



**CONTROL DATA**  
CORPORATION

**ZELLARS C. WEST**  
Sales Representative

EDP Sales  
875 Greentree Road, Pittsburgh, Pennsylvania 15220  
Telephone Area Code 412/922-5633

**Control Data<sup>®</sup>**  
**7600 Computer System**

**Preliminary System Description**



## 7600 SYSTEM DESCRIPTION (Preliminary)

### Introduction

The 7600 system is the result of a development program to provide computing capacity substantially beyond that of the 6600 systems. Central processor computation exceeds four times as fast as corresponding computation in the 6600 system. The 7600 system is designed to be machine code upward compatible with the 6400/6500/6600 systems in the area of central processor routines; it is not compatible on the machine code level in the area of system programs or input-output drivers. The 7600 system input-output provisions have been generalized and greatly expanded over those provided in the 6400/6500/6600 systems. Input-output data rates are not expected to average substantially higher on a per channel basis than the rates in the 6600 systems. A much larger volume of input-output data is handled in the 7600 system by a much larger number of input-output channels.

The 7600 system contains a central processing unit (CPU) and a number of peripheral processing units (PPU). Some of the PPU are physically located with the CPU and others may be remotely located. The PPU communicate with the CPU over high speed data links with the data buffered at the CPU end of the data link. The PPU provide a communication and message switching function between the CPU and individual peripheral equipment controllers. Each PPU has a number of high speed data links to individual peripheral equipment controllers in addition to the data link to the CPU. The PPU time shares the data link to the CPU among the peripheral equipment controllers on a record by record basis.

The 7600 system is designed to accommodate multiple operating stations. New peripheral equipment configurations are being developed which will operate from programable equipment controllers and are specifically intended for this application. These controllers will be able to communicate with the PPU on a record by record basis at higher rates than the existing 6400/6500/6600 equipment.

### 7600 system parameters

CPU computation section  
60 bit internal word  
binary computation in fixed point and floating point format  
nine independent arithmetic units  
twelve word instruction stack

synchronous internal logic with 27.5 nanosecond clock period

CPU small core memory  
65,536 words of coincident current memory (60 bit + 5 parity)  
32 independent banks  
2048 words per bank  
275 nanosecond read/write cycle time  
27.5 nanosecond per word maximum transfer rate

CPU large core memory  
512,000 words of linear select memory (60 bit + 4 parity)  
8 independent banks  
64,000 words per bank  
1760 nanosecond read/write cycle time  
8 words read simultaneously each reference  
27.5 nanosecond per word maximum transfer rate

CPU input-output section  
15 independent channels (asynchronous) each channel full duplex (60 bit)  
buffer areas of 64 or 128 words each channel  
55 nanosecond per 60 bit word transfer rate

PPU computation section  
12 bit internal word  
binary computation in fixed point  
synchronous internal logic with 27.5 nanosecond clock period

PPU core memory  
4096 words of coincident current memory (12 bit + 1 parity)  
two independent banks  
2048 words per bank  
275 nanosecond read/write cycle time

PPU input-output section  
8 independent channels (asynchronous) each channel full duplex (12 bit)

### System communication

The 7600 system is divided into a number of major sections which are interconnected as shown in figure 1. All input data enters the system at a programable peripheral equipment controller. All output data leaves the system at a programable peripheral equipment controller. The PPU serve to gather input data from these peripheral equipment controllers for delivery to the CPU for processing, and distribute processed data to these equipment controllers for output devices. The communication between PPU and equipment controller is generally limited by the rate at which the equipment controller can deliver or accept data. Most equipment controllers in the 7600 system will contain a core memory buffer capable of holding one record of

data for the attached input or output devices. The communication between PPU and equipment controller is then a one record burst of data followed by a relatively long period of inaction. The PPU is then able to time share its channel to the CPU among a number of equipment controllers without storing more than one record of data in its own memory at any one time.

Communication between a PPU and a peripheral equipment controller is over a 12 bit full duplex channel. Each channel has a 12 bit data path from PPU to controller, and a separate 12 bit data path from controller to PPU. These two data paths are independent and may operate simultaneously. Each path has two associated control lines carrying control information in the direction of data flow. These lines carry a "word flag" to indicate passage of each 12 bit word of data, and a "record flag" to indicate the completion of a record of data. Each path has one associated control line carrying control information against the direction of data flow. This line carries a "word resume" signal to indicate receipt of a data word.

Communication between a PPU and the CPU is over a 12 bit full duplex channel identical to that described above. The 12 bit data path from PPU to CPU includes a 60 bit assembly register at the CPU end of the data link. This register assembles five 12 bit words into a 60 bit word for entry into the CPU memory. The 12 bit data path from CPU to PPU includes a 60 bit disassembly register at the CPU end of the data link. This register disassembles a 60 bit word from the CPU memory into five 12 bit words for transmission over the data link.

A maximum of 15 PPU may be directly connected to the CPU. Each CPU channel has assembly and disassembly registers to convert from 60 bit to 12 bit word length. All 15 CPU input-output channels may be in operation at the same time. Data is transmitted to, or from, PPU on a record by record basis. The CPU program is interrupted at the end of a record transmission to exchange control information with the communicating PPU or to initiate transmission of another record. On very long records the CPU program is interrupted at prescribed intervals in the buffer data. The frequency of this interruption is a function of the buffer size and may be preset individually for each CPU channel.

### Operating system

The 7600 hardware was designed with a particular software approach in mind.

