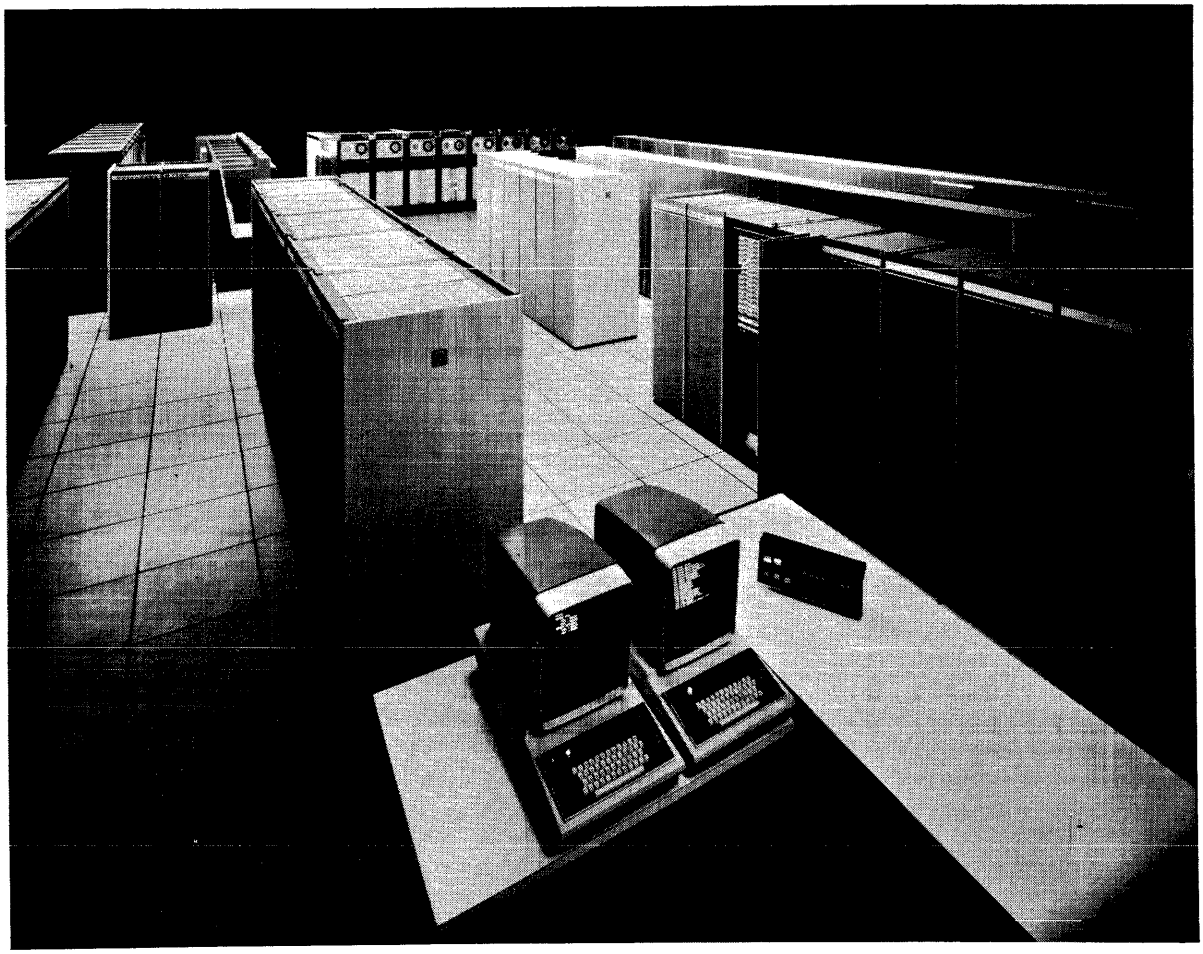
Burroughs

B 7700 SYSTEMS

CHARACTERISTICS MANUAL



Burroughs
B 7700
INFORMATION PROCESSING SYSTEMS

CHARACTERISTICS MANUAL

Burroughs Corporation
Detroit, Michigan 48232

Printed in U.S. America 1-73 1059979

Correspondence regarding this document should be forwarded using the Remarks Form at the back of the manual, or may be addressed directly to Burroughs Corporation, Technical Information Organization, 200 W. Lancaster Avenue, Wayne, Pennsylvania 19087.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

This volume covers the characteristics (both hardware and software) of the Burroughs B 7700 Information Processing System--the most advanced, the largest, and the most powerful member of the Burroughs family of 700 systems.

The four sections of this characteristics manual are as follows.

Section 1, General Description of B 7700 System: an introduction to the idea of the interaction of independently operating computing, input/output, and memory modules through an exchange and to the features that account for the high throughput and availability of the system.

Section 2, System Configuration: a discussion of the range of configurations of the B 7700.

Section 3, Components and Subsystems: summaries of the characteristics and features of the principal components and functional subsystems of the B 7700.

Section 4, Software: an introduction to the standard software of the B 7700, including a brief functional description of the master control program (MCP)--the unique executive program that automatically makes optimum use of all system resources.

The term "software" as used in this manual applies to that category of Burroughs Program Products defined as "Systems Software".

Other categories of Burroughs Program Products are:

Application Program Products
Program Product Development Aids
Program Product Conversion Aids

SECTION 1
GENERAL DESCRIPTION OF B 7700 SYSTEM

THE B 7700 SYSTEM

The Burroughs B 7700 Information Processing System is a large-scale, truly general-purpose, balanced, flexible, multiprogramming and multiprocessing computing system that is suitable for such diverse applications as time sharing, scientific problem solving, and business data processing. Carrying forward ideas proven successful in the Burroughs B 5700 and B 6700 information processing systems, the B 7700 is, in fact, fully code compatible with the B 6700 and affords Burroughs users the opportunity for growth without reprogramming or recompiling. In other words, object code of users' programs that can be executed successfully on the B 6700 can be executed without modification on the B 7700, and object code that can be executed on the B 7700 can be executed without modification on the B 6700. Nevertheless, the B 7700 is designed to satisfy the increasingly complex data processing needs of the years to come. The system is able to handle complex data structures and sophisticated program structures dictated both by higher-level languages now in use and by the requirements of advanced problems, is able to manage efficiently the massive on-line and archival storage requirements of large data bases, and is able to accommodate vast networks of data communications devices.

A very fast, modular parallel processing system with exceptional versatility in configuration, the B 7700 can be tailored to the processing needs of a user by arranging central processor modules, input/output modules, and memory modules on an electronic grid, or exchange (figure 1-1), in a variety of ways depending upon the exact needs of the user. If the high performance and adaptability of the B 7700 could be attributed to a single factor, it would be to the balance attained by means of controlled interaction of independently operating computing, input/output, and memory modules through the exchange. Thus, the throughput of the system as a whole is maximized, and the performance of no single element of the system is maximized to the neglect or detriment of others.

The key to the efficient and balanced use of the system is the Burroughs master control program (MCP)--a unique executive software operating system that automatically makes optimum use of all system resources. It is this operating system that makes multiprogramming and multiprocessing both functional and practical by dynamically controlling system resources and by scheduling jobs in the multiprogramming mix. In use, the master control program allocates system resources to meet the needs of the programs introduced into the computer. It continually and automatically reassigns resources, starts jobs, and monitors their performance.

Further implications of the modularity and flexibility of the system are its expandability (a capacity to add hardware modules without reprogramming) and its increased reliability (and thus increased availability to the user). This reliability is achieved by the use of fail-soft techniques that (in addition to providing for error detection and error correction, redundancy of data paths, and independence and redundancy of power supplies) exclude faulty modules from the system and permit processing to continue (again, without reprogramming) even with a temporarily reduced configuration.

Even though it is very large and immensely complicated and thus able to perform complex computations, the B 7700 is, nevertheless, comprehensible to the persons who use it: programming is done only in higher-level, problem-oriented languages; the control language used in entering jobs into the system is a simple, free-form, English-like language; and the messages that pass between the system and the operator are brief, clear, and easy to learn.

DISTINGUISHING FEATURES

Although the balanced use of the principal components of the system as a whole under the control and coordination of the master control program is the key to the high throughput of the B 7700, the high performance of the system is in large part achieved by improving the speed of execution of instructions, by reducing or masking the overhead associated with references to memory, by freeing the central processor from concern with input/output operations, and by employing fail-soft measures that minimize system degradation.

