS Coupler Interface Configuration

The read or write operation is initiated by instructions in the Central Processing Unit (CPU) of the computer system. The Central Memory Control (CMC) of the computer system controls the data in and out of CM and the ECS coupler. The ECS coupler controls and monitors the information to ensure a successful transfer operation. If errors occur during a transfer operation, the ECS coupler detects the errors or passes the error signal detected by other equipments to the CPU. The computer system master clock is slaved to the ECS controller clock of the ECS subsystem providing synchronization of the computer system and the optional equipment.

The ECS coupler performs the following internal functions.

- Generates and sends the ECS address, request controller signal, and the read or write signals to the ECS controller after receiving the initial request coupler signal and the starting ECS address from the CPU.
- Checks for odd parity on the word count and ECS starting address received from the CPU.
- Generates and sends the continue request signal to the CMC 400 nanoseconds ahead of the bank initiate signal for the central memory bank reservation operation.
- Generates and sends the bank initiate signal to the CMC with each central memory word to start a memory cycle.
- Receives the starting ECS address from the CPU. Increments the ECS address from the starting address for each ECS record (eight 60-bit words/record) transferred.
- Generates and sends an odd parity bit for the ECS address to the ECS controller.
- Compares the number of words transferred with the word count received from the CPU to ensure a complete transfer of data.
- Generates and sends the end of transfer signal to the CPU when the transfer is completed without an error.
- Generates and sends the error end of transfer signal to the CPU if an error is detected during a transfer. The particular error detected determines if the transfer is terminated immediately or allowed to continue until all words are transferred.
- Generates and sends the ECS transfer error signal and a 3-bit error code to the CPU to identify the error when an error is detected during a transfer. This signal and code are routed by the CPU to the S/C register in the peripheral processor subsystem (PPS) of the computer system.
- Interprets a flag operation instruction from the CPU and sends the instruction to the ECS controller.
- Generates and sends the OK exchange CPU signal to the CPU when the coupler has completed a transfer. A CPU exchange may proceed upon receipt of the signal by the CPU.



10377-1 CYBER 176 PERIPHERAL PROCESSOR INCREMENT

The CDC 10377-1 Peripheral Processor Increment adds four additional Peripheral Processors (PPs) and 12 input/output data channels to a CDC CYBER 170/Model 176 computer system containing, as standard, 10 PPs and 12 input/output data channels. The resulting system has 14 PPs and 24 input/output data channels.

Each processor has 4096 12-bit words MOS memory, and is independently programmable to handle system and input/output operations.

The data channels attached to peripheral devices are serviced by the peripheral processors and each bidirectional data channel has a maximum data rate of four million characters per second.



10380 COMPARE/MOVE UNIT

The CDC 10380 Compare/Move Unit adds to a CDC CYBER 170/Model 171 computer four 6-bit character-oriented compare/move instructions which operate on variable length fields in central memory. Characters can be moved from one central memory location to another, and fields of characters can be compared either directly or through a collation table.

MODELS

<u>Model</u>	<u>Description</u>
10380-1	This model adds one 10380 unit to a CDC CYBER 170/Model 171 system with one central processor.
10380-2	This model adds one 10380 unit to a CDC CYBER 170/Model 171 system with two central processors.



10381 DATA CHANNEL CONVERTER

The CDC 10381 Data Channel Converter allows CDC 3000 series peripheral equipment to be attached to a data channel of a CDC CYBER 170/Model 171 computer. The CDC CYBER 170/Model 171 computer can perform Connect, Function, Read, Write, and Status operations on these equipments via the 10381 converter.

DESCRIPTION

The CDC CYBER 170/Model 171 computer transmits codes to the 10381 converter prior to starting any operation on an external equipment. These codes establish conditions in the converter so that the proper 3000 type signals accompany the CDC CYBER 170/Model 171 computer input/output operations. Up to eight peripheral equipments may be connected to the output of the 10381 converter.

MODELS

<u>Model</u>	<u>Description</u>
10381-1	This model adds the first data channel converter to the Model 171 system.
10381-2	This model adds a second channel converter to a Model 171 system which has one 10381-1.



10400-10401 CLA CABLES

10400 ASYNCHRONOUS CLA CABLE

The 10400 Asynchronous Communications Line Adapter (CLA) modem cable connects the CLA to customer-provided modem or local terminal at circuit speeds up to 9600 bps at RS232 specifications. Cable length is 50 feet (or 15 meters).

10401 SYNCHRONOUS CLA CABLE

The 10401 Synchronous Communications Line Adapter (CLA) modem cable connects the CLA to customer-provided modem (to 10,800 bps) or local terminal (2400, 4800 or 9600 bps) at RS232 specifications. Cable length is 50 feet (or 15 meters).