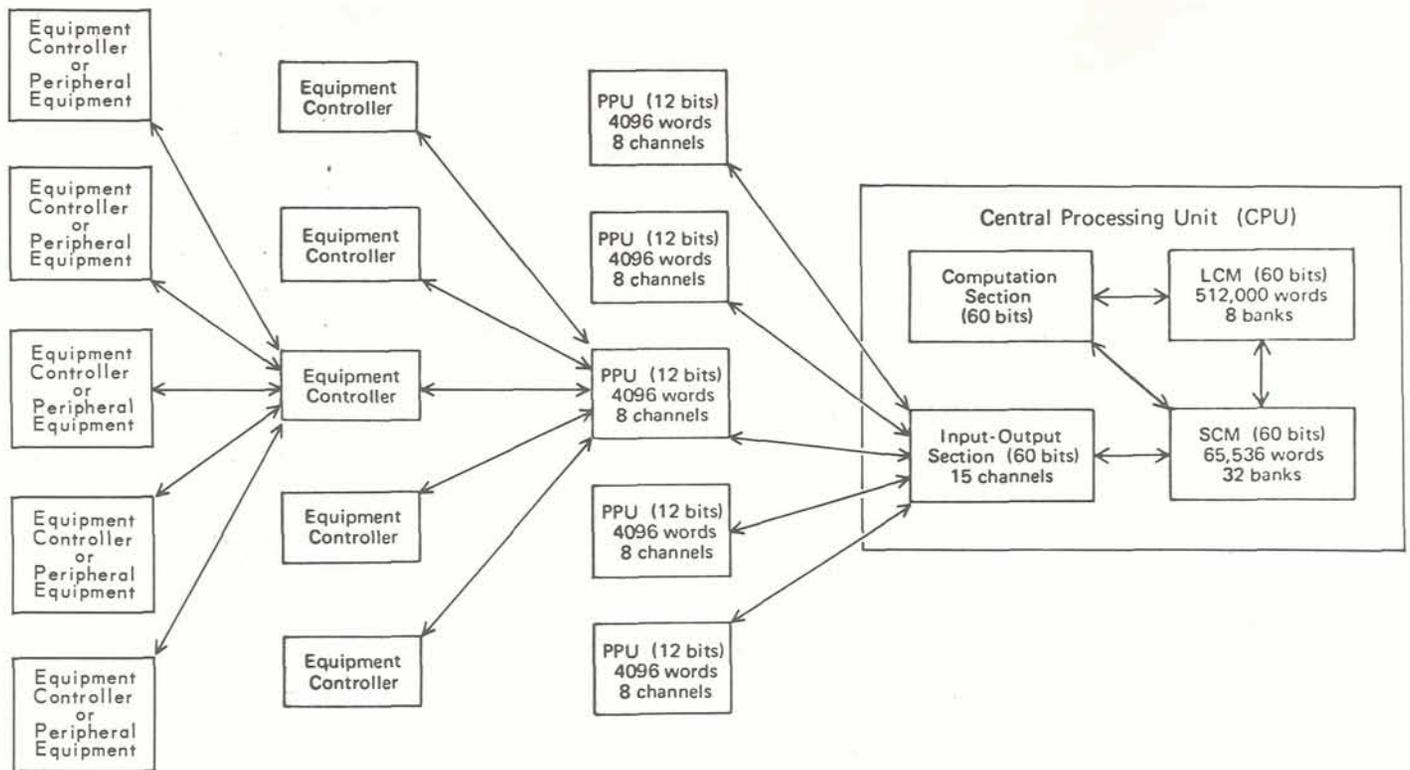


Fig. 1 7600 System Communication

This approach is an outgrowth of experience with the Chippewa and SCOPE Operating Systems for the 6600. The 7600 system software will be described in a following section of this manual.

#### System monitor

The system monitor is a CPU program in the 7600 system. This program is loaded with the operating system on a machine "dead start" and remains in the CPU core memory as long as the operating system is used. A portion of the system monitor resides permanently in the small core memory (SCM) section of the CPU. This portion of the monitor program is called the "resident monitor" program. The bulk of the system monitor resides in the large core memory (LCM) section of the CPU. This portion of the monitor program is called piecemeal into the SCM for execution as overlays on the resident monitor program.

#### Object program

An object program is defined in this manual to mean any CPU program other than the system monitor program. This term is used to describe generally a job oriented program. An object program may be a machine language program such as a FORTRAN compiler, or it may be a program which results from compiling FORTRAN statements with a compiler.

#### Central processing unit (CPU)

The CPU is a single integrated data processing unit. It consists of a computation section, small core memory, large core memory, and input-output section. These sections are all contained in one main frame cabinet and operate in a tightly synchronous mode with a clock period of 27.5 nanoseconds. Communication with equipment outside of the main frame cabinet is asynchronous.

#### CPU core memory

The CPU contains two types of internal core memory. One type, designated as the small core memory (SCM), is a many bank coincident current type memory with a total capacity of 64K words of 60 bit length ( $K = 1024$ ). The other type, designated as the large core memory (LCM) is a linear selection type of memory in which eight 60 bit words are addressed as a single unit. The LCM has a total capacity of 500K words of 60 bit length. These two types of internal memory have significantly different system functions in the CPU.

The SCM is arranged in 32 banks of 2K words each. Each bank is independent of the other 31 banks. Maximum data transfer rate between the SCM as a unit and other parts of the system is one word each clock period. Each SCM bank has a four clock period access time from arrival of the storage address to

readout of the 60 bit word. The total read/write cycle time for a SCM bank is ten clock periods. It is thus possible for a maximum of ten SCM banks to be in operation at one time. This maximum occurs during block copy instructions between SCM and LCM in which the addresses for sequential words cause no SCM bank conflicts. In random addressing of the SCM for CPU program data, CPU instructions, and input-output channel data, an average of four SCM banks in operation at one time is more normal.

The SCM performs certain basic functions in system operation which the LCM cannot effectively perform. These functions are essentially ones requiring rapid random access to unrelated fields of data. The first 4K addresses in SCM are reserved for input-output buffer and control areas. These areas are addressed by the CPU input-output section as required to service the communication channels to the PPU. CPU object programs do not have access to these areas. The next 1K addresses are reserved for the resident monitor program.

The remainder of the SCM is divided between fields of CPU program code and fields of data for the currently executing program.

The LCM is arranged in eight banks of

64K words each. Each bank is independent of the other seven banks. A storage reference to a LCM bank results in a read/write cycle which takes 64 clock periods. Eight 60 bit words are read simultaneously from a LCM bank whenever a read/write cycle occurs. These words are held in a 480 bit operand register for each LCM bank. Subsequent reference to a word residing in one of these operand registers allows either read or write function without the delay of a bank read/write cycle. Maximum data transfer rate between the LCM as a unit and other parts of the system is one word each clock period. This maximum transfer rate occurs during block copy instructions between LCM and SCM. LCM bank read/write cycles are anticipated in the block copy operation to avoid a bank access delay.

The LCM provides the basic working storage for the CPU. All object programs are assembled here for execution in

the SCM. All data files are buffered through LCM for the object programs. Small object programs are generally run to completion in SCM with the complete input file in LCM at the beginning of execution, and the complete output file in LCM at the end of execution.

The low order addresses in LCM are reserved for monitor program overlays, object time library, and compilers. These areas require approximately 32K of the 500K available storage. The remainder of LCM is divided into fields of the various operating stations in the system.

### Computation section

The computation section of the CPU contains nine segmented arithmetic units, 24 operating registers, and a 12 word instruction stack. These units work together to execute a CPU program stored in the SCM. Data moves into, and out of, the computation section of the CPU through the operating registers.

Data may be directly addressed in either the SCM or the LCM. The general information flow in this section is illustrated in figure 2.