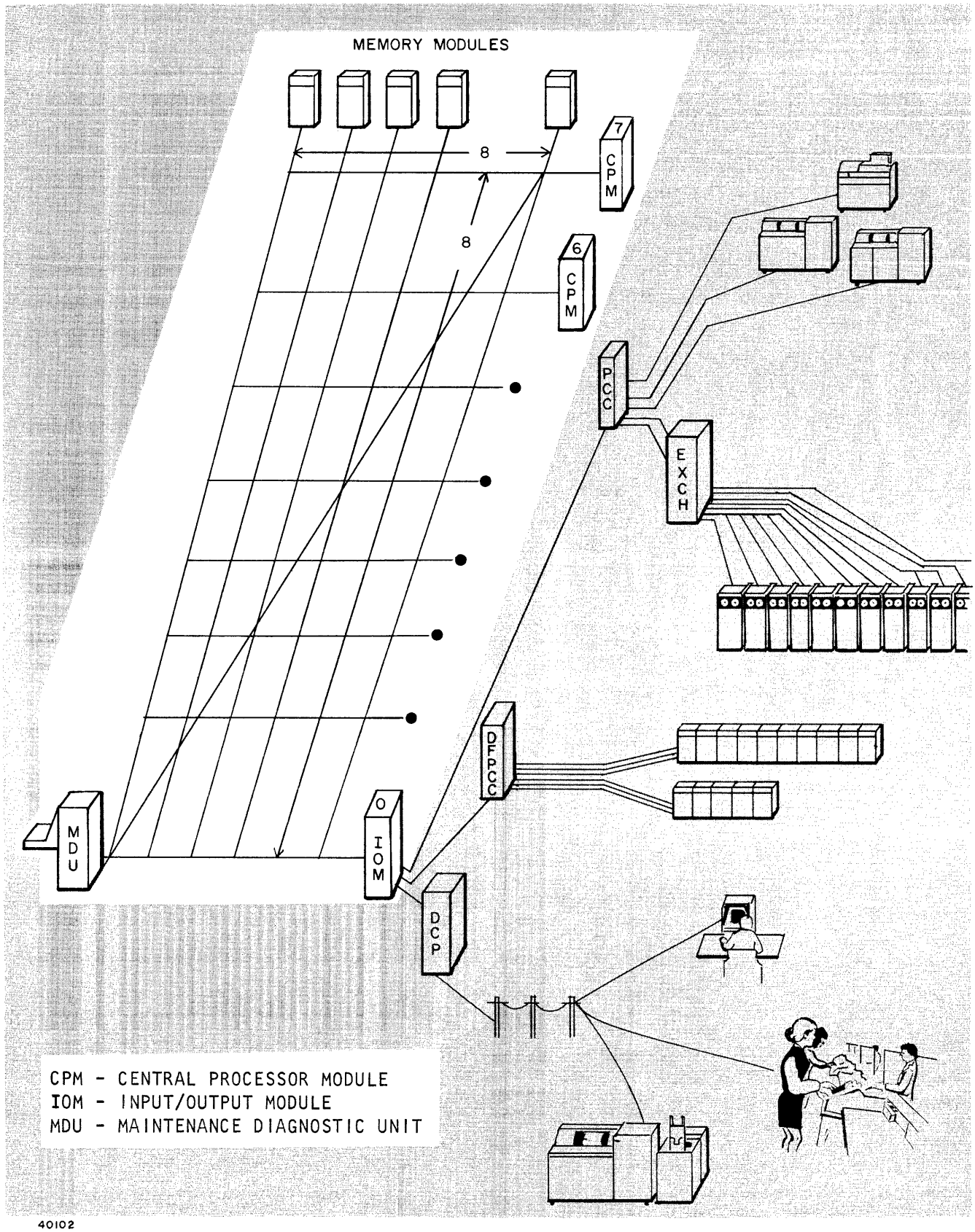


Figure 1-1. B 7700 Exchange

Because system main-frame hardware has been designed and built strictly according to stringent circuit and wiring rules and proven design and packaging techniques and because its processing elements incorporate monolithic integrated circuits, the B 7700 system performs consistently at high operating frequencies: the central processor module at a clock rate of 16 megahertz and the remainder of the system at 8 megahertz.

By combining the following features with the high internal operating frequencies, the performance of the system is further enhanced.

1. The parallel and independent operation of the three main sections (program, execution, and storage) of the central processor module. This parallelism (coupled with the high clock rate) makes possible the speeding up of arithmetic computations and data manipulations and the overlapping of these computations and manipulations with memory references.

2. A special high-speed integrated circuit memory (program, stack, and associative data buffers). This high-speed local memory permits multiword transfers between the central processor and main memory and makes possible the anticipation of the need for program and data words. Hence, the time spent waiting for the completion of transfers to and from memory is reduced and at times virtually eliminated.

3. The four-way interleaving of addresses in main memory and the capability for phased multiword transfers of information to and from memory in groups of up to four words. Consequently, memory access times for each user of memory are reduced, and memory is thus made more accessible to all users.

4. The asynchronous performance of input/output operations by the input/output module independent of the central processor, which is therefore freed to do other useful work.

The three goals of the fail-soft features of the B 7700 are to keep the system running 100 percent of the time, to minimize system degradation, and to provide the user with tools for performing his own data recovery. These goals are achieved by the artful combination of hardware and software throughout the system.
The first goal--to keep running--is achieved as follows.

1. By the high reliability of system hardware.

2. By the incorporation of error detection circuits throughout the system.

3. By single-bit error correction of errors in memory.

4. By recording errors for software analysis.

5. By modular design, by use of separate power supplies and redundant regulators for each module, and by use of redundant buses.

6. By the ability of the master control program to reconfigure the modules of the system to temporarily exclude a faulty one.

In short, the detection and reporting of errors is done by hardware, analysis of errors is done by software, and the reconfiguration of the system is done dynamically by software. Because of the modularity of power supplies and the use of redundant regulated supplies for critical voltages, the impact of a malfunctioning dc supply is minimized and does not result in a catastrophic failure.

The second goal--to minimize system degradation--is achieved by providing diagnostic programs and equipment for rapidly identifying and repairing faults and for reestablishing confidence in a repaired module before it is returned to the user's system. The diagnostic programs of the B 7700 system identify a faulty module. By the use of the maintenance diagnostic unit, a fault in any main-frame module or in a disk file optimizer is narrowed to a single clock period and to a flip-flop and its associated logical circuits. Finally, by the use of the card tester on the maintenance diagnostic unit, the faulty integrated circuit chip is identified.

The third goal--to provide the user with tools for performing his own data recovery--is achieved by the use of such features as installation allocated disk, protected disk files, duplicated disk files, and fault statements in the higher-level programming languages used on the system.

Installation allocated disk allows the user to specify the physical allocation of his critical disk files in order to facilitate the maintenance and reconstruction of these files. Protected disk files allow a user to gain access to the last portion of valid data written in a file before an unexpected system halt. The use of duplicated disk files is to avoid the problem of fatal disk file errors. The master control program maintains more than one copy of each disk file row, and, if access cannot be gained to a record, an attempt is made to gain access to a copy of the record. By the use of fault statements, the user can stipulate the actions to be taken by his programs in case certain errors occur.

SECTION 2
SYSTEM CONFIGURATION

Physically, the components of the B 7700 system fall into three categories, as follows:

1. Central components of the B 7700 system--the central processor module, the input/output module, the memory module, the maintenance diagnostic unit, and the operator's console (see table 2-1).
2. Standard Burroughs cabinets that contain peripheral controls and exchanges, the disk file optimizer, the data communications processor, and ac power supplies.
3. Standard peripheral devices that are joined to the central system by means of standard Burroughs peripheral controls, adapters, and exchanges and standard remote devices that are joined to the central system by means of line adapters and the data communications processor.

The arrangement of these components into a system and the size of the system depend on the application and workload of the user. In the following paragraphs, the range of configurations of the B 7700 --the maximum configuration, the minimum configuration, and the typical configuration with full fail-soft capabilities--is described.

Table 2-1. Central Components of the B 7700 System

Style Number	Name	Description
B 7700	Basic system	One 16-megahertz, parallel processing central processor module One asynchronously-operating input/output module containing four multiword channels for disk file controls, 20 word channels for peripheral controls, four word channels for data communications processors, and one disk file optimizer adapter Processor/memory exchange Operator's console and control One maintenance diagnostic unit
B 7701	Additional central processor module	See above.
B 7780	Additional input/output module	See above.
B 7780-1	Multiword channels	Four multiword channels for disk file controls and one disk file optimizer adapter

Table 2-1. Central Components of the B 7700 System (Cont)

Style Number	Name	Description
B 7001-2	Memory module	786,432 eight-bit bytes (131,072 words) of core memory storage 1.5-microsecond cycle time Two-way interleaving that permits two-word transfers to and from memory
B 7001-4	Memory module	1,572,864 eight-bit bytes (262,144 words) of core memory storage 1.5-microsecond cycle time Four-way interleaving that permits four-word transfers to and from memory
B 7341	Additional operator's display control	Controls a maximum of eight operator's display terminals
B 9342-1	Additional operator's display terminal	
B 9951-7	Console display stand	Low, without work table
B 9951-8	Console display stand	High, without work table
B 9951-9	Console display stand work table	Right or left

MAXIMUM CONFIGURATION

Figure 2-1 illustrates the theoretical maximum configuration of the B 7700 system. As many as eight memory modules may be arranged on the exchange with a combined total of up to eight requestors of memory--central processor modules and input/output modules. Any single requestor of memory may address and gain access to the entire contents of high-speed main memory (1,048,576 words, or 6,291,456 eight-bit bytes). On the maintenance bus (which services the memory control modules, central processor modules, input/output modules, and disk file optimizers) one or two maintenance diagnostic units may be placed.

At rates of up to 6.75 million bytes per second, a single input/output module is capable of transferring data simultaneously between main memory and 28 peripheral controls (including eight high-speed controls) and between main memory and as many as four data communications processors. It is also capable of handling as many as four disk file optimizers (devices that are used in improving the rate of transfer of data between main memory and disk files). At present, the maximum number of high-speed, medium-speed, and low-speed peripheral devices that may be attached through controls and exchanges to a single input/output module or that may be included in the input/output subsystem of the B 7700 is 255. (Each card reader, pseudoreader, card punch, line printer, paper tape reader, paper tape punch, operator's display terminal, and free-standing magnetic tape unit; each station on a magnetic tape cluster; and each electronics unit in a disk file subsystem is considered a device.) By suitable cross-connection through exchanges, it is possible to establish pathways between disk files, disk packs, or magnetic tape units and more than one input/output module; hence, these peripheral devices can be shared between the input/output modules in the system.

Among the peripheral devices available are disk file and disk pack memory modules that constitute a virtual memory that in effect greatly expands the storage capacity of the main memory of the system; these modules, which are interfaced with one input/output module through controls are as follows:

1. Head-per-track disk file modules that are combined under the control of disk file optimizers to form optimized-access memory banks capable of storing from 450 million to 8 billion eight-bit bytes of information per input/output module and whose access time is as low as 2 to 6 milliseconds or 4 to 10 milliseconds.

2. Head-per-track disk file modules that are combined (without the control of the optimizer) into random-access memory banks of from 15 million to 16 billion eight-bit bytes per input/output module and whose average access time is 23 or 40 milliseconds (dependent on the type of storage unit).

3. Disk pack memory modules that are combined into random-access memory banks with a capacity of from 121 million to many billions of eight-bit bytes of storage per input/output module and whose average access time is 30 milliseconds.

Besides the 255 peripheral devices that may be included in the input/output subsystem, there is a vast network of remote terminals, remote controllers, and remote computers that can be accommodated by the up to 1024 remote lines serviced by the four programmable data communications processors that can be controlled by a single input/output module. Normally, each line handles a number of remote devices, and, naturally, systems that have more than one input/output module can have more than one data communications network. The maximum number of data communications processors that may be included in a B 7700 system is 28.

MINIMUM CONFIGURATION

The smallest possible B 7700 system is composed of the central components listed below.

Central Components	Quantity
Central processor module (CPM)	1
Input/output module (IOM)	1

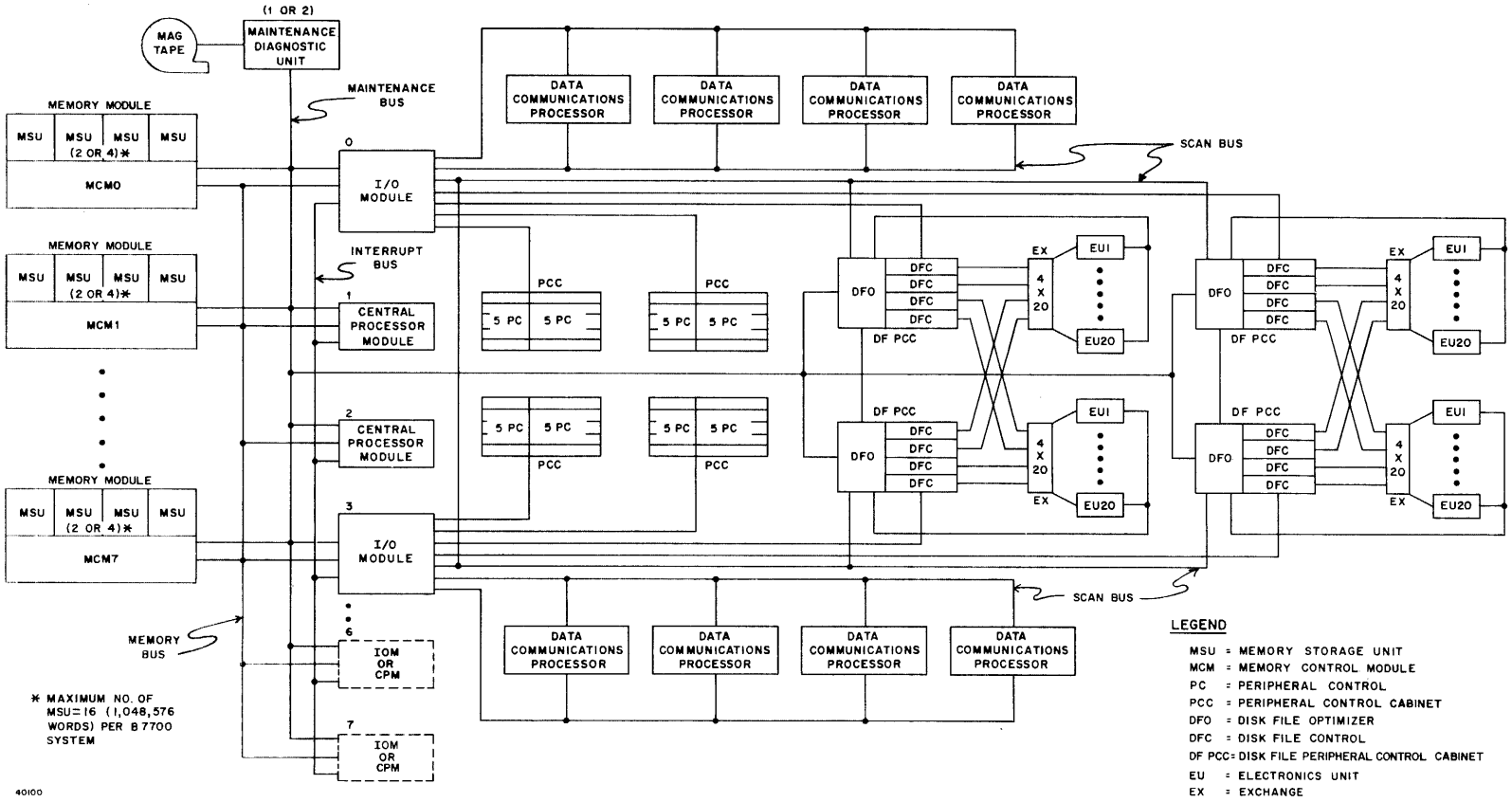


Figure 2-1. Maximum Configuration of the B 7700 System

Central Components	Quantity
Memory module	1
Memory control module (MCM)	1
Memory storage cabinet (MSC)	1
Memory storage unit (MSU)	2
Maintenance diagnostic unit (MDU) and its associated magnetic tape unit	1
Operator's console	1

Besides these central components, the minimum configuration must contain a disk file memory subsystem at least large enough to hold the master control program, a card reader, a line printer, a magnetic tape unit, peripheral controls, and ac power cabinets. In practice, other peripheral devices and their controls are used with this minimum configuration.

Naturally, this minimum system lacks the redundancy and power of larger configurations. First (lacking redundancy of main-frame modules) this configuration does not take full advantage of the fail-soft features possible with the B 7700 and second (because each memory control module controls but two storage units) two-word transfers, not four-word transfers, to and from memory are possible.

TYPICAL CONFIGURATION WITH FULL FAIL-SOFT CAPABILITIES

The power, speed, flexibility, and reliability of which the B 7700 is capable are fully realized in a system that includes the following central components.

Central Components	Quantity
Central processor module (CPM)	2
Input/output module (IOM)	2
Memory module	4
Memory control module (MCM)	4
Memory storage cabinet (MSC)	8 (2 per MCM)
Memory storage unit (MSU)	16 (2 per MSC)
Maintenance diagnostic unit and its associated magnetic tape unit	1
Operator's console	2 (1 per IOM)

Besides these central components, this typical fail-soft configuration must contain two disk file memory subsystems (one for each input/output module) or a single disk file subsystem that is shared by means of exchanges by the two input/output modules, peripheral controls, and ac power cabinets. Naturally, a complement of peripheral devices and their controls and exchanges, data communications processors, and remote devices suited to the application and workload of the system is also needed.

A system of the proportions described above incorporates fully the fail-soft features of the B 7700 and takes complete advantage of its capability of handling four-word transfers of data to and from main memory.

SECTION 3
COMPONENTS AND SUBSYSTEMS

The following paragraphs and tables contain very brief and general descriptions of the principal components and functional subsystems that—under the control of the master control program and arranged in configurations suited to particular data processing needs—make up a B 7700 system. These components and subsystems are as follows.

1. Central processor module.
2. Input/output subsystem.
3. Memory subsystem.
4. Maintenance diagnostic unit.
5. Operator's console.
6. Disk file subsystem.
7. Data communications subsystem.
8. Power subsystem.

CENTRAL PROCESSOR MODULE

Table 3-1 highlights the characteristics and features of the central processor module, the computational element of the B 7700 system.

Table 3-1. Characteristics And Features Of The Central Processor Module

Characteristics and Features	Description
Clock rate	16 megahertz
Number per system	At least 1, at most 7
Parallel processing	Three major, independent, asynchronously-operating sections—the program section, the execution section, and the storage section—make up the central processor module. Communication between these sections is carried out by means of queues of operations. Because of the parallelism of the central processor module, arithmetic computations and data manipulations, the calculation of addresses, and the transferring of data to and from memory may go on at the same time.
Local buffering	The use in the central processor module of special high-speed integrated circuit memories—the program buffer, the stack buffer, and the associative data buffer—reduces and at times virtually eliminates the time spent waiting for the completion of transfers of data to and from main memory. Because these buffers are filled autonomously (two or four words at a time, depending on the configuration of main memory) on the principle of anticipation rather than that of need followed by demand, the replenishment of their contents takes full advantage of normal main memory idle time.
Program buffer	The program buffer, local storage for up to 32 program words, permits tight loop capture of program loops of 30 words or less. A loop once in the buffer may be executed repeatedly without further fetching of program words from main memory.

Table 3-1. Characteristics And Features Of The Central Processor Module (Cont)

Characteristics and Features	Description
Stack buffer	The 32-word stack buffer provides local storage for (and hence quick access to) descriptors, variables, and control words at the top of the stack of a job that is being executed.
Associative data buffer	The associative data buffer, which is composed of 16 words, provides local high-speed storage for the operands and descriptors of a job that are most often used but that are not close enough to the top of the stack to be in the stack buffer.
Hardware stack mechanism	The stack structure of the B 7700 system is not merely a software fabrication imposed upon uncongenial hardware. Rather the hardware mechanism for structuring and manipulating the stack is intrinsic to the central processor module. This hardware stack mechanism makes possible the efficient handling of temporary storage and permits the control of subordinate routines, communication between processes, and the servicing of interrupts to be treated in a uniform and efficient way.
Memory protection	Memory protection (preventing a program's gaining access to or altering data not assigned to it) is made possible by a combination of hardware and software mechanisms. The hardware mechanisms include automatic detection of a program's attempt to index beyond an assigned data area and the use of control bits (set by software) that prevent a user program's changing program words, data descriptors, segment descriptors, memory links, indirect reference words, control words, and tables of the master control program.
Instruction set of four families of operators Arithmetic operators Word operators Control word operators String operators	The operators of the central processor module act upon vectors, entire words, characters, groups of bits, and single bits. The same set of operators is used in performing both single-precision and double-precision arithmetic.

Table 3-1. Characteristics and Features of the Central Processor Module (Cont)

Characteristics and Features	Description
Representation of data Single-precision and double-precision operands Strings of eight-bit EBCDIC, eight-bit USASCII, six-bit BCL (Burroughs common language), and four-bit packed numeric characters Descriptors, control words, indirect reference words and program words	Each B 7700 word contains 48 bits of information: three control, or tag, bits; and a variable number of parity bits and other error-checking bits. The control bits serve to distinguish the types of words.
Largest positive integers that can be expressed	The set of integers is symmetrical with respect to 0. Thus, the negative integer corresponding to any valid positive integer may also be expressed.
Single precision	549,755,813,887
Double precision	302,231,454,903,657,293, 676, 543
Range of positive normalized real numbers that can be expressed	The set of real numbers is symmetrical with respect to 0. Thus, the negative real number corresponding to any valid positive real number may also be expressed.
Single precision	$(8^{13} - 1) \times 8^{63}$ to 8^{-51} and 0
Double precision	$(1-8^{-26}) \times 8^{32780}$ to 8^{-32755} and 0
Handling of hardware interrupts	Interrupt conditions detected by the central processor module, the input/output module, or the memory control module are processed by the central processor module, which prepares the stack for entry into the interrupt-handling procedure of the master control program, places the needed parameters in the stack, and causes entry into the interrupt-handling procedure of the master control program. Thus, by automatically discontinuing (either temporarily or permanently, depending on the interrupt condition) the process being executed at the time the interrupt condition occurs, the B 7700 system is able to deal with nearly every condition (both normal and abnormal) that may arise in the multiprogramming, multiprocessing environment.

Table 3-1. Characteristics and Features of the Central Processor Module (Cont)

Characteristics and Features	Description
States of operation	The central processor module operates in either of two states: control state, used by the master control program, or normal state, used by both user programs and the master control program. The interrupt-handling procedure of the master control program is always executed in the control state. The differences between the two states are that in control state the processing of interrupt conditions arising outside the central processor module (external interrupts) is inhibited whereas in normal state it is not so inhibited and that in control state the central processor may execute privileged instructions (including Set Interval Timer, Idle Until Interrupt, Set Memory Inhibits, and Set Memory Limits) that it may not execute in normal state.
Modes of operation	In addition to the two states, the central processor module can operate at any one of four interrupt management levels: normal mode, control mode 1, control mode 2, and control mode 3. The central processor operates in normal mode until an interrupt condition is detected. The three control modes allow for repeated attempts to enter the hardware interrupt routine with different environments. If an interrupt is detected while the central processor is in control mode 3, the processor is halted.
Detection and reporting of errors	The use of residue checking in all arithmetic operations and address calculations and of parity checking in data transfers greatly facilitates the detection of errors within the central processor module. If a failure occurs within the central processor module, a processor internal interrupt is produced and the cause of the failure is denoted by the contents of the fail register of the processor.