### Instruction word stack

The instruction word stack is a group of twelve 60 bit registers in the CPU computation section which hold program instruction words for execution. The instruction stack information is essentially a moving window in the program code. The stack is filled two words ahead of the program address currently being executed. A small program loop may frequently be entirely contained within the instruction stack. When this happens the loop may be executed repeatedly without further references to SCM.

The current instruction word is contained in a special register in the CPU computation section. This register is designated the CIW register. Program instruction words are read one at a time from the instruction stack and are

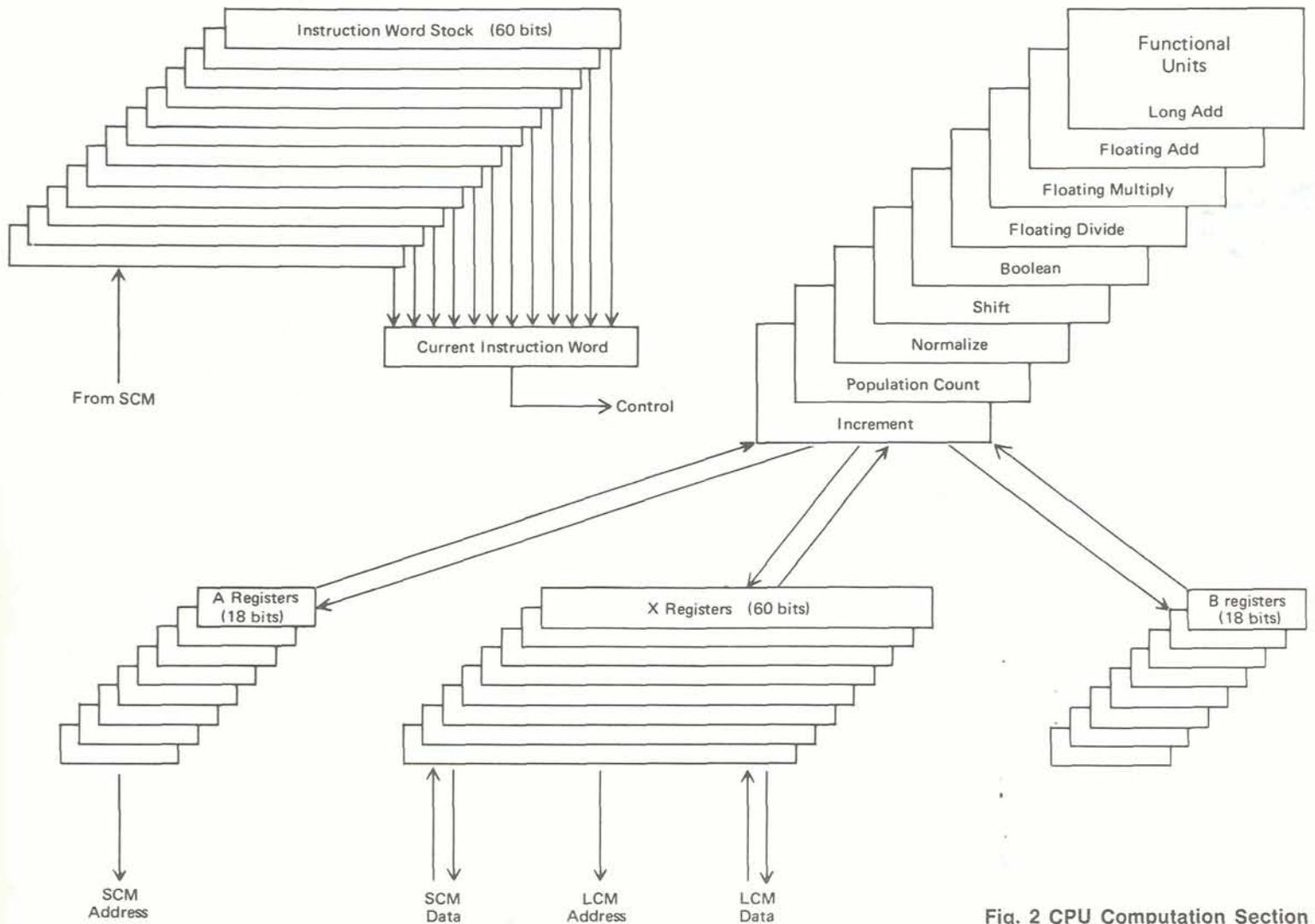


Fig. 2 CPU Computation Section

1000	Channel 16 input package	Channel 16 output package	Channel 17 input package	Channel 17 output package	
700	Channel 14 input package	Channel 14 output package	Channel 15 input package	Channel 15 output package	
600	Channel 12 input package	Channel 12 output package	Channel 13 input package	Channel 13 output package	
500	Channel 10 input package	Channel 10 output package	Channel 11 input package	Channel 11 output package	
400	Channel 6 input package	Channel 6 output package	Channel 7 input package	Channel 7 output package	
300	Channel 4 input package	Channel 4 output package	Channel 5 input package	Channel 5 output package	
200	Channel 2 input package	Channel 2 output package	Channel 3 input package	Channel 3 output package	
100	MCU package	Real time package	Channel 1 input package	Channel 1 output package	
0					
	0	20	40	60	100

(octal addresses)

Fig. 4 I/O Section Exchange Packages in SCM

(EEA) Error exit address — This is a SCM absolute address for an exchange jump or error termination.

(BPA) Breakpoint address — This is a SCM relative address for breakpointing an object program.

(PSD) Program status designation — This is a register of control information.

#### Program breakpoint

An object program may be executed in small sections during a debugging phase by using the breakpoint address register (BPA). This is a hardware register in the computation section of the CPU which is loaded from the object program exchange package. A coincidence test is made between (BPA) and the program address register (P) as each program instruction word is read from the IWS to the CIW register. When a coincidence occurs the program execution is terminated with an exchange jump to the error exit address (EEA).

The monitor program controls the breakpoint address for debugging an object program by altering the exchange package for the object program before each execution interval. The monitor program receives instructions for breakpoint control from an operator console or from control cards in a

job stack. It is possible to step through a program one instruction word at a time using an operator console to monitor the register values at each step.

#### Error Exits

Execution of an object program may be terminated by an exchange jump to the error exit address (EEA) under certain conditions. Some of these conditions may be selected by mode declarations through the monitor program, and some are unconditional. In general, errors due to arithmetic overflow, underflow, or indefinite results during computation may be allowed to proceed through the calculation, or may cause an error exit, depending on mode selection. Errors due to hardware failure or program addressing out of an assigned field in storage cause unconditional error exits. In any error exit case the monitor program has the ability to continue the object program where the error can be corrected or ignored.