INPUT/OUTPUT SUBSYSTEM

The input/output subsystem (see Figure 3-1) of the B 7700 may include from one to seven input/output modules (see table 3-2); low-speed and medium-speed peripheral devices and their controls and exchanges; the high-speed disk file subsystem; and the data communications subsystem, which is made up of as many as 28 data communications processors with associated adapters and remote devices.

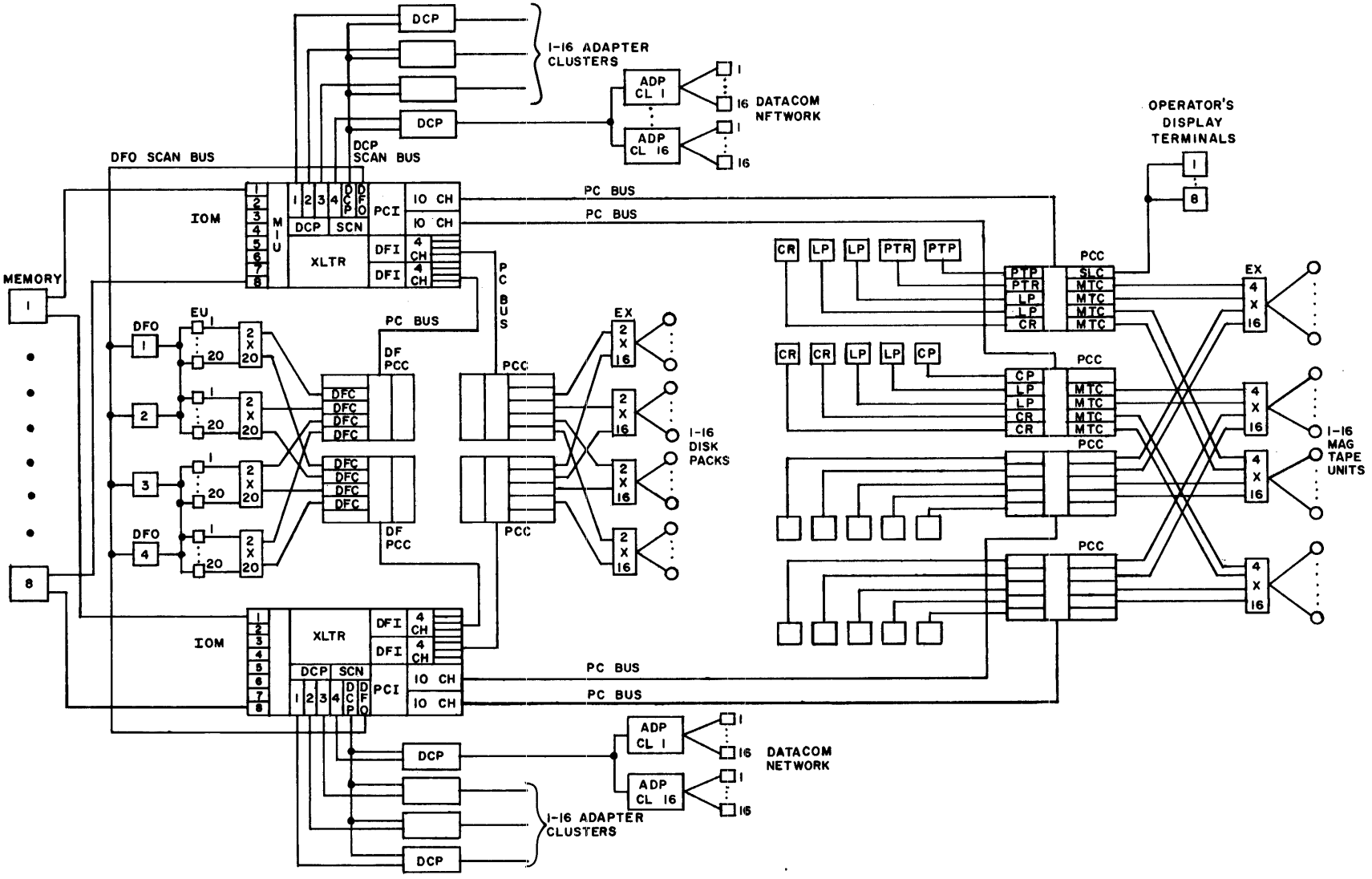


Figure 3-1. Input/Output Subsystem (Showing Theoretical Connectivity Of Devices In A Typical Two-IOM System)

Table 3-2. Characteristics And Features Of The Input/Output Module

Characteristics and Features	Description
Clock rate	8 megahertz
Number per system	At least 1, at most 7
Number of peripheral devices(including disk file optimizers but excluding data communications processors and remote terminals)	A maximum of 255 per input/output module: a maximum of 255 per input/output subsystem
Number of data communications processors (see table 3-8)	Maximum of 4 per input/output module; a maximum of 28 per input/output subsystem
Number of disk file optimizers (see table 3-7)	A maximum of 4 per input/output module
Asynchronous operation independent of the central processor module	The central processor module under the control of the master control program queues requests for input and output operations in a job stack in main memory and alerts input/output modules to the need for input and output operations; the input/output modules service these requests asynchronously and independent of the central processor module. In fact, if a number of input and output requests for the same peripheral device are queued in main memory, these requests are linked in such a way that only one interruption of the input/output module is needed for the servicing of all these requests. Hence, once a request for input or output has been queued, the central processor module is free to perform other useful work while the input/output module is performing input and output operations.
Connectivity and throughput	The data transfer rate of the input/output module equals the throughput capability of its memory port. This speed is achieved principally by concentrating the high-speed disk files on double-buffered channels that communicate with main memory in two-word phased transfers and by permitting the simultaneous transfer of data to 20 low-speed and medium-speed peripheral devices and to eight high-speed peripheral devices and to four data communications processors.
32 channels	20 channels for low-speed and medium-speed peripheral devices, 8 for high-speed disk file subsystems, and 4 for data communications processors. Also, 4 disk-file-optimizer interfaces are provided.
Maximum data-transfer rate between main memory and peripheral devices and data communications processors	6.75 million bytes per second (with all data service adapters and with a phased, 1.5-microsecond memory subsystem)

Table 3-2. Characteristics And Features Of The Input/Output Module (Cont)

Characteristics and Features	Description
Peripheral device compatibility	Standard Burroughs peripheral devices and controls, exchanges, disk file optimizers, and data communications processors used with the B 6700 are compatible with the input/output module of the B 7700 system.
Diversity of paths to and from peripheral devices	Through exchanges and redundant peripheral controls, alternative ways of gaining access to a peripheral device are provided.
Unique data and control paths and asynchronous control sections	There is a unique bus between each input/output module and each memory control module. There are also unique data and control buses from an input/output module to each peripheral control cabinet under its control and a unique memory data bus for each data communications processor. Moreover, there are unique asynchronous control sections for each interface within the input/output module.
Code translations	The input/output module is capable of performing the following code translations: EBCDIC to USASCII EBCDIC to BCL external USASCII to BCL external USASCII to EBCDIC BCL internal to BCL external BCL external to BCL internal BCL external to USASCII BCL external to EBCDIC
Interrupts generated by the input/output module	The input/output module generates interrupts in response to the following conditions:
Channel interrupts (related to specific requests and devices)	The completion of an input/output operation, if the software has requested that an interrupt be generated; otherwise, because the central processor module and the input/output module operate independently and asynchronously, this condition does not trigger an interrupt. A change in peripheral status from a single-line control device The need of a data communications processor for the attention of a central processor module

Table 3-2. Characteristics And Features Of The Input/Output Module (Cont)

Characteristics and Features	Description
	The detection of an error that is related to an input/output request or to a peripheral device, for example, a parity error on a magnetic tape unit. Errors of this type are reported in a result descriptor, prevent further servicing of the device on which the error occurred, and trigger interrupts only as requested by software.
IOM error interrupts (not related to a specific request or device)	The detection of an error that is not related to a specific input/output request or peripheral device. Errors of this type are reported in the fail register of the input/output module, the contents of which are placed in a result descriptor.
Detection and reporting of errors	There are facilities for detecting errors that may occur in any operation in which data are transferred into or out of the system. Among the error detecting features of the input/output module are parity checking of data transfers, residue checking of all arithmetic operations, parity checking of all local memory operations, timeout on memory transfers and scan bus operations, memory bounds checking, detection of illegal commands and conditions, and parity checking of register-to-register transfers. Particular care is taken in addressing main memory: residue checks are made in the calculation of memory addresses, and bounds checks are made each time an attempt is made to gain access to main memory. When a failure occurs in the input/output subsystem, it is reported in a result descriptor that pinpoints the fault. If the fault is not related to a specific request or device, an IOM error interrupt is produced.

MEMORY SUBSYSTEM

As many as eight memory modules (see table 3-3) with a maximum of 1,048,576 words of storage may make up a memory subsystem. A memory module may consist either of one memory control module and one memory storage cabinet or of one memory control module and two memory storage cabinets. These two types of memory modules may be intermixed in a memory subsystem.

Table 3-3. Characteristics And Features Of The Memory Module

Characteristics and Features	Description
Number per system	At least 1, at most 8 (a maximum of 1,048,576 words)
Capacity (of a memory module)	B 7001-2 (one memory control module and one memory storage cabinet): 131,072 words; B 7001-4 (one memory control module and two memory storage cabinets): 262,144 words
Capacity (of a memory storage cabinet)	131,072 words (786,432 eight-bit bytes). Each memory storage cabinet contains two stacks, or memory storage units of 65,536 words each.
Clock rate	8 megahertz
Cycle time	1.5 microseconds
Access time	0.8 microseconds
60-bit memory word	48 data bits, 3 tag bits, 1 parity bit for the first 52 bits, 7 check bits, and an overall parity bit. The check bits and the overall parity bit are used for the detection of one-bit and two-bit errors and for the correction of one-bit errors.
Phasing, or multiword transfers of data at the clock rate	The transfer of information between a memory module and a requestor occurs in bursts of two words or four words, depending on the configuration of the memory module. Thus, in effect, the time it takes to perform a memory operation on one word is greatly reduced. After the first word of data has been transferred, succeeding words are transferred at the rate of 1 each clock period. For example, whereas it would take 1.750 microseconds to read a single word in a single-word transfer, it takes on the average only 0.531 microsecond to read a word in a four-word transfer.