The error condition flags and mode selection flags are all contained in an 18 bit program status designation (PSD) register. This register is loaded from the exchange package for each object program. The mode selections are made in the exchange package prior to the execution interval by the monitor

program. If an error condition occurs during the execution interval the monitor program can determine the type of error by analyzing the terminating exchange package parameters. Each bit in the PSD register has significance either as a mode selection or an error condition flag.

#### CPU input-output section

The CPU input-output section includes the mechanism to buffer data to (or from) the directly connected PPU. Each PPU communicates with the CPU over a 12 bit full duplex channel. Each channel has assembly and disassembly registers to convert the 12 bit channel data to 60 bit CPU words. The function of the CPU input-output section is to deliver these 60 bit words to the SCM for incoming data, read 60 bit words from the SCM for outgoing data, and interrupt the CPU program for monitor action on the buffer data as required.

The input-output section is able to process a maximum of one 60 bit word each two clock periods. The effective processing rate is somewhat lower than this because of bank storage conflicts in SCM. Whenever a bank conflict occurs on an input-output section request, the communication path to the SCM is held up until the conflict is resolved. Channel requests for a SCM word reference are processed on a priority basis whenever the I/O section is not able to keep up with the channel requests. The priority is assigned in order by channel number, with the lowest order channels having the highest priority.

There are a total of 15 channels in the I/O section of the CPU. These channels are numbered in octal beginning with 01 and ending with 17. Each channel has a SCM buffer area for incoming data and a separate SCM buffer for outgoing data. In addition each channel has an exchange package for incoming data and an exchange package for outgoing data. Each buffer area is divided into two fields, a lower field and an upper field. Data is entered (or removed) from the buffer area in a circular mode. The last word in the lower field is followed by the first word in the upper field. The last word in the upper field is followed by the first word in the lower field. Whenever a buffer area has been filled (or emptied) to the point where a field boundary is crossed, the CPU is interrupted through the associated exchange package to process the buffer data. The channel continues to fill (or empty) the other buffer field while the CPU is processing this buffer data.

The I/O section exchange package areas are permanently assigned in the lowest

order addresses of SCM. These areas are arranged as shown in figure 4. The I/O section buffer areas are assigned in the next higher order address positions of SCM. These areas may be changed both in size and order (wiring change) to accommodate various types of channel volume. A typical arrangement for the buffer areas is shown in figure 5. Total I/O section space in SCM cannot exceed absolute address 10,000 octal.

### Real time clock

CPU programs may be timed precisely by using the CPU clock period counter. This counter is essentially a real time clock which is advanced one count each clock period of 27.5 nanoseconds. Since the clock is advanced synchronously with the program execution, the program may be timed to an exact number of CPU clock periods.

The CPU clock period counter contains a 17 bit register which can be read by a CPU program with an 016 instruction. This register contains the lowest order 17 bits of the real time count. An overflow of the highest order bit in this register sets a real time interrupt flag. This flag reads as an 18th bit on an 016 instruction. In addition, this flag causes an exchange jump to the real time exchange package at absolute address 20 in SCM.

### Real time interrupt

The CPU clock period counter causes an interrupt of the CPU program every 3.6 milliseconds (approx.). This corresponds to the modulus of the 17 bit register in the clock period counter. This interrupt causes an exchange jump to absolute address 20 in SCM. The real time exchange package at this SCM address executes a CPU program in monitor mode which advances the count in a 60 bit SCM storage location to continue the real time clock function of the clock period counter. This program also tests time limit for the currently active object program. The real time interrupt flag is cleared at the end of the execution interval for real time exchange package. This flag is read as an 18th bit on an 016 instruction in order to avoid erroneous timing indication where the reading of the clock period counter coincided with the setting of the interrupt flag. This bit is an indication that the counter has passed its modulus but the interrupt to advance the SCM word has not yet occurred.

### External interrupt

The CPU computation may be interrupted from an external source through the directly connected PPU. Each such PPU has the ability to interrupt the execution of an object program and call the system monitor by sending a control

message over the associated channel to the CPU. The interrupt occurs whenever a record flag arrives at the CPU input-output section on an incoming data link. Interpretation of message content is a function of the system monitor program.

### System dead start

The system is initially started through the maintenance control unit (MCU). This mechanism is used whenever power is turned on after an idle period or when the system is restarted after hardware failure. The MCU is essentially a PPU with specially adapted input-output channels. This unit is used exclusively for maintenance functions.

The dead start sequence begins with a deck of binary cards for the MCU program. These cards are loaded through the MCU card reader and activate the MCU. The MCU program then dead starts the CPU and all other PPU in sequence. A bootstrap program is entered directly into SCM from the MCU. The CPU program is then initiated by the MCU through the MCU exchange package at absolute address zero in SCM. This bootstrap program transmits a resident PPU program to all directly connected PPU to initiate their activity. The system is then loaded completely from a system library tape associated with one of the

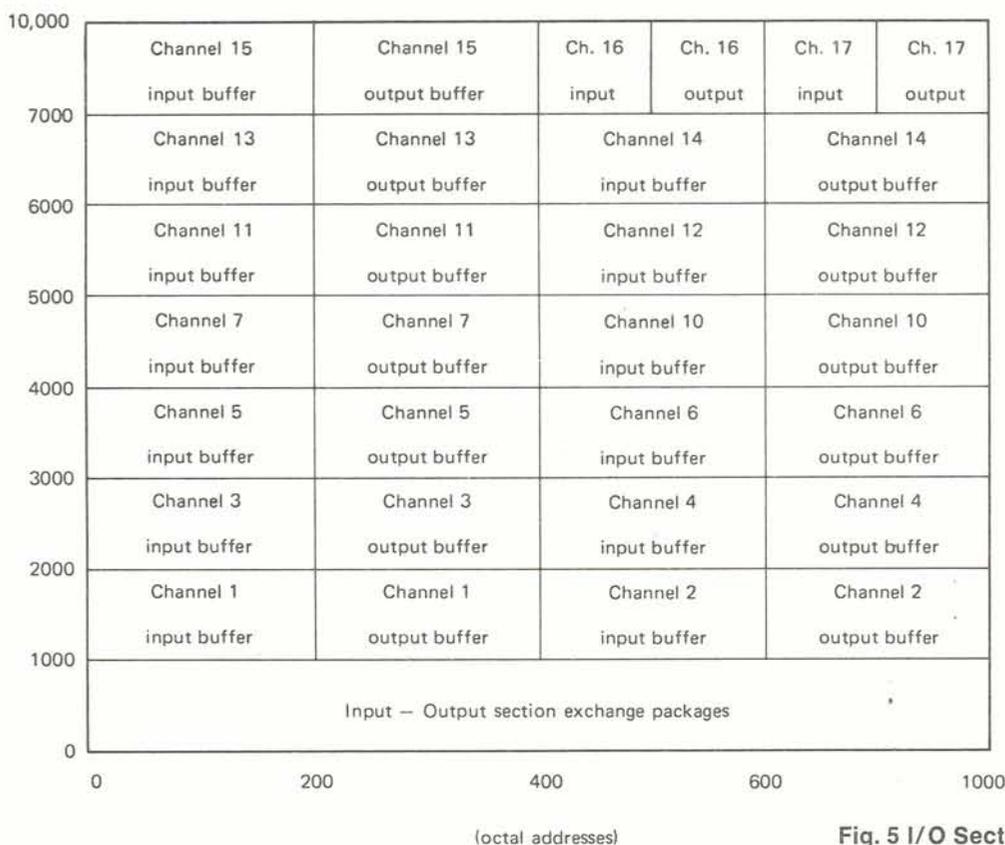


Fig. 5 I/O Section Buffer Areas in SCM

system PPU. This library tape may be any tape in the system and may be declared at dead start time through the maintenance console.

**System operation**

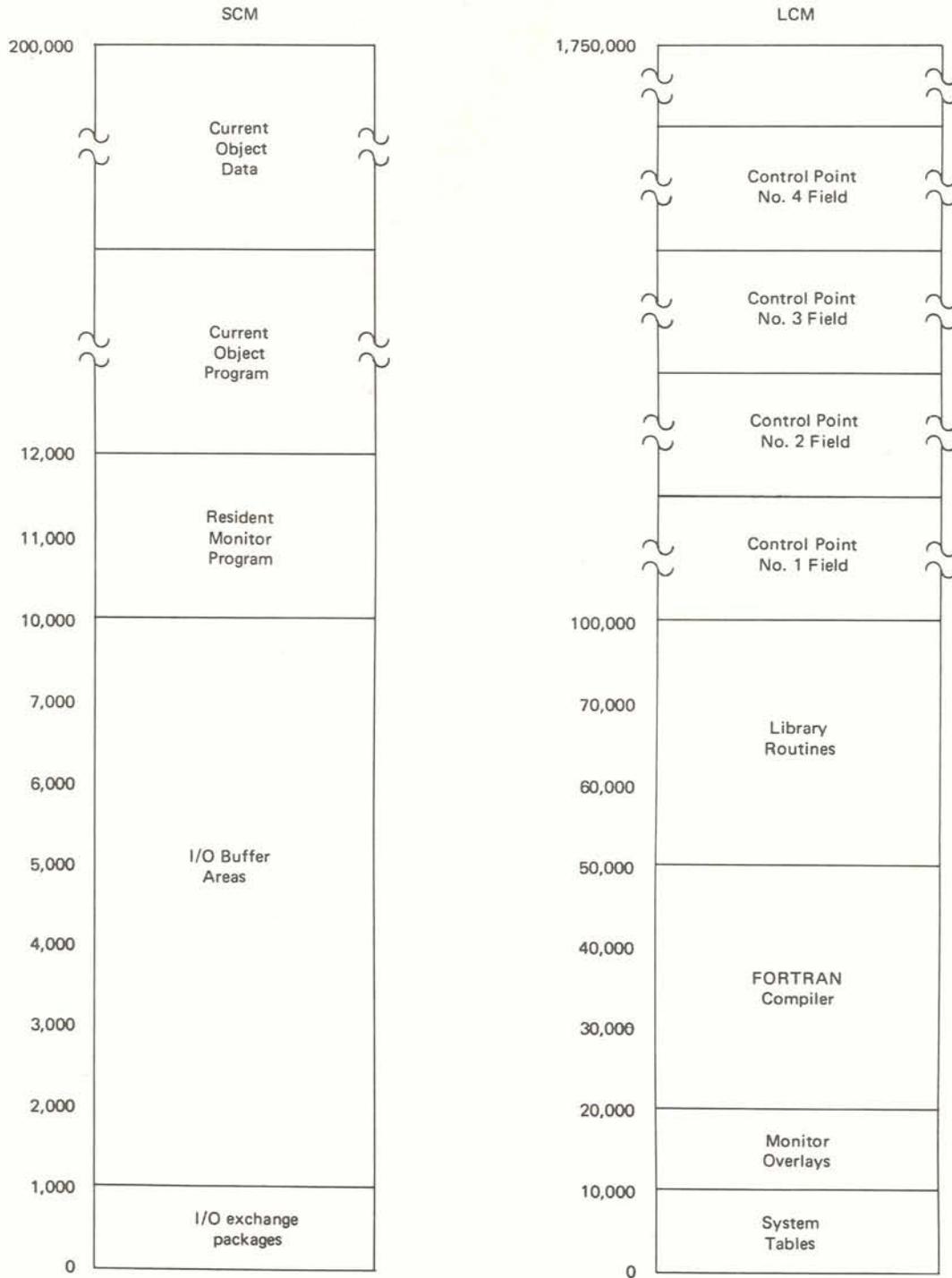
The operating system software consists of the CPU monitor program, with its overlays, plus the PPU programs to drive the peripheral equipment. The operating system forms a software framework in SCM and LCM to hold the object

programs for execution. Organization of CPU storage is illustrated in figure 6 on the following page. The I/O section storage areas are fixed by hardware addressing in SCM. The remaining areas are strictly software organization.

The resident monitor program resides immediately above the I/O section in SCM. This program is a permanent part of the operating system and is never moved during system execution. This

program handles all I/O section interrupt requests as well as object program requests. The remaining portion of SCM beginning at octal address 12,000 and continuing to 200,000 is available for object program code and data.

The low order addresses in LCM are used for permanent storage of frequently used programs and overlays. The system tables are kept here and are directly addressed by the monitor program.



(Octal addresses)

**Fig. 6 Operating System Storage Allocation**

Library routines and compiler code may vary with installation requirements. Approximately 100,000 octal words of LCM are expected to be used for these permanent storage requirements.

The remainder of LCM beginning at octal address 100,000 and continuing to address 1,750,000 is available for object program code and data. This area is divided into a number of control point fields. Each control point field contains a single job oriented program code and data. In addition, each control point contains the necessary control information for continuity from one program to the next. These areas vary in size as required for each job. When a job is completed and a new job is assigned to a control point, the storage areas are readjusted by moving the data in each control point field through the SCM as a buffer to a new LCM location.

Job execution proceeds through the system in three phases. In the first phase cards are read at an operating station and an input file is generated on a disk pack, tape unit, or disk file. This input file may physically reside at the operating station or it may reside in a central disk file.