Table 3-3. Characteristics And Features Of The Memory Module (Cont)

Characteristics and Features	Description
Two-way or four-way interleaving of addresses	Placing consecutive addresses in adjacent memory storage units, or stacks, makes it possible to transfer words to and from memory in bursts of two words (with two stacks per memory control module) or four words (with four stacks per memory control module). By arranging memory in this way, preparation for a memory operation in one stack goes on simultaneous with the performance of an operation in another stack.
Addressing	A central processor module or an input/output module can address up to 1,048,576 words of memory. The range of addresses within a memory module is designated by the contents of the memory limits register of the memory control module, and whether or not a particular requestor may have access to a memory module depends on the setting of the access mask of the memory control module. Both of these registers can be set programmatically by the master control program.
Reconfiguration	If a memory storage unit or a memory control module should fail, the master control program can, by changing the settings of the memory limits register and the requestor inhibits register, in effect eliminate the faulty module or unit from the system and permit processing to continue on a reconfigured system.
Masking	By appropriate setting of memory limits registers and access masks, the master control program can group memory modules into separate subsystems each of which is accessible only to certain central processor modules and input/output modules and in effect masked from others and each of which can contain the same range of addresses. Thus, a system can be divided into several separate subsystems, each with its own executive program and each dedicated to but a part of the total processing load of the system. In this way too, critical data and program code can be duplicated in memory in order to provide additional protection against system failures.
Interrupts generated by the memory module	The memory module generates interrupts under the following conditions:
Fail 1	If an irrecoverable error has occurred
Fail 2	If a one-bit error, which is correctable, has occurred
Fail S (a software interrupt)	If during a protected write operation an attempt is made to write data into a protected location

Table 3-3. Characteristics And Features Of The Memory Module (Cont)

Characteristics and Features	Description
Detection and correction of single-bit errors	All single-bit memory errors are detected and corrected; the fail register of the memory control module is loaded with information about the failure, and the requestor (central process module or input/output modules) is notified of the failure (a fail 2 interrupt is generated) and of the type of error that occurred.
Detection and reporting of two-bit errors	Two-bit errors are detected and reported but not corrected. Again, the fail register of the memory control module is loaded with information about the failure and the requestor is notified of the failure (a fail 1 interrupt is generated) and of the type of error that occurred.

MAINTENANCE DIAGNOSTIC UNIT

The maintenance diagnostic unit (see table 3-4) is a central console that in conjunction with a dedicated magnetic tape unit is used in off-line testing of the central processor module, the input/output module, the memory control module, and the disk file optimizer and in testing the cards of these components of the system. When by the use of on-line confidence and diagnostic programs a faulty module has been identified, the cause of the trouble is further traced first to the card level and finally to the circuit level by the use of module testing and card testing facilities of the maintenance diagnostic unit.

Table 3-4. Characteristics and Features of the Maintenance Diagnostic Unit

Characteristics and Features	Description
Number per system	At least 1, at most 2
Interface with modules that can be tested	Permanently, by means of unique cables between the maintenance diagnostic unit and each module that can be tested
Control and sampling of flip-flops of module under test	Because the modules that it tests have logical circuits dedicated to maintenance, the maintenance diagnostic unit is capable of controlling (setting, and resetting) and sampling all of the flip-flops of these modules.
Control of clock of module under test	The maintenance diagnostic unit controls the clock of the module under test; single clock pulses and trains of clock pulses can be used.
Off-line tracing of troubles to the card level	The strategy of testing modules on the maintenance diagnostic unit is to exercise a faulty module and clock period by clock period to compare the states of its flip-flops with a prerecorded norm. In this way a trouble is traced to a clock period and to a flip-flop and its associated logical circuits.

Table 3-4. Characteristics and Features of the Maintenance Diagnostic Unit (Cont)

Characteristics and Features	Description
Off-line tracing of troubles to the integrated circuit chip level	Similarly, the testing of faulty cards on the card tester of the maintenance diagnostic unit is carried out by providing input patterns to a card, sampling its outputs, and comparing them with predetermined norms.
Initiation of test routines	From the dedicated magnetic tape unit or manually from the panels of the maintenance diagnostic unit
Selectable test options	Among the test options are those of stopping on an error and of cycling.

OPERATOR'S CONSOLE

The operator's console is the center of communications between the operator and the B 7700 system; at least one operator's console is needed for each system. and two (one for each input/output module) are needed for full fail-soft capability. On the console is an operator's panel that contains five indicator-switches used in halting, clearing, or starting the system and in reading a loader deck, either from cards or from disk, into a system that is not under the control of the master control program. Hence, control of the cold-start, cool-start, and halt-load operations and of the loading of stand-alone test routines is exercised by the use of the switches on this console.

Besides the operator's panel, at least one operator's display terminal (consisting of a video output screen and an input keyboard) used in passing messages between the system and the operator must be located on the operator's console. A second display terminal is optional, but at least two display terminals (one on each input/output module) are needed for full fail-soft capability.

DISK FILE SUBSYSTEM

An extremely-high-speed, modular, random-access extension of main memory, the disk file subsystem of the B 7700 may include both head-per-track disk file memory modules (see table 3-5) and disk pack memory modules (see table 3-6) interfaced with the input/output module by the use of controls and exchanges.

Under the control of the disk file optimizer (see table 3-7), head-per-track disk file modules can be combined to form optimized-access memory banks capable of storing from 450 million to 8 billion eight-bit bytes of information per input/-output module and whose access time is as low as 2 to 6 milliseconds or 4 to 10 milliseconds.

As its name suggests, the disk file optimizer is used in optimizing the rate of transfer of data (through the input/output modules and disk file controls and exchanges) between main memory and disk file modules. For the transmission of control information, the optimizer is joined with the input/output module by means of a scan bus.

Without the disk file optimizer, head-per-track disk file modules can be combined into random-access memory banks with a storage capacity of from 15 million to 16 billion eight-bit bytes per input/output module and with an average access time of 23 or 40 milliseconds.

Disk pack memory modules can be combined into random-access memory banks with a capacity of from 121 million to many billions of eight-bit bytes of storage per input/output module and with an average access time of 30 milliseconds.

Table 3-5. Characteristics and Features of the Head-Per-Track Disk File Memory Module

Characteristics and Features	Description
Configuration of basic disk file memory module	1 disk file electronics unit (EU) and from 1 to 5 disk file storage units (SU)
Maximum storage capacity controlled by a disk file electronics unit	200 million eight-bit bytes
Storage capacity of a disk file storage unit	A minimum of 15 million eight-bit bytes and a maximum of 40 million bytes, depending on the model
Average access time	Under control of the disk file optimizer: as low as 2 to 6 milliseconds or 4 to 10 milliseconds; without the disk file optimizer: 23 milliseconds or 40 milliseconds

Table 3-6. Characteristics and Features of the Disk Pack Memory Module

Characteristics and Features	Description
Configuration of basic disk pack memory module	From 1 to 4 single-data-access or simultaneous-data-access dual drives
Maximum storage capacity of a basic disk pack memory module	968 million eight-bit bytes
Storage capacity of a dual drive	121 million or 242 million eight-bit bytes, depending on the model
Average access time	30 milliseconds
Average latency	12.5 milliseconds

Table 3-7. Characteristics and Features of the Disk File Optimizer

Characteristics and Features	Description
Number per input/output module	1 to 4
Number of disk file modules connected to a disk file optimizer	The disk file optimizer can be connected to as many as 40 disk file electronics units.
Accumulation, or queuing, of I/O control words for disk file operations	I/O control words for disk file operations are transferred by the input/output module to the disk file optimizer by the execution of a scan-out operation and are stored, or queued, in the stack of the disk file optimizer.
Determination of the control word most suited (the optimum control word) for execution	The control words stored in the stack are examined, and comparisons are made between the positions of the disk that would permit access to the desired addresses and the actual position of the disk. The comparison that shows the smallest difference between the desired position and the actual position results in the selection of the optimum control word.
Transferring the optimum control word to the input/output module	When a control word has been selected for execution, it is transferred to the input/output module, and the disk file input/output operation is carried out automatically by the input/output module.

DATA COMMUNICATIONS SUBSYSTEM

To add the capabilities for remote computing, remote inquiry, and on-line programming to the B 7700 system, a data communications subsystem is needed. This subsystem is made up of networks consisting of a data communications processor, adapter clusters, line adapters, and remote devices of virtually every kind; the heart of a data communications network is the data communications processor.

The data communications processor (see table 3-8) is a small, programmable, special-purpose computer devoted solely to sending and receiving data over a multitude of data communications lines.

Relative to the central processor module, the data communications processor operates asynchronously; once started, it operates independent of the central processor. Through the DCP-memory interface of the input/output module, the data communications processor has access to main memory; control of the data communications processor is exercised through the scan bus interface of the input/output module.

Table 3-8. Characteristics and Features of the Data Communications Processor

Characteristics and Features	Description
Number per B 7700 system	A maximum of 28
Number per input/output module	1 to 4, each of which operates independent of the others
Number of remote lines per data communications processor	As many as 256. From 1 to 16 adapter clusters can be attached to each data communications processor, and each adapter cluster contains up to 16 line adapters, each of which terminates 1 line: normally, each line handles a number of remote devices.
Clock rate	5 megahertz
Instruction set	A set of five groups of two-address and three-address instructions
Code translations	The data communications processor is capable of performing the following code translations: EBCDIC to USASCII EBCDIC to BCL USASCII to EBCDIC USASCII to internal BCL to EBCDIC Internal to USASCII Internal to BCL

POWER SUBSYSTEM

The power subsystem (see table 3-9) of the B 7700 system is characterized by the modularity of power regulators, the location of individual power supplies and regulators in each cabinet of the central B 7700 system, and the parallel operation and redundancy of power regulators.

Requirements for isolation of the B 7700 power subsystem from variations in the external power source are listed in the B 7700 Installation Planing Manual (form number 1060027).

Table 3-9. Characteristics and Features of the Power Subsystem

Characteristics and Features	Description
Regulation of current and voltage	The power subsystem provides overvoltage and undervoltage protection for direct current, overvoltage protection for alternating current, and current regulation.
Individual power supplies and regulators in the cabinet of each central component	AC power is brought to the cabinet of each central component, where it is converted to a 160-volt dc level—the base input to the power regulators (-12 volt, -6 volt, 4.8 volt, and -2 volt) in the cabinet.
Modularity of power regulators	The power regulators in each cabinet are modular and thus easily removed and replaced.
Redundancy of -2 volt and 4.8 volt power regulators	An optional fail-soft feature, redundant power regulators work in parallel with each other; if one should fail, the other assumes the full load. A redundant power kit is available for each of the following central components: the central processor module, the input/output module, the memory control module, and the maintenance diagnostic unit.