In the second phase the input file is copied into a control point field in LCM. If the input file is small the entire file may then reside in LCM. If the input file is large the first portion of the file is copied into a buffer area in the control point field. The control cards in the input file are then interpreted by the monitor program and the necessary compiler or library routines are read from outside the LCM if necessary for program execution. When the control point information is ready for execution the program code is transferred to SCM. If the program and associated data are too large to fit entirely in SCM a portion of the data must be retained in LCM and directly addressed there. This must be done by declaration at compile time in order to designate certain arrays of data to reside in LCM for execution.

Only one program at a time is executed in SCM. The entire SCM object program field may thus be used for each program. Data is read from the input file in LCM and results are stored in an output file in LCM. If the amount of input and output data is small the job may be run to completion in one execution interval. If job execution is delayed by buffer size or by intermediate file references the program code is returned to LCM and another control point uses the SCM while buffer data is transferred to (or from) LCM. The second phase is completed

when the output data has been delivered to the output file buffer in LCM and this buffer has been emptied onto a disk pack, tape, or disk file for listing.

The third phase consists of copying the output file from the magnetic storage to a printer at the operating station. During this phase the LCM control point field has been released for another job.

### System peripherals

The Disk/Drum Subsystem consists of one controller and one 7600 Disk File (two storage units) or up to two 7600 Drums. The 7600 Disk File contains two double-ended disk drive spindles (stacks) which rotate asynchronously at 1,800 rpm, less inductive slip. Each of the four half-spindles contains 18 disks (32 recording surfaces). There are two hydraulic positioning access assemblies, each one servicing a pair of half-spindles. Each of the two access assemblies is considered to be one file unit. Each access assembly contains two separate horizontally opposed groups of 16 head arms, one group for each half spindle (stack). Mounted on the end of each head arm are two head pads, each of which contains one read/write magnetic head. Thus, there is one head for each of the 32 surfaces on a half-spindle. Each unit is divided into 4 groups of 16 heads each (2 groups on each stack). Each access can be moved to any of the 512 data positions on each stack or a non-data position (retract) via a position function. There is only one data zone on the file. It has a nominal data rate of 380 nanoseconds per 16 bit word. Since the PPU interface receives/transmits 12 bit bytes, the nominal PPU data rate of the disk system is 285 nanoseconds per 12 bit byte. The maximum latency is approximately 35.5ms. There are 512 positions on each of the two stacks. The positioning times are:

Stack to stack position	15 msec.
Adjacent position	25 msec.
Random average position	100 msec.
Maximum position	145 msec.

Head switching time does not exceed 60 usec in the same stack. The recovery time for a read following a write is 60 usec.

### Disk File Characteristics (approximate figures)

Number of tracks/ disk	512
Bit capacity per track	86,930 bits
Sectors per track	40 sectors
Bits/head/sector	2,166 bits
preamble	165 bits

data	1,935 bits
post-amble	66 bits
Index	290 bits
Heads Parallel	16
Data/Sector/Head Group (516 60 bit words)	30,960 bits
Sectors/Rev.	40
Data/Rev./ Head Group	1,238,400 bits
Head Groups/Stack/ Unit	2
Devices/System	2
Total Storage/System	316,000,000 bits
Data/Position/ Stack/Unit	2,476,800 bits
Positions/Stack/Unit	512 track positions
Data/Stack/Unit	1,268,121,600 bits
Stacks/Unit	2
Data/Unit	2,536,243,200 bits
Units/Read-Write Control	1
Units/Device	2
Total Storage Device	5,072,486,400 bits
Devices/System	1
Total Storage/System	5,072,486,400 bits

The 7600 Drum Unit consists of a rotating drum coated with a magnetic material and the associated read, write, and head select electronics. The drum is driven by an integral three-phase induction motor. The speed of the drum is 1,800 rpm minus a maximum 5% induction slip. There are 128 groups of 16 heads. The nominal data rate of the drum is 380 nanoseconds per 16 bit word. Since the PPU interface receives/transmits 12 bit bytes, the nominal PPU data rate of the drum system is 285 nanoseconds per 12 bit byte.

### Drum Characteristics (approximate figures)

Bit capacity per track	85,400 bits
Sectors per track	40 sectors
Bits/head/sector	2,135 bits
pre-amble	165 bits
data	1,935 bits
post-amble	35 bits
Index	1,000 bits
Heads-Parallel	16
Data/Sector/ Head Group (516 60 bit words)	30,960 bits
Sectors/Rev.	40
Data/Rev./ Head Group	1,238,000 bits
Head Groups/ Stack/Unit	128
Data/Position/ Stack/Unit	158,000,000 bits

Positions/Stack/Unit	1
Stacks/Unit	1
Data/Unit	158,000,000 bits
Units/Read-Write	
Control	1
Units/Device	1
Total Storage/Device	158,000,000 bits

The controller contains two read/write control units. Each control unit can be operated by two 7600 PPU on a time-shared basis. Each PPU uses two input-output channels (one for data and one for control) to operate the controller. Each read/write control unit operates one disk file storage unit or one drum. The two PPU channels are identical, but the signals are defined differently. The control channel transmits output function codes and receives input status codes. After the necessary functions are issued on the control channel, the data transfer takes place on the data channel. The read/write control units do not contain any reservation logic for the two PPU interfaces. Thus, the PPU must reserve control units with software. The software ensures that only one channel of a read/write control unit is operating at one time. The only exception is that status can be read at any time. A 16 bit cyclic parity byte is generated by the controller for the data field of each record written on the peripheral unit. The parity byte is written on the storage unit at the end of the record. During a read operation, the parity byte is read and compared to the parity byte generated for the record being read. If the parity does not compare the parity error status bit sets. Small block data transfers can be accomplished by writing on alternate sectors and considering the PPU as being reserved for one storage unit. For large block data transfers, two PPU can operate one storage unit in a mode where PPU A operates on the even-numbered sectors and PPU B operates on the odd-numbered sectors. Thus the transfer rate of one block is twice as fast.

Other 7600 peripherals include card readers and punches, tape transports, line printers, removable disk storage drives, and CRT displays. These will communicate with the system PPU through programable equipment controllers.

Operating specifications will be comparable or identical to those of 3000 and 6000 Series peripherals.

#### System configurations

Typically, a 7600 system utilizes a small number of peripheral equipments used primarily for system functions, and a

larger number of peripheral equipments grouped as operating stations and used primarily for user functions.

The system-oriented peripheral equipments include a card reader and CRT display reserved for the maintenance function and used in conjunction with the MCU. Additionally, one or more Disk/Drum Subsystems and a variable number of PPU are utilized for system-oriented functions; of these, one Disk Subsystem configured as two independent units and four PPU constitute the minimum recommended configuration. The configuration of the operating stations will vary, and is subject to continued refinement as system operating experience accumulates. In general, both general purpose and special purpose stations may be utilized; the latter typified by a tape input-output station operated in conjunction with a centralized tape vault, and the former by the configuration of Figure 7. Operating stations may be located remote from the central complex through use of communications links and interfaces, and when employing programable equipment controllers, may be utilized for a variety of CPU-independent processing.