SECTION 4
SOFTWARE

To understand how the B 7700 system works, it is not enough to know its hardware alone, for the B 7700 is a true synthesis of both hardware and software. The software is not an afterthought. On the one hand, many functions conventionally handled by software are built into B 7700 hardware, and, on the other hand, the control and the balanced use of hardware resources of the system depend upon the software.

There are two notable facts about the B 7700 software:

1. It is all written in higher-level compiler languages to the exclusion of assembly languages and machine language. The principle of writing operating systems and compilers in higher-level languages has been successfully followed by Burroughs for over a decade. In this way, Burroughs has overcome the difficulties of man's communicating with the computer by providing languages that both he and the machine can understand. Not only has Burroughs closed the communications gap between man and machine, but it has done so efficiently by providing compiler-oriented hardware that speedily processes the results of compilations.

2. With no recompilation, it is possible to run on the B 7700 all applications programs, compilers, and utility programs that run on the B 6700. This is known as machine-code compatibility of software.

The principal items of B 7700 software are as follows:

1. The master control program (MCP): an extension of the unique Burroughs executive software operating system proven on the successful Burroughs 500-series systems and operational on the Burroughs B 6700 systems since 1969.

2. Compilers for higher-level, problem-oriented batch programming languages (COBOL, ALGOL, FORTRAN, PL/I, and ESPOL): the same compiler programs that have been running successfully on the Burroughs B 6700 since 1969.

3. Data communications software: the same assemblers for terminal computers, the same compilers, the same code generators, and the same message control systems that now run successfully on the B 6700.

4. Utility programs that perform general-purpose functions not performed by the operating system.

5. Confidence and diagnostic programs for both central components and peripheral devices.

6. Applications programs: users' programs including (without any modification of machine code) all programs that run successfully on the B 6700. Among these applications programs are data management systems, simulation languages, and mathematical programming systems.

MASTER CONTROL PROGRAM (MCP)

To make multiprogramming and multiprocessing a reality, a system must be capable of dynamically controlling its own resources and the scheduling of its jobs, and it must be capable of processing a number of jobs concurrently in less time than it takes to process the same jobs serially. The B 7700 system does these things by means of the master control program (MCP)--the software executive program that is the key to the effective, balanced use of the system. The B 7700 master control program is written in B 6700/B 7700 ESPOL, an extension of B 6700/B 7700 ALGOL.

By integrating users' object programs with the software-compatible B 7700 hardware system and with high-speed disk, the master control program optimizes the productivity of the B 7700 system. The master control program is, therefore, an essential part of the processing environment of the B 7700 system.

The integrative action of the master control program is achieved in the following three ways.

1. By its capability of coordinating the execution of many programs, or jobs, in the processor or processors.

2. By its capability of controlling both input and output so as to make optimal use of the relatively slow peripheral devices.

3. By its capability of taking executive action to meet virtually all processing conditions and to minimize the adverse effects of system degradation.

Thus, the overall rate and efficiency at which jobs can be processed under control of the master control program are increased, again, in three ways.

1. By increasing the speed of execution of individual users' programs. This increase in speed can be achieved by the use of a combination of several facilities, as follows:

   a. Parallel processing, or multiprocessing, with the introduction of more than one central processor module.

   b. Multiprogramming--the running of several jobs concurrently. The master control program maintains a list (ordered by priority) of jobs ready to run. When a running job must, for example, wait for an input or output operation to be completed, the master control program will start up the next job in the ready queue and restart the original job when the I/O operation has been completed and no jobs of higher priority are in the ready queue.

   c. Reentrant code--the means by which a single copy of a routine in main memory can be shared by several jobs.

   d. Tasking--the processing in a coordinated manner of whole families of tasks.

2. By increasing the speed of data handling. For this purpose, two facilities are provided in the utility section of the master control program.

   a. Loadcontrol. This enables card input to be transferred to disk or tape. The card image files so formed are assigned to different pseudo card readers which are then treated by the master control program as if they were real physical card readers.

   b. Printer and punch backup. This enables output to be placed on tape or disk and then printed or punched out at a later, more convenient time.

3. By increasing the ease of operating the machine. Simple English-like operator attention and error messages, automatic assignment of labeled files to jobs without operator intervention, a simplified control card language, and other features help increase throughput on the B 7700 system.

Except in two areas--in the interface between the master control program and unique B 7700 hardware and in the area of failsoft--the B 7700 master control program is identical with the B 6700 master control program that has been operating successfully since 1969. A new approach to the software control of a data processing system is the B 7700 multilevel master control program, which is expected to provide users with many advantages not offered by the more traditional master control program.

The multilevel master control program consists of a kernel, which has overall control of the B 7700 system and which is the sole interface between system software and system hardware, and one or more control programs, which run under control of the kernel and to which the kernel delegates many tasks of program supervision, system supervision, and input/output control.

There are two main reasons for adopting a multilevel approach to the software control of a data processing system. First, it is possible under control of the kernel to execute concurrently several control programs, each tailored to support a particular type of application, or job, be it batch work, testing of hardware modules, or time sharing. Each control program makes use of the strategies for resource allocation and scheduling most appropriate to a special kind of job and need not include irrelevant strategies. Thus, several control programs under the control of the kernel may share a hardware system, and each job running under control of a control program will benefit from the specialized facilities of the control program that controls it. Moreover, this arrangement permits the isolation of a user's production environment

from, for example, an environment in which faulty hardware is being tested or an environment in which experimental system software is being debugged.

Second, by making the operating system more modular, it becomes more understandable and more manageable, and thus, easier to write, maintain, and to extend. In fact, a user may write his own special control programs and still retain the use of the basic functions provided by the kernel and the standard control programs provided by Burroughs.

Although the general concepts presented in the following functional description of the B 7700 master control program apply as well to the multilevel version as to the single-level version, the details of implementation relate to the single-level master control program only.

System Startup

In order to place the MCP in control of the system, the MCP code file must be loaded onto disk, starting at disk address 0 of the *load* disk unit. In addition, the MCP information table and the disk directory must be present on disk. When these initial conditions have been satisfied, the Halt-Load operation is used to read the first 8192 words of the MCP code file into core memory, and the system begins to execute the MCP.

The functions of loading the MCP code file to disk from magnetic tape and of creating or revising the MCP information table and the disk directory are accomplished by the System Loader program. This program is in the form of a card deck containing the machine code instructions, followed by data cards that specify parameters for the initialization. Items that may be specified include the types and number of peripherals available and their configuration in the I/O subsystem, the size of disk areas to be used for the disk directory and for overlay, the disk units to be used for backup or reconstruction, the tables to be displayed on particular supervisory consoles, the tape from which the MCP is to be loaded, and various runtime system options.

Hardware Interrupts

The B 7700 processor hardware interrupt system is the primary interface between the MCP and the hardware. Hardware interrupts are generated automatically and under certain conditions by the B 7700 system and are handled by the MCP interrupt procedure. An interrupt is a means of diverting a processor from the job which it is doing if certain predetermined conditions occur. When a hardware interrupt has been processed by the MCP, the MCP will (if conditions then permit) reactivate the interrupted process.

When a processor is executing the interrupt handling procedure of the MCP, it is in *control state,* one of the two operating states of a processor. A processor can operate in either of two states, control state, used in executing the MCP, or normal state, used in executing user programs and certain MCP functions. In a multiprocessor system each processor handles its own interrupts; that is, all processors may be in control state at the same time.

Entry into control state occurs when the processor is started and as a result of certain interrupt conditions. In control state the processor can execute privileged instructions not available in normal state, and various classes of interrupts can be inhibited or allowed programmatically. Exit from control state into normal state occurs whenever the MCP initiates a normal state program or exits back to a normal state program following an interrupt. In the latter case, user program return may not be to the program in process when the interrupt occurred.

Normal state excludes use of privileged instructions required by the MCP, permits hardware detection of invalid operators, and enforces memory protect and security facilities. Exit from normal state occurs as a result of an interrupt condition or by a call to a control state program, for example, to execute I/O. Many MCP functions can be run in normal state. Interrupts to a normal state MCP function can be enabled.

Hardware interrupts may be classified as internal and external interrupts. For internal (syllable dependent and syllable independent) interrupts, each processor in a B 7700 system is pro-

vided with a private, internal network. Internal interrupts associated with a processor are fed directly into this network and are stacked local to the processor. External interrupts, on the other hand, may be serviced by any processor in a system.

Syllable dependent interrupts are detected by the processor operator logic. These include arithmetic error, *presence bit*, memory protect, and invalid operand interrupts. Except for arithmetic error interrupts, for which programmatic control may be supplied, and *presence bit* interrupts, interrupts of this group generally result in program termination.

Syllable independent (alarm) interrupt conditions are not normally anticipated by the processor operator logic. They serve to inform the processor of some detrimental change in environment and can result from hardware failure as well as programming errors. These interrupts include those for a faulty read from memory, an invalid address, and an invalid program instruction word; all result in termination of the process involved.

External interrupt conditions are similar to the alarm interrupts, in that they are not anticipated by the operator logic. However, they do not normally require immediate action and do not necessarily result in termination of the program. These include interchannel and IOM error interrupts.

When a hardware interrupt condition occurs, the interrupted processor enters the control state, marks the stack, and inserts three words in the top of the stack. The first entry is an indirect reference word which points to a register that contains a PCW which points to the MCP hardware interrupt procedure. The first entry is followed by two interrupt parameters, P1 and P2, which contain information indicating the nature of the interrupt condition. When the processor enters the MCP hardware interrupt procedure, it remains in control state in order to disable external interrupts. The processor execution state (control or normal) is determined by the control bit of the PCW. When the control bit is on the processor will execute a procedure in control state. Otherwise, it will execute in normal state.

Upon entry to the hardware interrupt procedure the parameter P1 is analyzed to determine the type of interrupt which occurred. For some interrupts, such as *presence bit* interrupts, P2 contains additional information to be used by the interrupt procedure. Then the appropriate action is initiated.

After the interrupt procedure has been entered, the program base register is pointing at the interrupt procedure, the program index register and the program syllable register are pointing at the interrupt procedure entry point, and the return control word for the interrupt procedure's exit is pointing back to the object program's code.

Storage Control

The MCP maintains records of storage availability through the use of memory links which are assigned within the areas they describe. Each type of memory link is linked to form a list which contains sufficient information for a single hardware operator to find the next memory and all succeeding links. Memory areas are classified as in-use or available according to their current state.

Specifically, in-use memory links contain the stack number of the requesting process, the length of the in-use area, an availability bit set in the off position, a code indicating the usage of the area, links to the last previously allocated and next in-use areas, and so on. Available memory links contain the length of the area, an availability bit set in the on position, links to the next and last available areas, and so forth.

The MCP performs dynamic storage allocation by use of the environment control routine for all system storage media: main memory, magnetic disk, and system library magnetic tape. As a result of considering the different system storage media as a hierarchy of memory, the MCP controls allocation and deallocation of all system memory, regardless of the type.

Memory protection is provided for by a combination of hardware and software devices. One of the hardware features is automatic detection of an attempt by a program to index beyond its designated data area. Another is the use of one

of the control bits in each word as a memory protect bit to prevent user programs from writing into words of memory which have the protect bit set. (The protect bit is set by the software.) Any attempt to perform such a write operation is inhibited, and an interrupt is generated, which results in termination of the program. Thus a user program cannot change program segments, data descriptors, or any program words or MCP tables during execution.

Control of Jobs

The MCP maintains control of jobs by the use of stacks, descriptors, and tables of system and process status.

One stack is associated with each job in the system. The stack, a contiguous area of memory, is assigned to a job to provide storage for basic program and data references. It also provides for temporary storage of data and job history. When a job is activated on a processor, two high-speed top-of-stack processor locations are linked to the job's stack memory area. This linkage is established by the stack-pointer register (S register), which contains the address of the last word placed in the stack. In addition, the 32-word stack buffer, an area of processor local IC memory, is made available to contain the top portion of the active stack, in order to provide quick access for stack manipulation by the processor.

Data are brought into and out of the stack through the top-of-stack locations according to the last-in, first-out principle. Total capacity of the top-of-stack locations is two operands. Loading a third operand into the top-of-stack locations causes the first operand to be pushed from the top-of-stack registers into the stack buffer. The stack-pointer register (S) is incremented by one as each word is placed into the stack or stack buffer and is decremented by one as each word is withdrawn and placed in the top-of-stack registers. As a result, the S register continually points to the last word placed into the job's stack.

A job's stack is bound, for memory protection, by two registers, the base-of-stack register (BOSR) and the limit-of-stack register (LOSR). The contents of BOSR define the base of the stack, and the LOSR defines the upper limit of the stack. The job is interrupted if the S register is set to the value contained in either LOSR or BOSR.

Descriptors are words used to locate data and program areas in memory and to describe these areas for control purposes. Descriptors are the only words containing absolute addresses which can be used by a user's program; however, the user's program cannot alter them. Descriptors are divided into three categories: data, string, and segment.

Data descriptors are used for referring to data areas, including input/output buffer areas. The data descriptor defines an area of memory starting at the base address contained in the descriptor. The size of the memory area in number of words is contained in the length field of the descriptor. Data descriptors may directly reference any memory word address.

String descriptors refer to data areas organized as 4-, 6-, or 8-bit characters: The descriptor defines an area of memory starting at the base address contained in the descriptor. The size of the memory area is defined by the length field.

Segment descriptors are used to locate program segments. These descriptors contain either the main memory or disk file address of a particular segment. All programs are entered and exited through the segment descriptors common in the segment dictionary stack; all references to those descriptors are relative. Entrance to or removal of any given program segment from memory is achieved by changing the *presence bit* in that segment descriptor. No stack search of any kind is required.

The MCP also maintains tables that summarize system and process status. The mix table contains the priority, status (scheduled, active, or suspended), and mix index of each job that has been entered in the system. The peripheral unit table has an entry for each peripheral unit in the system. Each entry contains the status of the corresponding unit and the file associated with that unit.

SCHEDULING AND INITIATION OF JOBS

The sequence of jobs to be run and the optimal program mix considering the priority ratings

and system requirements of each object program and considering the present system configuration are determined by the scheduling routine of the MCP. The MCP incorporates a dynamic scheduling algorithm, that is, one which reschedules the job sequence whenever a higher priority job is introduced into the system. Job priority may be programmer-defined by use of the PRIORITY statement. If no priority is specified by the programmer, a default value of one-half the maximum allowable priority is assigned by the MCP.

The calculation of the priorities is performed in a well-isolated section of the MCP. Thus, the user may easily tailor the Burroughs-supplied priority algorithms to his specific requirements.

As each job is read from the system input unit (card reader or pseudo card reader, i.e., magnetic tape or disk), the control card interpreting procedure, CONTROLCARD, makes an entry into the sheet queue to schedule each batch-mode process. The sheet queue is a linked list of processes which await execution. Each entry in the sheet queue is a partially built process stack. The information contained in this stack includes the estimated amount of main memory required by the process, priority, time of entry into the schedule, size and location of code segments, working storage stack size, and size and location of the process stack information. After CONTROLCARD completes its tasks and if sufficient system resources are free, the entry is moved from the sheet queue to a queue called the ready queue.

When sufficient system resources exist to allow another job into the mix, a procedure called RUN is started. RUN makes the segment dictionary for the job present in main memory and transfers control to the job.

Real-time and time-sharing applications entering the system by way of the data communication facilities merely become additions to the multiprogramming mix.

EXECUTION OF JOBS

As soon as control is transferred to a new job, an interrupt may occur because the outer block code segment is generally not present in main memory. This interrupt is handled by the PRESENCEBIT procedure of the MCP. PRESENCEBIT is entered and the following actions occur in order to bring the segment into memory: (1) PRESENCEBIT calls GETSPACE to allocate an area in main memory for the code segment; (2) after an area is allocated, PRESENCEBIT calls DISKIO, the disk input/output procedure, and waits for notification that the segment has been read in; and DISKIO links the request into the I/O queue. Upon completion of the disk I/O, PRESENCEBIT is notified that the segment is now available. PRESENCEBIT marks the segment descriptor present and exits back to the job at the point of interruption, and the job continues to run.

A program residing in memory occupies separately allocated areas; that is, each part of the program may reside anywhere in memory. The actual address is determined by the MCP. Also, the various parts are not necessarily assigned to contiguous memory areas. Registers within the processor and descriptors in the stack indicate the bases of the various areas during the execution of a program.

The separately allocated areas of a program are: (1) the program segment-sequences of instructions performed by the processor in executing the program; (2) the segment dictionary, a table containing one word for each program segment; this word tells whether the program segment is in main memory or on the disk, and gives its corresponding main memory or disk address; (3) the stack, which contains all the variables associated with the program, including control words that indicate the dynamic status of the job as it is being executed; (4) data areas used by the program, which are referenced by data descriptors or string descriptors in the program's stack; and (5) the MCP stacks and segment dictionary, which contain variables pertinent to the MCP and the MCP segment dictionary entries.

As a job runs, additional segments of program code and data will be needed. The job stack contains the storage locations for simple variables and array data descriptors, but program code segments and array rows are assigned their own areas of memory. This assignment of separate memory areas for code segments and array rows allows segments and

data to be absent from main memory until they are actually needed. Thus, in the B 7700 system a reference to data or code through a data descriptor or a segment descriptor causes the processor to check the presence bit in the descriptor.

If the presence bit is off, an interrupt occurs which transfers control to PRESENCEBIT. The nonpresent descriptor is passed as a parameter. PRESENCEBIT reads the address field of the descriptor and calls the GETSPACE procedure to allocate an area in main memory for the code segment. Parameters are supplied to GETSPACE so that an adequate-sized contiguous area of memory may be reserved for a particular stack. After GETSPACE satisfies the request for core space, it returns the memory address of the area it has allocated, and PRESENCEBIT causes the information to be read from disk into memory. When the disk read is finished, PRESENCEBIT stores the memory address of the information into the address field of the descriptor, turns the presence bit on, and updates the descriptor in the process stack. PRESENCEBIT then returns control to the interrupted process, and the information is accessed again by the process. Now the information is present in memory; the information is obtained and the process execution continues in the normal manner.

The storage required for the referenced data or code may be allocated at the front or rear of an adequate-sized area and marked as overlayable or nonoverlayable. When an in-use area is allocated, it is linked to the previously allocated in-use area by the left-off link and pointer fields in the memory links. These fields constitute the left-off list. A reference word pointing to the oldest entry in the left-off list allows the chronological history of in-use memory areas to be determined.

When there is insufficient available memory to satisfy a particular request, the overlay mechanism is invoked. The left-off list is searched, starting at the overlayable area that has been allocated for the longest period of time. If this area, combined with any adjacent available area is adequate to satisfy the request, it is overlaid. Otherwise, allocated areas with lower standing addresses are considered. If the request is satisfied and the area found is larger than the required size, the unused portion is made available by linking it to the available list. If the request is not satisfied, the next oldest overlayable area is obtained and the left-off list is searched as described above. This process is repeated until the left-off list has been exhausted. If the request cannot be satisfied, a no memory condition exists.

SOFTWARE INTERRUPTS AND EVENTS

Software interrupts are programmatically defined for use by the MCP and object program processes. Software interrupts allow processes to communicate with each other and with the MCP.

Software interrupts allow a process to stop running (thereby releasing the processor) until a specified event occurs, or continue running and be interrupted if the event does occur. A software interrupt occurs when a process is interrupted by the direct action of some other process. A process can be interrupted if it has an interrupt declaration (statement) within its scope.

A process may invoke the occurrence of an event by means of the CAUSE statement. The MCP scans the event interrupt queue to determine if the interrupt has been enabled. If the interrupt is not enabled and the event is caused, no action is taken by the MCP on that process, and it looks at the next process stack in the queue.

If interrupts are enabled in the next stack, the MCP makes an entry in the software interrupt queue. This queue is ordered by stack number. If the stack is active, that is, if another processor is working in the stack, the MCP will interrupt that processor with an interchannel interrupt. Next, the MCP forces a transfer of control to the statement related to the interrupt declaration. Upon completion of this statement, the process will return to its previous point of control unless a transfer of control is specified in the interrupt statement. In this case the process will not return the point of control before the interrupt but will transfer control as specified in the interrupt statement.

As the MCP scans the event interrupt queue finding enabled interrupts in inactive stacks,

it makes an entry in the software interrupt queue, doing nothing with that stack until it becomes active. Immediately after making the stack active, the MCP checks the software interrupt queue to see if there is an interrupt pointing to that stack. If an interrupt is found, the MCP forces a transfer of control to the statement referred to by the interrupt declaration. Upon completion of the statement, control is transferred as described above.

TERMINATION OF JOBS

When the execution of a job is terminated, the following actions occur: (1) any outstanding I/O requests are completed, if possible, and any open files are closed, the units released, and the buffer areas are returned to the available memory table; (2) all overlayable disk areas allocated to the job are returned to the available disk table; (3) all job object code and data array areas of main memory are returned to the available memory table; (4) an end-of-job entry is made in the system log for the job; (5) the jobs stack is linked into the terminate queue.

Input/Output Control

All input/output operations on the B 7700 system are performed by the MCP. The MCP automatically assigns peripheral units to symbolic files whenever possible in order to minimize the amount of operator attention needed by each job. When an input file is requested by a job, the MCP searches its tables for the appropriate peripheral unit which contains the file requested. If the file name specified by the job is found on a particular unit, that unit is marked in use and assigned to the job. Output files requested by a job are automatically assigned by the MCP if a suitable unit exists for the file. In the case of disk files, a disk file directory entry is made and the needed disk space is allocated for the file.

FILE HANDLING

In order for the MCP to associate peripheral units with symbolic files, the compilers that run on the B 7700 system must furnish the following information about files to the MCP: the symbolic file name, the actual file name (file title), the peripheral type (disk, magnetic tape, card, paper tape, etc.), the access type (serial or random), the file mode (alpha, binary, etc.), the buffer size, the number of buffers, and the logical record size. The actual file name is the file title which is associated with the unit that contains the file or the title in the disk file header. The actual file name will be identical with the symbolic file name unless otherwise specified by label equation control statements.

In order to allow dynamic specification of actual file names for a file, three tables are necessary: a process parameter block, a label equation block, and a file information block. The process parameter block is created by CONTROLCARD for all files in a job. It contains the symbolic file name and any compilations or execution time label equation information specified for this process. The label equation block and the file information block are specified for this process. The label equation block and the file information block are created by the compiler and maintained by I/O functions for each file in a process. The label equation block contains the current label equation and other file attribute information for a particular file, including any programmatic specification of file attributes. The file information block contains frequently used information concerning the file, such as the type of access required, type of unit assigned, physical unit being used, and attributes which depend upon the type of unit assigned. Incorporation of the file attributes in the file information block and label equation block allows modification of file specifications such as buffer size and blocking factors, at program execution time, without recompilation of the program.

TRANSFER OF DATA

Object program I/O operations on the B 7700 system involve the automatic transfer of logical records between a file and a job. A logical record consists of the information the job references with one Read or Write statement. The size of a logical record does not necessarily coincide with the size of the physical record or block accessed by the hardware I/O operations. When a physical record contains more than one logical record, the file is referred to as a blocked file.

When a file is accessed by a job, a physical record is written from or read to a memory area known as a buffer area for the file. If the file is blocked, the MCP maintains a record pointer into the buffer. This pointer is used by the process to access the current logical record. If the next record is not already present in a buffer, then the MCP automatically performs the required I/O operation.

Multiple buffers may be used to effectively increase throughput for jobs that require groups of physical records at one time. Since the MCP performs all object program I/O action, a job with multiple buffers allocated for a file allows the MCP to perform I/O operations independent of the status of the job. The determination of the number of buffers required for efficient execution of a job depends on the type of files being used, the particular hardware configuration being used, the processing characteristics of the job, the memory requirements of the job, and the mix of jobs which are typically multiprocessing. The MCP attempts to keep all input buffers full and all output buffers empty for each job, regardless of status, thereby minimizing the time that a process is suspended waiting for an I/O operation to be completed.

DATA COMMUNICATIONS

The MCP provides extensive data communication facilities, including time-sharing, remote computing, and remote inquiring. No terminal device interfaces directly with the control system. Instead, the necessary linkage is provided through a communications line, adapter and multiplexing devices, and the data communications processor.

Those aspects of the data communications system that are oriented toward applications are handled by the message control system (MCS) program. These aspects include remote file maintenance and job control. In addition, the message control system coordinates interprogram communications and provides message-switching capabilities. A single remote station may communicate with other remote stations or more than one object job.

Communication Between User and MCP

Communication between the user of the B 7700 system and the MCP is accomplished with a combination of display units, control units (display units with associated keyboards), control cards, and a comprehensive system log.

The status of the system and of the jobs in progress is presented on the display units. Specific questions requiring short answers may be entered by use of the keyboard. These questions and answers are displayed as they occur. Also, by entering the appropriate keyboard messages, various tables may be called for display. These tables include the job mix, peripheral unit, label, and disk director tables.

The operator communicates directly with the MCP by the use of input and output messages entered and received at the control units. The input messages include any control statement allowed on a control card, messages to enter jobs into the mix and to eliminate jobs from the mix, and messages to reactivate jobs that have been suspended. Output messages pertain to various functional areas of the MCP, to users' programs, and to system hardware modules.

A user submits a job to the system as a set of control cards and a source language deck. Alternatively, he may submit only a set of control cards or enter control statements at the input keyboard if he has previously stored on disk the programs that he wishes to run and has entered their names in the disk directory following an error-free compilation.

For a job requiring compilation, the first control card must be a compile statement, which specifies the compiler to be used and the type of compilation to be made. There are three forms: compile and execute, compile for the library, and compile for syntax check. The other types of control cards may be used for all jobs whether they do or do not require compilation. These include an execute statement, process time statement, priority statement, core requirement statement, I/O time statement, and I/O unit statements which associate file labels with particular I/O units.

The MCP maintains on disk a system log, which is a record of all activities on the system. Besides system error and maintenance statistics,

the log makes available to the user such data as the processing time for each job, the time at which each job was started, its elapsed running time, and its actual processor time.

COMPILERS

The B 7700 system rapidly generates efficient object code from COBOL, ALGOL, FORTRAN, PL/I, and ESPOL source language programs. The exclusive use of these program-oriented languages makes original programming simpler and less costly, and, because changes to programs are made only at the source level, the same is true of program maintenance.

Besides simplicity and economy of programming and program maintenance, Burroughs compiler-language programming also offers the user the benefits of automatic program segmentation (and thus of programmer-independent virtual memory), invariable object code (and thus of reentrant programs), and of binding (the intermixing of ALGOL, COBOL, and FORTRAN subprograms).

Segmentation (Virtual Memory)

At the time of compilation, the Burroughs compilers logically and automatically segment all programs and catalog these segments in a segment dictionary. The segments themselves are stored on disk files. The executive software refers to this dictionary at the time a program is being executed and calls segments into main memory from disk file storage automatically as they are needed. Thus, programmers can write without regard to the size of main memory, to segment locations, or to the nature of the multiprogramming mix.

Invariable Object Code (Reentrant Programs)

Because object code is invariable, a single copy of a program segment in main memory may be shared by several users. Hence, compilers, data communications programs, data management routines, and, in fact, all users programs can be shared. In this way, more efficient use is made of core memory, a limited and costly resource, and system productivity increases.

Binding

Because the machine code generated by all Burroughs compilers has a uniform structure, the binding, or intermixing, of ALGOL, COBOL, and FORTRAN subprograms is made possible.

DATA COMMUNICATIONS SOFTWARE

The standard data communications software for the B 7700 system is identical with that for the B 6700. This software includes an assembler for the DC 1000 terminal computer: compilers such as DCALGOL, and BASIC; code generators such as the Network Definition Language (NDL) and the DCP Program Generator (DCPPROGEN) that are used together to produce code for the data communications processor; and message control systems such as Message Control System II (MCSII), Remote Job Entry (RJE), and Command and Edit (CANDE) Language.

The principal functions of the programs generated for the data communications processors are to equip the data communications processor to poll communications lines, to answer and terminate calls, to format messages, to translate and manipulate data, and to control line discipline functions. Hence, the central B 7700 system is relieved of most data communications overhead, and transaction processing routines can be executed in parallel and asynchronously.

To generate code for the data communications processor, the user need only describe the characteristics of his terminals and the line disciplines associated with them. These specifications are punched into cards, and the NDL and DCPROGEN code generators produce the code needed to drive the data communications processor and to establish the interface between the processor and its terminals. As terminals are added to a system or removed from it or as line networks change, new programs can be generated for the data communications processor without affecting applications programs.

The message control system operates in conjunction with the master control program; the main functions of a message control system are to handle all message queuing to and from the data communications processor and to control communications between data communications

programs and users programs that run on the central system under the control of the master control program.

The message control system may accept input from a variable number of remote stations through a program communication file in core; may react to the activation and deactivation of stations (for example, it may allow a dial-line, ring indicator to be answered and send the first message); may perform command and edit functions as required by the input data stream; may perform command functions upon occurrence of exception conditions, including controlling subsequent I/O of other devices attached to the line; initiate object jobs as independent processes and handle certain command requests from object programs such as file-open attachment of files to given lines; and may maintain file security restrictions and check remote user security.

UTILITY PROGRAMS

Most of the utility functions of the B 7700 system are built into the operating system. A few, however, such as the memory dump analyzer and the ALGOL and COBOL filters, are furnished separately.

CONFIDENCE AND DIAGNOSTIC PROGRAMS

Both on-line and stand-alone confidence and diagnostic programs for both the central components of the B 7700 system and peripheral devices are provided as standard B 7700 system software. In addition, test tapes used with the maintenance diagnostic unit are provided for the off-line testing of the central processor module, the input/output module, the memory control module, and the disk file optimizer.

APPLICATIONS PROGRAMS

Without recompilation, all users programs that run on the B 6700 system will run on the B 7700.

BURROUGHS CORPORATION
DATA PROCESSING PUBLICATIONS
REMARKS FORM

TITLE: **B 7700 REFERENCE LIBRARY CHARACTERISTICS MANUAL**

FORM: **1059979**
DATE: **1-73**

CHECK TYPE OF SUGGESTION:

☐ ADDITION ☐ DELETION ☐ REVISION ☐ ERROR

GENERAL COMMENTS AND/OR SUGGESTIONS FOR IMPROVEMENT OF PUBLICATION:

FROM: NAME _____ DATE _____
TITLE _____
COMPANY _____
ADDRESS _____
_____

tear along dotted line

STAPLE

FOLD DOWN SECOND FOLD DOWN

Postage Will Be Paid by Addressee

No Postage Stamp Necessary If Mailed in the United States

BUSINESS REPLY MAIL

First Class Permit No. 46, Wayne, Pennsylvania

Burroughs Corporation

200 West Lancaster Avenue
Wayne, Pennsylvania 19087

Attn: Sales Technical Services

FOLD UP FIRST FOLD UP

1059979 1-73 Printed in U.S. America