#### System software concepts

In general, 7600 software releases will provide source language compatibility with their then current 6000 Series counterparts. Exceptions will be made only in the interest of improved system performance or compliance with evolving language standards.

Product Set releases will provide an increasing variety of user facilities, with each retaining all capabilities of previous releases. Facilities will include batch and time shared system operation, and comprehensive sub-systems for information management, on-line diagnostics, and job or system resource accounting. Additionally, the system will emphasize fail safe methods for recording and include automatic facility for restart and recovery from a variety of hardware and user errors. Specifications and schedules will be published separately, but will be consistent with the following general descriptions and will be oriented toward the use of programable equipment controllers:

Assembly languages, FORTRAN, and loader

COMPASS: CONTROL DATA 7600 COMPASS is a COMPrehensive ASsembly System which provides a symbolic programming language for the 7600 central processor. It is designed to

efficiently use computer resources while giving the user maximum flexibility in program construction.

The COMPASS language allows all hardware CPU functions to be expressed symbolically. In addition, many features are included which enable the programmer to control the assembly process itself. COMPASS is executed in the CPU.

#### Features provided by COMPASS include:

- Free-field source format
- Selective assembly of code sequences based upon assembly-time tests
- Assembly-time access to symbol table information
- Programmer control over local and common code blocks
- Subprogram linkage with external symbols and entry points
- Data generation deferred to load time
- Symbol equation and redefinition
- Macro coding; both program and system defined
- Micro coding
- Extensive listing control
- Thorough language diagnostics

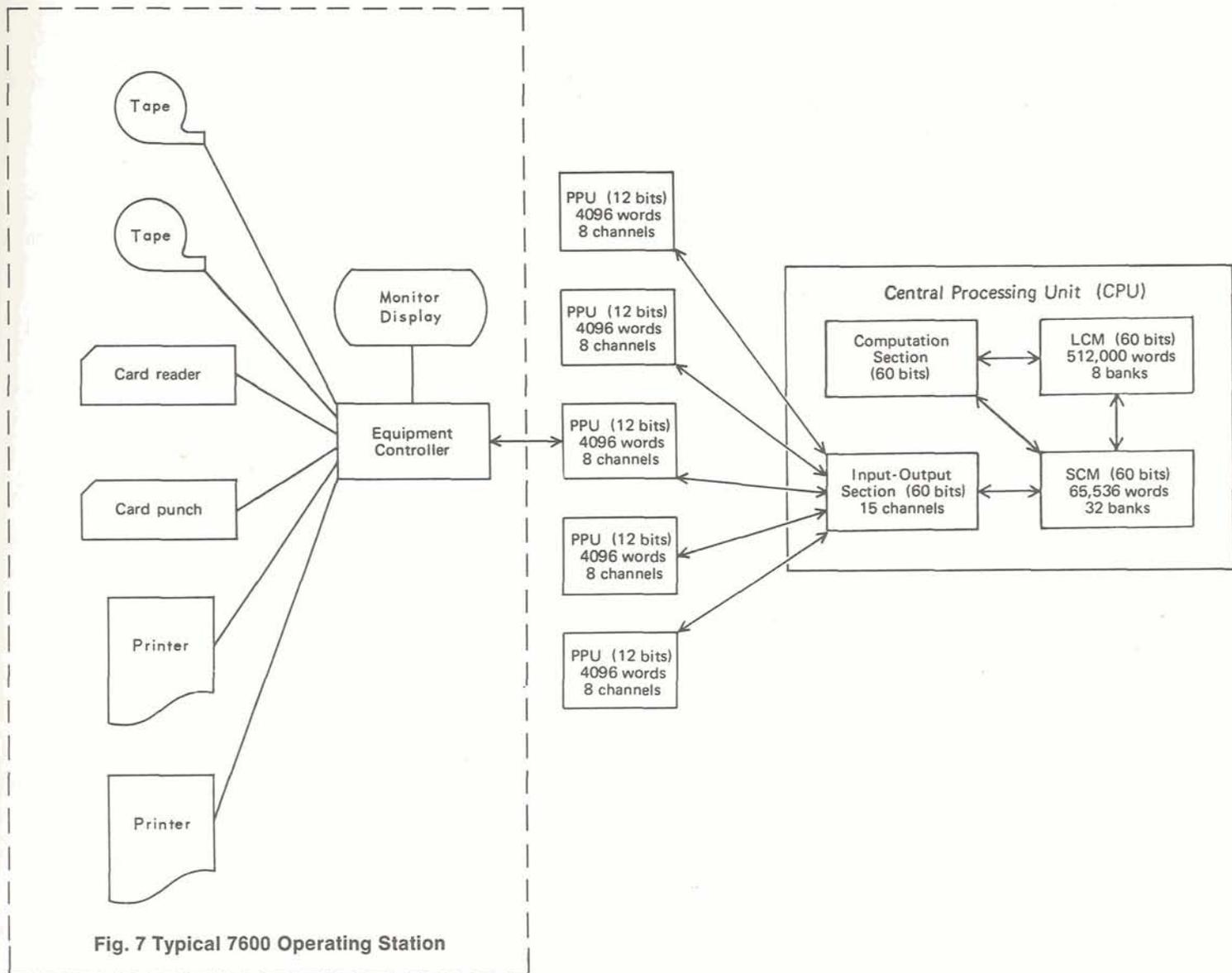
**FORTRAN:** FORTRAN is a procedural language designed for solving problems of a mathematical or scientific nature. CONTROL DATA 7600 FORTRAN utilizes an extremely fast compiler which is compatible with FORTRAN IV and USASI FORTRAN with extensions. Compilations may be produced simultaneously with other compilations or with execution of other programs in the multi-programming mode. A program consisting of mixed FORTRAN and COMPASS subprograms may be compiled, assembled and executed as a unit.

Emphasis is placed on efficiently producing relocatable binary output and on USASI compatibility. Subprograms are compiled independently, and a file consisting of relocatable binary subprograms is produced. Options are available to produce a source listing, an object code listing, a cross reference listing, and a relocatable binary deck.

The relocatable binary subprograms are processed by the 7600 loader.

The compiler can operate as a load-and-go unit and produce machine language output. It operates as an independent program under control of the operating system and can be called to use only the storage required for compilation of a particular program.

**7600 LOADER:** The 7600 CPU loader performs the following functions: loads absolute and relocatable binary



programs, links separately compiled or assembled programs, loads library subprograms and links them to user programs, detects errors and provides diagnostics, outputs a memory map, and generates overlays.

Three types of loading are performed: normal, segment, and overlay. In normal loading, absolute or relocatable binary subprograms which have been compiled or assembled independently are brought together in LCM for execution in SCM.

Segments and overlays allow programs that exceed storage to be organized so that portions or groups of programs may be called, executed, and unloaded as needed.

Overlays are subprograms in absolute form and are loaded at execution time

without relocation for high-speed loading. There are several levels of overlays. Segments are groups of subprograms in relocatable form and are loaded and unloaded as units. Each segment is assigned a level number when it is loaded. Level zero is reserved for the main segment. Loading a segment displaces segments with level numbers greater than or equal to its level number. Segments with level numbers lower than the segment loaded are left intact. When a segment is loaded, its references are linked to subprograms and common blocks of previously loaded segments. When a segment is displaced by another segment of equal or lower level number, the displaced segments are delinked from the segments that are to remain. References left unsatisfied may be satisfied by entry points in a newly loaded segment.

#### Commercial data processing programs

The 7600 software product set includes COBOL and a set of problem oriented languages which allow the commercial data processing user to define, maintain, sort, and retrieve data from files. These programs adhere to USASI standards where applicable. The systems which comprise the commercial data processing product set include:

COBOL  
SORT  
Information System  
Table Generator  
Report Generator

These systems will be augmented by a data manager which will organize the transmission of data to and from peripheral devices. This provides the user with task flexibility, ease of definition for a range of problems, and

efficient utilization of mainframe and peripherals by centralization of I/O.

**DATA MANAGER:** Although I/O requests can be made directly to a physical record processor, the data manager is the lowest entry point in the hierarchy for generalized I/O requests. The data manager is logically divided into two segments — file manager and logical record processor. The file manager catalogs permanent files, allocates storage, and performs label handling, security checks, and job accounting. The logical record processor formats I/O and checkpoint requests, blocks and deblocks logical records, provides access methods (e.g., sequential, indexed sequential, and direct), and accumulates data transfer statistics such as block counts, record counts, and error counts. The data manager will have a modular structure such that only those modules required will reside in core.

**COBOL:** USASI standards separate the language into a nucleus and a number of modules each performing a specific task. These modules are:

Table Handling	SORT
Sequential Access	Report Writer
Random Access	Segmentation
Random Processing	Library

The nucleus and each of the modules adhere to USASI specifications, and, as such, form a subset of 7600 COBOL. Where no conflicts exist with these specifications, COBOL is externally compatible with 6000 COBOL. For example, 7600 COBOL incorporates INCLUDE and ENTRY statements, and other items currently utilized by 6000 COBOL. Many of the above modules are not restricted to COBOL. For example, sequential and random access are handled by the file manager, while the SORT and report writer modules can function as stand-alone systems.

**SORT:** 7600 SORT assumes wider responsibilities than its 6000 counterpart. Run, file, and record attributes as well as control cards differ from 6000 SORT. The user needs only to describe input and output files and sort keys. The system itself selects the most efficient sorting algorithm and merging techniques.

**REPORT GENERATOR:** A superset of the COBOL report writer module, the report generator formats raw data as it is being retrieved from files, establishes line and page images, appends headings and page numbers, and selectively prints data according to user furnished criteria.

**INFORMATION SYSTEM:** The information system is a set of problem oriented languages which enables the user to create, amend, and retrieve data from data files. A feature of the system is the ability to retrieve data as a function of several variables, either by control card command or in an interactive mode. When the requirement is to retrieve and possibly structure large amounts of data from sequential files, control cards are normally used. When, on the other hand, the requirement is to retrieve limited amounts of data from a random file, it becomes advantageous to use an interactive mode with limited response time.

**TABLE GENERATOR:** A table, as interpreted by the table generator, is a multi-dimensional array, each dimension of which is defined by a single variable or a number of catenated variables. The table generator is a generalized program which creates any number of tables from a file consisting of fixed and/or variable length records. Up to 25 record types may appear in any one file and may be structured into a hierarchy. The items or data fields within a record may be alphanumeric codes or binary information, and they may be resequenced as desired. The resulting tables can assume a wide variety of output formats.

The system first retrieves the necessary information for the tables, sorts it into the desired sequence, and then prepares output in the form of table elements on a set of files on a disk, tape, or equivalent medium. These files subsequently serve as input to the report generator program.

#### Remote Access Software

**CONTROL DATA 7600 EXPORT/IMPORT** programs allow multi-access to a 7600 computer from remotely located operating stations in a batch job processing mode. Users at remote sites may submit jobs and receive printed output in the same way they would at stations local to the computer. Each remote terminal connects to the central computer via communications lines.

The software for this remote system consists basically of the EXecutive Processor Of Remote Tasks (EXPORT), which resides in a Peripheral Processor assigned to remote communications at the central facility, and the Input/Output Monitor for Processing Of Remote Tasks (IMPORT), which resides in each remote operating station.

The software control program, EXPORT, handles several remote stations and

performs the following functions for each:

- Monitors all lines connected to the controller to which it is assigned.
- Loads jobs for the central facility from the remote station.
- Outputs data to the appropriate remote station.
- Changes program parameters (priority, run time, etc.) as requested by the remote operator.
- Monitors each remote job and, upon request, sends the status of the job to the remote station.
- Monitors the status of jobs at the central facility so that remote output is handled properly.

The operator at the remote site initiates the remote communication operation. By entering commands, the remote operator can interrogate the system for status information, change priorities and times limits or terminate jobs.

**EXPORT/IMPORT** communicates with the remote operator by informing him of status on the flow of jobs through the system and certain error conditions.

The software control program, IMPORT, performs the following functions:

- Inputs jobs from the card reader in the same format as jobs loaded at local operating stations.
- Transmits card images via the communication equipment to EXPORT at the central site.
- Receives the results of jobs from the central site and outputs them on the appropriate remote station peripheral device.
- Communicates with EXPORT by transmitting and receiving information concerning current job processing functions.

During transmission of each data block, a cyclic code word is generated. At the receiving end, this code word is used in an error detection scheme; when an error-free data block is received, the next data block is transmitted. This error detection code word provides a reliability factor in excess of 99.9 percent.

The EXPORT/IMPORT system is connected to the central computer via voice grade or broad band communication links, depending on the nature of the application being performed at the remote station.

#### Additional products

Additional software products will be announced and specified at a later date. These will include interactive terminal access facilities, mathematical and simulation languages, PERT, APT, and other technical applications packages.

60258400

**CONTROL DATA**  
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440  
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD