Burroughs

B5500

Information
Processing Systems

REFERENCE MANUAL

Burroughs

B 5500

INFORMATION PROCESSING SYSTEMS

REFERENCE MANUAL

Business Machines Group

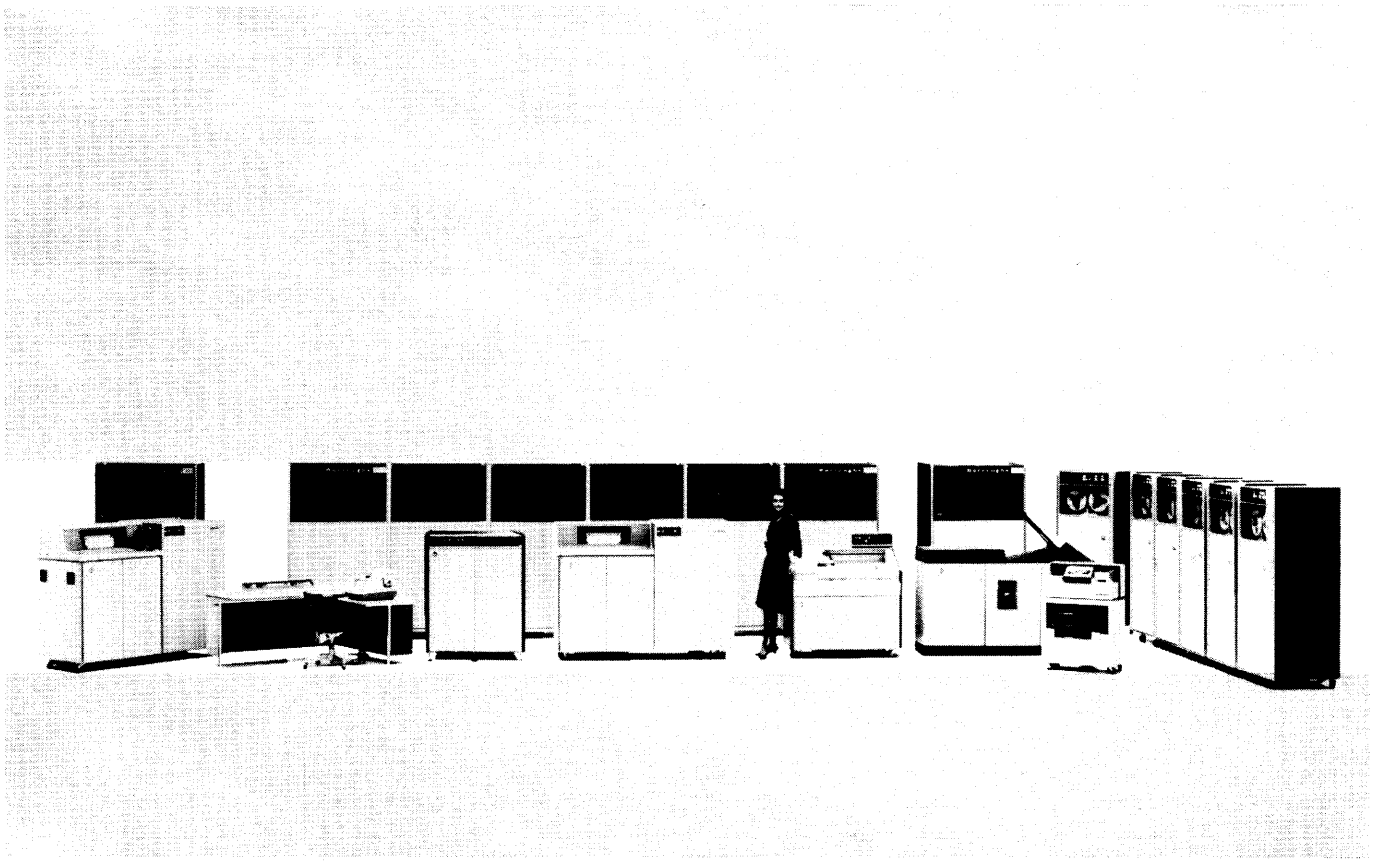
Sales Technical Services

Systems Documentation

Burroughs Corporation

Detroit, Michigan 48232

Contains material from 200-21014 and B 5000.55B, both



Burroughs B 5500 Information Processing System

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT.)

LIST OF TABLES

INTRODUCTION

This reference manual describes the hardware characteristics of the Burroughs B 5500 Information Processing System by presenting detailed information concerning the functional operation of the entire system. The B 5500 is a large-scale, high-speed, solid-state computer which represents a departure from the conventional computer system concept. It is a problem language oriented system rather than the conventional hardware oriented system. Because of the design concept of the B 5500, there exists a strong interdependence between the hardware and the Master Control Program which directs the system. The material presented herein pertains only to the hardware considerations, whereas the Master Control Program is discussed under separate cover.

SECTION 1
SYSTEMS DESCRIPTION

GENERAL

1-1. Without delving into such elements of computer design as circuitry and machine logic, this manual explains how the B 5500 achieves its flexibility and efficiency through a comprehensive systems approach to problem solving. The B 5500 is designed as a complete system, combining components and built-in aids, to bring simplified programing, ease of operation, and complete freedom of system expansion to the user. The B 5500 has compiler oriented machine language and logic which accept the common languages; ALGOL, FORTRAN, and COBOL. The machine language of the B 5500 is designed specifically for these problem languages, reducing compilation time and eliminating object program redundancies. The B 5500 automatically handles memory assignments, input/output unit assignments, segmentation of programs, and subroutine linkages, eliminating many arduous programing tasks and reducing the likelihood of error. Programs may be corrected at the source language level and are simplified by integral debugging aids.

MASTER CONTROL PROGRAM

1-2. The Master Control Program provides the over-all coordination and control of processing that is so important to total production through the maximum use of all B 5500 components. Operator intervention is nearly eliminated because complete management of the system is assumed by the Master Control Program, a comprehensive operating system that provides simultaneous input/output operations and multiprocessing. By controlling the sequence of processing, initiating all input/output operations and providing automatic handling procedures to meet virtually all processing conditions, the Master Control Program can obtain maximum use of the system components at all times. Because so many functions are performed under this centralized control, changes in schedule, system configuration, and program sizes can be readily accommodated; thus achieving greater over-all production and efficiency.

1-3. The configuration for a B 5500 system may vary considerably, depending on specific applications and expected workload. Both the possible components and the maximum configuration are listed in table 1-1.

DESCRIPTION AND FUNCTION OF MAJOR UNITS

1-4. The major units of a maximum B 5500 configuration include central control unit, processors A and B, memory modules 0 to 7, input/output control units 0 to 3, and display and distribution unit. These major units are contained in a major system cabinet and must be arranged in a specific order (figure 1-1).

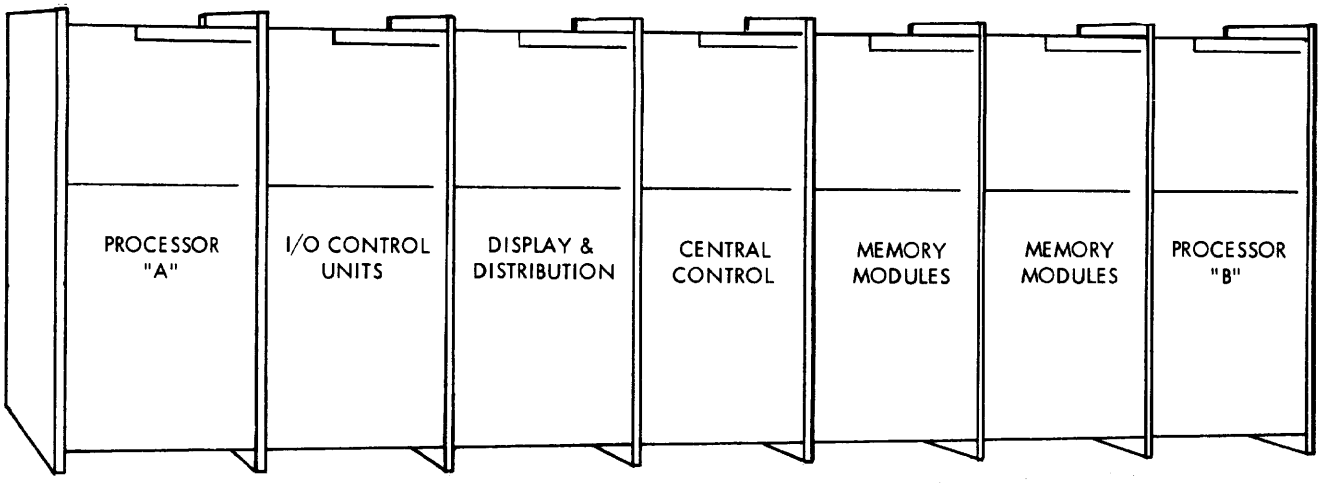


Figure 1-1. Major System Cabinet Configuration

TABLE 1-1

B 5500 Configuration Chart

Unit Number	Description	Max.	Ability of One Unit	Notes
B 5280	Processor A	1		
B 5281	Processor B	1		
B 460	Memory Module	8	4096 words 6 μs cycle time	B 460 and B 461 modules cannot be mixed on any one system.
B 461	Memory Module	8	4096 words 4 μs cycle time	
B 430	Storage Drum	2	32,768 words	
B 5283	Input/Output Channels	4		16 Tape units maximum B 422 and B 423 units cannot be mixed. B 422 and B 424 can be mixed if the B 422 operates at 120 inches per second.
B 422	Magnetic Tape Unit	16	24-66KC	
B 423	Magnetic Tape Unit	16	24KC	
B 424	Magnetic Tape Unit	16	66KC	
B 425*	Magnetic Tape Unit	16	18-50-72KC	
B 320	Line Printer	2	475 LPM, 120 chr.	Two Line Printers Max.
B 321	Line Printer	2	700 LPM, 120 chr.	
B 325	Line Printer	2	700 LPM, 132 chr.	
B 328	Line Printer	2	1040 LPM, 120 chr.	
B 329	Line Printer	2	1040 LPM, 132 chr.	
B 122	Card Reader	2	200 CPM	Two Card Readers Max.
B 123	Card Reader	2	475 CPM	

* Requires Feature 872 Extended Magnetic Tape Capability.

TABLE 1-1 (Cont)

B 5500 Configuration Chart

Unit Number	Description	Max.	Ability of One Unit	Notes
B 124	Card Reader	2	800 CPM	Two Card Readers Max.
B 129	Card Reader	2	1400 CPM	
B 303	Card Punch	1	100 CPM	One Card Punch Max.
B 304	Card Punch	1	300 CPM	
B 141	Paper Tape Reader*	2		Three Paper Tape Units Maximum
B 341	Paper Tape Punch*	2		
B 450	Disk/File/Data Communications Basic Control	2		If system has two B 5470's and one B 5480, then two B 450's are are needed.
B 452	Disk File/Data Transmission Basic Control	8	Can hold 2 B 487 DTTU's	Required for B 487 DTTU's.
B 5470	Disk File Control (DFC) Unit	2		
B 5480	Data Communication Control (DCC) Unit	1	Can service up to 15 Terminal Units in any combination	B 487 DTTU's not allowed
B 249	Data Transmission Control Unit	1	Can service up to 15 Terminal Units	All types of Terminal Units
B 451	Disk File Expanded Control	4	Service up to 5 5 DFE Units	Two per each B 5470
B 471	Disk File Electronics (DFE) Unit	20		10 DFE Units Max. per each B5470 DFC Unit
B 475	Disk File Storage (DFS) Module	100	9,600,000 BCL Characters	5 DFS Modules Max. per each B 471 DFE Unit
B 481	Teletype (TTY) Terminal Unit	15	Can service up to 399 teletype units	15 Terminal Units Max.
B 483	Typewriter (TYP) Terminal Unit	15	Can service up to 8 B 493 TIS stations simultaneously	
B 484	TWX (TWX) Terminal Unit	15	Can service up to 8 AT&T Data-sets simultaneously	

* Feature 920 is available for code translation.

TABLE 1-1 (Cont)

B 5500 Configuration Chart

Unit Number	Description	Max.	Ability of One Unit	Notes
B 487	Data Transmission Terminal Unit (DTTU)	15	Can service up to 16 adaptors	15 Terminal Units Max.
980	TYP/TWX Adaptor		1-TIS or 1-TWX network	
981	TTY Adaptor		1-TTY network	
982	Data Speed II Adaptor			
983	801 Auto Call Unit			
984	U1004 Adaptor		1-1004 Station	
985	IBM 1050 Adaptor		25-1050 Stations	
B 493	Typewriter Inquiry Station (TIS)	120		Only 8 TIS per each B 483 TYP Unit

Central Control Unit

1-5. The heart of every B 5500 system is the central control unit which is comprised of three sections: memory exchange, input/output exchange, and system control. A description of the basic function of each section follows.

1-6. MEMORY EXCHANGE. The memory exchange section provides access routes for requesting units (processors and input/output control units) to core memory modules. In order to connect the proper memory module to the requesting unit, the desired address is examined to determine which memory module is being addressed, then the proper routing is established. The design of the memory exchange allows the simultaneous access of different memory modules by multiple requesting units. (One requesting unit per one memory module.) After the route is established, all bits of a word are transferred in parallel to or from the requesting unit.

1-7. INPUT/OUTPUT EXCHANGE. The input/output exchange section provides the connection between any input/output device and any input/output control unit. Like the memory exchange, the input/output exchange permits simultaneous parallel transfer of information, thus providing multiple input/output operations. The routine established enables the flow of both information and control logic to each peripheral unit.

1-8. SYSTEM CONTROL. The system control section contains the master clock, a one-sixtieth of a second timer, and an interrupt system. The one-sixtieth second timer is included to provide a means of recording job run time. The interrupt system is the major portion of the system control section and will initiate Master Control Program routines to handle "exception conditions" when they are detected.

Processors A and B

1-9. The processors of the B 5500 system are designed for maximum use of problem oriented languages. A B 5500 system may have two processors for simultaneous operation. Each processor is independent of the other in normal operation and may operate in one of two modes, word mode or character mode. Only one processor in each system can operate in either of two states, normal state or control state; the other processor must operate in normal state only.

1-10. WORD MODE. The usual mode of operation is word mode. In word mode, operators are primarily concerned with one or two complete words. Each word contains 48 bits. Registers are arranged (for display) in groups of three bits (an octade), coded in octal notation. Word mode operation uses a full parallel adder, with automatic floating point logic.

1-11. CHARACTER MODE. In character mode, a word consists of eight characters of six bits each. Operators primarily work with one character or one to six bits of a character. Character mode operation uses a serial adder and does not have automatic floating point logic. Numbers are represented in binary coded decimal (BCD).

1-12. NORMAL STATE. Instructions in this state are concerned with the conventional aspects of computation (adding, subtracting, information transfer, etc.). Any interrupt (detection of an "exceptional condition") that occurs while processing in normal state will suspend the normal state operation and transfer operation to control state in processor one.

1-13. CONTROL STATE. All operations that can be performed in normal state can also be performed in control state. Additional control operations are available while in control state. Unlike normal state, interrupts occurring in control state will not suspend processing. Most MCP (Master Control Program) routines are written to operate in the control state because of the additional control operators that are available.

1-14. OPERATORS (INSTRUCTIONS). In both normal and control state there are four operators in a word. The act of obtaining the next operator can occur simultaneously with the performance of other operations. This simultaneity of operations results in minimal time loss due to the lack of an instruction.

1-15. CLOCK. A one megacycle clock rate is used throughout the entire system. The clock rate plus the extremely efficient logic results in an average add time for two words of about 3 μ s. and an average multiply time for two words of about 30 μ s., both in floating-point.

Core Memory Module

1-16. A B 5500 system can have up to eight memory modules. (All eight must be either model B 460 or model B 461.) Each core memory module operates independently of any other module. Information stored in a core memory module is accessed through the memory exchange section of the central control unit. As mentioned before, memory exchange permits parallel accesses to multiple modules (allows more than one core memory module to be accessed at the same time). Each module is a high speed (cycle time of either 6 μ s. or 4 μ s.), coincident current magnetic core memory. Every core memory module has its own address register and information buffer register. Each module contains 4096 words resulting in a maximum core memory size (for one system) of 32,768 words. One word of core memory consists of 48 bits plus one bit reserved for parity check. For the B 460 module and the B 461 module, memory cycle times are 6 μ s. and 4 μ s. respectively. When a word from either memory module B 460 or B 461 is read, the information is available for use in either 3 μ s. or 2 μ s. respectively.

Input/Output Control Unit

1-17. The I/O (input/output) control unit(s) control all peripheral I/O devices in the B 5500 system. A B 5500 system may have up to four I/O control units. Each I/O control unit operates independently of the other, thus allowing a maximum of four simultaneous I/O information transfer operations. The I/O control unit contains a one word buffer. This buffer is used to hold information being translated or transferred to or from an I/O device. The I/O control unit will translate each word of information from the internal system code to the output code, or vice versa. The I/O control unit communicates with a core memory module through the memory exchange section of the central control unit and communicates with any I/O device through the I/O exchange section of the central control unit. The use of these exchanges allows a maximum number of simultaneous I/O information transfer operations equal to the number of I/O control units.

1-18. I/O CHANNEL. One I/O control unit plus the established routing in the central control unit (using memory exchange and I/O exchange) is referred to as an I/O channel. A B 5500 system can have a maximum of four I/O channels.

Display and Distribution Unit

1-19. The display and distribution unit contains an indicator panel for each processor,

for each I/O control unit, and for central control unit. The panel displays the contents of most registers, flip flops, and certain memory areas. Using the proper request technique, any area of memory can be seen as it actually appears in memory. Some portions of memory can be altered during processing via an indicator panel in the display and distribution unit. Besides the indicator panels, this unit contains control indicators for the power supply and a Power-Off switch. The display and distribution unit is the distribution center for the peripheral devices and for the power from the power supply.

INFORMATION FLOW BETWEEN UNITS

1-20. The three phases of information flow are input, processing, and output (figure 1-2). Core memory is a fast, random access, temporary storage device for information to be used during processing. During the input phase, information is brought into core memory. During the processing phase, the information in core memory is worked upon. During the output phase, the information is taken out of core memory either as output or as information to be re-entered into the computer at a later time.

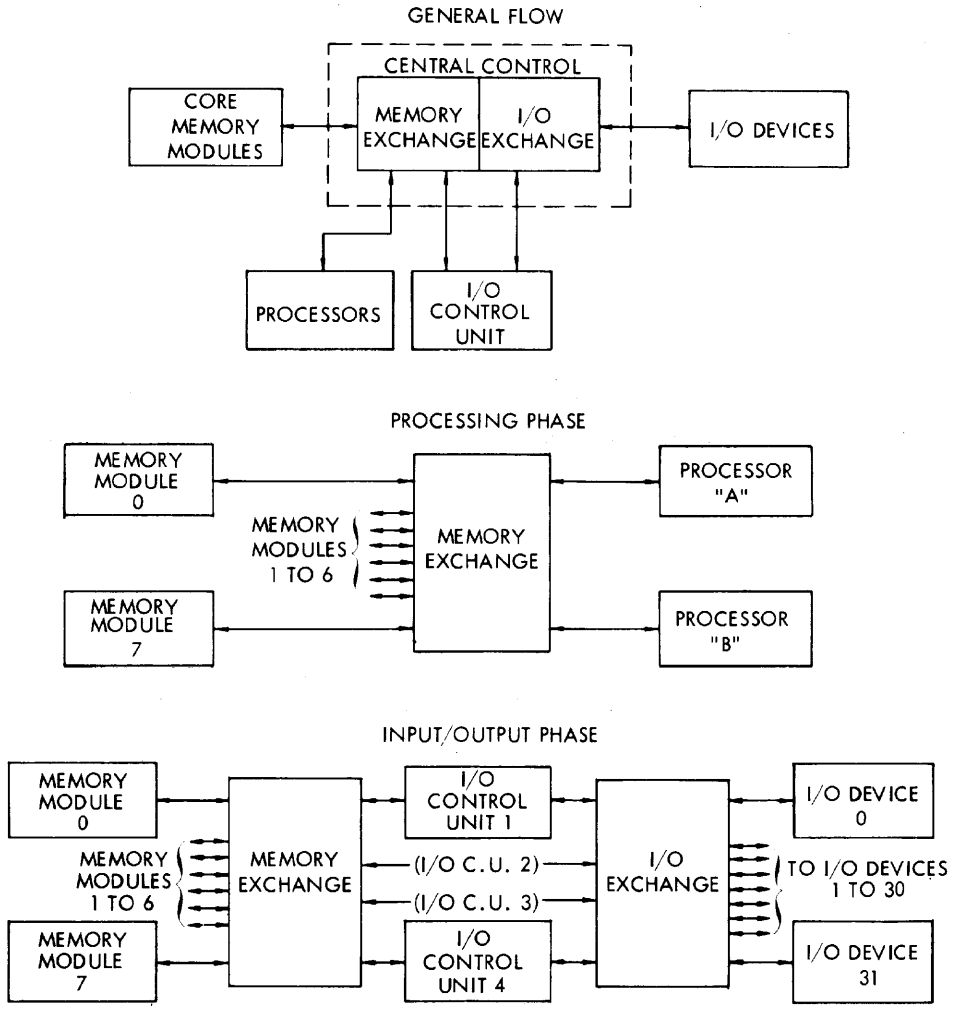


Figure 1-2. Information Flow (General)

Memory Access

1-21. The only components which may access core memory modules are processors or I/O control units. Each core memory module is a self-contained unit independent in operation. This independence permits the simultaneous memory accesses of different modules. There is a maximum of six requesting units (two processors and four I/O control units) which can access a maximum of eight memory modules. A total of six simultaneous memory accesses are possible when each requesting unit is accessing a different memory module. This condition of simultaneous memory accesses is made possible by the memory exchange section (in the central control unit) which provides multiple routes so that any requesting unit may be connected to any memory module that is not in the access cycle (being accessed). The connection (routing) is made for each access, and each access is for one word only.

Input/Output Access

1-22. For I/O operations, the I/O devices are connected to the I/O control units by the I/O exchange section of the central control unit. Then the I/O control units are connected to memory by the memory exchange. The I/O exchange can connect any I/O device to any I/O control unit. Multiple routes can be established to enable simultaneous I/O operations. The connections are made by the I/O exchange for each I/O operation and are maintained for the duration of the input or output cycle.

Information Transfer

1-23. Information is transferred as 48 bit words in parallel between the following: processors and memory exchange, I/O control units and memory exchange, memory exchange and memory modules. Information is transferred seven bits at a time (6 bit information character plus a parity bit) by way of routing between the following: I/O control units and I/O exchange, I/O exchange and I/O devices.

Communication Between Memory and Processor

1-24. MEMORY EXCHANGE. When a word in a processor is to be stored in core memory, the transfer of a word of information is from the processor, through the memory exchange, to the required core memory module (see double line, figure 1-3). Memory exchange contains a memory write exchange and a memory read exchange. These two features provide for a transfer in either direction. Every word of information going to a core memory module from a processor goes through the memory write exchange (see double line). Every word of information going to a processor from a memory module goes through the memory read exchange (see single line).

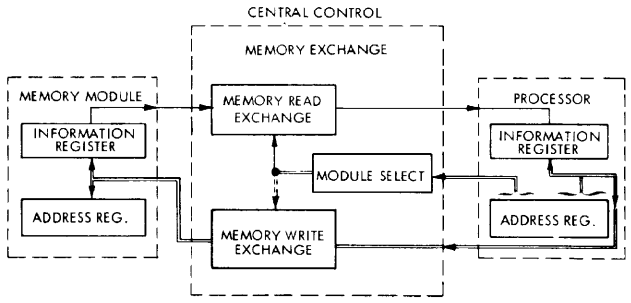


Figure 1-3. Memory-Processor Communication

1-25. MEMORY ADDRESSING. Each core memory module has 4096 words. Each address, within the module, uses 12 bits. A bit represents a binary condition (off or on) which is related to the binary system of counting in which $2^{12} = 4096$. The representation of the address for the 4096th word would be 12 bits all on; the first word, all bits off. Three bits are added to the address field of each word to designate which memory module. There is a maximum of eight modules and $2^3 = 8$. The first module is represented by 3 bits off; the eighth module, by all 3 bits on. The total address field for merely one word of information consists of 15 bits. The three high order (left-most) designate which (if more than one) memory module. The remaining 12 bits designate the particular word in the module. When a memory access is requested by a processor (a processor requests one word of information), the three high-order bits of the address are examined

by the memory exchange module select (see figure 1-3) and the route between the processor and the module is set up. Then, the 12 low order (remaining) bits of the address are transferred to the memory module where the word location is selected and the word of information is transferred either to or from the processor via the route just established.

Communication Between Memory and I/O Control Unit and I/O Device

1-26. I/O EXCHANGE. The communication between a core memory module and an I/O control unit is identical to processor-memory communication. The I/O exchange, like the memory exchange, allows for a transfer of information in either direction. The routing procedure permits any of the four I/O control units to be connected to any of the I/O devices. Generally, information is transferred to and from the I/O devices as 6 bit information characters.

1-27. INPUT. During an input operation, the information is transferred from the input device, through the I/O exchange, to the I/O control unit, through the memory exchange, to core memory. Referring to figure 1-4, the input source character string is read and transferred, one character (6 bits) at a time, to the I/O control unit. A full (48 bit) word of eight characters is built up in the one word buffer contained in the I/O control unit. The first character of the one word buffer is the left-most and is the most significant character of the word. When the last character of a word has been read and placed in the buffer, and before the next character of a word has been read by the input device, the word in the buffer is transferred to core memory. This transfer is effected when the I/O control unit requests a memory access. The proper memory module is selected by the memory exchange and the information word is then transferred to the word address of the memory module through the memory write exchange. As soon as the next source string character is read, the input operation starts building a new word of information in the one word buffer of the I/O control unit. This process will apply to all input except the instance of a magnetic tape backward read operation. In this operation, the first character to be read is the least significant (right-most) character and will be placed in the least significant character position of the least significant word out of all the word positions in core memory that are reserved for this input.

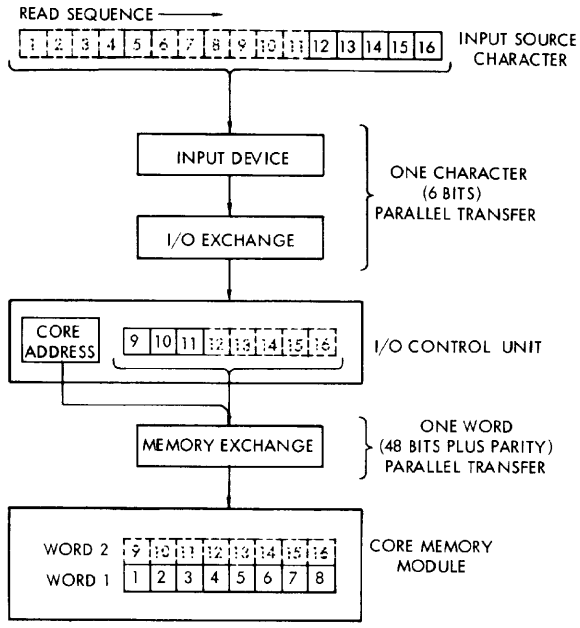


Figure 1-4. Input Information Transfer

1-28. OUTPUT. An output operation transfers information from core memory, through memory exchange, to an I/O control unit, then from the I/O control unit, through the I/O exchange, to the output device. As each new word is needed in the I/O control unit, a memory request is made, the core memory module is selected, and the proper word is accessed. Each character of the word (in the one word buffer of the I/O control unit) is transferred serially (one character at a time) from the I/O control unit, through the I/O exchange, to the output device. As in most I/O transfers, the first character to be transferred to the output device is the most significant and will be in the most significant character position of the output string (figure 1-5).

Interrupt System

1-29. The interrupt system is a process whereby certain conditions occurring during processing will initiate a specific routine contained in the MCP (Master Control Program).

Since the MCP maintains a centralized communications control, an interrupt condition causes a transfer of control from the program to the MCP which may initiate certain types of operations that can proceed simultaneously with computation. In essence, interrupt conditions provide opportunities either for the MCP to transfer control to the area of processing which will use the equipment most effectively, or for the MCP to respond to any error conditions. Because it has over-all control of the system, the MCP is able to make effective use of the available system. There are two classes of interrupt conditions: processor independent or processor dependent.

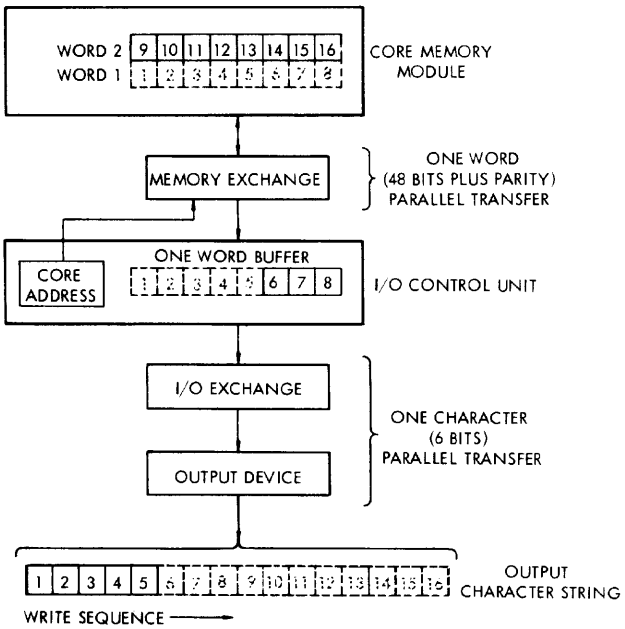


Figure 1-5. Output Information Transfer

1-30. PROCESSOR INDEPENDENT INTERRUPTS. Processor independent interrupts are those which are not initiated or generated by any program code, but which are received from an external source. These are:

a. Time interval - used to collect log information and to check program running time.

b. Processor B busy - used to determine the presence of, or to indicate, a malfunction of Processor B (all work will be shifted to Processor A).

c. Printer finished - used to indicate that one line has been printed.

d. I/O channel busy - used to determine which, if any, channels are available or to indicate a malfunction of a channel.

e. I/O channel finished - used to indicate that a channel is now available.

f. Keyboard request - used to indicate that the system operator has a request to enter via the keyboard on the console.

g. Disk file check operation finished - used to indicate that a disk file check operation is finished.

1-31. PROCESSOR DEPENDENT INTERRUPTS. Processor dependent interrupts are initiated or generated by a program code operating within a processor. They are either a result of a programing error or a result of requiring the MCP to perform a function such as initiating an I/O operation, allocating memory, etc. These are:

a. Memory parity - indicates a parity error in a word read from memory.

b. Invalid address - used to indicate a malfunction of a memory module or program error.

c. Communication operator - used by a program to enter the control state.

d. Flag bit - used to indicate the presence of a construct other than an operand, when an operand is required.

e. Continuity bit - used to indicate that a multiple input/output area is now available.

f. Invalid index - indicates that an index value exceeds a predesignated size.

g. Exponent underflow - indicates that an arithmetic operation has resulted in an exponent value less than -63 (operand less than 8^{-51}).

h. Exponent overflow - indicates that an arithmetic operation has resulted in an exponent value greater than +63 (operand greater than 8^{+76}).

i. Integer overflow - indicates that an operand exceeds 8^{13} when a floating point number is being converted to an integer.

j. Divide by zero - indicates that the divisor is zero when a divide operation is executed.

k. Program release - indicates an input/output area is available to receive or transfer information.

l. Stack overflow - indicates that the stack is about to exceed its allocated area.

m. Presence bit - indicates that a program has referred either to information that is not present in memory or to input/output information that is not available.

1-32. INTERRUPT HANDLING. Whenever an interrupt occurs, the interrupt will be retained until an MCP routine has been initiated to process the specific interrupt. Each interrupt has an assigned priority that causes the MCP to react to the interrupt with the highest priority whenever more than one interrupt occurs at one time. Remaining interrupts will be handled according to this priority sequence. A more thorough explanation of interrupts is presented later.

SECTION 2
DATA REPRESENTATION

GENERAL

2-1. Several methods of data representation are used with the B 5500 Information Processing System. Two systems are used internally, and three systems are used for input/output. The internal systems are the binary system and an alphanumeric code. Input/output devices use Burroughs Common Language (BCL) code, binary, or standard punched card code. A number is composed of an integral part (left of decimal point) and a fractional part (right of decimal point).

BINARY NOTATION

2-2. The decimal system is based upon the ten digits, 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, and upon the powers of ten. Similarly, the binary system is based upon the two digits, 0 and 1, and upon the powers of two. Every binary digit is represented by one flip flop, thus a binary digit is one bit. A number will be represented internally as a series of bits either off or on. When a bit is on (1), its position determines the value. Consider an example of five bits.

2-3. The least significant bit, if on (1), has a value of 2^0, or 1; the next most significant bit ot the left of the binary point has the value of 2^1, or 2; the third bit (count from right to left) has the value of 2^2, or 4; etc. In this manner, any integer can be represented in binary form. The example below illustrates some integers. Fractions in binary are much the same as integers. Here, though, the powers are negative powers with the first power to the right of the binary point having the value of 2^{-1}, or 1/2; the second bit has the value of 2^{-2}, or 1/4; the third bit, 2^{-3}, or 1/8; the fourth bit, 2^{-4}, or 1/16; etc. It is apparent that while some fractions are represented correctly, others can only be approximated. However, the degree of error is very small when a sufficient number of bits are used.

$2^0 = 1$

0 = off bit

1 = on bit

value of position = $2^4\ 2^3\ 2^2\ 2^1\ 2^0$.

...0 0 0 0 1 = 0 +0 +0 +0 + 1 = decimal 1

...0 0 0 1 0 = 0 +0 +0 +2 + 0 = decimal 2

...0 0 0 1 1 = 0 +0 +0 +2 + 1 = decimal 3

.

.

.

...1 1 1 1 1 = $2^4+2^3+2^2+2^1+$ 1 = 16 + 8 + 4 + 2 + 1 = decimal 31

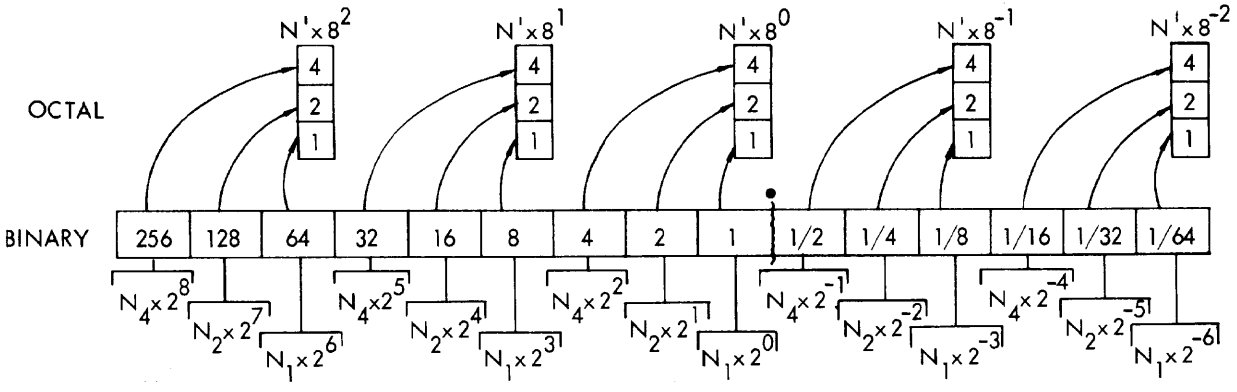


Figure 2-1. Binary to Octal Conversion

OCTAL NOTATION

2-4. When 2 is raised to the third power, 2^3, the result is 8. Eight is the base of the octal system, just as two is the base of the binary system and ten is the base of the decimal system. The relationship between the three systems is shown using the decimal number 325_{10} (the subscript $_{10}$ means "to the base ten"), and its binary and octal equivalents.

$$325_{10} = 3\times10^2 + 2\times10^1 + 5\times10^0 = 300 + 5 = 325$$

$$101\ 000\ 101 = 1\times2^8 + 0\times2^7 + 1\times2^6 + 0\times2^5 + 0\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 =$$
$$256 + 0 + 64 + 0 + 0 + 0 + 4 + 0 + 1 = 325$$

$$505_8 = 5\times8^2 + 0\times8^1 + 5\times8^0 = 320 + 0 + 5 = 325$$

2-5. Note that when the binary number 101000101 is broken into groups of three, working from right to left from the decimal point, it looks like 101 000 101. The result of 101 binary is $1\times2^2 + 0\times2^1 + 1\times2^0 = 4 + 0 + 1 = 5$. Thus, after breaking 101000101, the binary equivalent of 325_{10} into groups of three, and substituting each group by its value, 101 000 101 will become 505. 505 is the octal equivalent of 325_{10}. In the octal system, three bits are used to represent one octal digit; hence, as illustrated, a direct conversion from binary to octal or octal to binary is quite simple. The equivalence of these two systems is shown in figure 2-1 where N = either 0 or 1, and where $N' = N_4 \times 2^2 + N_2 \times 2^1 + N_1 \times 2^0$. (The squares have been crossed out because they were multiplied by zero.) The location of the decimal point, binary point, and octal point is indicated by the serrated line. Similar to figure 2-1, figure 2-2 illustrates the relationship of the systems using the number 325.75.

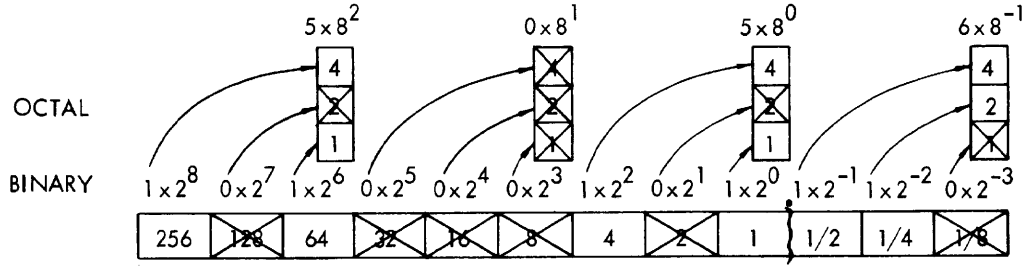


Figure 2-2. Schematic of Binary to Octal Conversion of $325.75_{10} = 505.6_8 = 101\ 000\ 101\ .110$

NUMBER CONVERSION

Binary to Decimal Conversion

2-6. INTEGRAL. This conversion is effected by adding together the value of each bit that is on. In this way, the binary number 11010011 would be equal to

$$1x2^7+1x2^6+0x2^5+1x2^4+0x2^3+0x2^2+1x2^1+1x2^0=$$

$$1x2^7+1x2^6+\ 0\ \ +1x2^4+\ 0^{+0}+1x2^1+1x2^0=$$

$$128\ +\ 64\ +\ 16\ +\ 2\ +\ 1\quad 211_{10}$$

2-7. A second method of effecting a binary to decimal conversion is the "double dabble" method. In this procedure, the high-order bit is doubled (multiplied by 2) and then added to the next lower-order bit. This sum is then doubled and again added to the next lower bit. This process is continued until the entire binary number has been expended (figure 2-3A). The correct result is obtained after the low-order bit (units) has been added.

2-8. FRACTIONAL. The above process will work for integral numbers and for the integral part of fractional numbers, but it will not work for the fractional part of fractional numbers. To convert binary fractions to decimal fractions, division is used. As was previously stated, the bits to the right of the binary point have the decreasing values of 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4},etc., or, as fractions 1/2, 1/4, 1/8, 1/16, etc., respectively. N1 is the binary digit in the first position to the right of the binary point, N2 is the digit in the second position to the right of the binary point, etc. Thus, a binary fraction would be expressed in the decimal equivalent as:

$$.N1x2 + N2x2 + N3x2 = \frac{\frac{N3}{2} + N2}{2} + N1 = \text{decimal equivalent fraction}$$

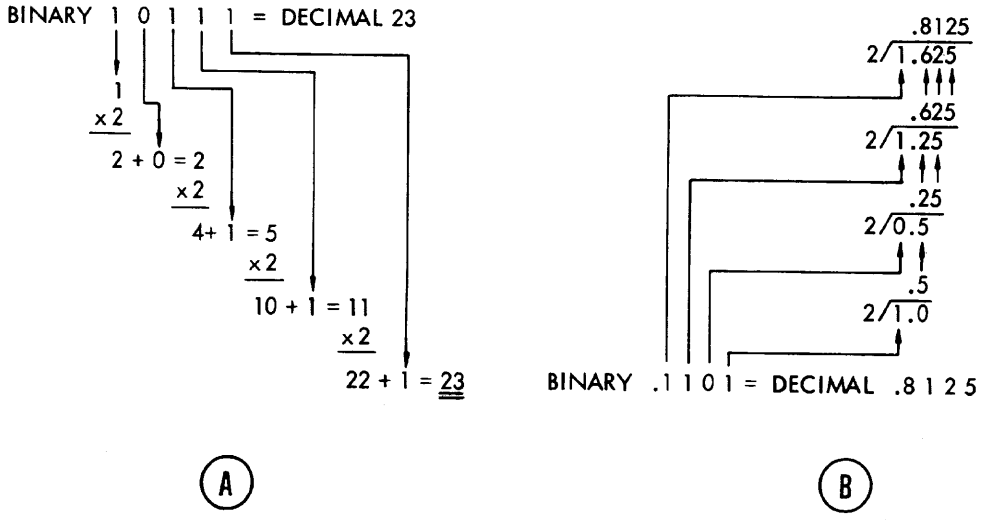


Figure 2-3. Binary to Decimal Conversion

This can be used to find the decimal equivalent of a binary fraction. In this process, the lowest order significant bit is taken as the integer 1 and divided by 2. The next higher-order bit is then added into the unit's position of the resulting quotient, and the division is repeated. This is repeated until the binary point is reached. The result is complete when the bit to the immediate right of the binary point has been added into the unit's position and the result divided by 2. This process is shown in figure 2-3B.

Decimal to Binary Conversion

2-9. INTEGRAL. This may be effected in several ways. If the powers of 2 are known, then the binary equivalent can be found by subtracting from the number the largest power of 2, which is smaller than the decimal number, and then recording a bit for that power of two. The largest power of 2, which is smaller than the result of the preceding subtraction, is then found, subtracted, and the corresponding binary bit recorded. In effect, this is the reverse of the first method of converting from binary to decimal.

2-10. A second method of conversion is accomplished by successive division. The decimal number to be converted is divided by 2 and the quotient and remainder are noted. The remainder will always be either 0 or 1. Then the quotient is divided by 2, resulting in another quotient and remainder. This is repeated until the quotient is 0. The remainder, resulting from the first division, is the low order bit; and the last remainder is the high order bit. This process is valid for the integral part of a number (figure 2-4A).

2-11. FRACTIONAL. The fractional part of a number may be converted in a method somewhat similar to the preceding method of division. The fraction is multiplied by 2 and, if the result is greater than 1, the 1 is recorded in the binary string as a 1 bit. If the product remains less than 1, the binary bit is 0. The fractional part of the product is carried down and again multiplied by 2. This is repeated until the fractional part is equal to 0, or the required degree of accuracy is attained. This process is shown in figure 2-4B.

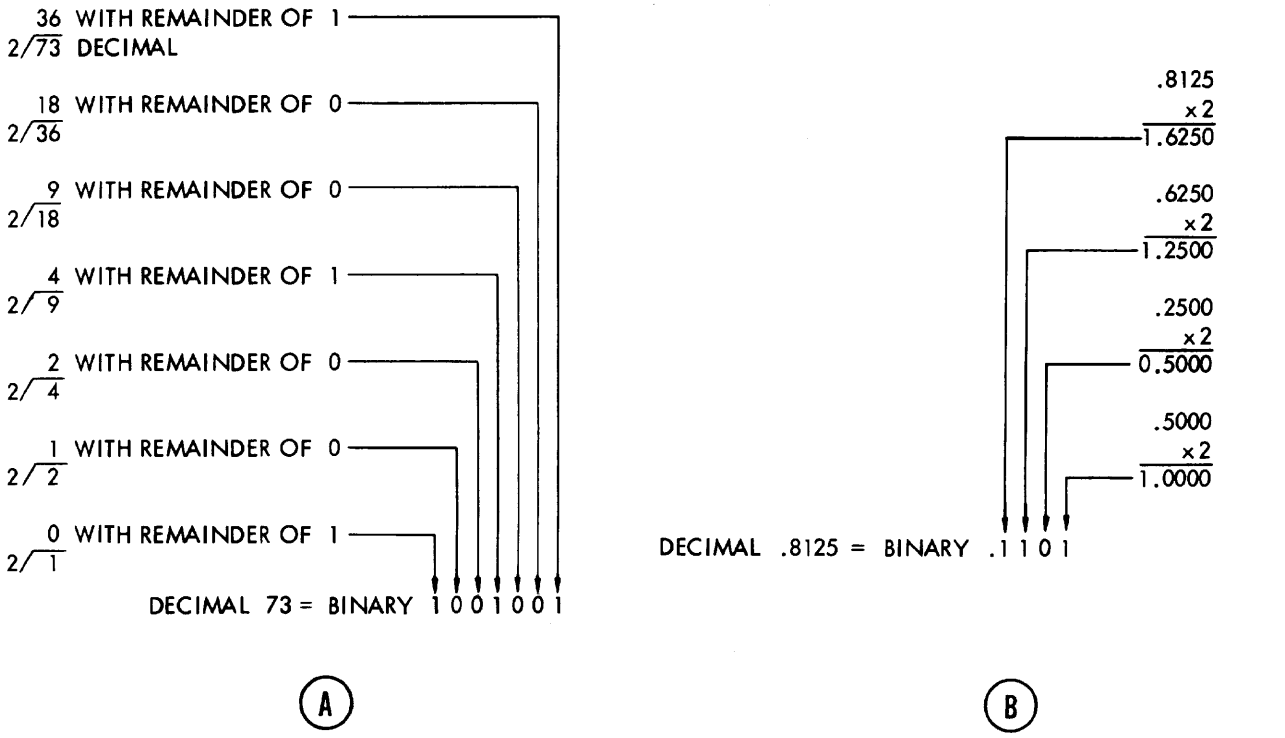


Figure 2-4. Decimal to Binary Conversion

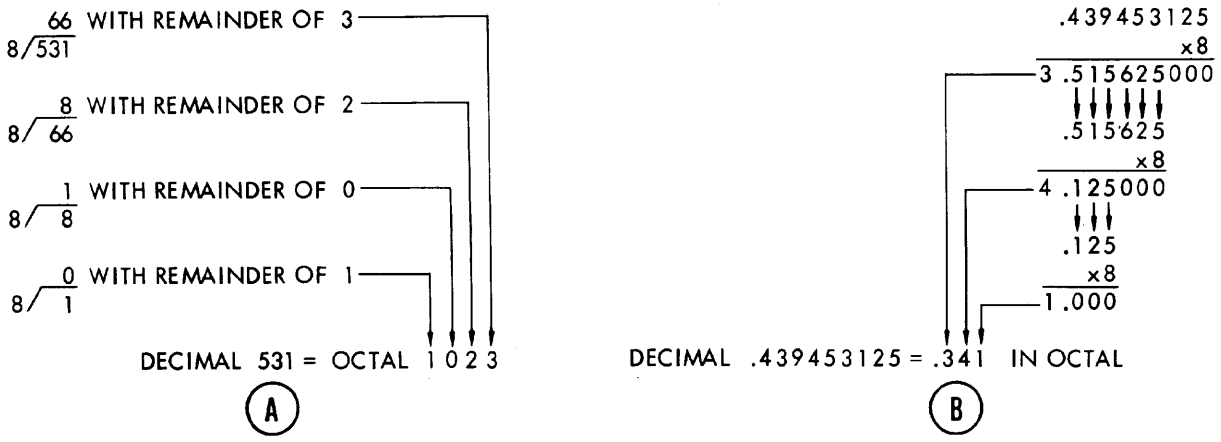


Figure 2-5. Decimal to Octal Conversion

Decimal to Octal Conversion

2-12. INTEGRAL. When it is desirable to convert a decimal number to its octal form, the powers of eight may be used. Another method is to divide the number by eight. The remainder is the low-order octal digit. The quotient is then again divided by eight, and the remainder resulting is the next higher-order octal digit. This process is repeated until the quotient is zero. This method is used for the integral part of numbers (figure 2-5A).

2-13. FRACTIONAL. When a fractional part of the number is to be converted, multiplication is used. Here the fraction is multiplied by eight and the integral portion formed is the first octal digit to the right of the octal point. This process is repeated until either the fraction is zero, or the desired degree of accuracy is attained. This is shown in figure 2-5B.

Octal to Decimal Conversion

2-14. OCTADE. In octal to decimal or decimal to octal conversions, if the powers of 8 are known, then the procedure is much the same as the corresponding subtraction method of binary. The difference is the digital multiplier (N) which will have a value of from 0 through 7 in octal. Each octal digit will be referred to as an octade. The values of the octades are as follows (the top-most relates to the least significant octade):

$Nx8^{13}$	= Nx	549,755,813,888
$Nx8^{12}$	= Nx	68,719,476,736
$Nx8^{11}$	= Nx	8,589,934,592
$Nx8^{10}$	= Nx	1,073,741,824
$Nx8^{9}$	= Nx	134,217,728
$Nx8^{8}$	= Nx	16,177,216
$Nx8^{7}$	= Nx	2,097,152
$Nx8^{6}$	= Nx	262,144
$Nx8^{5}$	= Nx	32,768
$Nx8^{4}$	= Nx	4,096
$Nx8^{3}$	= Nx	512
$Nx8^{2}$	= Nx	64
$Nx8^{1}$	= Nx	8
$Nx8^{0}$	= Nx	1

$Nx8^{-1}$ = Nx .12500

$Nx8^{-2}$ = Nx .01562 50000

$Nx8^{-3}$ = Nx .00195 31250

$Nx8^{-4}$ = Nx .00024 41406 25000

$Nx8^{-5}$ = Nx .00003 05175 78125

$Nx8^{-6}$ = Nx .00000 38145 72265 62500

$Nx8^{-7}$ = Nx .00000 04768 21533 20312 50000

$Nx8^{-8}$ = Nx .00000 00596 02691 65039 06250

$Nx8^{-9}$ = Nx .00000 00073 25336 45629 88281 25000

$Nx8^{-10}$ = Nx .00000 00009 15667 05703 73535 15625

$Nx8^{-11}$ = Nx .00000 00001 14458 38212 96991 89453 12500

$Nx8^{-12}$ = Nx .00000 00000 14307 29776 62086 48681 64062 50000

$Nx8^{-13}$ = Nx .00000 00000 01788 41222 07760 81085 20507 81250

2-15. INTEGRAL. On the conversion from octal to decimal, a method very similar to "double dabble" may be used. Here the higher-order octade is multiplied by 8 and then added to the next lower octade. This sum is then multiplied by 8 and again added to the next lower octade. This is continued until the first octade to the left of the octal point is reached. After the unit's octade has been added, the result should be complete (figure 2-6A).

2-16. FRACTIONAL. The above is valid for the integral part of a number, but for the fractional part of a number, the following must be used. The lowest order octade is considered to be an integer. As such, it is divided by 8. The next higher octade is then added to this quotient in the unit's position and the sum again divided by 8. This continues until the first octade to the right of the octal point has been added and the result divided by 8. See figure 2-6B which is the implementation of:

$$\frac{\frac{\frac{N3}{8} + N2}{8} + N1}{8} = \text{decimal equivalent fraction}$$

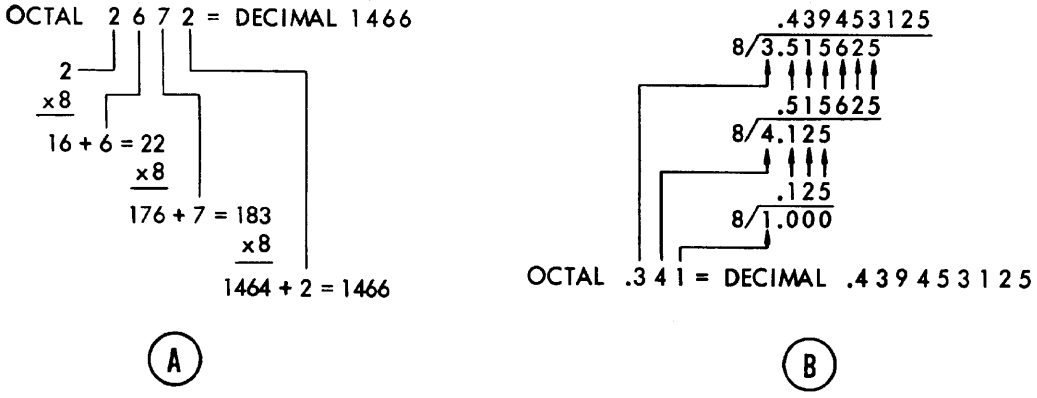


Figure 2-6. Octal to Decimal Conversion

Binary Code Decimal (BCD)

2-17. BCD is numerical representation as it appears in the alphanumeric internal code. In this code, each decimal digit is represented by six bits: 1, 2, 4, 8, A, B. Combinations of the numeric bits with the A and B bits off, are used to represent the digits 0 through 9. As an example, the decimal number 39 would have the 1 and 8 bits on in the low order (units) digit; and the tens digit (3) would have the 1 and 2 bits on. The representation of numbers by this system is shown in figure 2-7.

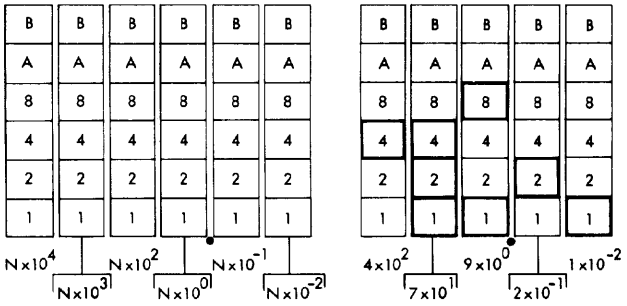


Figure 2-7. Binary Coded Decimal Representation

DATA TYPES AND PHYSICAL LAYOUT

Characters

2-18. When operating in the word mode, data/words of 48 bits are normally referred to as operands and may represent either numeric or logical information. In character mode, data is represented in internal alphanumeric code and is in the form of continuous strings of characters. The portion of a character string ...IS + 34... is illustrated in figure 2-8 as it would appear in character mode. Each BA8421 represents one character expressed in internal alphanumeric code. (See Appendix A.)

Operands

2-19. Operands are the words of information that are worked with when processing in word mode. Bit number 0 being off indicates that the word is an operand. Operands can represent either numeric or logical information.

2-20. NUMERIC OPERANDS (NUMBERS). The sign of a numeric quantity is represented by bit number 1 (figure 2-9). When the bit is off the quantity is positive; when the bit is on, the quantity is negative. All numeric operands are expressed in floating point form, where each numeric operand has both a mantissa and an exponent. This form may be related to power of ten notation where 13297. is the mantissa and -3, the exponent in a representation of the number 13.297 (13297. x 10^{-3}). The mantissa is comprised of 39 bits which make up 13 octades. The mantissa of a numeric operand is considered to be an integer and is treated as such; i.e., the octal point is considered to be to the right of the least significant octade. The exponent of the number is represented by 6 bits (bits 3 through 8) which form two octades. Bit number two is the sign of the exponent. When bit 2 is off, the exponent is positive; when on, negative. The 6 bit exponent field along with the exponent sign provides for a maximum positive exponent of +77 in octal (all bits on in both octades) which is +63 in decimal ($7x8^1 + 7x8^0 = 56 + 7 = 63$). The smallest exponent that could be represented would be -77 in octal which is -63 decimal.

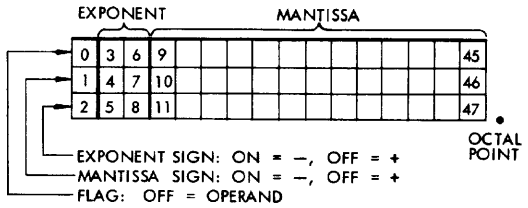


Figure 2-9. Numeric Operand

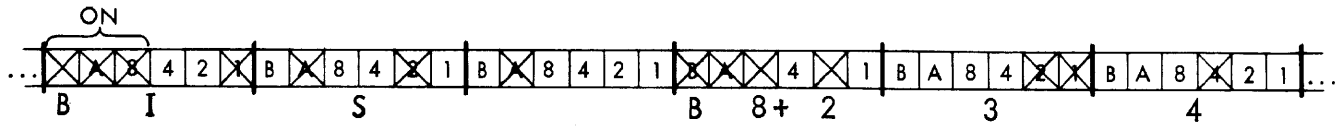


Figure 2-8. Character Mode Representation

2-21. An integer is any number that has an exponent equal to zero. An integer is expressed only in the mantissa which contains 13 octades. Thus, the maximum integer is +7777777777777 in octal which is 549,755, 813,887 in decimal. The exponent must be zero. As an example, the decimal number 12 (14 octal) might be represented in any of the following forms. All notation is octal.

Exponent Sign	Exponent	Mantissa	
+	00	0000000000014	(integer)
-	01	0000000000140	(floating point)
-	02	0000000001400	
.	.	.	
.	.	.	
.	.	.	
-	13	1400000000000	

2-22. LOGICAL OPERANDS. Logical operands are considered to have one of two values: true (on) or false (off). Logical values are a result of Boolean operations or relational operations. Relational operators generate a logical value as a result of an algebraic comparison of two arithmetic expressions. Bit number 47 is used to represent a logical value. Relational operators set only bit number 47, and conditional operators use bit 47 for a decision. Boolean operators consider each bit, from bit 1 to bit 47, as an individual logical value thereby operating on the whole word. A logical value is expressed in the following form (figure 2-10):

Bit 0	off to flag the value as an operand
Bits 1-46	off
Bit 47	off for false, on for true.

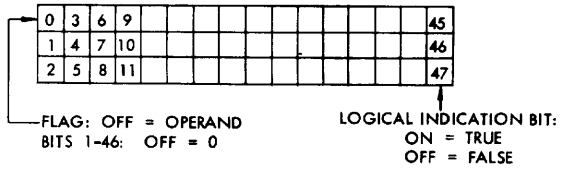


Figure 2-10. Logical Operand

SECTION 3

POLISH NOTATION AND STACK

GENERAL

3-1. To facilitate the understanding of the B 5500 stack concept, a method of notation (Polish notation) must be understood. A problem that exists with most forms of mathematical notation is clarifying the boundaries of specific terms. This has been eliminated with the use of parentheses, brackets, and braces. However, with a complex equation, it becomes necessary to duplicate the use of the few types of delimiters that exist. It might be noted that it is common to encounter mathematical equations such as Y = 5Z + 7/2Z and Y = (5Z + 7)/2Z. Two equations express different functions of Z, but one could easily be used when the other was intended. From this it can be seen that an error in notation can change the whole problem, because the parentheses have definite meaning.

3-2. Polish notation is an arithmetical or logical notational system using only operands and operators arranged in a sequence or string which eliminates the necessity of factor boundaries.

POLISH NOTATION

Polish String

3-3. The essential difference between Polish notation and conventional notation is that operators are written to the right of a pair of operands instead of between them. For example, the conventional B + C would be written B C + in Polish notation. Looking at the example, A = 7 (B + C), it could be written as follows:

B C + 7 x A =

Any expression written in Polish notation is called a Polish string. In order to fully understand this concept, the rule for evaluating a Polish string should be known.

General Rules for Generation of Polish String

3-4. The source of expression is:

Name	Action
Variable	Place variable in string being built and examine next symbol.
Operator -Separator	Place in delimiter list and examine next symbol.
-Arithmetic or Boolean operator and last entered delimiter list symbol was: a. an operator of lower priority. b. a left bracket "[" or paren "(".	Place operator in the delimiter list and examine next source symbol.

Name	Action
c. a separator. d. nothing (delimiter list empty.	
-Arithmetic or Boolean operator and last entered delimiter list symbol was: an operator of priority equal to or greater than the symbol in the source	Remove the operator from the delimiter list and place in the string being built. Then compare the next symbol in the delimiter list against the source expression symbol.

Rule for Evaluating Polish String

3-5. The rule can be summarized in a few steps:

a. Scan the string from left to right.

b. Remember the operands and the order in which they occur.

c. When an operator is encountered do the following:

1) Take the two operands which are last in order.

2) Operate upon them according to the type of operator encountered.

3) Eliminate these two operands from further consideration.

4) Remember the result of (2) and consider it as the last operand in order.

Following this rule through the Polish string step by step, B C + 7 x A = would result in A assuming the value 7 (B + C) (figure 3-1).

Step	Symbol Being Examined	Symbol Type	Operands Being Remembered and Their Order of Occurrence (1 or 2) Before Operation	Operation Taking Place	Results of Operation
a	B	Operand			
b	C	Operand	1 B		
c	+	Add Operator	2 C 1 B	B + C	(B + C)
d	7	Operand	1 (B + C)		
e	x	Multiply Operator	2 7 1 (B + C)	7 x (B + C)	7 x (B + C)
f	A	Operand	1 7(B + C)		
g	=	Replace Operator	2 A 1 7(B + C)	A←7(B + C)	A = 7(B + C)

Figure 3-1. Evaluation of Polish String BC + 7 × A=

3-6. COMPILATION USING POLISH NOTATION. Polish notation is used as the base for the B 5500 ALGOL compilation algorithm. An ALGOL arithmetic or Boolean expression or assignment statement may be translated to Polish notation in much the same way as the arithmetic (or algebraic) expression that already has been considered. In compiler translation, the source expression is examined one symbol at a time with a left to right scan and is combined into logical entities. As each logical entity is examined, a specific procedure is followed so that the Polish notation expression is constructed in its finalized form with one scan of the source expression.

STACK CONCEPT DESCRIPTION

3-7. When a computer is utilized, the problem is expressed in a source language used by the programmer. Portions of the source language program will fall into one of two categories. One will describe the constants and variables that will be used in the program, and the other will be the computational part.

3-8. When the source program is compiled, certain constants and variables are assigned a location within a table. The number of this location, within the table, is referred to as an index. In the B 5500, the area where these constants and variables are placed is called the program reference table (PRT). The computational part of the source program will be converted into a machine language string of instructions. An example of this is the source language plus sign (+) which will be directly replaced by the machine language Add instruction. The machine language string, resembling a Polish notation string, will be referred to as the program segment string and will, as in the case of the PRT, be assigned an area of memory at running time. There are four "instructions" (12 bits each) in each 48 bit program word of the program segment string. Each "instruction" is called a syllable.

3-9. The stack is purely a working area and a temporary storage area used only in word mode. The stack can be thought of as analogous to a physical stack where the last item placed on the stack is the top of the stack. When items are removed (one at a time) from the stack, the item on the top of the stack is the first item to be removed. The item at the bottom of the stack remains at the bottom of the stack until all other items have been removed from the stack. Generally, the stack will be used as a temporary storage area for the constants or variables.

Program Reference Table (PRT)

3-10. Basically, three areas will be associated with the computational part of every program. These are the PRT, which is the storage area of certain constants and variables; the program segment string, for the storage of the actual machine language string of instructions that have been generated through the actions of the compiler program; and the stack which is used as a temporary storage area to combine the operands and operators needed to develop the results of given problems or source language statements.

3-11. RELATIVE ADDRESSING. When information within the PRT is to be referenced, it will be called upon by a method of relative addressing. The relative address requires two elements: one element is the address of the PRT, the other element is the location number (index) which will be added to the address. For example, consider data that has been assigned the forty-second word location of the PRT. This word would have an index of +41 counting from 0. Note that this 41 is not an absolute address, but is just a position within the PRT. An index of zero would be the first word of the PRT. At the time a statement is compiled that references the information at +41, the instructions created by the compiler to reference this information will have the index of +41 a part of the instruction. Each time the program is run, the MCP assigns an absolute core address as a base address for the PRT. The PRT contents are then read into an area beginning with the address and continuing with successive words above this base address. Therefore, the information in the forty-second word of the PRT is located using an absolute address formed by adding 41 to the base address. By storing the PRT in a different area of memory, the same information can still be referenced, even though it has a different absolute address. Relative addressing is used to access information from two areas other than the PRT.

These are the program segment string and the stack. Relative addressing within the stack may be both positive and negative with respect to an absolute base address.

3-12. SIMPLE STACK OPERATION. The stack is usually used for a temporary storage or working area. For example, the add operator (instruction) will add the two top words of the stack together, delete the top word, and place the sum in the second word of the stack. The sum subsequently becomes the top word of the stack. As in the add operator, the other operators also manipulate the stack to achieve their desired functions. Before further stack discussions can be considered, the instructions necessary for bringing information to the stack will be discussed briefly.

Program Segment String Syllables

3-13. OPERAND CALL SYLLABLE. The program segment string contains specific types of syllables that place information into the stack. When data is desired in the stack, the word of data may be brought to the stack with an operand call syllable. As mentioned in relative addressing, this syllable construction will contain an index value to be used to increment an address. Assume the PRT will be referenced for the following problem:

It is desired to add together the operands X and Y. In order to add these two operands, X would be brought from the PRT and placed in the stack and then the same operation for Y will be completed. The Polish notation sequence is XY+. The actual program segment string would appear as follows (Note the similarity of the Polish notation sequence to the program segment string sequence.):

(OPDC = Operand Call syllable)

Operation Sequence

1) OPDC X
2) OPDC Y
3) ADD

Executing this program will leave the sum of X + Y as the top word in the stack (figure 3-2). Advancing this example further, assume that at compilation time X was assigned an index of +32 and Y an index of +45. When the compiler compiles the expression X + Y, the resulting machine language will be:

OPDC32 OPDC45 ADD

Except for the conversion of these syllables to their respective internal octal codes, this program segment string is shown as it would appear in the computer.

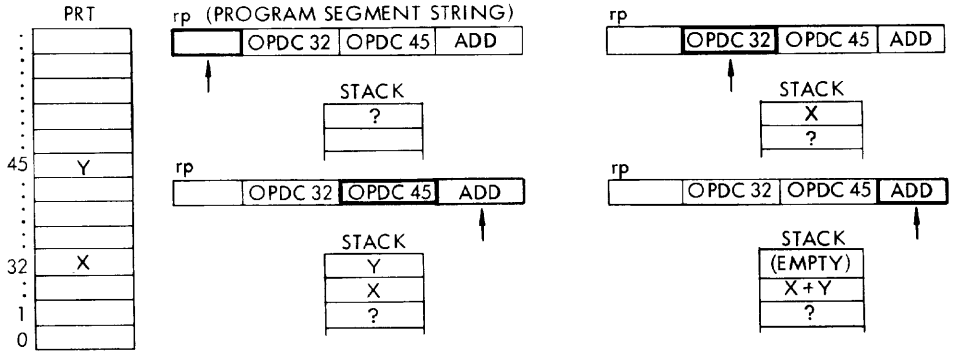


Figure 3-2. Execution Sequence and Stack Movement of X + Y

3-14. DESCRIPTOR CALL SYLLABLE. When executed, the Descriptor Call syllable (DESC) will place in the stack a word containing the absolute address of the referenced operand. The word generated and placed in the stack is referred to as a data descriptor. The absolute address is contained in the lower 15 bits of the word. For example, DESC for the previously discussed "X" would place, in the top of the stack, a data descriptor containing the absolute address of "X." The DESC places the address of the operand in the stack, whereas the Operand Call syllable places the operand itself in the stack. In the example Z←X + Y, a DESC may be used to reference "Z" since "Z" receives the value X + Y. A Store syllable would be executed to store the sum of X + Y in "Z." Assume that "Z" has an index of +100, the program could have the following sequence (the Polish notation string is ZXY + ←):

```
DESC = Descriptor Call syllable
DESC:  100, OPDC: 32, OPDC: 45, ADD,
       EXCHANGE, STORE;
```

3-15. The Exchange syllable is necessary to arrange the stack for the proper operation of the Store syllable. Since the Store syllable requires the address of "Z" as the top word of the stack and the sum of X + Y as the second word of the stack, they must be exchanged prior to executing the Store operation. The following is a brief discussion (illustrated in figure 3-3) of the operation of Z←X + Y. The first syllable, DESC: 100, will place the absolute address of Z in the stack. For example, use a PRT address of 10300. After executing the DESC, the top word of the stack would be a data descriptor containing an absolute address of 10400. The second syllable, OPDC: 32, would place the value of X (brought from 10332) in the top of the stack and force the data descriptor into the second word of the stack. The third syllable, OPDC: 45, would place the operand Y on the top of the stack and push the operand X and the absolute address of Z (data descriptor) down to the second and third words, respectively, of the stack. The Add syllable would then add X and Y. The sum now becomes the top of the stack, because the Add syllable deletes the operands X and Y from the top of the stack and leaves only the sum of X and Y. The Exchange syllable will switch the two top words of the stack. This will place the sum of X + Y in the second word and the absolute address of Z in the top word of the stack.

3-16. The Store syllable will then place the second word of the stack in the address specified by the top word, or in this example, the sum of X + Y will be placed in the absolute address 10400, the location of Z.

3-17. LITERAL SYLLABLE. Another type of syllable is the Literal syllable. This type of syllable is used whenever a positive integer is required having a value less than the decimal value of 1024 and when executed places the integer in the top of the stack. Within the 12 bits that make up a syllable, two bits are reserved to identify the type of syllable. The remaining ten bits could contain a maximum binary value equivalent to 1023. The remaining ten bits of the Literal syllable will be the value of the constant. When this syllable is executed, the ten bits are used to form an integer operand word as the top word of the stack.

Stack Area Description

3-18. STACK LOCATION. The MCP allocates, to each operational program, an area of core memory to be used as a stack. The lowest addressed location forms the bottom of the stack, and successively higher memory locations are used when words are put into the stack. The first word of the PRT immediately follows the last word (top) of the stack area.

3-19. STACK REGISTERS (A,B,S). Two arithmetic registers are associated with the stack in the B 5500. These are the A and the B registers. They are the two arithmetic registers contained by each processor. These are associated with the stack in such a way that all information entering or leaving the stack must go through one or both of these registers. The A and/or B registers contain part of the stack only if the information in the respective register is valid. It is thus possible for the two top words of the stack to be in the arithmetic registers A and B, and for the balance of the stack to be in core memory. An address register S, contains the

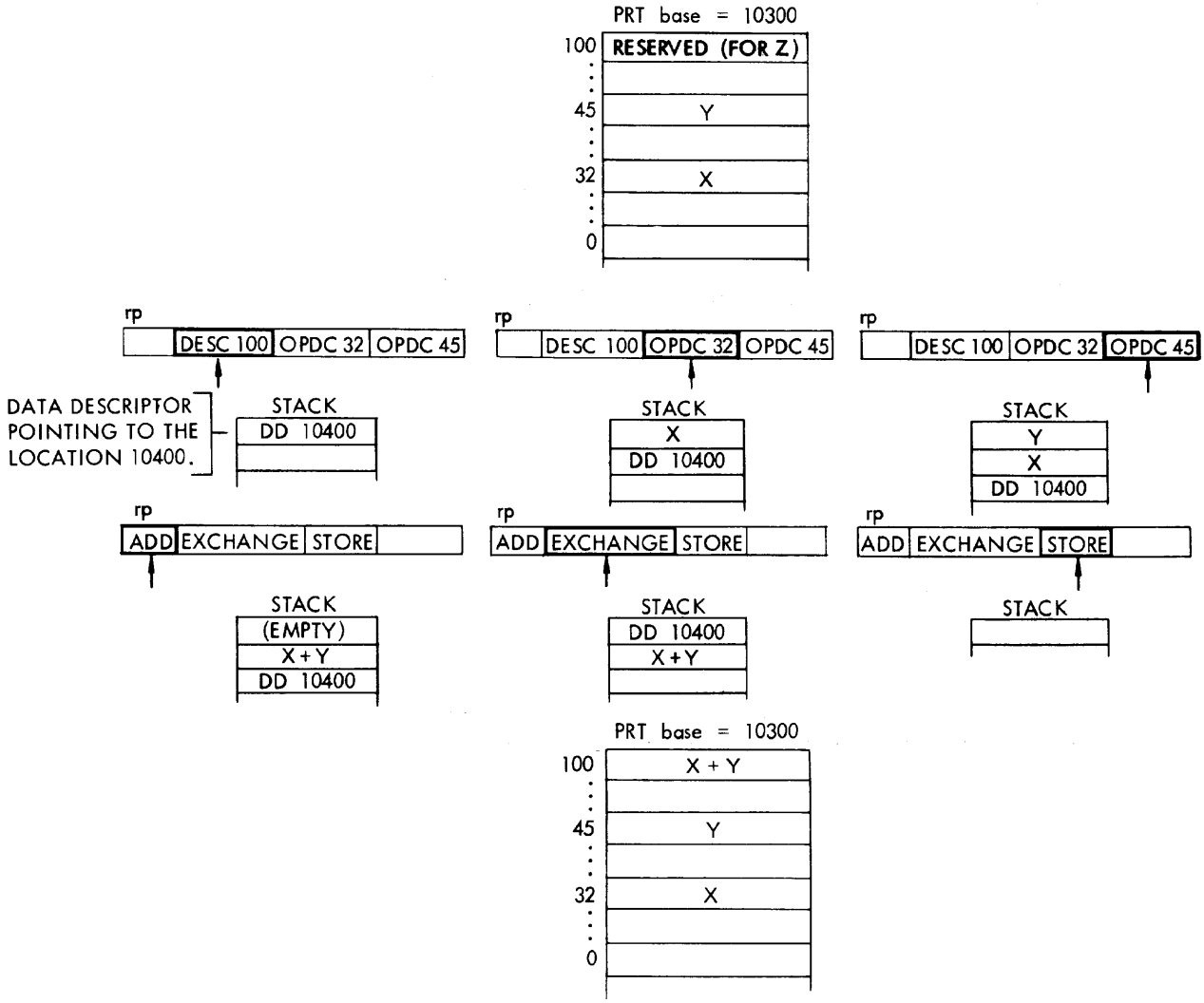


Figure 3-3. Execution Sequence and Stack Movement of Z←X+Y

address of the top word of the stack that is in core memory. The S register is counted up or down as words are added to, or taken from, the core memory portion of the stack. When one word is added to the core memory portion of the stack, the S register is counted up by 1 prior to storing the word into the core stack.

3-20. TOP OF STACK. If the A and B registers do not contain valid information, then the word addressed by the S register is the top word in the stack and is referred to as the "top of the stack." If the A register does not contain valid information, and the B register does contain valid information, then the word in the B register is the top of the stack. The word addressed by the S register is the second word in the stack. When register A contents are valid and register B contents are not valid, then the top of the stack is in A and, again, the second word of the stack is addressed by register S. When both A and B registers contain valid information, then the top of the stack is in register A; the second word, in register B; and the third word of the stack is addressed by register S. These conditions are listed in table 3-1.

Stack Adjustment

3-21. Whenever a word is transferred from the A or B registers to the memory portion of the stack, the S register is counted up one and the word is stored in the new S register

TABLE 3-1

Arithmetic Registers Relative to Core Portion of Stack

A Register Contents	B Register Contents	Top of Stack In	Second Word of Stack In	Third Word of Stack In
Valid	Valid	A register	B register	Word addressed by S register
Valid	Invalid	A register	Word addressed by S register	Word addressed by (S-1)
Invalid	Valid	B register	Word addressed by S register	Word addressed by (S-1)
Invalid	Invalid	Word addressed by S register	Word addressed by (S-1)	Word addressed by (S-2)

address. When the word pointed to by S is to be transferred to the A or B register, the word is transferred and the S register is counted down by one to point to the next valid word in the core memory portion of the stack. This transfer of words between the arithmetic registers and the core portion of the stack is referred to as an adjustment of the stack. The normal procedure for an adjustment of the stack when the arithmetic registers are to be filled from the core portion of the stack is such that register B is filled. B is then shifted to A and B is filled again from the top word in the core portion of the stack. S is counted down twice in the process.

3-22. RELATIVE ADDRESSING IN STACK (F REGISTER). When an operator is encountered that involves two operands, the stack is adjusted so that A and B both contain valid information and then the operation is performed. The result is left in the B register, and the A register is left empty. All operators work with either one word or two words from the top of the stack. Sometimes it is desirable to use a word that is not at the top of the stack. Under this situation, this word is accessed from within the stack and placed in the top of the stack. To permit this, relative addressing is allowed in the stack. The base address of a working area of the stack is contained in an address register labeled F. This base address is used in conjunction with an index to access a word in the stack.

3-23. STACK IN OPERATION. It must be realized that all information a program has to work with must be brought into the system before it can be worked upon. An area of memory must be set aside for both input and output. Information that is being introduced to the system is brought into the system through the input area. Information leaving the system exits through the output area. Arrays and tables are used in conjunction with the storage of certain types of data. Each of these arrays and tables are allocated an area in core memory. This location is determined by core memory space available when the area is to be first used. Thus data is stored in several different areas: the program reference table, input/output areas, data tables (arrays), and the stack. In addition to these, constants will be contained as words or syllables in the program string. Since all work is done in the arithmetic registers, all information or data is transferred to the arithmetic registers and the stack. Results which are generated must be transferred from the stack to one of the preceding areas.

3-24. At this point, an ALGOL assignment statement and the Polish notation equivalent will be related to the stack concept of operation. The example is $Z \leftarrow Y + 2x(W+V)$, where

← means "is replaced by." In terms of a computer program, this assignment statement indicates that the value resulting from the evaluation of the arithmetic expression is to be stored in the location representing the variable Z.

3-25. When Z←Y + 2x(W+V) is translated to Polish notation, the result is ZY2WV+x+←. Each element of the example expression causes a certain type of syllable to be inserted in the machine language program when the source problem is compiled. The following is a detailed description of each element of the example, the type of syllable compiled, and the resulting operation (see figure 3-4 and table 3-2). Since "Z" is to be the recipient of a value, the address of "Z" must be placed in the stack so that at a later time in the program the computed value of the problem may be stored. In order to place an address in the stack, a DESC is executed on "Z." At compile time Z, W, and V would have been reserved a location within the PRT. In the problem, Y is to be added to a quantity; therefore, Y will have to be brought to the stack as an operand. This is accomplished with an Operand Call syllable on Y. The value 2 is a positive integer with a value less than 1024, in magnitude. Instead of reserving a 48-bit word within the PRT for this quantity, a Literal syllable with the value 2 is compiled. W is to be added to a value and, therefore, must be brought to the stack as an operand. V is also an operand and, therefore, is called to the stack by the Operand Call syllable. The next operator encountered in the Polish notation string is the add symbol. This symbol will cause the compilation of the add syllable for the add operator. When this operator is executed, it will cause the value of V to be added to W. V and W no longer exist in the stack and the top of the stack contains the sum of V and W. The next syllable encountered in the Polish notation string is the multiply symbol. For this symbol a multiply operator will be compiled, and when executed, would multiply the results of V + W by 2. The "2" and "sum V + W" are deleted from the top of the stack and the product "(2x(V+W)" is left as the top word of the stack. The next syllable encountered in the Polish notation string is another add symbol. For this symbol, the add operated is compiled, and when executed, would add the results of 2 times "W plus V" to Y. At this time all necessary arithmetic symbols have been replaced with their respective operator and, if the program segment string were executed, the arithmetic result would be found in the top of the stack. It is now necessary to store the result just generated in the location of "Z." The store operator will take the second word of the stack (B register) and store it into the area pointed to by the top word (data descriptor) of the stack (A register). The configuration of the stack at this time has the result of the computation in the top of the stack, while the address that the result is to be stored in is the second word of the stack. In order to satisfy the requirements of the store operator, these two words must be exchanged. The XCH (exchange) operator is the next syllable in sequence. After the execution of the Exchange syllable, it is then possible to store the results of the arithmetic operations into the area pointed to by the data descriptor that was placed into the stack by the descriptor call on "Z." This is accomplished with the Store syllable. This completes the compilation of the statement Z←Y + 2x(W+V). The PRT, PST (program segment string) and stack could then be assigned an area of memory, and the program run. The PRT would contain reserved locations for Z, Y, W and V. The PST would contain the following syllables for the specified statements:

DESC: Z; OPDC: Y; LITC: 2; OPDC: W;
OPDC: V; +; x; +; XCH; STD

The stack, at completion of the problem, would be empty.

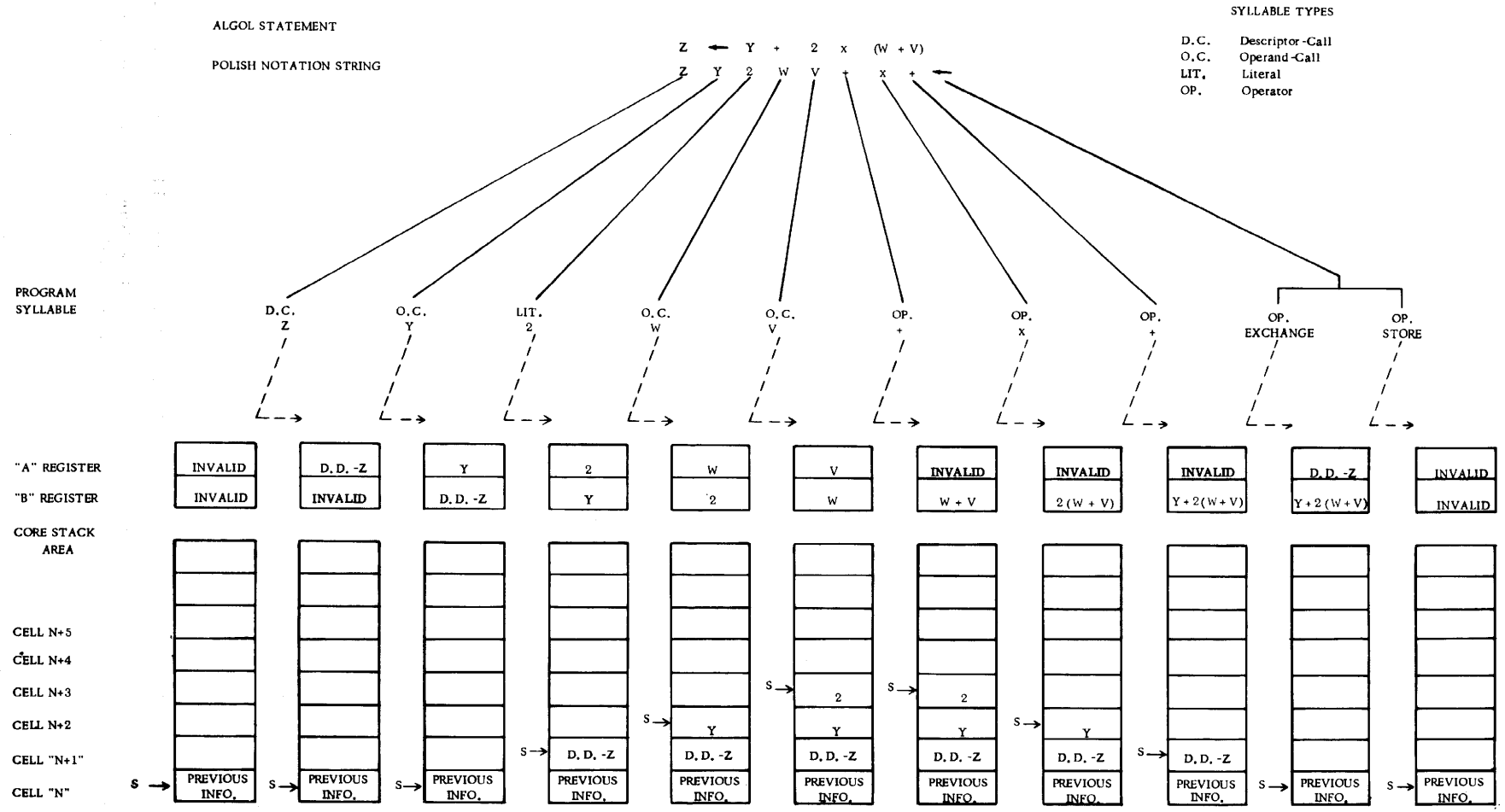


Figure 3-4. Stack Operation

TABLE 3-2

Description for Example of Stack Generation

Execution Sequence	Polish Notation Element	Syllable Type Compiled	Function of Syllable During Running of Program
1	Z	Descriptor Call for Z	Place the address of Z in the top of the stack and push down the stack.
2	Y	Operand Call for Y	Place the value of Y in the top of the stack and push down the stack.
3	2	Literal with Value of 2	Place a 2 in the top of the stack and push down the stack.
4	W	Operand Call for W	Place the value of W in the top of the stack and push down the stack.
5	V	Operand Call for V	Place the value of V in the top of the stack and push down the stack.
6	+	Operator - ADD	Add the two top words in the stack and place the result in the B register as the top of the stack.
7	x	Operator - MULTIPLY	Multiply the top of the stack by the second word of the stack (A register contents multiplied by B register contents). Place the result in the B register as the top of the stack.
8	+	Operator - ADD	Add the two top words in the stack and place the result in the B register as the top of the stack.
9	←	Operator - EXCHANGE Operator - STORE	Since the store operators store the second word of the stack in the location specified by the top of the stack, the data descriptor and the result must be interchanged before storing. The store operator causes the second word in the stack to be stored in the location specified by the top of the stack.

SECTION 4

MAJOR REGISTERS AND CONTROL FLIP FLOPS

GENERAL

4-1. Some of the basic concepts of the B 5500 Information Processing System have been introduced. Each of the major units of the B 5500 are presented along with the actual registers and control flip flops that are included in the units. These registers and flip flops are explained in terms of their major usage within the system. If a system has two processors, the circuitry of the second processor is a duplicate of the first processor. In addition to being displayed on various units, a flip flop indicator may be set or reset. Each flip flop indicator is a manual push button; the setting is changed merely by pushing. The display panel also contains switches for maintenance purposes.

PROCESSOR

Registers and Flip Flops

4-2. A REGISTER. The A register is a 48-bit information register that holds one complete word of information. In word mode, the A register may contain the top word of the stack and is one of the arithmetic registers. In character mode, the A register contains one word (8 characters) of a source string of information.

4-3. B REGISTER. The B register is a 48-bit information register which holds one complete word of information. In word mode, the B register may contain either the top word or the second word of the stack. In the word mode, the B register is the other arithmetic register (the A register is the first arithmetic register). In character mode, the B register contains one word (8 characters) of a destination string of information.

4-4. AROF. This is a control flip flop used to indicate the validity of the information in the A register. If AROF is on, the A register contains valid information. This means that the A register is the top of the stack.

4-5. BROF. This is a control flip flop used to indicate the validity of the information in the B register. If BROF is on, the B register contains valid information. It therefore may be either the top or the second word in the stack.

4-6. Y REGISTER. The Y register is a 6-bit register that is part of the serial adder. The bits of the Y register are labeled as B, A, 8, 4, 2, and 1. The Y register contains one character that is shifted to it from the A register.

4-7. Z REGISTER. The Z register is a 6-bit register and is part of the serial adder. The bits of the Z register are labeled as B, A, 8, 4, 2, and 1. The Z register contains one character that is shifted to it from the B register. The output of the serial adder is placed in the Z register before shifting the result back to the B register.

4-8. G REGISTER. The G register is a 3-bit register that indicates a source string character position in the A register. The G register counts from 0 to 7, and when equal to 0, it indicates the most significant character position.

4-9. K REGISTER. The K register is a 3-bit register that indicates a destination string character position in the B register. The K register counts from 0 to 7. When at 0 it indicates the most significant character position.

4-10. H REGISTER. The H register is a 3-bit register that is used to indicate the particular bit of the character selected by the G register. The H register counts from 0 through 5 and back to 0. When the H register contains a 0, it indicates the B bit of the selected character (character bits being designated as B-0, A-1, 8-2, 4-3, 2-4, 1-5).

4-11. V REGISTER. The V register is a 3-bit register that is used to indicate a particular bit of the character selected by the K register. The V register counts from 0 through 5 and back to 0. When the V register contains a 0, it indicates the B bit (of B, A, 8, 4, 2, and 1) of the selected character which is in the B register.

4-12. N REGISTER. The N register is a 4-bit counter which is used to record the octal shifts performed on the contents of the B register. When the contents of the B register are shifted to the left, the N register is counted up; when the contents of the B register are shifted to the right, the N register is counted down. The N register counts from 0 to 15 and back to 0. It is 0 when the contents of the B register are in the normal unshifted position.

4-13. X REGISTER. The X register is a 39-bit information register which is used as an extension of the B register mantissa in some of the word mode arithmetic operations. This extension allows for an additional accuracy of approximately twelve decimal digits. When in character mode, the X register is used to contain a loop control word. A loop control word contains information about the return point, the number of times to be repeated, repeat count, etc.

4-14. M REGISTER. The M register is a 15-bit address register that is used in word mode for all relative address accessing of information. During word mode arithmetic operation, fields of the M register are used as high order extensions of the A and B register mantissas and exponents and as several control counters. In character mode, the M register contains the address of the source string word being processed, and is used to access the source string words.

4-15. S REGISTER. The S register is a 15-bit address register that is used in word mode to address the top word in the core portion of the stack. In character mode, the S register is used to address and access the destination string words.

4-16. R REGISTER. The R register is a 9-bit register that is used to contain the address of the base of the program reference table. The 9 bits are used as the high order bits of a 15-bit address, the low order 6 being 0. The R register is not used to access memory. Its contents are shifted to the M register and added to the index for "relative addressing." During character mode, the six low order bits of the R register form the Tally register.

4-17. F REGISTER. The F register is a 15-bit address register that contains the address of the last return control word or mark stack control word (MSCW) that has been placed in the stack. In character mode, the F register always contains the address of a return control word (RCW) so that control may be returned to word mode.

4-18. E REGISTER. The E register is a 6-bit register that is used for memory access control. Individual bits determine whether to read a word out of core memory or to write a word into core memory, which address register to use, and the destination or source register (A register, B register, M register, S register, or P register).

4-19. P REGISTER. The P register is a 48-bit register that contains the current program word of the program string segment. Each program word is comprised of four 12-bit program syllables which are numbered 0, 1, 2, and 3.

4-20. T REGISTER. The T register is a 12-bit register that contains the program syllable currently being executed.

4-21. C REGISTER. The C register is a 15-bit address register that contains the address of the program word that is contained in the P register.

4-22. L REGISTER. The L register is a 2-bit register that selects the syllables from the program word in the P register. The L

register counts from 0 to 3 and back to 0. It may point to the syllable that is presently in the T register or as is the usual case, to the next syllable to be executed.

4-23. TROF. This is a control flip flop used to indicate that the T register contains a valid syllable. It permits execution of the syllable. When TROF is off, a new syllable is required in the T register.

4-24. PROF. This is a control flip flop used as a part of the control for shifting a syllable from the P register to the T register in order to be executed.

4-25. NCSF. This is the normal control state flip flop. When it is on, it indicates that the processor is operating in normal state.

4-26. SALF. This is a sub-program level flip flop. When it is on, it indicates that the processor is in the sub-program level of operation.

4-27. CWMF. This is the character-word mode flip flop. When it is on, it indicates that the processor is operating in character mode.

4-28. HLTF. This is the halt flip flop, which is used to stop clock pulses to the processor for maintenance purposes. It is not used during normal processing.

4-29. MSFF (TFFF OR Q12F). The mark stack flip flop, the true-false flip flop, and Q12F are all the same flip flop. In word mode, it is the mark stack flip flop. When it is on, it indicates that the last control word placed in the stack was a mark stack control word. It is turned off when a RCW is placed in the stack. In character mode, the flip flop is called the true-false flip flop (TFFF). When the TFFF is on, it indicates that the result of the last conditional test was true.

4-30. Q REGISTER. The Q resister is actually a number of logical flip flops that are used to control some of the processor operations. As noted above, the MSFF is one of the Q register flip flops.

4-31. I REGISTER. The I register is composed of 7 bits. The bits are numbered 1, 2, 3, and 5 through 8. The first three bits are set individually as the result of a specified interrupt condition. The last four (5 through 8) are set to a specific combination as the result of other interrupt conditions. If any of these bits are set while operating in normal state, the processor generates control words using the information in specific registers and then enters control state. This operation will be performed automatically at the end of the execution of the program syllable in the T register.

4-32. J REGISTER. The J register is a 4-bit counter that is used for primary logical control during the execution of each syllable.

4-33. MEMORY ACCESS CONTROL FLIP FLOPS (MROF, MRAF, MWOF). There are three flip flops in each processor that are used within the processor for memory timing purposes. MROF (memory read obtained flip flop) is set to the on status to indicate that information has been obtained during the core memory read cycle. MRAF (memory read access flip flop) is set to the on status to indicate that a program word has been obtained when accessing a new program word to be shifted to the P register. MWOF (memory write obtained flip flop) is used to indicate that a word has been written in core memory during a memory write operation.

4-34. REGISTER DISPLAY. The display for all processor registers is in the display and distribution unit. All of the registers and control switches for each processor are on one panel. Figure 4-1 shows this panel layout. Each indicator light on the display panel contains a switch so that flip flops may be manually set or reset. There is a REG CLEAR switch to clear the entire register. In addition to the register display, there are a number of control switches for maintenance purposes.

Program Syllable Access

4-35. Four registers and two flip flops are associated with the access and execution of program syllables. These are the C, L, P,

and T registers, and the TROF and PROF flip flops. The P register holds 4 syllables. One syllable at a time is transferred to the T register for execution. The L register contains the syllable number, within the program word, of the next syllable to be transferred to the T register. It holds this number until the syllable is transferred to the T register and execution of the syllable is started. After execution of the syllable is started, the L register is counted up by one. When the last syllable word is transferred to the T register, the L register has been counted up to 0. At this time the C register is counted up by one and a new program word is brought into the P register. The C register contains the address of the program word. TROF and PROF control the counting of the L and the C registers and the transfer of the next syllable to the T register.

Information Access

4-36. Two registers are used to contain an address for access to core memory for information. These are the S and M registers. When an access is obtained, the information is shifted to or from the core memory module for either the A or B register. If the information is read from memory, then it is shifted to either the A, B, or in a special case, the M register. When the M register is used to receive information, only bits 18 through 32 of the word are used.

4-37. E REGISTER. The E register of the processor is used whenever a memory access is needed. For information, only the four low-order bits of the E register will be used. The coding for the usage is as follows:

E set to 2 — Use the contents of the S register for the address of a core memory read access. When the word is obtained, place it in the A register.

E set to 3 — Use the contents of the S register for the address of a core memory read access. When the word is obtained, place it in the B register.

E set to 4 — Use the contents of the M register for the address of a core memory read access. When the word is obtained, place it in the A register.

E set to 5 — Use the contents of the M register for the address of a core memory read access. When the word is obtained, place it in the B register.

E set to 6 — Use the contents of the M register for the address of a core memory read access. When the word is obtained, place the bits, 18 through 32, of the word into the M register.

E set to 10 — Use the contents of the S register for the address of a core memory write access. Store the word in the A register in the address specified.

E set to 11 — Use the contents of the S register for the address of a core memory write access. Store the word in the B register in the address specified.

E set to 12 — Use the contents of the M register for the address of a core memory write access. Store the word in the A register in the address specified.

E set to 13 — Use the contents of the M register for the address of a core memory write access. Store the word in the B register in the address specified.

Processor Interrupt

4-38. I REGISTER. The interrupt register (or I register) of each processor contains 7 bits. Three of these represent specific interrupt conditions. These three conditions have separate flip flops because they may all occur at the same time. The other 4 bits are coded, and only one interrupt may be present at any one time. The interrupt condition and the bits that they set are as follows:

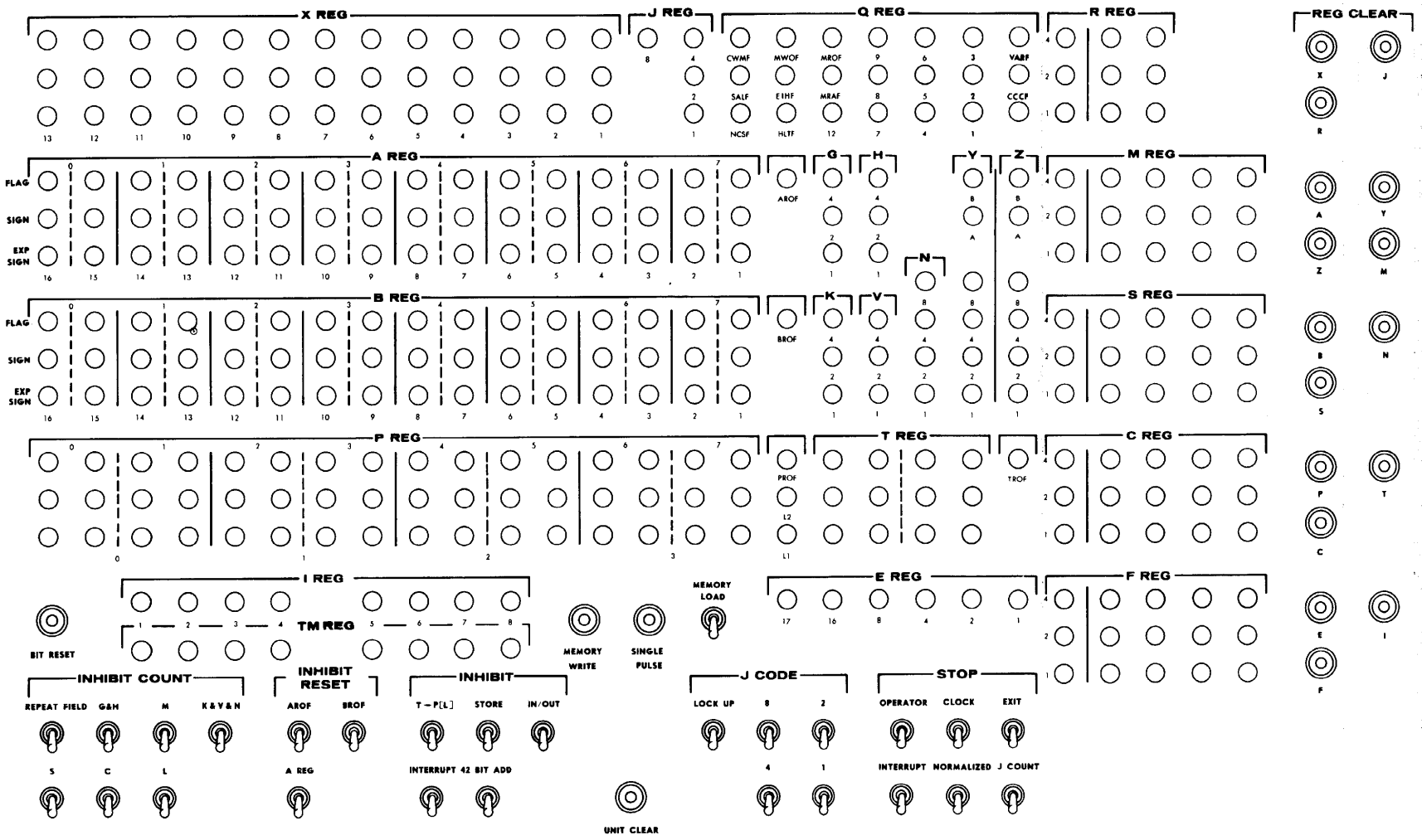


Figure 4-1. Processor Display Panel

I01	ON	Memory parity error. This indicates that a parity error has occurred as a result of a processor memory access. The word in error will be in either the A, B, or P register. This interrupt can be set while in normal or control state. If this interrupt occurs on the processor that is designated as number 1, while it is operating in a control state, clock pulses to the processor will be inhibited and the processor will idle.
I02	ON	Invalid address. This interrupt is set as a result of the processor attempting to access a nonexistent core memory location; or while operating in normal state, when the C, S, or M registers are addressing the first 1000 (octal) words of a core memory area. The latter case is to protect the MCP area of memory from destruction by an object program.
I03	ON	Stack overflow. This interrupt may be set only while operating in normal state and indicates that the program stack has exceed its allocated size. Since the stack is located directly below the program reference table, this condition is sensed when the three high-order octades of the S register are equal to the R register.

The following interrupts may only be set while operating in normal state unless otherwise indicated by the syllable causing the interrupt.

I05	I06	I07	I08	
OFF	OFF	ON	OFF	Communication operator. This interrupt is set by a communication operator and is used to prove a means of entry to the MCP.
ON	OFF	ON	OFF	Program release. This interrupt is used to signal the MCP that an input/output area is free to be filled or emptied by an input/output operation.
OFF	ON	ON	OFF	Continuity bit. This interrupt is used where I/O descriptors indicate that multiple input/output areas are linked by multiple I/O descriptors. The I/O descriptors must be rotated and an I/O operation initiated.
ON	ON	ON	OFF	Presence bit. This indicates that the information associated with a descriptor is not present in core memory. The information must be brought into core memory before processing can continue.
OFF	OFF	OFF	ON	Flag bit. This interrupt indicates that an Operand Call syllable (OPDC) accessed a descriptor. This is an error condition.
ON	OFF	OFF	ON	Invalid index. This interrupt indicates that an index is larger than the size field of the descriptor or is negative. This is an error condition.

I05	I06	I07	I08	
OFF	ON	OFF	ON	Exponent underflow. This interrupt indicates that an arithmetic operation has resulted in an exponent which is less than -77 (octal). This is an error condition.
ON	ON	OFF	ON	Exponent overflow. This interrupt indicates that an arithmetic operation has resulted in an exponent which is greater than +77 (octal). This is an error condition.
OFF	OFF	ON	ON	Integer overflow. This interrupt indicates that a numeric value, which should be stored integer, cannot be made into an integer. This is an error condition.
ON	OFF	ON	ON	Divide by zero. This interrupt indicates that a divide operation was attempted with a divisor equal to 0 in the A register. This is an error condition.

Description of Interrupt Control

4-39. DETECTING AND PROCESSING. The interrupt control section of central control provides a central point for system interrupts. The interrupts are sensed and sent into a priority area where the interrupt address register (IAR) is set to a predetermined value for this particular interrupt. When processor 1 is in normal state and completes the execution of an operator, automatic interrupt detection takes place. If an interrupt exists, processor 1 will store its registers into control words and proceed to process the interrupt. When processor 1 interrogates the interrupt, the contents of the IAR are transferred to the C register. Processor 1 will then access memory to obtain an MCP routine for processing the interrupt. If several interrupts exist, as soon as processor 1 reads the present setting of the IAR, the IAR will be set to the address of the next priority interrupt. These interrupts are processed independently until all of them are processed, before processor 1 can return to normal state.

4-40. CATEGORIES. Interrupts are of three categories:

1. External interrupts - these include I/O units finished, time clock, etc.
2. Processor 1 interrupts - from processor 1 computing operations.
3. Processor 2 interrupts - from processor 2 computing operations.

4-41. PRIORITIES. Priority of interrupts are broken into 6 major categories, and are listed in priority sequence.

1. Processor 1 memory errors.
2. External interrupts.
3. Processor 1 stack errors.
4. Processor 1 syllable interrupts.
5. Processor 2, memory or stack errors.
6. Processor 2 syllable interrupts.

4-42. It is the duty of the interrupt control section to detect interrupts and to set address location number. Each address location number is set one at a time according to a priority scale and is stored into the interrupt address register.

4-43. Once an interrupt condition develops somewhere in the system, it transmits an interrupt signal to the interrupt control area. Each interrupt signal is generated by a flip flop. External interrupt flip flops are located in the central control frame. The processor interrupt flip flops are located in the processor unit, and are cabled to the interrupt area in central control.

4-44. EXTERNAL INTERRUPT FLIP FLOPS. There are 14 interrupt flip flops contained in the central control unit. These interrupts are set for various system interrupt conditions other than processor interrupts and are displayed on the central control display panel (figure 4-2). The external interrupt flip flops are:

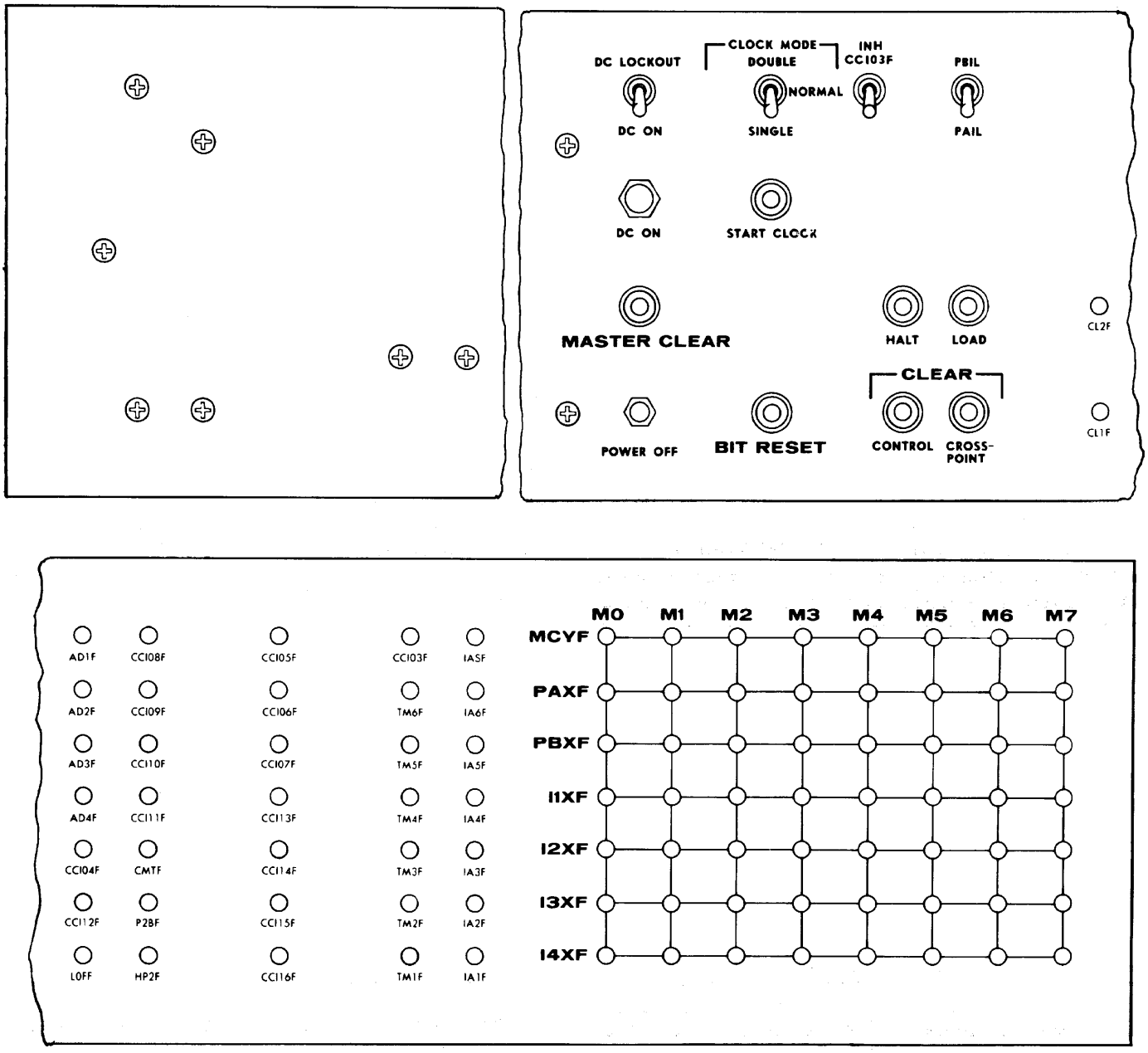


Figure 4-2. Central Control Display Panel

CCI03F ON Time Interval. This interrupt is set once each 64/60 (1.07) seconds when the real time clock counts through 63 to 0. When the interrupt is processed, the MCP will add the 64/60 seconds to the system elapsed time of each program that is in the MIX. (Bits TM1F through TM6F constitute the real time clock.) CCI03F is also used as the most significant bit of the real time clock.

CCI04F ON I/O busy. This interrupt is set when an input/output operation is attempted and all available I/O control units are busy.

CCI05F ON Keyboard request. This is set as a result of the system operator pressing the KEYBOARD REQUEST button on the keyboard. It notifies the MCP that the operator wants to communicate with the system.

As a result, the keyboard input will be accepted by the MCP.

CCI06F ON Printer 1 finished. This interrupt denotes that the line printer unit 1 has completed the print cycle. Since the printer contains a buffer for a complete line of print, an I/O control unit finished interrupt occurs as soon as the buffer is filled. When the print cycle is completed, a printer finished interrupt occurs to signal the MCP that the unit is free.

CCI07F ON Printer 2 finished. This interrupt is the same as printer 1 finished except that it is for print 2.

CCI08F ON I/O 1 finished. This interrupt notifies the MCP that an I/O operation has been completed (including storing the result descriptor in core memory) and that the I/O unit is free. I/O control unit 1 stores the result descriptor in cell 14 (octal).

CCI09F ON I/O 2 finished. This interrupt is the same as I/O 1 finished except that I/O control unit 2 stores the result descriptor in cell 15 (octal).

CCI10F ON I/O 3 finished. This interrupt is the same as I/O 1 finished except that I/O control unit 3 stores the result descriptor in cell 16 (octal).

CCI11F ON I/O 4 finished. This interrupt is the same as I/O 1 finished except that I/O control unit 4 stores the result descriptor in cell 17 (octal).

CCI12F ON Processor 2 busy. This interrupt is set when an attempt is made to initiate processor 2, and processor 2 is either busy or not available.

CCI13F ON Inquiry request. This interrupt is set for the inquiry interrupt level. It notifies the MCP that a remote inquiry request is desired.

CCI14F ON Special interrupt 1. Not assigned.

CCI15F ON Disk file read check finished-disk file control unit 1. This interrupt indicates that the initiated read check operation has now been completed.

CCI16F ON Disk file read check finished-disk file control unit 2. This interrupt is the same as disk file read check finished - disk file control unit 1 except that the control unit is 2.

4-45. INTERRUPT ADDRESS REGISTER. IA1F through IA6F forms the interrupt address register. When an interrogate interrupt operator is executed by processor 1 (in control state), the interrupt address register is set to a specific address, that of the highest priority interrupt. This address is placed in the C register of processor 1 and contains the syllables to branch to the handling routine of the MCP. The interrupt bit, whose generated address had been set into the interrupt address register, is reset when the address is transferred to the C register. In memory module 0, the addresses 20 through 75 (octal) are reserved for the words (interrupt control words) containing MCP branch operators to branch to the proper handling routine (figure 4-3).

4-46. REAL TIME CLOCK. The real time clock consists of 6 flip-flops forming a binary counter that is pulled up by one every 1/60th of a second. The flip-flops are labeled as TM1F to TM6F. When the counter overflows (counts past 63 to 0) the time interval interrupt (CCI03F) is set. CCI03F is used as the most significant digit of the real time clock.

4-47. HALT PROCESSOR 2 FLIP FLOP. When processor 1 executes a halt P2 operator while operating control state, halt processor 2 flip flop (HP2F) is set. Processor 2 will store its registers just as if a processor 2 dependent interrupt had occurred. Processor 2 will then idle. Processor 2 busy flip flops (P2BF) is on when processor 2 is in operation.

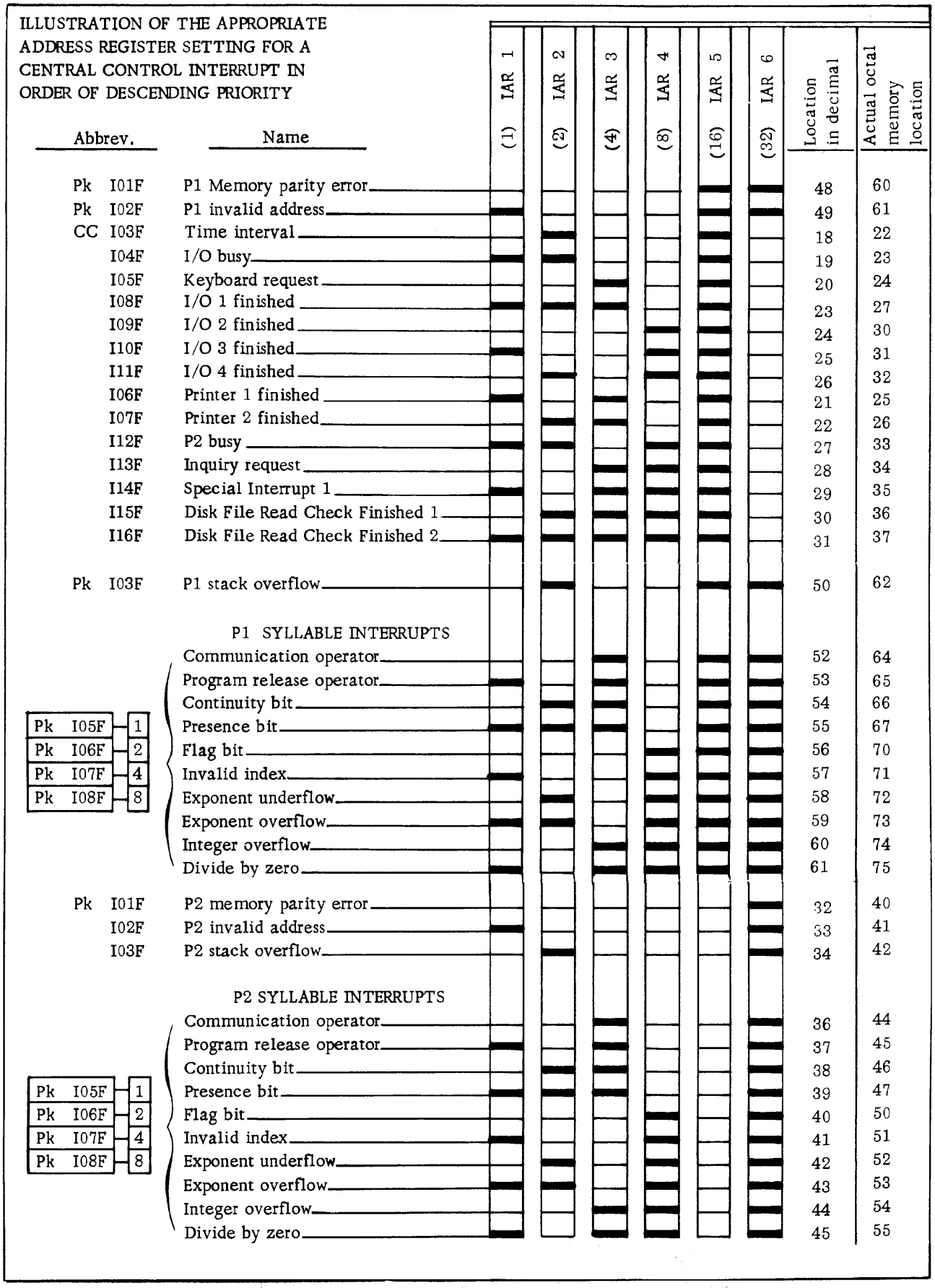


Figure 4-3. Interrupt Priority and Addressing

4-48. COMMENCE TIMING AND LOAD FLIP FLOPS. CMTF (commence timing flip flop) is used to initiate processor 2 or an I/O operation. LOFF (load flip flop) is set when the LOAD button on the console is pressed. LOFF is used to initiate the load operation which causes a portion of the MCP to be loaded into core memory. This loading starts initial operation after the system power has been down or after the HALT button on the console has been pressed.

CORE MEMORY MODULE REGISTER

4-49. Each core memory module of the B 5500 contains 4096 words of information. Each of these words contains 48 information bits and one parity bit. Odd parity is used and the detection of parity errors is accomplished in the individual memory modules. The generation of the odd parity is accomplished when the information is stored in the memory module. In order to address 4096 words, the memory address register (MAR) must contain 12 bits reflecting binary code. The information register in the core memory module is called the memory information register (MIR) and contains 49 bits.

I/O CONTROL UNIT REGISTERS AND FLIP FLOPS

Registers and Flip Flops

4-50. As mentioned previously, the I/O control unit contains a one word buffer that is used for input/output operations of the B 5500. Transfer of information between core memory and an I/O control unit is accomplished by a parallel transfer of all 48 bits of a word at one time. The information transferred to or from the peripheral unit is accomplished by a transfer of 6 bits (one character) at a time. Thus, the I/O control unit serves as a buffer between the two transfer areas.

4-51. W REGISTER. The W register in the I/O control unit is a 48-bit register which contains one word. The word is transferred to or received from the peripheral unit as 6-bit characters. On the display panel, the register is shown as being split in the middle. The high-order 24 bits are on top, and the low order 24 bits are on the bottom. This is shown in figure 4-4.

4-52. D REGISTER. The D register is used for control purposes. This is a 42-bit register whose bits are labeled as D01F through D45F. There are no flip flops for positions 23, 28 and 29. When an I/O descriptor is brought into the I/O control unit, it is first placed in the W register and then transferred to the D register. The D register has the same general format as the I/O descriptor. The bits that are missing from the D register are not used by the I/O control unit. The fields of the D register are as follows (the bit numbering corresponds to the etching on the I/O display panel):

Bits 45 through 41	Peripheral unit designation.
Bits 40 through 31	This is the word count field. If the operation is to read or write an integral number of words, then the word count field is used to terminate the transfer of information to or from core memory.
Bit 30	Memory inhibit bit. This bit is on for operations that do not require core memory communication for information.
Bit 27	Binary/Alpha bit. When bit 27 is on, the information is considered to be in binary form and an integral number of words are read or written. When bit 27 is off, the information is considered to be in alphanumeric form. The information may or may not be an integral number of words.
Bit 26	Magnetic tape direction bit. This bit is used in connection with magnetic tape operation and specifies the direction of tape movement. Off indicates forward motion and on indicates backward motion.

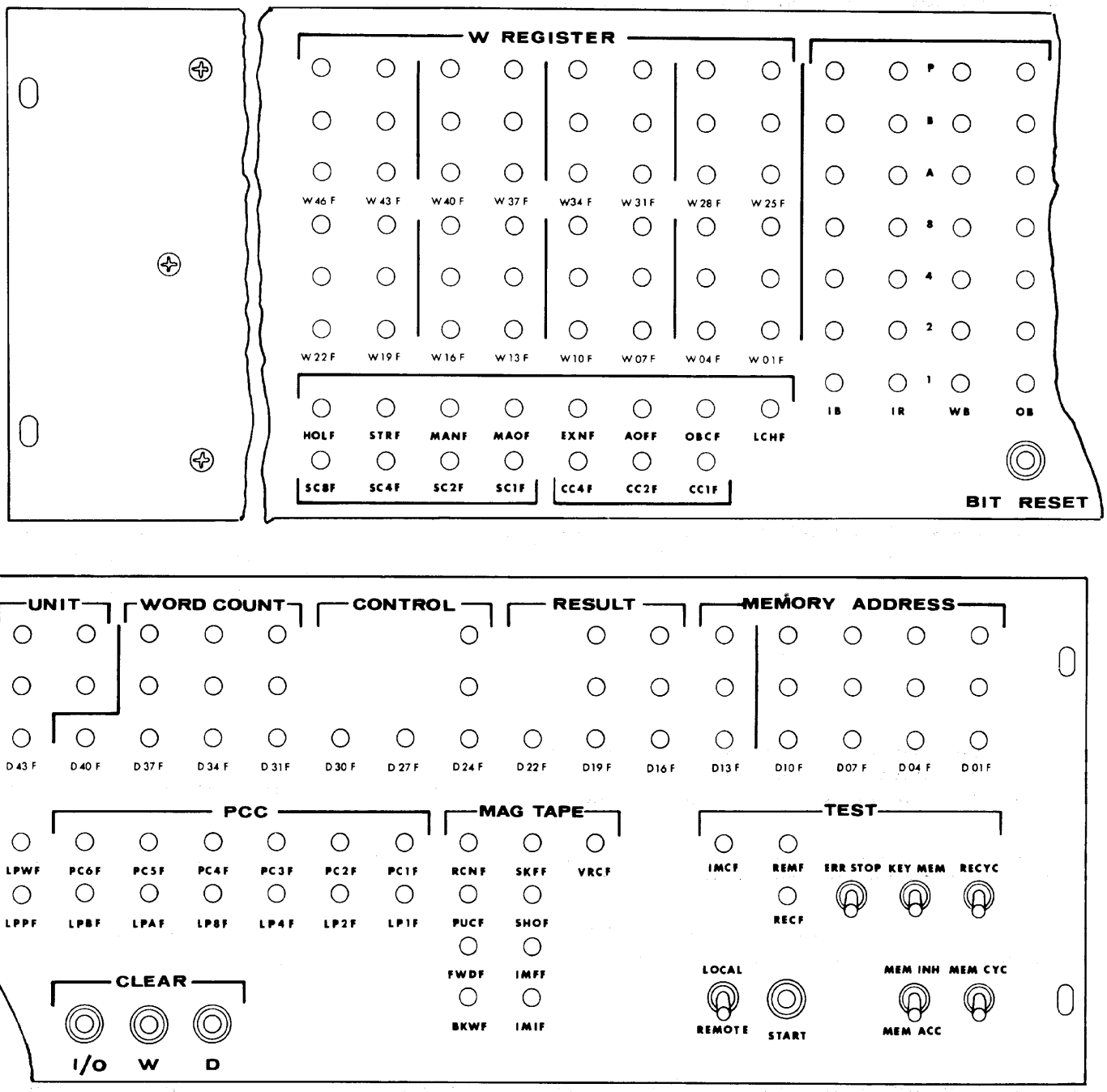


Figure 4-4. I/O Display Panel

Bit 25	Word counter bit. This bit is used for operations that may be either by word or by character. In these operations, if the operation is by word, then bit 25 is on; otherwise, it is off. It should be on if bit 27 is on.
Bit 24	Input/output bit. When this bit is on, the operation is an input operation, and when off, an output operation.
Bits 22 through 16	These bits are used for the error field of the result descriptor and as logical flip flops.
Bits 15 through 1	These bits contain the core memory address. Whenever a memory access is required, the address is shifted through the memory exchange in the central control unit to the memory module.

4-53. In addition to the W and D registers, the I/O control unit contains the following registers and flip flops.

4-54. CHARACTER COUNTER. The character counter, labeled CC, contains three flip flops which are CC1F, CC2F, and CC4F. Which character is to be read into or out of the W register is indicated by the character counter. Character 0 is the most significant character position of the W register.

4-55. INPUT BUFFER REGISTER. The input buffer register, IB, contains 7 bits. The bits are labeled as IB1F, IB2F, IB4F, IB8F, IBAF, IBBF, and IBPF. The IBPF is the parity bit. Information in the IB register is encoded and set into the W register character position specified by the character counter CC. The IB register is set from the tape information read register, or is read into from any input device other than magnetic tape.

4-56. TAPE INFORMATION READ BUFFER REGISTER. The tape information read register, IR, contains 7 bits labeled as IR1F through IRPF. One character plus a parity bit can be contained in the IR register. This register is set from information read from magnetic tape units. The output of the IR register is shifted to the IB register. The basic function of this register is to provide temporary storage for incoming tape information. This is necessary to insure proper timing.

4-57. OUTPUT BUFFER REGISTER. The output buffer register, OB, contains 6 bits labeled OB1F through OBBF. The register contains one character and the parity level is generated according to D27F. Information is placed in the output buffer register from the character position of the W register that is specified by the character counter. The output of the OB register is sent to peripheral units, other than magnetic tape units, or is shifted to the tape write buffer. Information is decoded between the W register and the OB register.

4-58. TAPE INFORMATION WRITE BUFFER REGISTER. The tape write buffer register, WB, contains 7 bits labeled WB1F through WBPF. The register holds one character plus a parity bit. Information is set into the WB register from the output buffer, and the output of the WB register is sent to the magnetic tape unit by way of the I/O exchange in central control. The function of this register is to provide for a temporary storage for tape information to insure proper timing.

4-59. LONGITUDINAL PARITY REGISTER. The longitudinal parity register, LP, contains 7 bits labeled LP1F through LPPF. This register contains one character plus a parity bit. When information is shifted from the input buffer to the W register, those bits of LP are used to generate a longitudinal parity character; it is also used for temporary storage.

4-60. SEQUENCE COUNTER. The sequence counter, SC, is used for primary logic control. Four bits are included in the register: SC8F, SC4F, SC2F, and SC1F. The sequence counter may be counted up in binary form or may be set to a particular count.

4-61. PULSE COUNTER. The pulse counter, PC, is used for timing purposes. This counter includes 6 bits labeled PC1F through PC6F. The pulse counter is counted at a one megacycle rate. A specific pulse count is then used for logical timing purposes.

4-62. LOGICAL CONTROL FLIP FLOPS. The rest of the flip flops of the I/O control unit are used for logical control of various operations. Grouping on the display panel is by major usage. Each indicator on the display panel is also a switch so that the flip flop can be manually set or reset. The display panel includes controls for maintenance purposes.

Information Flow

4-63. As previously stated, the W register is the information register used for memory communication in conjunction with the D register. The 15 low-order bits of the D register are used for memory addressing. Two words enter the D register. One word is the address of the I/O descriptor, and the other word is the I/O descriptor itself. These words are brought into the W register and then transferred to the D register (figure 4-5).

4-64. INPUT INFORMATION FLOW. When magnetic tape is used, information is read into the tape Information Read register. At the time the information is sent from IR to IB, the complete character should be in IR. All other peripheral units are read into the IB register. Information in the IB register is decoded and sent to the W register character position which is specified by the character counter. When a complete word is built up in the W register, the W register contents are stored in core memory and a new word is started.

4-65. OUTPUT INFORMATION FLOW. When information is read out of memory, it is sent to the W register. One character at a time is decoded and shifted from the W register to the output buffer register (OB). All peripheral units, except magnetic tape units, use the output of the OB register for the information source. In the case of the magnetic tape operation, the contents of the OB register are shifted to the tape write buffer (WB). The output of the WB register is used by magnetic tape for writing.

4-66. CARD READER INPUT. The card readers for the B 5500 have the ability to read cards which are punched in either normal card code or binary code. When the cards are in normal card code, the operation is the same as for any other peripheral unit which sends BCL code to the I/O control unit. The card readers contain the necessary logic for changing the normal card code to BCL code. When a card is coded in binary, each card column represents 12 bits. The 12 bits of a card column form 4 octades which are equivalent to two characters (figure 4-6). When a binary column is read, the bits punched in rows 12 through 3 of that column are placed in the IB register. These are the high-order bits of the column. Next, as the IB contents are shifted to the W register, the bits punched in rows 4 through 9 of the same column are placed in the IB register. The character counter is counted up by 1 and the low-order bits (4 through 9) of the card column are then sent into the W register from the IB register. In the case of binary coding, there is no encoding between IB and W registers. This sequence takes place for each column. With 80 columns per card, a card coded in binary will fill 24 octal words of core memory. The word with the lowest address will contain column 1 in the high-order 4 octades. The word with the highest address will contain column 80 in the four low-order octades.

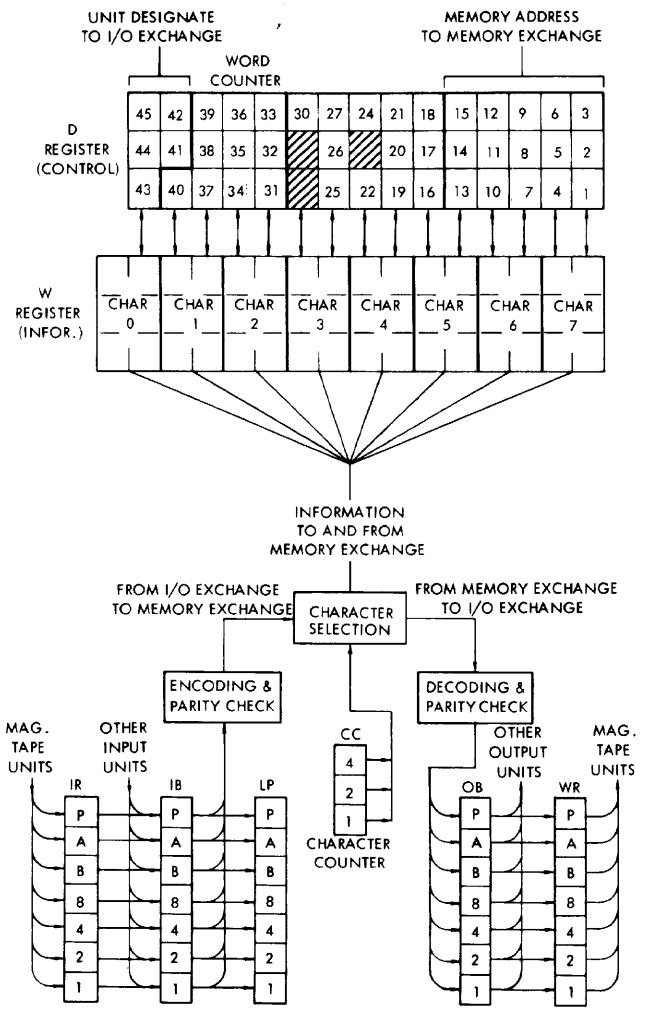


Figure 4-5. Basic I/O Control Unit Data Flow

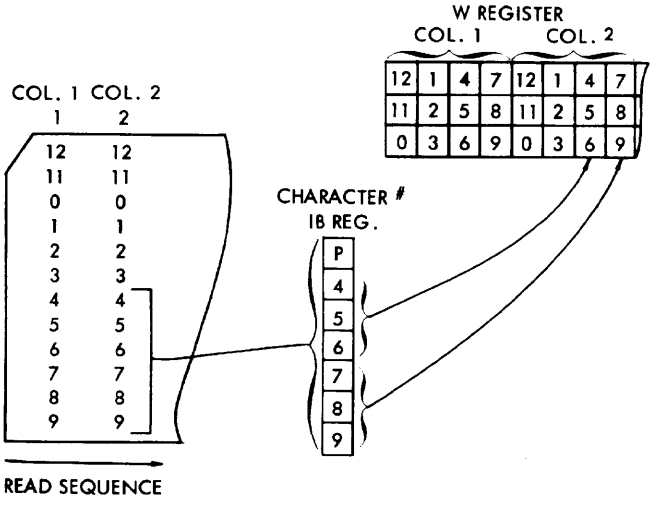


Figure 4-6. Binary Card Read

SECTION 5
WORD MODE OPERATION

GENERAL

5-1. This section describes the machine language program which is in the form of syllables and arranged in a program segment string.

SYLLABLE ADDRESSING AND SYLLABLE IDENTIFICATION

Syllable Addressing and Format

5-2. A machine language program is in the form of syllables which are arranged in a string (program segment string) and executed sequentially. These syllables are packed four to a core memory word (12 bits for each program syllable). The first syllable for a program word is formed from bits 0 through 11, and is labeled syllable 0 (figure 5-1).

SYLLABLE #0				SYLLABLE #1				SYLLABLE #2				SYLLABLE #3			
0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
2	5	8	11	14	17	20	23	26	29	32	35	38	41	44	47

Figure 5-1. Program Word in P Register

5-3. P AND T REGISTERS. The syllable being executed is contained in a 12-bit register named the T register. The T register is the control (instruction) register. The P register contains the 48 bit program word divided into 4 syllables. When the syllable in the T register is completely executed, the next syllable is shifted from the P register to the T register. When the last syllable of the word in the P register has been shifted to the T register, another program word is brought into the P register (figure 5-2).

5-4. L AND C REGISTERS. The 2 bit L register contains the number of the next syllable to be executed. This next syllable to be executed will be selected from one of four syllables currently within the program word within the P register. The core memory address of the word currently in the P register is kept in the 15 bit C register. When the last syllable of a program word in P has been shifted to the T register, the L register is overflowed from 3 to 0, and the C register is then counted up by one. The next program word is then brought from memory and placed in the P register (figure 5-3).

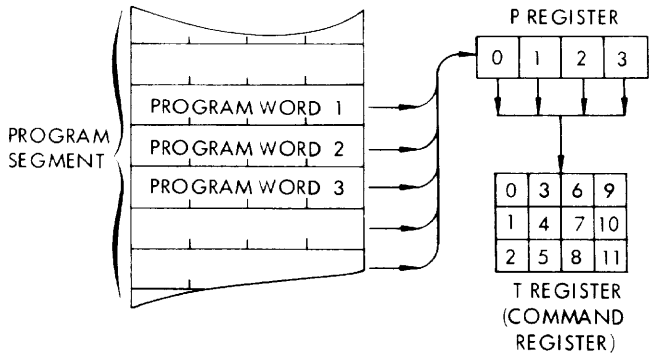


Figure 5-2. Program Syllable Sequence

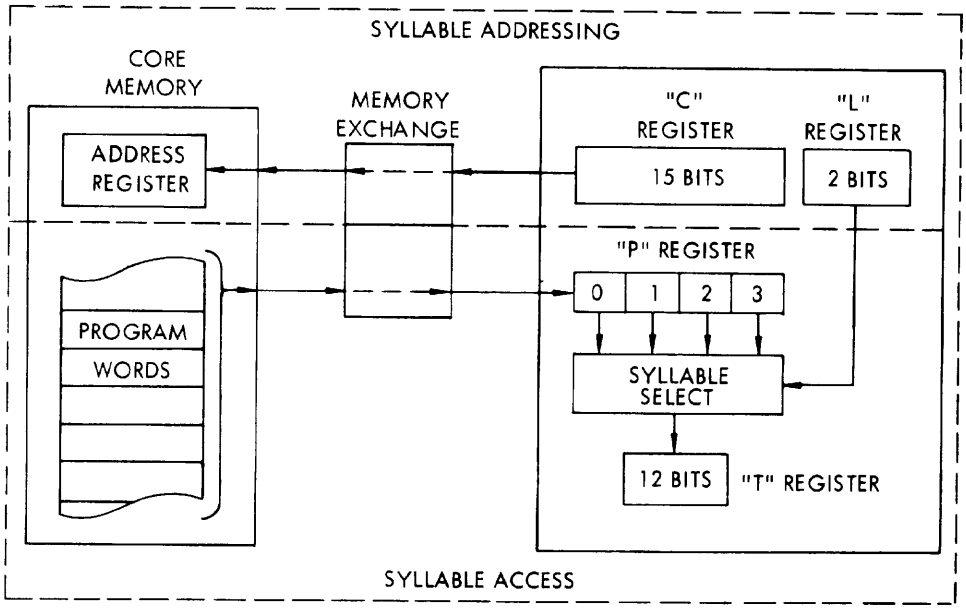


Figure 5-3. Syllable Access and Address

Word Mode Syllable Identification

5-5. SYLLABLE TYPE. In word mode, syllables are grouped into four categories: Descriptor Call, Operand Call, Literal Call, and Operator syllables. The format of the different types of syllables will be explained by relating to figure 5-4. The four types of syllables are distinguished by the two least significant bits of the syllable in the T register. The status of bits 11 and 10 and the syllable type are:

Bit 11	Bit 10	Syllable Type	Octal Value Ending In
off	off	Literal Call	0 or 4
on	off	Operator	1 or 5
off	on	Operand Call	2 or 6
on	on	Descriptor Call	3 or 7

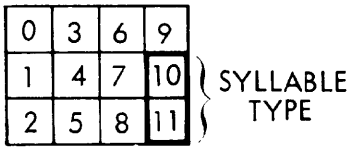


Figure 5-4. Word Mode Syllable Format

5-6. BITS 0-9. The Literal Call syllable (LITC) uses bit 0 through 9 to contain the integer value. Bit 9 in the T register, is the units bit of the integer value. The Operator syllables use bits 0 through 9 for determining the type of operator syllable and, for certain syllables, to contain such things as field length or other count. OPDC and DESC use bits 0 through 9 to contain the index for relative addressing[1] and to indicate the base address of the area which will be referenced. For example, consider the following syllable examples (SALF[2] = 0):

Octal Code In T Register	Type of Syllable and Value or Index
1232	OPDC: R+246
0253	DESC: R+52
0040	LITC: 10
0101	ADD (Add instruction, no decoding necessary)
0002	OPDC: R+0
0003	DESC: R+0

1. See paragraph 3-11.
2. Sub-program level flip flop.

As shown in the example, the value or index of the syllable is not readily apparent, except the Operator syllable type. However, the type of syllable is immediately known by observing the lower two bits of the T register.

5-7. CODING A SYLLABLE USING OCTAL MULTIPLICATION BY FOUR. This system may be employed by the machine language programmer to convert an octal number to a required syllable code. Multiplication by 4 will result in an answer that will be identical to that of bit shifting the octal number two places to the left, or in effect, multiplying it by 4. A faster method of multiplying the number is available than that of shifting two bits to the left. In order to employ this method, it is very helpful if the octal multiplication table of 4 is known. This table is as follows:

0 x 4 = 0
1 x 4 = 4
2 x 4 = 10
3 x 4 = 14
4 x 4 = 20
5 x 4 = 24
6 x 4 = 30
7 x 4 = 34

Note that the answers resulting from multiplication always end in 0 or 4. This may be a simple reminder to aid in rembering the table. Multiplication of the octal number 123 by the number 4 is shown in figure 5-5.

```
11          4 x 3 = 1 4
123
 x4         4 x 2 = 10 + 1 = 1 1
514
            4 x 1 = 4 + 1 = 0 5 - 0514
          "T" Register Code
```

Figure 5-5. Generating a Syllable Using Octal Multiplication by Four

Note that the multiplication is performed first, then the previous carry is added as in decimal multiplication. At the completion of the multiplication, a Literal syllable has been constructed having the value of 123. It is known to be a Literal syllable because the lower two bits equal 0. By adding 2 to the product of 4 x 123, an OPDC is constructed with an index of 123. By adding 3 to the product of 4 x 123, a DESC is constructed with an index of 123. This then, is a method which can be used to generate syllable codes for a desired value.

5-8. DECODING A SYLLABLE USING OCTAL DIVISION BY FOUR. Division by 4 may be used to determine the index or value of syllables other than operators. This can also be done by shifting the syllable two bits to the right. To divide the octal number 0514 by 4, use of the octal table of multiplication of 4 will be necessary. The division is illustrated in figure 5-6. The remainder of the division indicates the syllable type. In the example, the syllable type is the remainder 0 which indicates a Literal Call syllable. The syllable 0515 would generate a value of 0123 with a remainder of 1 to indicate an operator syllable type with the operator code 0123. The syllable 0516 would also generate a value of 0123 with a remainder of 2 to indicate an Operand Call syllable type, with an index of 123. The syllable 0517 would generate a value of 0123 with a remainder of 3 which indicates that this is a DESC with an index of 123.

```
Usable value = 0123          CODE IN
                             "T" REGISTER
          4   0514
              4
              11
              10
               14
               14
Syllable Type =  0
```

Figure 5-6. Decoding a Syllable Using Octal Division by Four

5-9. RELATIVE ADDRESSING. Relative addressing, when in program level (SALF = 0), always uses the base address contained in the R register. The index will be the 10 high-order bits of the T register or the 10 low-order bits of the A register depending upon the syllable being executed. The address contained within the R register is set into the 9 high-order bits of the M register and the index is set into the A register. The M and A registers are then added, and the sum is placed in the M register. The M register then contains the absolute address of the word being accessed. In program level, the index will always be positive with a maximum value of 1,023 (10 bits). When a processor is operating in sub-program level (SALF = 1), the number of bits available for an index is reduced because the high-order bits determine the sign and base address for relative addressing. The MSFF and the three high-order bits of the index field (T0, T1, and T2, or A38, A39, and A40) together determine the base to use and the sign of the index. The bits not required to determine the sign and the base may be used for the index value. This is given below and shown in table 5-1. In the following description, the SALF is on. The T register bits are used in the explanation; however, the same statements are valid for corresponding bits of the A register. The Relative Addressing Table, table 5-1, is valid for any syllable that specifies relative addressing capabilities, unless limiting specifications are outlined within the specific syllable.

TABLE 5-1

Relative Addressing Table

SALF	T0 A38	T1 A39	T2 A40	MSFF	Base	Index Sign	Index Bits	Maximum Index In Decimal/Octal
OFF	-	-	-	-	R	+	T(00 => 9) A(38 =>47)	1,023/1777
ON	OFF	-	-	-	R	+	T(02 => 9) A(39 =>47)	511/777
ON	ON	OFF	-	OFF	F	+	T(02 => 9) A(40 =>47)	255/377
ON	ON	OFF	-	ON	(R+7) *	+	T(02 => 9) A(40 =>47)	255/377
ON	ON	ON	OFF	-	C **	+	T(03 => 9) A(41 =>47)	127/177
ON	ON	ON	ON	OFF	F	-	T(03 => 9) A(41 =>47)	127/177
ON	ON	ON	ON	ON	(R+7) *	-	T(03 => 9) A(41 =>47)	127/177

\- Irrelevant (part of the index).

* Relative addressing using as the base, bits 18 thru 32 of the word stored in the program's PRT at R+7.

** "C" relative coding is forced to "R" relative for the store, program and I/O release operators.

5-10. T0 OFF - (A38 OFF). With T0 in the off state, the R register contains the base address. The index is positive and contained in bits T1 through T9 of the T register. The maximum index is 511.

5-11. T0 ON, T1 OFF - (A38 ON, A39 OFF). With T0 and T1 in the given conditions, the index will be contained in bits T2 through T9, and is always positive. The maximum index is 255. The base will depend on the state of the MSFF. If MSFF is off, then the F register contents will be used as the base address. If MSFF is on, then an F register would have been stored in bits 18 through 32 of the mark stack control word (MSCW) located in R + 7 of the program reference table (PRT). In this case, a preliminary memory access using relative addressing would use the base address contained in the R register. The index of 7 must be used, so that the M register can be set to the contents of the 8th word of the PRT. The latter relative address is known as "(R+7)+."

5-12. T0 ON, T1 ON, T2 OFF - (A38 ON, A39 ON, A40 OFF). With these conditions, the index is positive and is comprised of bits T3 through T9. The base address will be in either the C register or the R register, depending on the operator contained in the T register. The maximum index is 127.

5-13. T0 ON, T1 ON, T2 ON - (A38 ON, A39 ON, A40 ON). With these conditions, the index is negative and is comprised of bits T3 through T9. Again, the maximum index is 127. With the MSFF in the off state, the base address will be contents of the F register, and is known as "F-." With MSFF in the on state, the base address will be the contents of the F register that were stored in R + 7. This is known as "(R+7)F-".

5-14. When decoding a syllable that is using relative address code (SALF is on), do not include the relative address code bits as part of the syllable value. The inverse of this is also true when coding syllables.

5-15. NORMAL WORD MODE ADDRESSING. In word mode operation, the S register is used to address the top valid word of the core portion of the stack. This is always true in word mode. Any other memory accessing for information uses the address contained in the M register. The S register is used for all stack adjustment; and, with few exceptions, the information register used is the B register. Thus, the B register contents are stored in the address specified by the S register; or the B register is filled from the location address by the S register. Occasionally, the A register is filled from the location addressed by the S register. When the M register is used as the address register for accessing information, there are three methods of setting an address into the M register. In the first method, the low-order 15 bits of the A register contain an absolute address such as is found in a data descriptor. In this case, the address is shifted from the A register to the M register, and the access is performed. The specific operator determines which information register, A or B, is used. In the second method, the M register is set to the 15 bits (18 through 32) of an accessed word. Here, the M register is used to address the access word, and must have been originally set by one of the other two methods. The third method is used when relative addressing is required. In the third method, a base address is used with an index. The index is in the 10 high-order bits of the T register, or is in the 10 low-order bits of the A register. The current condition, along with a program syllable, determines which bits will be used as a base address. This base address will be one of the following:

1. Base address of the program reference table. This address is in the R register.

2. Address of the return control word in the stack. This address is in the F register.

3. The F register setting stored in the MSCW. The MSCW is the eight word of the program reference table (R+7). The F register setting will be in bits 18 through 32. To access this MSCW by relative addressing, an index of 7 is added to the base address of the program reference table. The M register contains the new address and addresses the cell (word). When the word is accessed, the M register receives bits 18 through 32.

4. Current word address of the program segment string. This address will be the contents of the C register. This is a special case and is not normally used.

REFERENCING A WORD WITH THE OPERAND/DESCRIPTOR CALL SYLLABLE

5-16. There are three types of words (operands, descriptors, and program descriptors) that may be found in the PRT or stack. Two syllables reference these areas and place data in the stack or remove it from the stack. These are the Operand Call and Descriptor Call syllables. Figures 5-7 and 5-8 show block diagrams for the resulting access operations.

5-17. OPERAND CALL SYLLABLE. If this syllable accesses a word from either the PRT or the stack, the following actions can occur:

1. If an operand is accessed, the operand is placed in the stack.
2. If a control word is accessed, the control word will be placed in the top of the stack, and treated as an operand.
3. If a data descriptor is accessed, the word addressed by the descriptor will be placed in the stack. If this word is not an operand, a "flag-bit" interrupt is produced.
4. If a program descriptor is accessed, the program will branch to a sub-routine within the program. In this case a special word, the return control word, will be placed in the stack. (Accessing a special program descriptor causes it to be placed in the top of the stack and no branch is performed.)

5-18. DESCRIPTOR CALL SYLLABLE. When this syllable references one of the four types, the following action can occur:

1. Referencing an operand will cause a data descriptor to be generated that contains the absolute address of the operand. This data descriptor will then be placed in the stack as the top word.
2. Referencing a data descriptor by this syllable will cause the data descriptor to be placed in the stack.

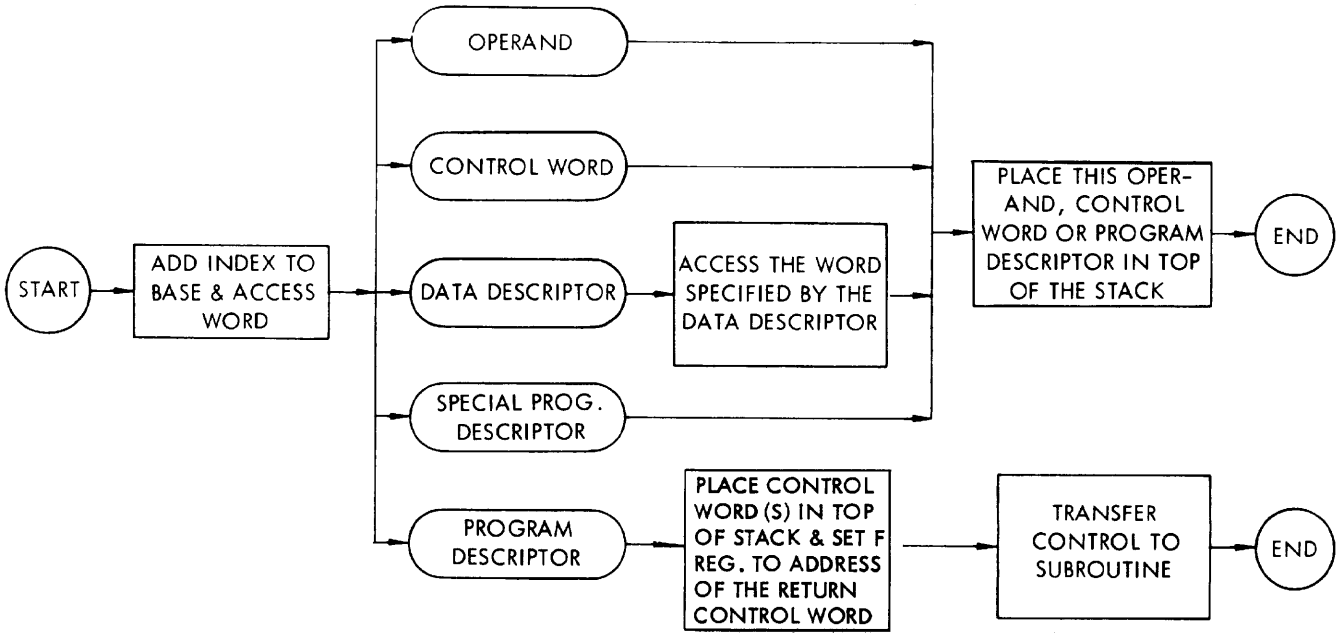


Figure 5-7. General Flow for Operand Call Syllable

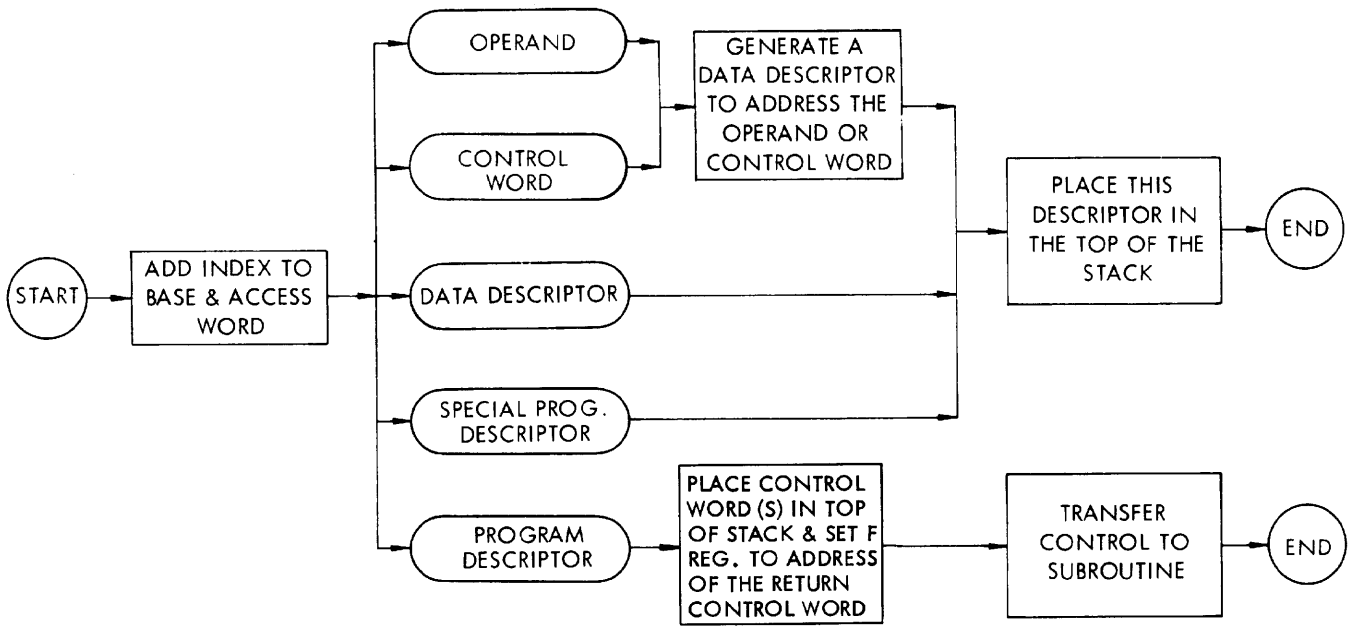


Figure 5-8. General Flow for Descriptor Call Syllable

3. Referencing a program descriptor will cause the same operation as though an OPDC accessed a program descriptor. A RCW will be generated and placed in the stack and the program will branch to the subroutine.

5-19. The primary function of the OPDC is to place an operand in the stack. The primary function of the DESC is to place an address in the stack. The exception to this rule is when either syllable references a program descriptor. This will cause the program to branch to the subroutine indicated by the program descriptor and also cause subsequent generation of a RCW and the placement of this generated word in the top of the stack.

5-20. DATA DESCRIPTORS. Descriptors have a function very similar to the implication of the name: they describe some data, a program, or an operation. All descriptors are used in word mode. There are four types of descriptors: data descriptors, program descriptors, I/O descriptors, and I/O result descriptors. The I/O result descriptors are generated at the completion of an I/O operation and reflect the results of the operation. The I/O descriptors and their application are discussed in section 9. All of the descriptors except the I/O result descriptor, have the flag bit (bit 0) on to mark them as descriptors. The remaining bits of each type of descriptors are as follows:

Bit 0	On marks this word as being a control word or a descriptor.
Bit 1	Off marks this word as a data descriptor.
Bit 2	Used to reflect the availability of the associated information, or space for the information. This is the presence bit. When bit 2 is on, the information is available; or, if information is to be stored, the space is available.
Bits 3 - 7	Not used. (Off)
Bits 8 - 17	This is the word count field and will be used when the data descriptor is associated with a group of words. When the word count field is equal to 0, the descriptor is associated with only one word.
Bit 18	Not used. (Off)

Bit 19 This is the integer bit, and is on if the data descriptor is associated with an operand that has been declared as an integer.

Bit 20 This is the continuity bit, and is on if there is more than one data descriptor that references multiple areas for the same type of information, such as multiple I/O buffers.

Bits 21-32 Not used. (Off)

Bits 33-47 These bits contain the absolute address of the information referenced. This absolute address is filled when the information has been allocated space in core memory.

5-21. APPLICATION OF DATA DESCRIPTORS. Data areas are referenced indirectly through data descriptors (see figure 5-9) located in the PRT. Each descriptor references a unique area in memory. If an area consists of a single word, the data descriptor contains the address of that word. However, if an area consists of data arranged in serial sequence (a single dimensional array), the data descriptor contains the base address of the array. This base address is incremented by an index to obtain the desired word from the array. The word count field within the data descriptor is not equal to 0 when referencing an array. Multiple indexing is required for an array which has more than one dimension. In this phase, the descriptor in the PRT has a word count field equal to the size of the first dimension. The first absolute address (called the mother vector) is the base address of a table of descriptors each called a dope vector. The number of descriptors in the table is dependent on the first dimension. Each of these descriptors has a word count field equal to the size of the second dimension. The absolute address of each of these descriptors in the first table is the base of another table. The second table will contain the actual data elements of the array (if the array was declared as a two dimensional array). If the array was declared as a three dimensional array, then the second table would have had descriptors in it, and these descriptors would have pointed to the actual data elements of the array. See figure 5-10.

NOTE

Any number of dimensions are allowed in the B 5500.

Subroutines

5-22. A subroutine can be defined as a series of operations that are repeated many times during the execution of a program. The subroutine is entered by transferring control from the main body of the program to the subroutine. Upon completion of the subroutine, control is returned to the syllable following the syllable that caused entry to the subroutine. The structure of the B 5500 programing language extends the usefulness of this type of program organization. Procedure declarations in ALGOL, and sections and paragraphs in COBOL, are considered to be subroutines. Therefore, the subroutine mode of operation in the B 5500 has been designed to assure efficient generalized handling of subroutines to any depth, including recursively defined subroutines (a subroutine which calls itself from within the subroutine.)

5-23. SALF. Although the subroutine simplifies the programing task, it creates new problems of stack control, accessing data for the subroutine, etc. To alleviate some of these problems, each processor contains a flip flop called SALF. When a subroutine has been entered, the SALF is set to indicate that at least one subroutine has been entered.

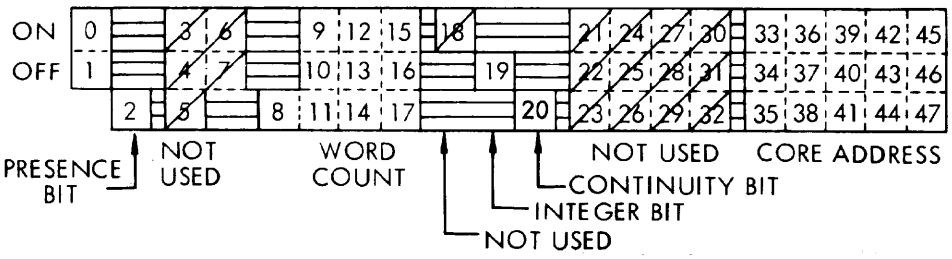


Figure 5-9. Data Descriptor Exploded

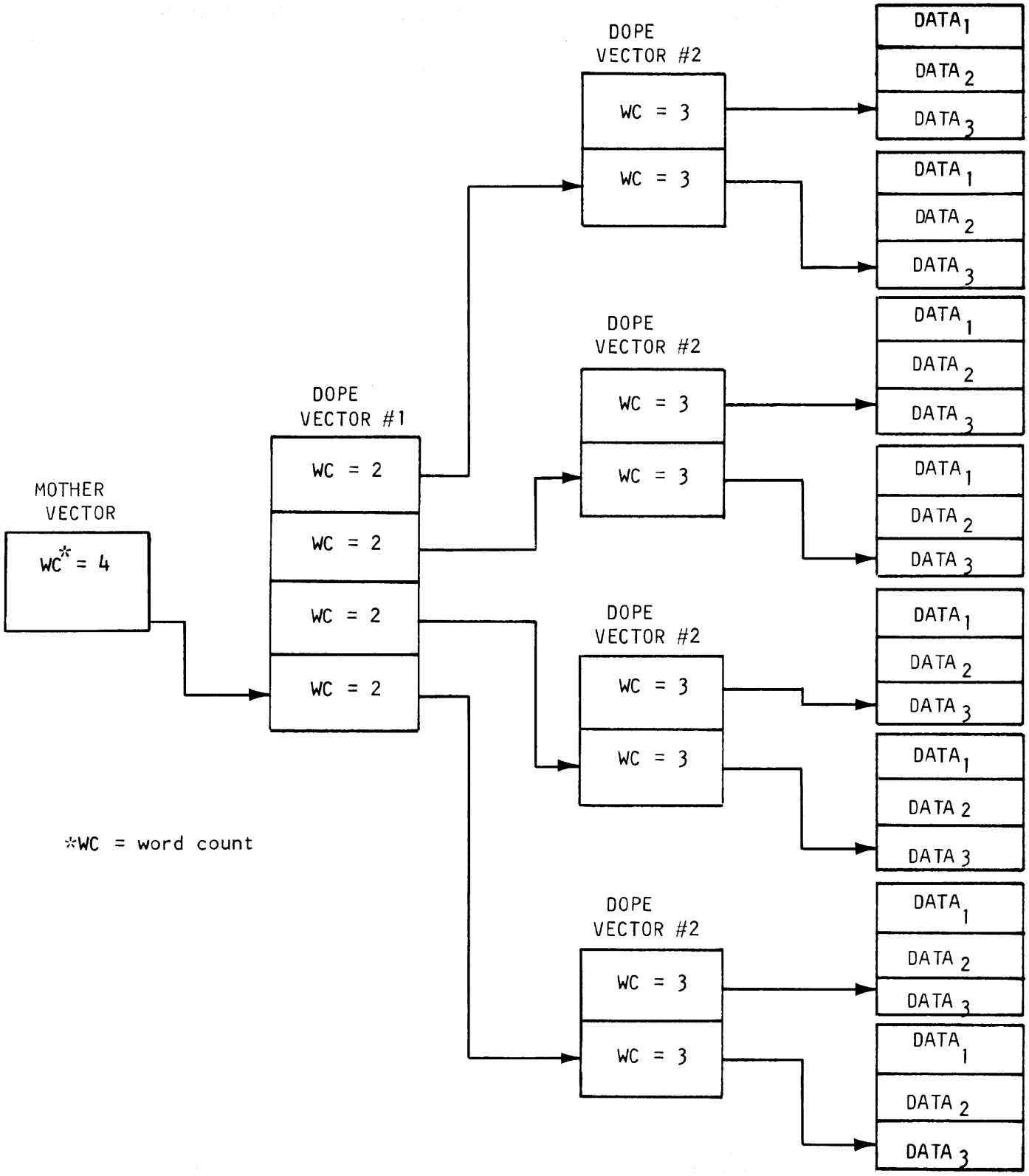


Figure 5-10. Example of 3-Dimensional Array (0:3, 0:1, 0:2)

With SALF in the on status, the processor is said to be operating in sub-level. The functional purpose of this flip flop is to allow additional relative addressing capability. For example, if SALF is off, only R register relative addressing can be used; however, if SALF is on, then R, F, C, and R+7 relative addressing can be used.

5-24. SUBROUTINE ENTRY AND EXIT. Normally, the actual parameters required by each subroutine are stored in, or referenced through, the Program Reference Table. Before the subroutine is entered, these parameters are brought to the stack. Control is then transfered to the subroutine, which can access the stack to use the parameters. However, before any parameter for the subroutine is placed in the stack, the top of the stack must be marked. This indicates the top of the stack before entry to the subroutine. This is accomplished with a Mark Stack Control Word (MSCW). This word contains the value of certain registers last used by the prior program. The F register is set to the address of the MSCW when its current setting is placed in the stack. The parameters for the subroutine are then placed in the stack and a RCW is placed in the stack, on top of the parameters, just before entering the subroutine. The F register is set to the address of the RCW when its contents are placed in the stack. Each control word contains the address of the previous control word which was addressed by the F register. The RCW contains the value of the rest of the registers that are necessary to allow control to return to the program at the point following the syllable that caused a branch to the subroutine. Return is accomplished automatically by executing a return normal or exit operator when the subroutine is completed. When exit from a subroutine is executed, the RCW is accessed and the registers are restored from the contents of this control word. The F register is set to the address of the previous control word, which is the MSCW. Access of the MSCW is then made, and the rest of the registers are restored. This restoration will allow the program, from which entry to the subroutine was made, to continue processing.

5-25. SPECIAL SUBROUTINE ENTRY AND EXIT. A special case arises when a subroutine (procedure) is to be entered that requires no actual parameters, yet performs work on data that is global to itself; but this data is not in, or accessible through, the PRT. This situation could best be illustrated by examining a procedure within a procedure. When this occurs rF does not point at the RCW at the top of the stack, but at a previous RCW lower in the stack. When this special procedure is to be exited, rS is pointing at the RCW at the top of the stack, therefore, a "return special" operator is executed.

5-26. MARK STACK CONTROL WORD (MSCW) DESCRIPTION. In most cases, the actual parameters needed by the subroutine are placed in the stack between a MSCW and a RCW. In these cases, the MSCW is generated and placed in the stack by a Mark Stack operator. The execution of the Mark Stack operator causes the contents, if any, of the A and B registers to be pushed into the core portion of the stack. The MSCW is then constructed and stored to mark the top of the stack. The setting of the R register, the state of MSFF, SALF, and the setting of the F register are contained in the MSCW. The F register is set to the address of the word in which the MSCW is stored. If the MSFF was off, and the processor is in sub-program level (SALF = 1), at the time the MSCW is constructed, the constructed MSCW is also stored in the cell address by R+7. The MSFF is then set to 1.

5-27. MARK STACK CONTROL WORD FORMAT. The format of the MSCW (see figure 5-11) is as follows:

Bit 0	On marks this word as being a control word or a descriptor.
Bit 1	On marks this word as being either a control word or a program descriptor.
Bit 2	Not used. (Off)

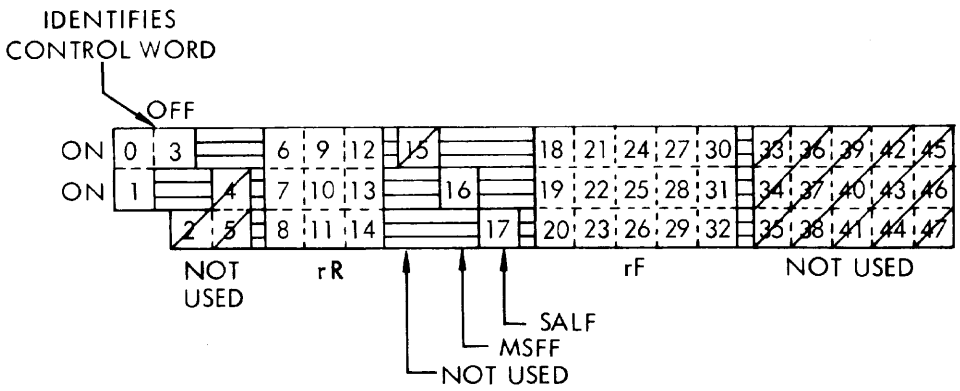


Figure 5-11. Mark Stack Control Word Exploded

Bit 3 — Off marks this word as a control word as opposed to a program descriptor.

Bits 4 - 5 — Not used. (Off)

Bits 6 - 14 — These bits contain the contents of the R register just prior to entering the subroutine. This is the base address of the program reference table.

Bit 15 — Not used. (Off)

Bit 16 — This bit represents the state of the MSFF just prior to the time the MSCW is generated. This bit is turned on if the MSFF is on.

Bit 17 — This bit represents the state of the SALF just prior to the time the MCSW is generated. This bit is turned on if the SALF is on.

Bits 18 - 32 — These bits are set to the contents of the F register when the MSCW is generated. This F register setting will be the address of a previous control word that has been placed in the stack, or 0

Bits 33 - 47 — Not used and will always be equal to 0.

5-28. PROGRAM DESCRIPTOR DESCRIPTION. A subroutine is entered by a program descriptor (often called a "procedure descriptor"). Program descriptors are generated during compilation and are placed in the program reference table. When entry into a subroutine is desired, an OPDC or DESC accesses the program descriptor. This causes a RCW to be placed in the stack and then the subroutine to be entered. One of the bits of the program descriptor is a presence bit. This bit is used to specify if the subroutine program segment is in core memory. If the desired program segment is not in core memory, then the program must be interrupted until the segment can be brought into core memory by the MCP. The interrupt generated by this bit is called the presence bit interrupt. Subroutines can be used in either word mode or character mode operations. A bit, called the mode bit is used to provide this information. The mode bit is sensed when the subroutine is entered, and the subroutines will operate in the mode specified by the mode bit. The program descriptor contains a core address field. The core address field contains the absolute address of the program word containing the first syllable of the subroutine. The subroutine will start with the first syllable of that program word. One bit of the program descriptor is used as an argument bit. This bit is on for every subroutine that requires actual parameters to be stored in the stack between the MSCW and the RCW. The program descriptor contains a field to store a setting of the F register. If the argument bit is off, the contents of the F register will be replaced by the contents of the F register field in the program descriptor when the subroutine is entered. This is called "accidental entry".

The new address contained in the F register is a prior setting of the F register, and will be the base for relative addressing used in the subroutine.

5-29. PROGRAM DESCRIPTOR FORMAT. The format of the program descriptor (see figure 5-12) is as follows:

Bit 0	On marks this word as being a control word or a descriptor.
Bit 1	On marks this word as a control word or program descriptor.
Bit 2	This is the presence bit and will be on if the program segment is available in core memory.
Bit 3	On marks this word as a program descriptor, as opposed to being a control word.
Bit 4	This is the mode bit. If on, the subroutine is to operate in character mode. If off, the subroutine is to operate in word mode.
Bit 5	This is the argument bit. It will be on if actual parameters are to be placed between the MSCW and the RCW.
Bits 6 - 17	Not used.
Bits 18-32	This field contains the F register setting that will be used during the operation of the subroutine if the argument bit (bit 5) is off.
Bits 33 - 47	This is the address of the program word that contains the first syllable of the subroutine. This address is filled in by the MCP when the subroutine program segment is brought into core memory.

5-30. RETURN CONTROL WORD (RCW) DESCRIPTION. The RCW is placed in the stack when an operand descriptor call is executed on a program descriptor. It is the last entry made in the stack before entering the subroutine.

5-31. RETURN CONTROL WORD FORMAT. The format of the RCW (see figure 5-13) is as follows:

Bit 0	On marks this word as not being an operand.
Bit 1	On marks this word as being either a control word or a program descriptor.
Bit 2	When off, this return control word was placed in the stack as a result of executing an OPDC. When on, this RCW was placed in the stack as a result of executing DESC.
Bit 3	Off marks this word as a control word, as opposed to a program descriptor.
Bits 4 - 6	These bits contain the contents of the H register at the time the subroutine is entered.

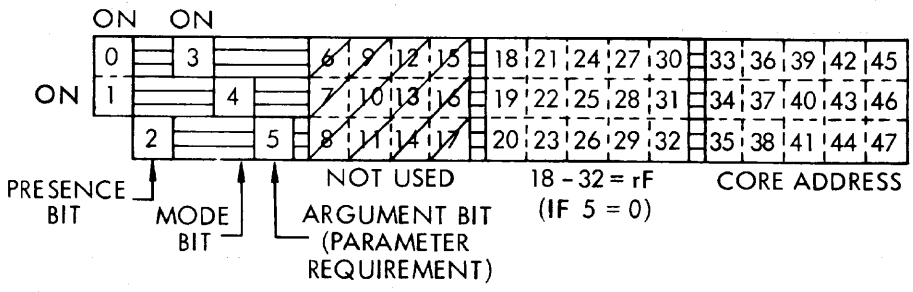


Figure 5-12. Program Descriptor Exploded

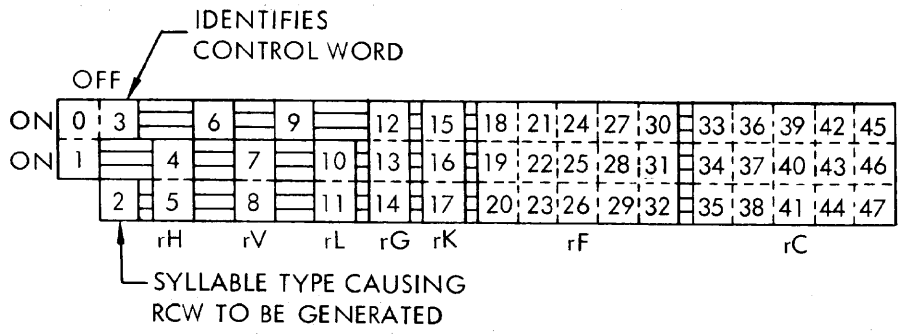


Figure 5-13. Return Control Word Exploded

Bits 7 - 9 These bits contain the contents of the V register at the time the subroutine was entered.

Bits 10 - 11 These bits contain the contents of the L register. These contents are the address of the syllable which follows the OPDC/DESC that caused the subroutine to be entered.

Bits 12 - 14 These bits contain the contents of the G register when the subroutine was entered.

Bits 15 - 17 These bits contain the contents of the K register when the subroutine was entered.

Bits 18 - 32 These bits contain the contents of the F register when the subroutine was entered. This will be the address of a previous control word.

Bits 33 - 47 These bits contain the contents of the C register. These contents are the address of the program word that contains the syllable indicated by bits 10 and 11 of this word.

STATE AND MODE

State

5-32. CONTROL STATE. The B 5500 can operate in either control state or normal state. Routines of the MCP are executed mainly while in control state, whereas object programs are executed mainly in normal state. In control state, automatic interrupt detection is suppressed and some machine operators are activated that are treated as no-ops while in normal state. The suppression of automatic interrupt detection in control state allows the MCP to complete its particular function without further interrupts. At the conclusion of any particular function, the MCP can interrogate for any other interrupt and handle them individually according to priority.

5-33. NORMAL STATE. In normal state, automatic interrupt detection occurs at the end of each syllable execution. This automatic interrupt detection has priority. Since the interrupt causes a transfer to control state, this is the only means of entering the MCP from an object program. A normal-control state flip flop (NCSF) in the processor indicates the state of the operation, whether normal or control. In a two processor system, only one processor can operate in control state. Processor 1 is the control state processor. Either processor can be designated as processor 1 by means of a physical switch located on the display and distribution panel. Processor 2 can generate interrupts, but each interrupt must be handled by processor 1.

Mode

5-34. Each processor of the B 5500 can operate in either of two modes, word mode or character mode. In word mode, the operators primarily function with one or more entire words. In character mode, the operators primarily function with individual characters or parts of a character. The functions of the various registers are differ-

ent in the two modes and each mode has an independent set of operators. Each processor contains a flip flop to control the mode of operation. This flip flop is the character-word mode flip flop (CWMF). It is on for character mode operation, and off for word mode operation. Although word mode operations may occur at either the program or sub-program level, any character mode operation is at the sub-program level.

5-35. WORD MODE. Using the stack concept as a basis, when a program is operating in word mode there must be several areas of core memory reserved for various parts of the program. The areas include the stack, the program reference table, the program segment string, and the I/O areas. Information is brought into the input area. This information will then be brought into the stack from the program reference table and from the program reference string. After processing, the results are stored in the output area. The results are later transferred to the output unit. Thus, word mode uses four basic areas of core memory.

5-36. CHARACTER MODE. In character mode, the processing is concerned with individual characters or parts of characters. Therefore, the primary concern is the two areas of memory; the source string area and the destination string area. Both of these strings may be thought of as strings of characters or character positions. The function of character mode operations is to take characters of information from the source string, process them, and transfer them to the destination string. Actually, processing in character mode is much more versatile than the basic concepts just presented. It is possible to skip characters in either string, compare fields of the two strings which are of equal length, add two fields (one in each string), place the result in the destination string, etc. Unless programed otherwise, characters are always handled sequentially from the high-order character to the low-order character. A program operating in character mode will reserve basic areas in core memory for the destination string, source string, stack, program reference table, program segment strings, and I/O areas. The destination string and source string areas may be reserved in addition to, or as part of the last four areas mentioned.

5-37. Any combination of mode and level that occurs in normal state may occur in control state. A processor in normal state may be in either sub-program or program level of operation. Program level occurs only in word mode. Sub-program level may occur in either word mode or character mode (figure 5-14). Since control state operation is not considered part of an object program, an object program is processed in normal state and may be in either program or sub-program level. The sub-program level operation can be in either word mode or character mode.

PROCESSOR INITIATION

5-38. To initiate a processor in normal state, the Initiate P1 or Initiate P2 syllable must be executed. Which syllable is executed depends on which processor is to be placed in normal state. A processor will be designated as #1 by the designate switch that is located on the display and distribution unit. Consider the initiation of this processor. Through the use of this switch, either processor A or processor B may be designated as processor 1 and the other processor automatically becomes processor 2. The Initiate syllable is only active if encountered while the processor is operating in control state. Therefore, processor 1 may only be initiated by itself. Processor 2 may never initiate processor 1, because processor 2 cannot operate in control state. If the Initiate syllable is encountered in normal state, it is treated as a no-op.

INTERRUPT OCCURRING WHILE IN NORMAL STATE (NCSF=1)

5-39. The Interrupt Address Register (IAR) in central control is set by the presence of an interrupt in the system. This includes the interrupts of processor 2, all external interrupts, and the interrupt register of processor 1. When an interrupt is present in the system, processor 1 must go into control state in order to handle the interrupt (assuming it is in the normal state when the interrupt occurs). If it is already in the control state, the interrupt is "captured" in the IAR and, therefore, will simply wait until processor 1 executes an Interrogate Interrupt (ITI) operator. Should processor 1 be in normal state when any interrupt occurs, a Store For Interrupt

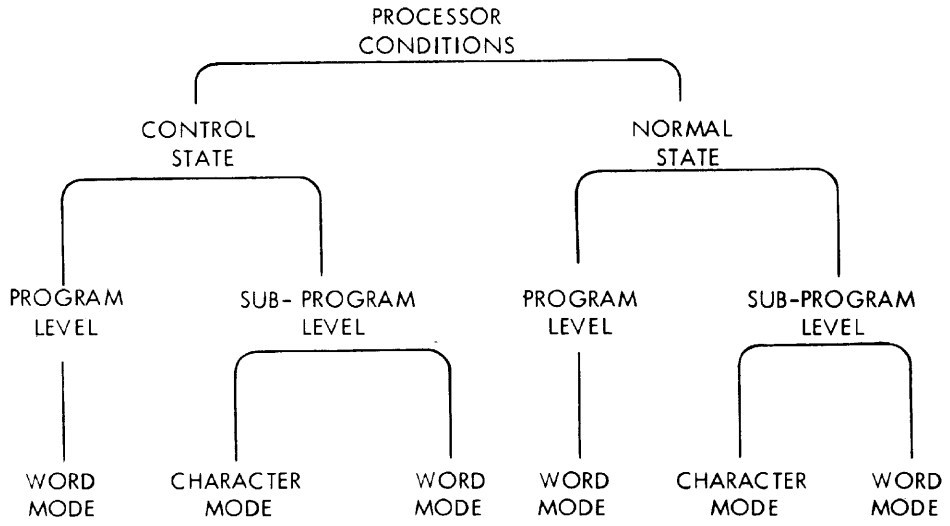


Figure 5-14. Permissible Combinations of State, Level and Mode

(SFI) is forced into its T register at the completion of the current operator, and immediately executed as the next syllable.

5-40. "Control words" are produced by a SFI operator to record the settings of all necessary registers. This is done so that after the interrupt has been handled, processing may continue at the exact point where it was interrupted. All of the control words except one are stored in the top of the stack. That one control word contains the absolute address of the top of the stack. The processor contains some 18 different registers and the contents of each are placed into the control words as they are produced and placed in the stack. The order that the control words are placed in the stack depends upon the mode when the interrupt occurs. See figures 5-15 and 5-16 for the order and contents of the control words.

5-41. WORD MODE. If the processor was operating in word mode when the interrupt was detected, the contents of the B register, if valid, are stored in the stack. The contents of the A register, if valid, are stored in the stack. The interrupt control word (ICW) and interrupt return control word (IRCW) are then generated and stored in the stack. An initiate control word (INCW) is generated and stored in R + 10 (octal).

5-42. CHARACTER MODE. If the processor was operating in character mode when the interrupt was detected, the contents of the A register, if valid, are stored in the stack. The contents of the B register, if valid, are then stored in the stack. The contents of the X register are then transferred to the stack, forming an interrupt loop control word. The ICW and IRCW are then generated and placed in the stack. An INCW is generated and stored in R+10 (octal) of the program that was being executed when the interrupt occurred.

5-43. When generation of the new INCW is completed, the NCSF is reset. The R register is cleared to 0.

5-44. PROCESSOR 1. An Interrogate Interrupt syllable is forced into the T register. The function of this syllable is to interrogate the IAR register in central control for an interrupt, set the contents of IAR into the C register, and set the S register to the value octal 00100. The processor then branches to the address contained in the C register.

5-45. PROCESSOR 2. Only processor 2 interrogates its own interrupt register for the presence of an interrupt. This interrogation occurs after the execution of each syllable. If a processor 2 interrupt is present, processor 2 will execute a Store For Interrupt syllable in the same manner as does processor 1. However, at the termination of the Store For Interrupt syllable, when NCSF is

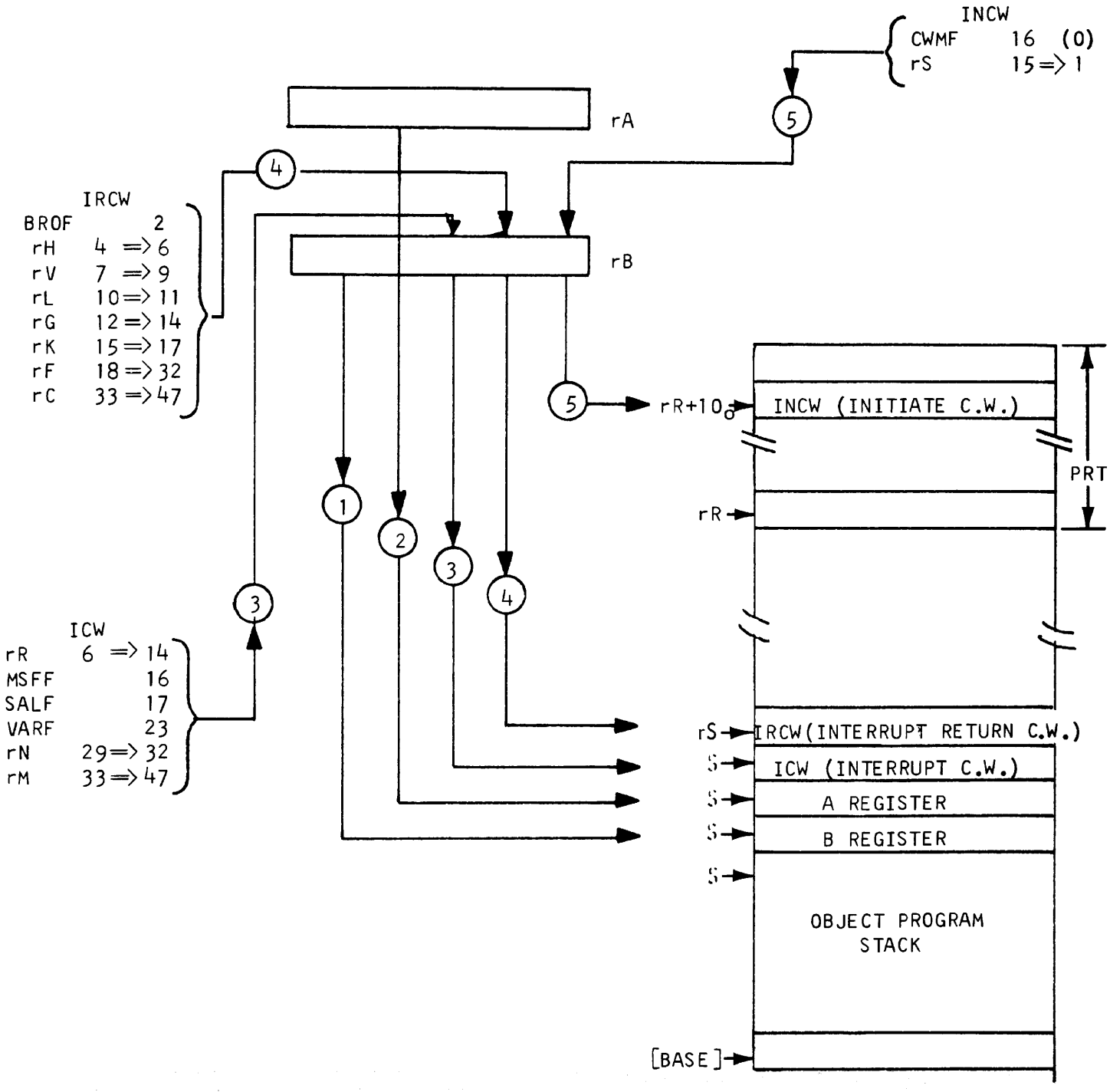


Figure 5-15. Store for Interrupt Word (Word Mode)

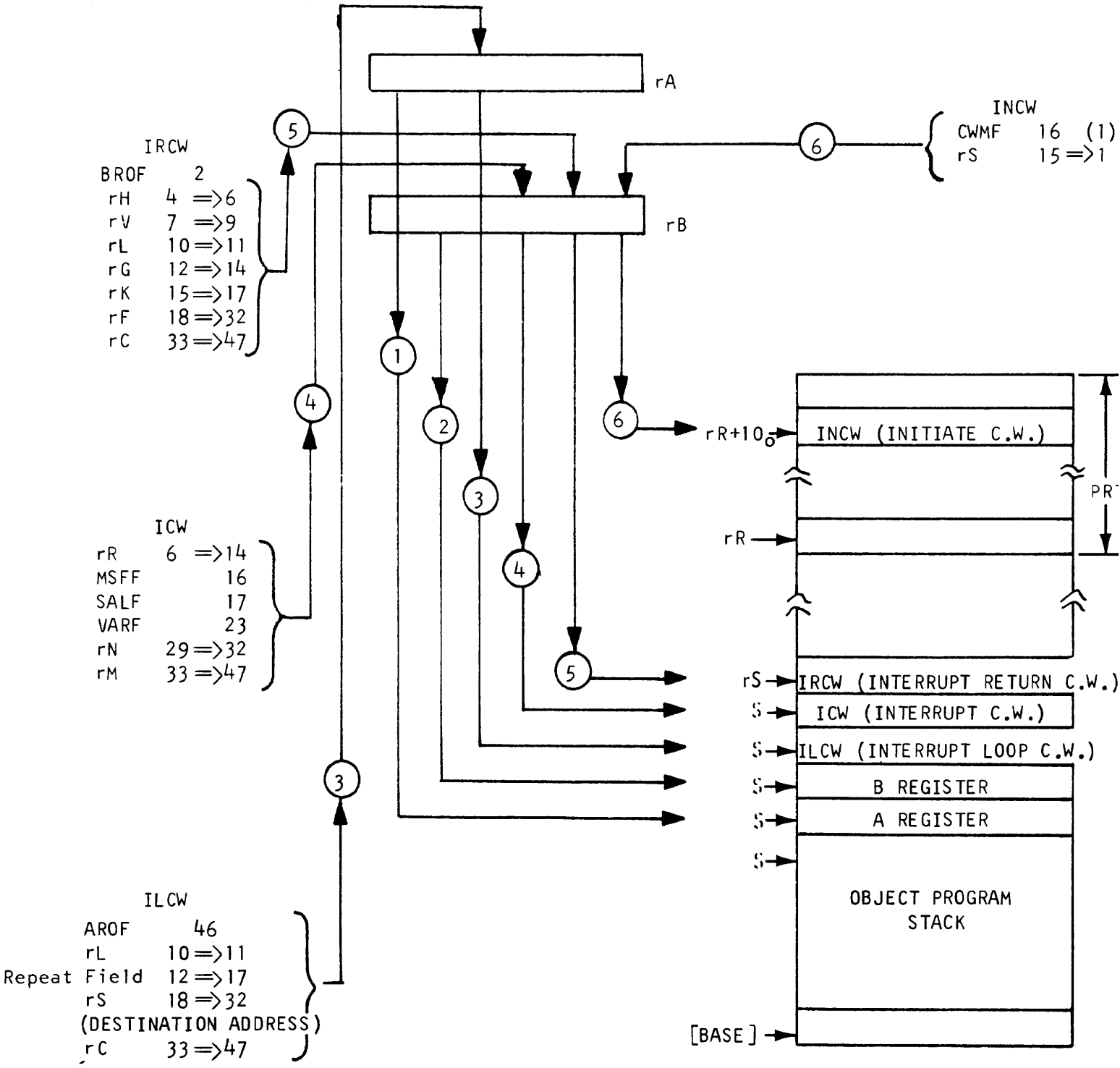


Figure 5-16. Store for Interrupt Word (Character Mode)

reset, an Interrogate Interrupt syllable is not forced in the T register. Instead, processor 2 is placed in an idle state by resetting TROF and PROF. The processor will then remain in this state until it is reinitiated by processor 1. Processor 1 reinitiates processor 2 by an Initiate P2 syllable.

5-46. INTERRUPT CONTROL WORD FORMAT. The format and significance of the field of the ICW (see figure 5-17) are as follows:

Bit 0	On marks this word as being an operand.
Bit 1	On marks this word as being a control word or a program descriptor.
Bit 2	Not used.
Bit 3	Off marks this word as a control word as opposed to a program descriptor.
Bits 4 - 5	Not used.
Bits 6 - 14	These bits contain the contents of the R register. If the interrupt occurred while in word mode, this will be the base address of the PRT. If the interrupt occurred while in character mode, this will be the "tally" register value.
Bit 15	Not used.
Bit 16	This bit reflects the state of the MSFF when this word is generated. If the MSFF is on, then bit 16 is on.
Bit 17	This bit reflects the state of the sub-program level flip flop at the time this word is generated. If operation is in a subroutine level (SALF on), then bit 17 is on.
Bits 18 - 28	Not used.
Bits 29 - 32	Used to store the count contained in the N register.
Bits 33 - 47	Used to store the contents of the M register. If the interrupt occurred while in character mode, this will be the current source string word address. If this interrupt occurred while in word mode, the filed will be 0.

5-47. INTERRUPT RETURN CONTROL WORD FORMAT. The format of the IRCW (see figure 5-18) is as follows:

Bit 0	On marks this word as not being an operand.
Bit 1	On marks this word as being a control word or program descriptor.

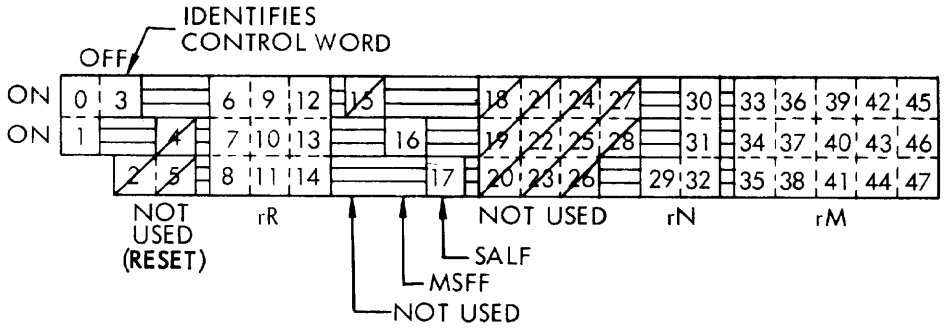


Figure 5-17. Interrupt Control Word Exploded

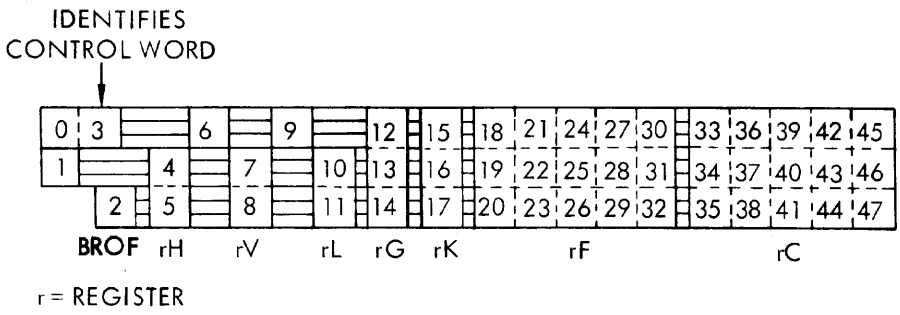


Figure 5-18. Interrupt Return Control Word Exploded

Bit 2 — This bit indicates that the B register may have contained a valid word. This is significant if the interrupt occurred while operating in character mode. In this situation, the B register will contain a word of the destination string that has been partially processed and must be stored to retain the information. When the program is restarted, this situation is recognized, the B register is filled again from the stack, and operation continues in a normal manner.

Bit 3 — Off marks this word as control word as opposed to a program descriptor.

Bits 4 - 6 — Used to store the contents of the H register which is the bit pointer associated with the A register.

Bits 7 - 9 — Used to store the contents of the V register which is the bit pointer associated with the B register.

Bits 10 - 11 — Used to store the contents of the L register. This will normally be the syllable number of the next syllable to be executed when control is returned to this portion of the program.

Bits 12 - 14 — Used to store the contents of the G register. This is the character pointer associated with the A register.

Bits 15 - 17 — Used to store the contents of the K register. This is the character pointer associated with the B register.

Bits 18 - 32 — Used to store the contents of F register.

Bits 33 - 47 — These bits store the contents of the C register. This is the address of the program word containing the next syllable to be executed when control is returned to this portion of the program.

5-48. INITIATE CONTROL WORD (INCW) DESCRIPTION. All of the control words that have been discussed are stored in the stack after being formed. (See figures 5-15 and 5-16.) The INCW is stored in the 10th octal cell (word) of the Program Reference Table for the particular program. It is used when initiating the program. The main purpose of the INCW is to contain the address of the top word in the stack of the program to be reinitiated. The top word will be the IRCW. Thus, when a processor is initiated, the processor accesses the INCW. In initiating processor 2, the MCP has placed the INCW in cell 10 (octal) of core memory and processor 1 executes an Initiate P2 syllable. In initiating processor 1, the MCP placed the INCW in the top of its "personal" stack. From the INCW, the processor obtains the address of the top word of the stack of the program to be initiated. From this point, the processor can restore registers so that the program can be continued. Since the mode of operation is also stored, when the program is reinitiated, adjustments can be made for the different conditions that exist during character mode operation.

5-49. INITIATE CONTROL WORD FORMAT. The format of the INCW is as follows:

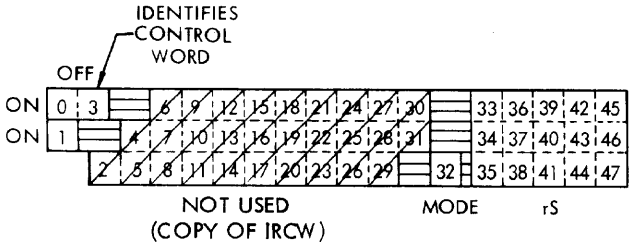


Figure 5-19. Initiate Control Word Exploded

Bit 0 — On marks this word as being a control word or a descriptor.

Bit 1 — On marks this word as being a control word or a program descriptor.

Bit 2 — Not used.

Bit 3 — Off marks this word as a control word as opposed to a program descriptor.

Bits 4 - 31 — No used.

Bit 32 — Used to store the mode of operation at the time of interrupt.

Bit 32 would be on if operation has been in character mode (CWMF on).

Bits 33 - 47 — This is the address of the IRCW. At the time the INCW was formed, this address was the S register.

Initiating an Input/Output Operation

5-50. GENERAL. When a program requires an I/O operation, the MCP must execute an initiate I/O syllable. This syllable may only be initiated by processor 1. The initiate I/O syllable will store the address of the I/O descriptor (I/O instruction) in cell 10 (octal) of memory module 0. The processor will then transmit an "initiate I/O" signal to central control. The MCP is then free to continue processing. When central control receives the initiate signal, it begins the operation on the lowest numbered I/O control unit that is not busy. Central control has then completed its function. The I/O control unit that was initiated will obtain the I/O descriptor using the address stored in cell 10. Using this descriptor, it will initiate the desired I/O operation. When the I/O operation is completed, the I/O control unit will set its respective "I/O control unit finished interrupt" flip flop.

5-51. DETAIL. When an I/O operation is required, an initiate I/O syllable must be executed. However, before this syllable is executed, the address of the desired I/O descriptor must be in the top of the stack. For example, suppose a card reader I/O descriptor has been stored in cell 142 (octal). A card reader operation is required; therefore, 142 must be placed in the top of the stack before the initiate I/O syllable is executed. The initiate I/O syllable stores the top word of the stack into cell 10 of memory module 0. When the store operation is completed, the processor sends a signal to central control to indicate that an I/O operation is requested. This signal will set the commence timing flip flop in the central control unit. This flip flop will then release the processor and initiate the operation using the lowest numbered I/O control unit that is available and not busy. I/O control selection priority is as follows:

I/O control unit 1 - first

I/O control unit 2 - second

I/O control unit 3 - third

I/O control unit 4 - fourth

Each I/O control unit has a corresponding admit descriptor flip flop (AD1F through AD4F) in the central control unit. When the I/O control unit is selected and initiated, the corresponding admit descriptor flip flop is set. This flip flop indicates that the corresponding I/O control unit is currently busy. The I/O control unit will then access the word in cell 10 which contains the address of the I/O descriptor. Using this address, the I/O control unit obtains the desired I/O descriptor. In the example, the card read I/O descriptor would be accessed from cell 142.

After the I/O descriptor has been transferred to the I/O control unit, the I/O operation can be initiated. When the I/O control unit has completed the I/O operation, the respective admit descriptor flip flop is reset. However, the I/O control unit finished interrupt flip flop is set. Setting this flip flop accomplishes two functions. First, while the flip flop remains set, the I/O control unit is marked busy; second, the presence of this flip flop being set is transferred to the interrupt section of central control. This will set the value of this interrupt in the Interrupt Address Register in central control (providing that there are no interrupts present on the system with a higher priority). The I/O finished interrupt flip flop will remain set until the interrupt is handled by the processor, through the initiation of an "interrogate interrupt" syllable (0211). Therefore, the I/O control unit will be marked as busy at the time the unit is initiated and continued to be marked as busy until the corresponding I/O finished interrupt flip flop is reset.

Load Operation

5-52. The LOAD switch on the console initiates the load operation. This is used to load, initially, a "loader" routine into core memory. This is the only way an operator has of initiating the system after power is first turned on or after the HALT button on the console has been pushed. The "loader" will call in portions of the MCP that are needed to start operation. The source of the "loader" may be from either the drum memory or the card reader. This is selected by the LOAD SELECT switch on the console. If the card reader is selected, one binary card is read into core memory. The load operation is controlled by the central control unit which forces an I/O descriptor into an I/O control unit in order to read from the drum or from the card reader into core memory. Memory location 20 is the first address into which information is read. After the input operation is completed without error, the central control unit sends an initiating signal to the processors. Processor 2 ignores the signal, while processor 1 responds as follows: The C register is set to 20 (octal) and a fetch operation is performed. The initial word of the program read into core memory is transferred from memory location 20 into the P register. When the operation is terminated, control of the processor continues with the execution of the first syllable of the program word in the P register. Operation is in the control state with all the registers cleared except the C, P, and T registers.

SECTION 6

WORD MODE SYLLABLES AND OPERATORS

GENERAL

6-1. In this section each operator or syllable description will be preceded by the OSIL/ESPOL mnemonic, name, engineering mnemonic, and octal form.

6-2. In word mode, there are four types of syllables identified by the two low-order bits of the syllable. The four types of syllables and their identification are:

Next Low Order Bit	Low Order Bit	Syllable Type	Octade Equivalent
0	0	Literal Call	0 or 4
0	1	Operator	1 or 5
1	0	Operand Call	2 or 6
1	1	Descriptor Call	3 or 7

In character mode, there is only one type of syllable which is an operator.

LITC LITERAL CALL SYLLABLE (LTSL) XXX0 or XXX4

6-3. The purpose of this syllable is to place a positive value between 0 and 1023 decimal (1777 octal) on the top of the stack. This is accomplished in the following manner. The ten high-order bits of the T register (which contains the syllable) are placed in the low-order 10 bits of the A register (bits 38 => 47). All other bit positions of the A register are cleared to zero. Prior to the transfer of the ten bits from T to A, the stack is pushed down if the contents of the A register are marked valid.

OPDC OPERAND CALL SYLLABLE (OCSL) XXX2 or XXX6

6-4. The objective of the Operand Call syllable is to bring an operand into the stack. Since operands may be stored in the program reference table, or in an area specified by a descriptor which is in the PRT, the PRT will be referenced most of the time through "R" relative addressing. It is possible that operands might also be in the stack as a result of previous operations. In this case, the stack must be referenced and an image of the operand brought to the top of the stack where it can be worked upon. This usually is accomplished with "F-" relative addressing. The program segment string may also be referenced through "C" relative addressing. The first action of the Operand Call syllable is to reference the PRT, stack, or program segment string, through relative addressing that is determined from:

1. the status of the SALF
2. the ten high order bits of the syllable
3. the status of the MSFF.

The designated word is then placed in the A register. The relative addressing scheme is shown in table 6-1. Prior to the execution of the syllable, if the A register is marked valid, it will be pushed down in the stack. For a flow chart of the following discussion refer to figure 6-1. Once the stack, PRT, or program segment string has been accessed, the following actions are taken according to the type of word brought to the A register.

Operand or Control Word (Operand = Bit 0 Off; Control Word = Bits 0, 1 On; 2 Off)

6-5. If the word accessed is an operand or control word, the operation is terminated as

soon as the word is placed in the A register. The A register is marked as valid.

Data Descriptor (Bit 0 On)

6-6. PRESENCE BIT OFF (BIT 2 OFF). This indicates that the area referenced by this descriptor is not present in core memory. If the processor is in normal state, the presence bit interrupt is set. The A register is marked as valid and the syllable terminated. If the processor is in control state, the A register is also marked as valid and the syllable terminated.

TABLE 6-1

Relative Addressing Table

This table is valid for any syllable that specifies relative addressing capabilities, unless limiting specifications are outlined within the specific syllable.

SALF	T0 A38	T1 A39	T2 A40	MSFF	BASE	Index Sign	Index Bits	Addressable Area Size In Decimal
OFF	-	-	-	-	R	+	T(0 => 9) A(38 => 47)	(1,024)
ON	OFF	-	-	-	R	+	T(1 => 9) A(39 => 47)	(512)
ON	ON	OFF	-	OFF	F	+	T(2 => 9) A(40 => 47)	(256)
ON	ON	OFF	-	ON	(R+7)*	+	T(2 => 9) A(40 => 47)	(256)
ON	ON	ON	OFF	-	C **	+	T(3 => 9) A(41 => 47)	(128)
ON	ON	ON	ON	OFF	F	-	T(3 => 9) A(41 => 47)	(128)
ON	ON	ON	ON	ON	(R+7)*	-	T(3 => 9) A(41 => 47)	(128)

\- Irrelevant (part of the index).

* Relative addressing using as the base, bits 18 thru 32 of the word stored in the programs PRT at R+7.

** "C" relative coding is forced to "R" relative for the store, program and I/O release operators.

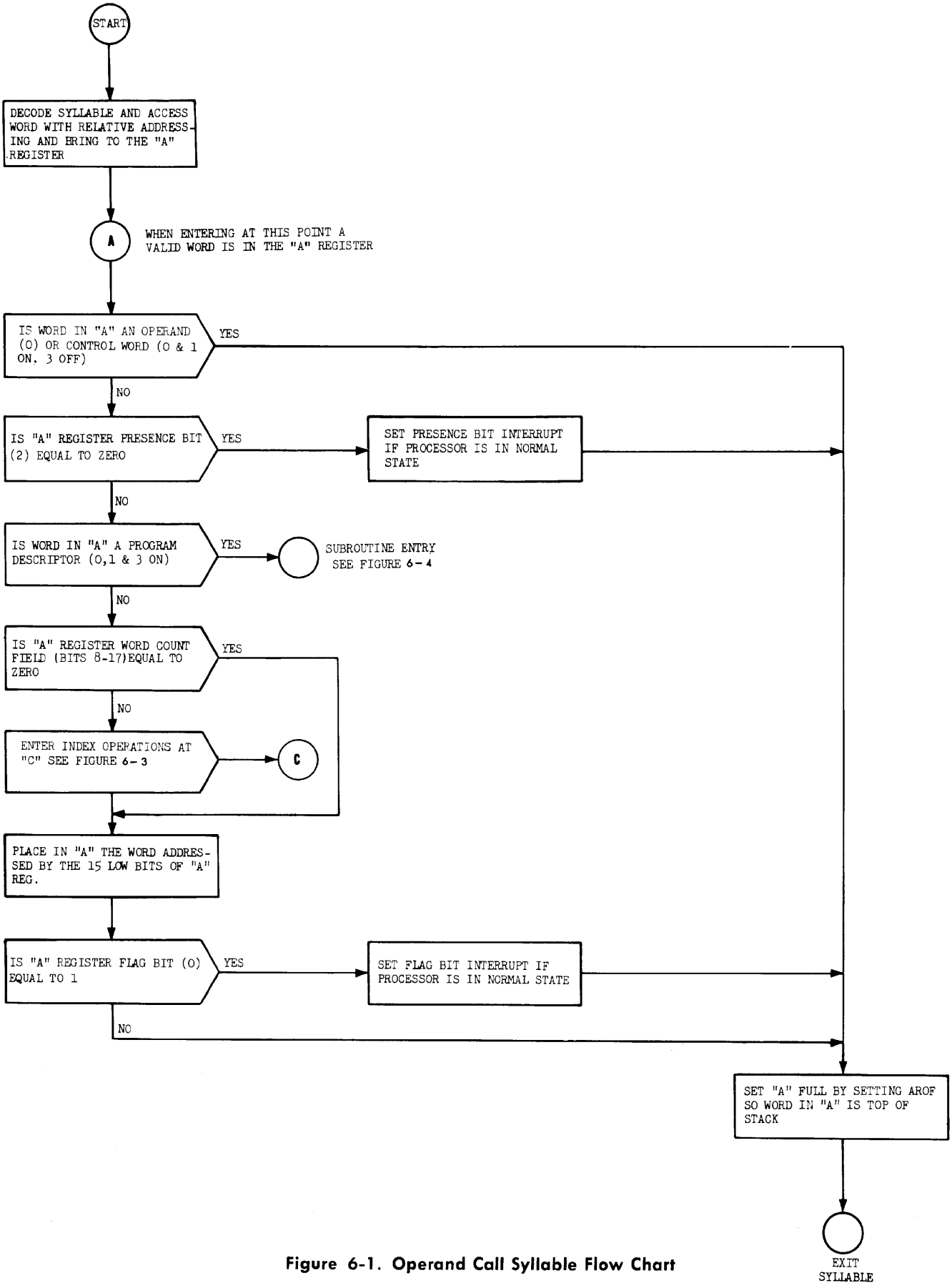


Figure 6-1. Operand Call Syllable Flow Chart

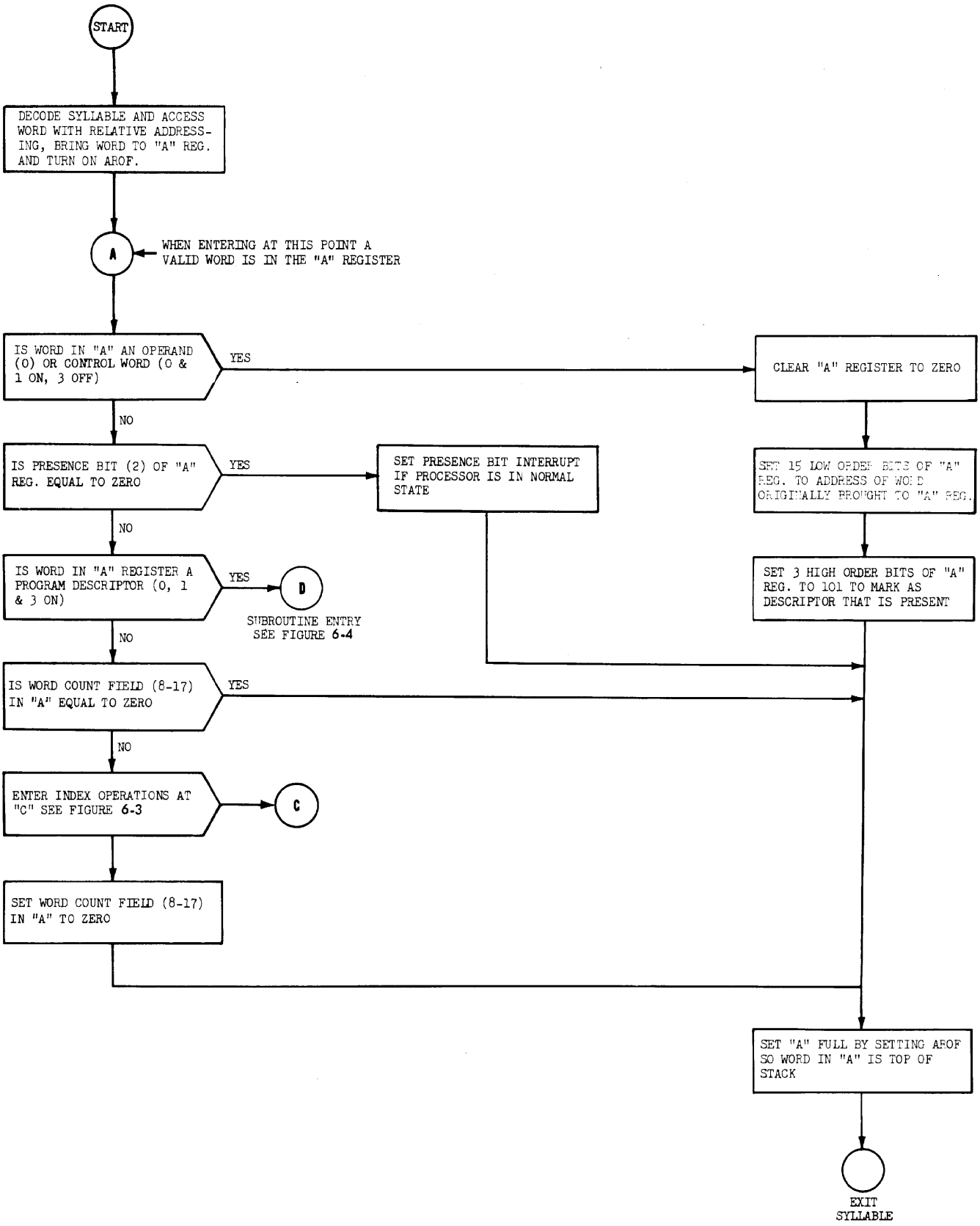


Figure 6-2. Descriptor Call Syllable Flow Chart

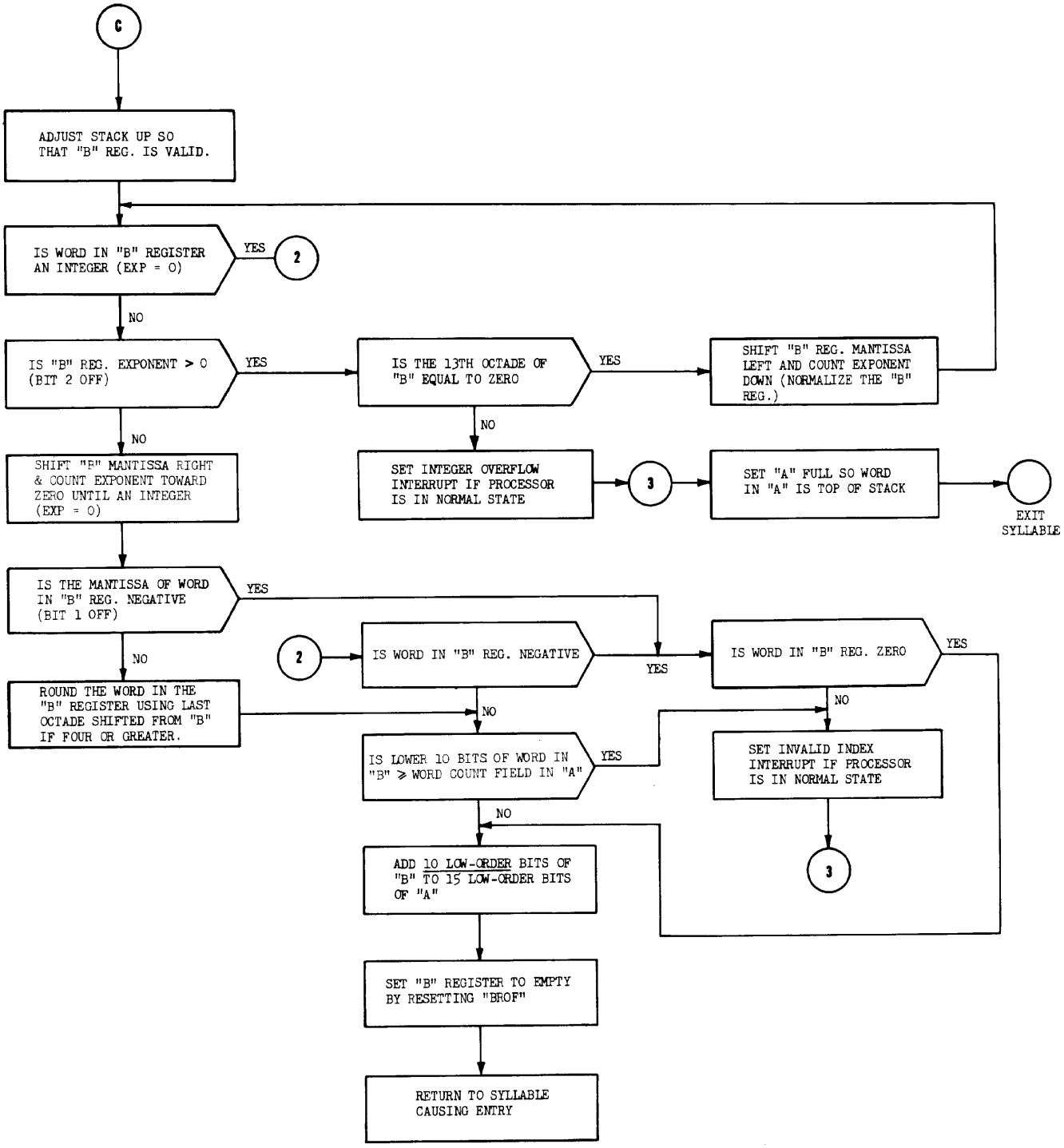


Figure 6-3. Index Operations—Operand and Descriptor Call Syllable

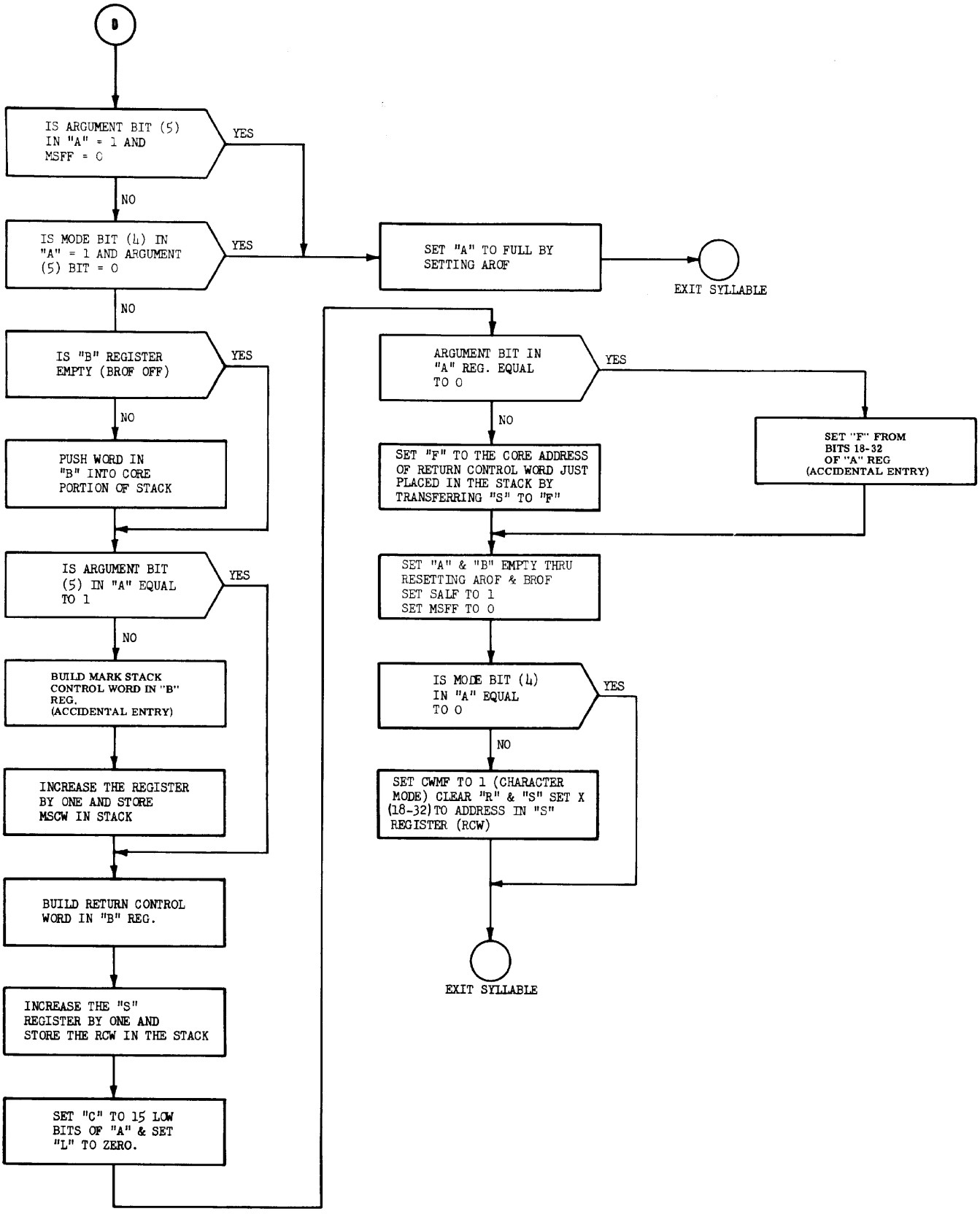


Figure 6-4. Subroutine Entry—Operand or Descriptor Call Syllable

6-7. PRESENCE BIT ON (BIT 2 ON). When the presence bit is on, the word count field is interrogated to see if it equals zero.

6-8. When the word count field is equal to zero (Bits 8-17 = 0) then the word specified by the data descriptor is accessed and placed in the A register. If this word is an operand or control word, the A register is marked as valid and the syllable terminated. If the word accessed was a descriptor (bit 0 on), and if the processor is in normal state, then a flag bit interrupt is set and the syllable terminated.

6-9. When the word count field is not equal to zero (Bits 8-17 $\neq$ 0) the data descriptor is referencing an area such as an array where indexing is required. The index is the second word of the stack. If the index contains an exponent value then it must be converted into an integer. In this operation it is possible to have an integer overflow interrupt, at which time the operation is terminated with the data descriptor in the top of the stack and the index as the second word of the stack. The integer overflow interrupt may only be set if the processor is operating in normal state. With an integer index, the low 10 bits of the index are compared to the word count field. The value of the integer must be a positive value and its magnitude less than that of the word count field. If either of the conditions do not exist and if the processor is operating in normal state, an invalid index interrupt is set, the operation is ended with the data descriptor in the top of the stack and with the index as the second word. At this point, the base address (contained in the lower 15 bits of the data descriptor) is added to the index (10 bits), and the word referenced with the resulting address is accessed and placed in the A register. If this accessed word is an operand or control word, the word is marked as valid and left as the top word of the stack. The index value is deleted from the stack. In either case, this completes the operation. If the word accessed is a descriptor (10 bit on) and the processor is in normal state, a flag bit interrupt is set.

Program Descriptor (Bits 0, 1, 3 On)

6-10. PRESENCE BIT OFF (BIT 2 OFF). This indicates that the program segment string which this descriptor references is not present in memory. The presence bit interrupt is set if the processor is in the normal state, and the syllable is terminated. The A register is marked as valid.

6-11. PRESENCE BIT ON (BIT 2 ON). When the presence bit is on, the argument bit is interrogated to see if formal parameters are needed.

6-12. ARGUMENT BIT OFF. If the argument bit is off (Bit 5 Off), these formal parameters are not required. At this point the CWMF is interrogated.

6-13. If the CWMF bit is on (Bit 4 On), the program descriptor is left in the top of the stack as a valid word and the syllable is terminated. This is called a special program descriptor.

6-14. If the CWMF bit is off (Bit 4 Off), the processor is entering word mode. This is called an "accidental entry." A MSCW and an RCW are generated and placed in the top of the stack. The C register is set from bits 33 through 47 of the program descriptor. The F register is set from the contents of bits 18 through 32 of the program descriptor. The SALF is turned on. An exit is made from this syllable with the MSFF in the off state. The next syllable to be executed will be the first syllable of the referenced program segment string.

6-15. ARGUMENT BIT ON. If the argument bit is on (Bit 5 On), then formal parameters are required. When formal parameters are required, the MSFF is interrogated.

6-16. If the MSFF is off, then the referenced program descriptor is placed in the top of the stack and the syllable is terminated.

6-17. If the MSFF is on, the mode must be determined.

6-18. If CWMF is on (Bit 4 On), the processor is entering character mode. Since the MSFF is on, a MSCW has already been placed in the stack. At this time, a RCW is generated and placed in the top of the stack. The C register is set from the program descriptor. The F register is set from the contents of the S register. This leaves the F register pointing at the RCW. The contents of the S register (which are identical to F at this time) are stored in bits 18 through 32 of the X register. The X register then contains

the address of the last previous RCW or loop control word placed in the stack when in character mode. The CWMF and SALF are then turned on. The first syllable of the referenced program segment string is then executed. This program would be in character mode.

6-19. If CWMF is off (Bit 4 Off), the processor is entering word mode. Again, since the MSFF is on, a MSCW has already been placed in the stack. At this time, an RCW is generated and placed in the top of the stack. The C register is set from the program descriptor. F register is set from the contents of the S register. This leaves the F register pointing at the RCW. The SALF is turned on, and an exit is made with the MSFF in the off state. The next syllable to be executed will be the first syllable of the sub-program.

DESC DESCRIPTOR CALL SYLLABLE (DCSL) XXX3 OR XXX7

6-20. The function of this syllable is to reference the stack program reference table or program segment string for a descriptor and to place this descriptor in the stack. When the stack or PRT is referenced, either an operand or a descriptor may be obtained. In the case of an operand, a data descriptor is generated which contains the address of the referenced operand. When a program descriptor is referenced, control is transferred to the program segment specified by the program descriptor and operation is subsequently in the sub-program level. The first action of the DESC is to reference the stack, PRT, or program segment string through relative addressing, and bring the word addressed to the A register. If the A register is valid prior to this operation, its contents will be pushed down into the stack. Once the PRT, stack, or program segment string have been accessed, and the word brought to the A register, the following actions occur according to the type of word accessed (figure 6-2).

Operand or Control Word (Operand = Bit 0 Off; Control Word = Bits 0, 1 On; 3 Off)

6-21. A data descriptor is generated which contains the absolute address of the referenced word. This data descriptor is placed in the top of the stack as a valid word.

Data Descriptor (Bit 0 On)

6-22. PRESENCE BIT OFF (BIT 2 OFF). This indicates that the word referenced by this descriptor is not present in memory. The presence bit interrupt is set if the processor is in normal state and the A register is marked as valid. The syllable is terminated.

6-23. PRESENCE BIT ON (BIT 2 ON). When this bit is on, the word count field is interrogated.

6-24. If the word count field is equal to zero (Bits 8-17 = 0), then the referenced data descriptor with a word count field equal to zero is referencing a one word operand and requires no index. The data descriptor is placed in the top of the stack and the syllable is terminated.

6-25. If the word count field is not equal to zero (Bits 8-17 $\neq$ 0), then the data descriptor is referencing an area such as an array when indexing is required. The index is the second word of the stack. If the index is a fractional number (exponent not equal to zero), then it must be converted into an integer. In this operation it is possible to have an integer overflow interrupt if the processor is in normal state. The integer 10 low-order bits are compared to the word count field. They must be a positive number, and have an absolute value of less than the value found in the word count field. If either of the previous conditions do not exist an invalid index interrupt is set. If the processor is operating in normal state, the operation is ended with the data descriptor in the top of the stack and the index as the second word of the stack. At this point the base address (contained in bits 33 through 47 of the data descriptor) is added to the top word of the stack to create the absolute address. The absolute address formed by the indexing operation is placed back into bits 33 through 47 of the data descriptor. The word count field is set to zero. The index is eliminated from the stack and the altered data descriptor is placed in the top of the stack as a valid word. The syllable is terminated.

Program Discriptor (Bits 0, 1, 3 On)

6-26. The actions that take place when a DESC accesses this type of word are the

same as those described under the program descriptor access of the Operand Call syllable. For this action refer to that syllable.

OPERATOR SYLLABLES

6-27. The word mode operators are specified and the description of the operators assumes any necessary stack adjustments. For certain operators an operand is made into an integer as follows:

a. If the exponent is zero, the operand is not changed.

b. If the exponent is positive and non-zero, the operand is normalized and the exponent reduced. If the exponent is not reduced to zero as a result of normalizing, the integer overflow interrupt bit is set.

c. If the exponent is negative and non-zero, the operand is shifted to the right until the exponent equals zero, and the mantissa is rounded according to the following rules:

   1. For positive operands, the mantissa is increased in magnitude by one, if the fractional part of the operand (the digits shifted out of the register) was greater than or equal to one-half.
   2. For negative operands, the mantissa is increased in magnitude by one, if fractional part of the operand was greater in magnitude than one-half.

ARITHMETIC OPERATORS—SINGLE PRECISION

6-28. The single precision arithmetic operators operate on the two top operands in the stack. They remove the two top operands from the stack and leave the result in the top of the stack.

ADD Single Precision Add (AD1L) 0101

6-29. The operands in the A and B registers are added algebraically and the sum left in the B register. For all conditions, at the end of the operation the A register is set to empty, the B register is set to full, and the B register flag bit is set to zero.

6-30. If either operand has a mantissa of zero, the non-zero operand is the result. If both operands have a mantissa of zero, the B register is set to all zeros. In either case, the operation is terminated.

6-31. If the mantissa signs and the exponents of the operands are equal, the mantissas are added and the sum placed in the B register. If the sum exceeds 13 octal digits, the mantissa of the sum is shifted right one octal place, rounded and the exponent algebraically increased by one.

6-32. If the exponents of the operands are equal but the mantissa signs are unequal, the difference of the mantissa with appropriate sign is placed in the B register. If the difference is equal to zero, the B register is set to all zeros.

6-33. If the exponents of the operands are unequal, the operands are first aligned. If the alignment causes the smaller operand to be shifted right 14 octal places, the larger operand is the result.

6-34. If the alignment causes the smaller operand to be shifted right, but less than 14 octal places, the digits of the smaller operand shifted out of the register are used to obtain the result.

6-35. If the signs of the operands are equal, the mantissas are added and the sum placed in the B register. If the sum does not exceed 13 octal digits, the last digit shifted out of the register is used for rounding the result. If the sum is 14 digits, the mantissa in B is rounded to 13 digits.

6-36. If the signs of the operands are unequal, the digits are complemented as they are shifted out of the register during alignment. In effect, the equivalent of the 15 digit subtraction occurs in this latter case and the result is rounded to the 13 most significant digits of the 15 digit result.

6-37. If the result has an exponent greater than +63, the exponent overflow bit is set in the interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

SUB Single Precision Subtract (SU1L) 0301

6-38. The operand in the A register is algebraically subtracted from the operand in the B register and the difference left in the B register. An add operation is performed as specified for the ADD operator except for the conditions of sign comparison.

MUL Single Precision Multiply (MU1L) 0401

6-39. The operands in the A and B registers are algebraically multiplied and the product left in the B register. For all conditions, at the end of the operation the A register is set to empty; the B register is set to full and the B register flat bit is set to zero.

6-40. If the mantissa of either operand is zero, the B register is set to all zeros.

6-41. If the exponents of the operands are both zero, the 26 digit product of the mantissas is computed. If the 13 most significant digits of the product are all zero, the 13 least significant digits are the mantissa of the result and the exponent of the result is zero. If the 13 most significant digits of the product are not all zero, the product is normalized and rounded to 13 digits. A mantissa of thirteen sevens is not rounded.

6-42. If the exponents of the operands are not both zero, the operands are normalized. The 26 digits product of the mantissas is computed, normalized and rounded to 13 digits.

6-43. If the result has an exponent greater than +63 or less than -63 the exponent overflow bit or exponent underflow bit respectively, is set in the interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

DIV Single Precision Divide (DV1L) 1001

6-44. The operand in the B register is algebraically divided by the operand in the A register and the quotient left in the B register. For all conditions, after the operation the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

6-45. If the mantissa of the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit in the interrupt register is set. In either case, the operation is terminated.

6-46. If the mantissa of neither operand is zero, both operands are normalized and the operand in the B register is divided by the operand in the A register. Fourteen significant quotient digits are developed. The quotient is rounded to thirteen significant digits and left in the B register.

6-47. If the result has an exponent greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

IDV Integer Divide (DV3L) 3001

6-48. The operand in the B register is algebraically divided by the operand in the A register and the integer part of the quotient is left in the B register. For all conditions, after the operation in the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

6-49. If the mantissa for the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit is set in the interrupt register. In either case, the operation is terminated.

6-50. If the mantissa of neither operand is zero, both operands are normalized. If the exponent of the B register is algebraically less than the exponent of the A register after both operands have been normalized, the B register is set to all zeros. If the exponent of the B register is algebraically equal to or greater than the exponent of the A register, the divide operation proceeds until an integer quotient or a quotient of 13 significant digits is calculated.

6-51. If an integer quotient is developed, the quotient is left in the B register with zero exponent. If a non-integer quotient is developed, the integer overflow bit is set in the interrupt register and the 13 significant digit quotient with the correct exponent modulo 64 is left in the B register.

RDV Remainder Divide (DV4L) 7001

6-52. The operand in the B register is algebraically divided by the operand in the A register to develop an integer quotient. The remainder after the division is left in the B register. For all conditions, after the operation the A register is set to empty, the B register is set to full and the B register flag bit is set to zero.

6-53. If the mantissa of the B register is zero, the B register is set to all zeros. If the mantissa of the A register is zero, the divide by zero bit is set in the interrupt register. In either case, the operation is terminated.

6-54. If the mantissa of neither operand is zero, both operands are normalized. If the exponent of the B register is algebraically less than the exponent of the A register after both operands have been normalized, the operand in the B register is the result. If the exponent of the B register is algebraically equal to or greater than the exponent of the A register; the divide operation proceeds until an integer quotient or a quotient of 13 significant digits is calculated.

6-55. If an integer quotient is developed and the mantissa of the remainder is not zero, the remainder with exponent modulo 64 is placed in the B register. The sign of the remainder is specified by the sign of the dividend. If an integer quotient is developed and the mantissa of the remainder is zero, the B register is set to all zeros.

6-56. If a non-integer quotient is developed, the integer overflow bit is set in the interrupt register and the B register is set to all zeros.

6-57. If the result has an exponent less than -63 and there is no integer overflow, the exponent underflow bit is set in the interrupt register. The B register contains the correct mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the B register modulo 64.

ARITHMETIC OPERATORS—DOUBLE PRECISION

6-58. For the double precision arithmetic operators, an operand occupies two words. The mantissa of the second word of an operand is considered an extension of the mantissa of the first word of an operand, i.e., the mantissa of the first word of an operand is an integer and the mantissa of the second word of the operand is a fraction. When in the stack, the first word of a double precision operand is in the top of the stack and the second word of a double precision operand is in the second word of the stack. Therefore, double precision arithmetic operators operate on four words in the stack, removing those words from the stack and leaving the result as two words in the stack.

6-59. For the add, subtract and multiply operators, two integer operands yield an integer result if no overflow occurs. If one or both operands is non-integer or the result overflows, the result is non-integer.

6-60. During the execution of all arithmetic operators, the exponents of the operands are carried to seven bits. At the completion of the execution of an arithmetic operator, the exponent is reduced to six bits. If the exponent of the result required more than six bits, the appropriate interrupt bit is set in the interrupt register.

DLA Double Precision Add (AD2L) 0105

6-61. The double length operand in the A and B registers is algebraically added to the double length operand addressed by the S register. The double length result is left in the A and B registers and the S register is reduced by two. Bit positions 48 through 40 of the least significant word of the double length result are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

6-62. If the exponents of the two operands are equal, the double length mantissas are algebraically added. If the double length mantissa exceeds 26 octal digits, the double length mantissa is shifted right one place and the exponent algebraically increased by one. The result is normalized. If the mantissa of the result is zero, the A and B registers are set to all zeros and the operation is terminated.

6-63. If the exponents of the operands are not equal, alignment of the double length operands occurs. The alignment of double length operands is equivalent to the alignment of single length operands. When shifting a double length mantissa, digits are shifted between the least significant digit of the most significant word of the operand to the most significant digit of the mantissa of the least significant word of the operand.

6-64. If the mantissa of the smaller operand is shifted right 26 or more octal digits during alignment, the larger operand, normalized, is the result.

6-65. After alignment of operands with unequal exponents, the mantissas of the operands are algebraically added. If the double length mantissa exceeds 26 octal digits, the double length mantissa of the result is shifted to the right one place and the exponent algebraically increased by one. The result is normalized.

6-66. If the exponent of the result is greater than +63 or less than -63 the exponent overflow bit or exponent underflow bit, respectively, is set in the interrupt register. The result in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register modulo 64.

DLS Double Precision Subtract (SU2L) 0305

6-67. The double length operand in the A and B registers is algebraically subtracted from the double length operand addressed by the S register. An addition operation is performed as specified for the add double length operator, (see paragraphs 6-61 through 6-66) except that the conditions of sign comparison are changed to effect an algebraic subtraction.

DLM Double Precision Multiply (MU2L) 0405

6-68. The double length operand in the A and B registers is algebraically multiplied by the double length operand addressed by the S register. The double length result is left in the A and B registers and the S register is reduced by two. Bit positions 48 through 40 of the least significant word of the double length result are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

6-69. Both double length operands are normalized. If either operand has a mantissa of zero, the A and B registers are set to all zeros and the operation is terminated.

6-70. If neither double length operand has a mantissa of zero, the normalized double length mantissas of the two operands are multiplied. Twenty seven digits of the 52 digit product are retained. The product is normalized and truncated to a 26 digit result. The two least significant digits are not considered a precise part of the result because there may be a maximum error of 1 in the twenty-fifth digit position.

6-71. If the exponent of the result is greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register, modulo 64.

DLD Double Precision Divide (DV2L) 1005

6-72. The double length operand addressed by the S register is algebraically divided by the double length operand in the A and B registers. The double length result is left in the A and B registers and the S register is reduced by two. Bit positions 48 through 40 of the least significant word of the double length result are set to zero. The flag bit of the most significant word of the double length result is set to zero. All non-zero results are normalized. The A and B registers are both set to full.

6-73. The double length operand in the A and B registers is normalized. If the operand in

the A and B registers has a mantissa of zero, the double length operand addressed by the S register is placed in the A and B registers as the result, the divide by zero bit is set in the interrupt register and the operation is terminated.

6-74. If the double length operand in the A and B registers does not have a mantissa of zero, the double length operand addressed by the S register is normalized. If the double length operand addressed by the S register has a mantissa of zero, the A and B registers are set to all zeros and the operation is terminated.

6-75. If neither double length operand has a mantissa of zero, the divide operation on the normalized operands takes place. A result of twenty six significant quotient digits is developed.

6-76. If the exponent of the result is greater than +63 or less than -63, the exponent overflow bit or exponent underflow bit, respectively, is set in the interrupt register. The result in the A and B registers contains the correct double length mantissa, mantissa sign and exponent sign. The magnitude of the correct exponent is contained in the exponent field of the A register modulo 64.

LOGICAL OPERATORS

LND Logical And (LOAL) 0415

6-77. Set a one in each position of the B register, except the flag bit, when a one appears in the corresponding bit positions in both the A and the B registers. Set a zero in the corresponding position of the B register except the flag bit, when a one is not in the corresponding positions of both the A and the B registers. The flag bit of the word in the B register is unaltered. The A register is set to empty.

LOR Logical Or (LOOL) 0215

6-78. For all bit positions of the B register, except the flag bit, set the bit to one if the corresponding bit position in either the A or B registers is one, otherwise set the bit to zero. The flag bit of the word in the B register is unaltered. The B register is set to empty.

LQV Logical Equivalence (LOEL) 1015

6-79. Set a one in each position of the B register, except the flag bit, when the corresponding bit positions of the A and B registers are equal. Set a zero in each position of the B register, except the flag bit, when the corresponding bit positions of the A and B registers are not equal. The flag bit of the B register is unaltered. The A register is set to empty.

LNG Logical Negate (LONL) 0115

6-80. Complement every bit position of the A register except the flag bit which is unaltered.

RELATIONAL OPERATORS

6-81. The relational operators perform comparisons on the two top operands in the stack. The operands are removed from the stack and the result of the comparison is placed in the top of the stack. The operands may be in an unnormalized form and the required normalizing/scaling will take place in the comparison operation. For the relational operators, operands of zero, minus zero and a zero mantissa with non-zero exponent are considered equal. Flag bits are ignored.

GTR B Greater Than A (BGAL) 0225

6-82. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically greater than the value of the operand in the A register, the low order bit of the B register is set to one and all other bits of the B register are set to zero; otherwise, all bits of the B register are set to zero. The A register is set to empty.

GEQ B Greater Than or Equal to A (BGEL) 0125

6-83. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically greater than or equal to the value of the operand in the A register, the low-order bit of the B register is set to one and all other bits of the B register are set to zero; otherwise, all bits of the B register are set to zero. The A register is set to empty.

EQL B Equal to A (BEQL) 4425

6-84. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically equal to the value of the operand in the A register, the low-order bit of the B register is set to one and all other bits of the B register are set to zero; otherwise, all bits of the B register are set to zero. The A register is set to empty.

LEQ B Less Than or Equal to A (BLEL) 4125

6-85. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically less than or equal to the value of the operand in the A register, the low-order bit of the B register is set to one and all other bits of the B register are set to zero; otherwise, all bits of the B register are set to zero. The A register is set to empty.

LSS B Less Than A (BLAL) 4225

6-86. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically less than the value of the operand in the A register, the low-order bit of the B register is set to one and all other bits of the B register are set to zero; otherwise, all bits of the B register are set to zero. The A register is set to empty.

NEQ B Not Equal to A (BNEL) 0425

6-87. The operand in the B register is algebraically compared with the operand in the A register. If the value of the operand in the B register is algebraically not equal to the value of the operand in the A register, the low-order bit of the B register is set to one and all other bits of the B register are set to zero. The A register is set to empty.

BRANCH OPERATORS

6-88. For the branch operators, the top of the stack specifies the cell or syllable to which branching occurs. Branching is either relative to the location of the branch operator or to an absolute address, as determined by the type of word in the top of the stack. An operand specifies the number of syllables (for syllable branches) or words (for word branches) to be jumped, either forward or backward; a descriptor specifies an address to which branching occurs. Conditional branch operators use the low-order bit of the second word in the stack as the true-false condition on which to branch. Conditional branches take place on a false condition.

BFW Branch Forward Unconditional (BFUL) 4231

6-89. If the flag bit of the word in the A register is zero, the C and L registers are increased by the 12 low-order bits of the word in the A register. If the flag bit and the presence bit of the word in the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases the A register is set to empty.

6-90. If the flag bit of the word in the A register is a one and the presence bit is a zero, the presence bit is set in the interrupt register, the contents of the A register are retained and the operation terminated.

BBW Branch Backward Unconditional (BBUL) 4131

6-91. If the flag bit of the word in the A register is zero, the C and L registers are decreased by the 12 low-order bits of the word in the A register. If the flag bit and the presence bit of the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases the A register is set to empty.

6-92. If the flag bit of the word in the A register is a one and the presence bit is a zero, the presence bit is set in the interrupt register, the contents of the A register are retained and the operation terminated.

BFC Branch Forward Conditional (BFCL) 0231

6-93. If the low-order of the word in the B register is a one, the A and B registers are set to empty and the operation terminated.

6-94. If the low-order bit of the word in the B register is a zero and the flag bit of the word in the A register is a zero, the C and L registers are increased by the 12 low-order bits of the word in the A register. If the low-order bit of the word in the B register is a zero and the flag bit and the presence bit of the word in the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases, the A and B registers are set to empty.

6-95. If the low-order bit of the word in the B register is a zero, the flag bit of the word in the A register is a one and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register, the contents of the A and B registers are retained and the operation is terminated.

BBC Branch Backward Conditional (BBCL) 0131

6-96. If the low order bit of the word in the B register is a one, the A and B registers are set to empty and the operation terminated.

6-97. If the low-order bit of the word in the B register is a zero and the flag bit of the word in the A register is a zero, the C and L registers are decreased by the 12 low-order bits of the word in the A register. If the low-order bit of the word in the B register is a zero and the flag bit and the presence bit of the word in the A register are both one, the 15 low-ordered bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases, the A and B registers are set to empty.

6-98. If the low-order bit of the word in the B register is a zero, the flag bit of the word in the A register is a one and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register, the contents of the A and B registers are retained and the operation is terminated.

BRT Branch Return (RJPL) 0135

6-99. If the presence bit of the word in the A register is zero, the presence bit is set in the interrupt register and the operation is terminated.

6-100. If the presence bit of the word in the A register is on, the operation is continued. The S register is set to the contents of bit positions 18 through 32 of the word in the A register. The C register is set to the contents of bit positions 33 through 47 of the word in the A register. The L register is set to zero. The contents of the cell addressed by the S register, the MSCW, is read from memory. The R and F registers are set to the contents of their respective fields of the MSCW. The MSFF and the SALF are set to the contents of their respective positions of the MSCW. The S register is decreased by one. The A and B registers are set to empty.

LFU Word Branch Forward Unconditional (JFUL) 6231

6-101. If the flag bit of the word in the A register is zero, the C register is increased by the 10 low-order bits of the word in the A register, and the L register is set to zero. If the flag bit and the presence bit of the word in the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases, the A register is set to empty.

6-102. If the flag bit of the word in the A register is a one and the presence bit is zero, the presence bit is set in the interrupt register, the contents of the A register are retained and the operation terminated.

LBU Word Branch Backward Unconditional (JBUL) 6131

6-103. If the flag bit of the word in the A register is zero, the C register is decreased by the 10 low-order bits of the word in the A register, and the L register is set to zero. If the flag bit and the presence bit of the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases, the A register is set to empty.

6-104. If the flag bit of the word in the A register is a one and the presence bit is a zero, the presence bit is set in the interrupt register, the contents of the A register are retained and the operation terminated.

LFC Word Branch Forward Conditional (JFCL) 2231

6-105. If the low-order bit of the word in the B register is a one, the A and B registers are set to empty and the operation terminated.

6-106. If the low-order bit of the word in the B register is a zero and the flag bit of the word in the A register is a zero, the C register is increased by the 10 low-order bits of the word in the A register, and the L register is set to zero. If the low-order bit of the word in the B register is a zero and the flag bit and the presence bit of the word in the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both of the cases the A and B registers are set to empty. If the low-order bit of the word in the B register is a zero, the flag bit of the word in the A register is a one, and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register, the contents of the A and B registers are retained and the operation is terminated.

LBC Word Branch Backward Conditional (JBCL) 2131

6-107. If the low-order bit of the word in the B register is a one, the A and B registers are set to empty and the operation terminated.

6-108. If the low-order bit of the word in the B register is a zero and the flag bit of the word in the A register is a zero, the C register is decreased by the 10 low-order bits of the word in the A register, and the L register is set to zero. If the low-order bit of the word in the B register is a zero and the flag bit and the presence bit of the word in the A register are both one, the 15 low-order bits of the word in the A register are transferred to the C register and the L register is set to zero. In both cases, the A and B registers are set to empty.

6-109. If the low-order bit of the word in the B register is a zero, the flag bit of the word in the A register is a one and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register, the contents of the A and B registers are retained and the operation is terminated.

CBD Non-Zero Field Branch Backward, Destructive (ZBDL) XX51[1]

6-110. Pushup occurs if necessary to fill the A and B registers. This operator tests a field of the word in the B register for zero. The starting bit of the field is determined by the G and H registers. The length of the field is determined by the 4 high-order bits in the operator code. The field may be 1 through 15 bits in length.

6-111. If the field is zero, both the A and B registers are marked empty and the operator is terminated. If the field is not zero, the B register is marked empty and the T register is changed to a syllable branch backwards unconditional operator which is immediately executed.

CBN Non-Zero Field Branch Backward, Non-Destructive (ZBNL) XX51[1]

6-112. A field in the B register is tested for zero as described in paragraph 6-110.

6-113. If the field is zero, the A register only is marked empty and the operator is terminated. If the field is not zero, the T register is changed to a syllable branch backwards unconditional operator, which is immediately executed.

CFD Non-Zero Field Branch Forward, Destructive (ZFDL) XX51[1]

6-114. A field in the B register is tested for zero as described in paragraph 6-110.

---

[1]The bits of the two most significant octades of the following four operators are defined as follows:

Non-Zero Field Branch Backward, Destructive XXX X11
Non-Zero Field Branch Backward, Non-Destructive XXX X01
Non-Zero Field Branch Forward, Destructive XXX X10
Non-Zero Field Branch Forward, Non-Destructive XXX X00

6-115. If the field is zero, both the A and B registers are marked empty and the operator is terminated. If the field is not zero, the B register is marked empty and the T register is changed to a syllable branch forward unconditional operator, which is immediately executed.

CFN Non-Zero Branch Forward Non-Destructive (ZFNL) XX51[1]

6-116. A field in the B register is tested for zero as described in paragraph 6-110.

6-117. If the field is zero, the A register only is marked empty and the operator is terminated.

6-118. If the field is not zero, the T register is changed to a syllable branch forward unconditional operator, which is immediately executed.

STORE OPERATORS

6-119. The store operators operate on the two top words in the stack. The top word in the stack specifies the address into which the second word in the stack is stored. This address may be relative or absolute, depending on whether the flag bit is zero or one respectively. The destructive store operators remove both the address and the information stored from the stack. The non-destructive store operators remove only the address from the stack.

STD "B" Store Destructive (BSDL) 0421

6-120. If the flag bit and the presence bit of the word in the A register are both one, the contents of the B register are stored in memory cell addressed by the 15 low order bits of the A register. The A and B registers are set to empty.

6-121. If the flag bit of the word in the A register is one and the presence bit is zero, the presence code is set in the interrupt register and the operation terminated.

6-122. If the flag bit of the word in the A register is zero, the ten low-order bits of the word in the A register are used as a relative address except that no addressing relative to the C register takes place. If the syllable calls for addressing relative to the C register, the absolute address is constructed relative to the R register instead. The contents of the B register are stored in the memory cell addressed after appropriate indexing of the relative address. The A and B registers are set to empty.

6-123. If the VARF is set, the processor is set to sub-program level, after the relative address operation and VARF is reset.

SND "B" Store Non-Destructive (BSNL) 1021

6-124. If the flag bit and the presence bit of the word in the A register are both one, the contents of the B register are stored in the memory cell addressed by the 15 low-order bits of the A register. The A register is set to empty.

6-125. If the flag bit of the word in the A register is a one and the presence bit is zero, the presence code is set in the interrupt register and the operation terminated.

6-126. If the flag bit of the word in the A register is a zero, the 10 low-order bits of the word in the A register are used as relative address except that no addressing relative to the C register takes place. If the syllable calls for addressing relative to the C register, the absolute address is constructed relative to the R register instead. The contents of the B register are stored in the memory cell addressed after appropriate indexing of the relative address. The A register is set to empty. If the VARF is set, the processor is set to sub-program level after the relative address operation and VARF is then reset.

---

[1] The bits of the two most significant octades of the following four operators are defined as follows:

Non-Zero Field Branch Backward, Destructive XXX X11
Non-Zero Field Branch Backward, Non-Destructive XXX X01
Non-Zero Field Branch Forward, Destructive XXX X10
Non-Zero Field Branch Forward, Non-Destructive XXX X00

ISD Integer Store Destructive (ISDL) 4121

6-127. If the flag bit and the presence bit of the word in the A register are both one, or if the flag bit of the word in the A register is zero, the word in the B register is made an integer as specified in paragraph 6-27.

6-128. If an integer overflow occurs, the integer overflow bit is set in the interrupt register. The contents of the A and B registers are retained and the operation is terminated. If integer overflow does not occur, a store operation, as specified for the store destructive operator, is performed.

6-129. If the flag bit of the word in the A register is one and the presence bit of the word in the A register is zero, the presence bit is set in the interrupt register and the operation terminated.

6-130. If the VARF is set, the processor is set to sub-program level after the relative address operation and VARF is then reset.

ISN Integer Store Non-Destructive (ISNL) 4221

6-131. If the flag bit and the presence bit of the word in the A register are both one or if the flag bit of the word in the A register is a zero, the word in the B register is made an integer as specified in paragraph 6-27.

6-132. If an integer overflow occurs, the integer overflow bit is set in the interrupt register, the contents of the A and B register is retained and the operation is terminated. If integer overflow does not occur, a store operation, as specified for the store non-destructive operator, is performed.

6-133. If the flag bit of the word in the A register is a one and the presence bit of the word in the A register is a zero, the presence bit is set in the interrupt register and the operation terminated.

6-134. If the VARF is set, the processor is set to sub-program level after the relative address operation and VARF is then reset.

CID Conditional Integer Store Destructive (CSDL) 0121

6-135. If the integer bit of the word in the A register is a one, an integer store destructive operator is performed as specified in paragraphs 6-127 through 6-134.

If the integer bit of the word in the A register is a zero, a store destructive operator is performed as specified in paragraphs 6-120 through 6-123.

CND Conditional Integer Store Non-Destructive (CSNL) 0221

6-136. If the integer bit of the word in the A register is a one, an integer store non-destructive operator is performed as specified in paragraphs 6-130 through 6-134.

If the integer bit of the word in the A register is a zero, a store non-destructive operator is performed as specified in paragraphs 6-124 through 6-126.

NOP Word Mode NO-OP (NOPL) 0055

6-137. Performs no operation except to count up the L register by one.

BIT OPERATORS

DIA Dial A (DIAL) XX55

6-138. If the six high-order bits of the operator are not zero, the three most significant bits of the operator are placed in the G register and the three next most significant bits of the operator are placed in the H register. If all of the six high-order bits are zero, no action takes place. If the H register is set to 110 or 111, the operation of the subsequent operators using this register is not specified.

DIB Dial B (DIBL) XX61

6-139. If the six high-order bits of the operator are not zero, the three most significant bits of the operator are placed in the K register and the three next most significant bits of the operator are placed in the V register. If all of the six high-order bits of the operator are zero, a set variant operator

takes place as described in paragraphs 6-225 through 6-231. If the V register is set to 110 or 111 the operation of subsequent operators using this register is not specified.

TRB Transfer Bits (TRFL) XX65

6-140. A field in the A register, starting at the bit position addressed by the G and H registers, replaces a corresponding length field in the B register, at the bit position addressed by the K and V registers, and proceeding towards the low-order bit positions.

6-141. The length of the field transferred is specified by the six high-order bits of the operator. The transfer of bits is terminated by the transfer of the specified number of bits or when either the A or B registers have been exhausted.

6-142. The contents of the G, H, K and V registers after the operation are the same as prior to the operation. The A register is set to empty.

FCE Compare Field Equal (CFEL) XX75

6-143. A field in the A register, starting at the bit position addressed by the G and H registers, is compared with a corresponding length field in the B register, starting at the bit position addressed by the K and V registers, and proceeding towards the low-order bit positions.

6-144. The length of the fields in the registers is specified by the six high-order bits of the operator. The comparison is terminated by the comparison of the number of bits specified or by the comparison of the low-order bit of either register.

6-145. If all of the corresponding bits of the field compared are equal, the low-order bit of the A register is set to one and all other bit positions of the A register are set to zero. If any of the corresponding bit positions of the fields compared are not equal, all bit positions of the A register are set to zero. The contents of the B, G, H, K and V registers after the operation are the same as prior to the operation.

FCL Compare Field Low (CFLL) XX71

6-146. A field in the A register, starting at the bit position addressed by G and H registers, is compared with a field in the B register, starting at the bit position addressed by the K and V registers, and proceeding towards the low-order bit positions.

6-147. The length of the fields in the registers is specified by the six high-order bits of the operator. The comparison is terminated by the comparison of the number of bits specified or by the comparison of the low-order bit position of either register.

6-148. If the magnitude of the field compared in the B register is less than the magnitude of the field compared in the A register, the low-order bit of the A register is set to one and all other bit positions of the A register are set to zero; otherwise, all bit positions of the A register are set to zero. The contents of the B, G, H, K, and V registers after the operation are the same as prior to the operation.

MOP Reset Flag Bit (RFBL) 2015

6-149. Set the flag bit of the word in the A register to zero.

MDS Set Flag Bit (SFBL) 4015

6-150. Set the flag bit of the word in the A register to one.

TOP Test Flag Bit (TFBL) 2031

6-151. If the flag bit of the word in the B register is zero, the low-order bit of the word in the A register is set to one and all other bits of the A register are set to zero; otherwise, all bits of the A register are set to zero. The A register is set to full.

SSP Reset Sign Bit (MSPL) 4431

6-152. Set the sign bit of the word in the A register to zero.

SSN Set Sign Bit (MSNL) 0431

6-153. Set the sign bit of the word in the A register to one.

CHS Change Sign Bit (CSSL) 1031

6-154. Complement the sign bit of the word in the A register.

ISO Variable Field Isolate (VFIL) XX45

6-155. This operator selects a field from the top word in the stack. The selected field is placed in a word right justified, and the rest of the word is set to zero. This new word replaces the top word in the stack.

6-156. The selected field may be 1 to 39 bits in length as determined by the L and S fields in the operator.

6-157. The starting bit position in the source field is defined by the G and H registers. The length of the field in characters is defined by the L field of the operator (see paragraph 6-159). The number of characters should include the characters positions which contain the first and last bits. If L = 7, then $3 \geq H \geq 5$, selecting not more than three bits of the most significant character position. The S field of the operator (see paragraph 6-159) specifies the number of bits that the resultant field is shifted to the right, thus deleting bits from the final characters transferred.

6-158. If G + L>8, no new word is accessed. End-around operation on the same word occurs.

AT THE END OF THE OPERATION N = 0, G = FINAL CHARACTER POSITION +1.

L INCLUDED CHARACTERS	S BITS SHIFTED	OPERATOR CODE

CTC Transfer "CORE" Field to "CORE" Field (CCXL) 5425

6-159. Pushup occurs, if required, into the A and B registers. The contents of bits 33 through 47 of the A register are transferred to bits 33 through 47 of the B register. The rest of the B register remains unchanged. The A register is marked empty.

CTF Transfer "CORE" Field to "F" Field (CFXL) 7425

6-160. Pushup occurs if required into the A and B registers. The contents of bits 33 through 47 of the A register are transferred to bits 18 through 32 of the B register. The rest of the B register remains unchanged. The A register is marked empty.

FTF Transfer "F" Field to "F" Field (FFXL) 3425

6-161. Pushup occurs if required into the A and B registers. The contents of bits 18 through 32 of the A register are transferred to bits 18 through 32 of the B register. The rest of the B register remains unchanged. The A register is marked empty.

SUBROUTINE OPERATORS

MKS Mark Stack (MSOL) 0441

6-162. The contents, if any, of the A and B register are pushed into the stack in memory. The MSCW is constructed and stored in the top of the stack in memory. The F register is set to the address of the cell in which the MSCW has been stored. If the MSFF is zero and the processor is in the sub-program level, the MSCW is stored in the cell addressed by the contents of the R register plus seven. The MSFF is set to one.

XIT Exit (REWL) 0435

6-163. Registers A and B are marked empty. The word addressed by the F register, the return control word, is placed in the B register.

6-164. If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the processor is in the normal state, the flag bit interrupt is set and the operator exited with the return control word left at the top of the stack. If the flag bit is 0 and the processor is in the control state, the operator is terminated, but the interrupt is not set.

6-165. The C, L, G, H, K and V registers are set to the contents of their respective fields of the return control word in the B register. The S register is set to the contents of the F register field of the return control word in the B register.

6-166. The word now addressed by the S register, the mark stack control word, is read from memory into the B register. The R and F registers are set to the contents of their respective fields of the MSCW. The MSFF and the SALF are set to the contents of their respective positions of the MSCW. The S register is decreased by one. The A and B registers are set to empty. The mark stack bit of the word in the B register is examined. If this bit is zero, the operation is completed.

6-167. If the mark stack bit is one, the program level bit (SALF) is examined. If the program level bit is zero, indicating program level, the operation is completed.

6-168. If the SALF bit is one, indicating sub-program level, the word addressed by the F register field of the MSCW, the previous MSCW, is placed in the B register. The MSFF is examined. If the MSFF is set, the process of reading the previous MSCW and examining the state of its MSFF is repeated until a MSCW with the MSFF reset (zero) is placed in the B register. The contents of the B register is stored in the cell addressed by the contents of the R register plus seven. The operation is completed.

RTN Return Normal (RNML) 0235

6-169. If the A register is empty, a word is placed in the A register by stack adjustment and the A register set to full. If both the A register and the B register are full, the B register is set to empty.

6-170. If the flag bit of the word in the A register is one and the presence bit zero, the presence bit interrupt is set and the operation immediately terminated.

6-171. The word addressed by the F register, the RCW, is placed in the B register.

6-172. If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the processor is in the normal state, the flag bit interrupt is set and the operator exited, with the A and B registers marked full. If the flag bit is 0 and the processor is in the control state, the operator is terminated but the interrupt is not set.

6-173. The C, L, G, H, K and V registers are set to the contents of their respective fields of the return control word in the B register. The S register is set to the contents of the F register field of the RCW in the B register.

6-174. The word addressed by the S register, the MSCW, is read from memory. The R and F registers are set to the contents of their respective fields of the MSCW. The MSFF and SALF are set from the word. The S register is decreased by one.

6-175. The mark stack bit of the word in the B register is examined. If this bit is zero the operation is completed.

6-176. If the mark stack bit is one, the program level bit is examined. If the program level bit is zero, indicating program level, the operation is completed.

6-177. If the program level bit is one, indicating sub-program level, the word addressed by the F register field of the MSCW (which is the previous mark stack control word) is placed in the B register and its mark stack bit is examined. If the bit is one, the process of reading the previous mark stack control word and the examination of its mark stack bit is repeated until a MSCW with the mark stack bit equal to zero is placed in the B register. The word in the B register is stored in the cell addressed by the contents of the R register plus seven. The operation is completed.

6-178. The subsequent action of the return operation is similar to that of the Operand/Descriptor Call syllable. If the syllable indication in the return control word indicates an OPDC, the subsequent action performed is described by the operand call flow chart, figure 6-1. If the syllable indication on the RCW indicates a DESC, the subsequent action performed is described by the descriptor call flow chart, figure 6-2.

RTS Return Special (RSPL) 1235

6-179. If the A register is empty, a word is placed in the A register by stack adjustment and the A register set to full. If both the A register and the B register are full, the B register is set to empty.

6-180. If the flag bit of the word in the A register is one and the presence bit zero, the presence bit interrupt is set and the operation immediately terminated. The word addressed by the S register, the RCW, is placed in the B register.

6-181. If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the processor is in the normal state, the flag bit interrupt is set and the operator exited, with the A and B registers marked full. If the flag bit is 0 and the processor is in the control state, the operator is terminated but the interrupt is not set.

6-182. The C, L, G, H, K and V registers are set to the contents of their respective fields of the RCW in the B register. The S register is set to the contents of the F register field of the RCW in the B register.

6-183. The word addressed by the S register, the MSCW, is read from memory. The R and F registers are set to the contents of their respective fields of the MSCW. The MSFF and the SALF are set to the contents of their respective positions of the MSCW. The S register is decreased by one.

6-184. The mark stack bit of the word in the B register is examined. If it is zero the operation is completed.

6-185. If the mark stack bit is one, the program level bit is examined. If the program level bit is zero, indicating program level, the operation is completed.

6-186. If the program level bit is one, indicating sub-program level, the word addressed by the F register field of the MSCW, the previous MSCW, is placed in the B register. The mark stack bit is examined. If it is one, the process of reading the previous mark stack control word and examination of its mark stack bit set is repeated until a MSCW with its mark stack bit equal to zero is placed in the B register. The word in the B register is stored in the cell addressed by the contents of the R register plus seven. The operation is completed.

6-187. The subsequent action of the return operation is similar to that of the Operand/ Descriptor Call syllable. If the syllable indication in the RCW indicates an OPDC, the subsequent action performed is described by the operand call flow chart, figure 6-1. If the syllable indicator in the return control word indicates a DESC, the subsequent action performed is described by the descriptor call flow chart, figure 6-2.

CMN Enter Character Mode In Line (ECML) 4441

6-188. The contents of the A register (presumed to be a destination address) and the B register are pushed down into the stack and both registers are marked empty.

6-189. A RCW is constructed and pushed into the stack. The C and L registers are stored in the RCW in the normal way, although they are not intended to be used for return.

6-190. The contents of the S register is transferred to both the F and X registers. The word below the RCW (presumed to be a destination address) is examined. If it is descriptor which is not present, the presence bit interrupt is set. If it is a descriptor which is present, the S register is set to the low order 15 bits of the descriptor. If the word below the return control word is an operand, the S register is set to the low order 15 bits and the K register is set from bits 30 through 32. The SALF is set to one. The MSFF is set to zero. The R register is set to zero. The CWMF is set to character mode.

STACK OPERATORS

XCH Exchange (EXCL) 1025

6-191. The contents of the A and B registers are exchanged.

DUP Duplicate (DUPL) 2025

6-192. Stack adjustment occurs to make one register empty and the other full. The contents of the full register is copied into the empty register and the empty register marked full.

DEL Delete Top of Stack (DELL) 0065

6-193. The top word in the stack is deleted in the following manner, depending on the condition of the A register occupancy flip flop (AROF) and the B register occupancy flip flops (BROF).

AROF	BROF	Action
0	0	S ← S - 1
0	1	BROF ← 0
1	0	AROF ← 0
1	1	AROF ← 0

MISCELLANEOUS OPERATORS

LOD Load Operator (LODL) 2021

6-194. If the flag bit and the presence bit of the word in the A register are both one, the word in the A register is replaced by the contents of the cell addressed by the 15 low-order bits of the A register.

6-195. If the flag bit of the word in the A register is zero, the 10 low-order bits of the word in the A register are used as a relative address. The contents of the A register are replaced by the contents of the memory cell addressed after appropriate indexing of the relative address.

6-196. If the flag bit of the word in the A register is one and the presence bit zero, the presence bit in the interrupt register is set and the operation is terminated.

6-197. If the VARF is set, the processor is set to sub-program level after the relative address operation and the VARF is reset.

INX Index (INDL) 0141

6-198. The 15 low-order bits of the word in the B register are arithmetically added to the 15 low-order bits of the word in the A register. Positions 0 through 32 of the A register are unchanged. Overflow is lost. The B register is set to empty.

COC Construct Operand Call (MDVL) 0241

6-199. The contents of the A and B registers are exchanged. The flag bit of the word in the A register is set to one. The subsequent action of this operator is identical to that of an OPDC after it has caused a word to be read from memory. For a complete description of the subsequent action of this operator see figure 6-1.

CDC Construct Descriptor Call (MDAL) 1241

6-200. The contents of the A and B registers are exchanged. The flag bit of the word in the A register is set to one. The subsequent action of this operator is identical to that of a DESC after it has caused a word to be read from memory. For a complete description of this subsequent action see figure 6-2.

COM Communication Operator (COML) 1011

6-201. The word at the top of the stack is stored in R plus nine (11 octal). The word is deleted from the stack. The communication interrupt bit is set. The operator is a No-Op in the control state.

PRL Program Release (PREL) 0111

6-202. If the flag bit and the presence bit of the word in the A register are both one, the contents of the cell addressed by the 15 low-order bits of the word in the A register is placed in the A register. If the processor is in the control state, the presence bit of the word obtained from memory is set to zero and the word is stored back into the cell from which it was fetched.

6-203. If the flag bit of the word in the A register is one and the presence bit zero, set the presence bit in the interrupt register and terminate the operation.

6-204. If the flag bit of the word in the A register is a zero, the ten low-order bits of the word in the A register are used as a relative address, except that no addressing relative to the C register takes place. If the syllable calls for addressing relative to the C register, the absolute address is constructed relative to the R register instead. The contents of the cell addressed after appropriate indexing of the relative address are placed in the A register. If

the processor is in the control state, the presence bit of the word obtained from memory is set to zero and the word is stored back into the cell from which it was fetched.

6-205. If the processor is in the normal state, the continuity bit of the word obtained from memory is inspected. If the continuity bit is a one, the continuity bit is set in the interrupt register. If the continuity bit is a zero, the program release bit is set in the interrupt register. The A register is set to empty.

6-206. If the processor is in normal state, the address just used is stored in R plus nine (11 octal). The address is stored in the fifteen low-order bits of the word, and all other bits are cleared to zero.

SFI Store for Interrupt Operator (SFIL) 3011

6-207. This operator can be used in any mode and state. If the processor is in word mode, the word mode interrupt operation takes place. If the processor is in character mode, a character mode interrupt operation takes place. (See paragraphs 5-39 through 5-41.)

6-208. If the operator occurs in processor 1 in either state, the store of registers and control words occurs as previously described and an interrogate interrupt operator is forced at the end of the store.

6-209. If the operator occurs in either processor while in the normal state, the processor is not changed to control state by the operator.

6-210. GENERAL. When operating in the normal state and an interrupt occurs, all necessary registers and flip flops are stored in the stack to allow the program to be continued after the interrupt has been processed. Following the interrupt, processor 1 is placed in the control state and the address of the cell assigned to the interrupt is transferred to the C register. All interrupts are processed on a priority basis. The address assigned to the interrupt is given and they are placed in priority sequence. These interrupts are given as follows in priority sequence:

Interrupt	Interrupt Flip Flop	Octal Cell Address
Processor 1 Memory Parity Error	PkI01F*	60
Processor 1 Invalid Address	PkI02F*	61
Time Interval	CCI03F	22
I/O Busy	CCI04F	23
Keyboard Request	CCI05F	24
I/O Control Unit 1 Finished	CCI08F	27
I/O Control Unit 2 Finished	CCI09F	30
I/O Control Unit 3 Finished	CCI10F	31
I/O Control Unit 4 Finished	CCI11F	32
Printer 1 Finished	CCI06F	25
Printer 2 Finished	CCI07F	26

* PkI indicates the processor unit number Interrupt register.

Interrupt	Interrupt Flip Flop	Octal Cell Address
Processor 2 Busy	CCI12F	33
Data-Communication/Transmission Interrupt	CCI13F	34
Not Assigned	CCI14F	35
Disk File #1/Read Check Finished	CCI15F	36
Disk File #2/Read Check Finished	CCI16F	37
Processor 1 Stack Overflow	PkI03F	62
Processor 1 Communication Operator	PkI07F	64
Processor 1 Program Release Operator	PkI05F.PkI07F	65
Processor 1 Continuity Bit	PkI06F.PkI07F	66
Processor 1 Presence Bit	PkI05F.PkI06F.PkI07F	67
Processor 1 Flag Bit	PkI08F	70
Process 1 Invalid Index	PkI05F.PkI08F	71
Processor 1 Exponent Underflow	PkI06F.PkI08F	72
Processor 1 Exponent Overflow	PkI05F.PkI06F.PkI08F	73
Processor 1 Integer Overflow	PkI07F.PkI08F	74
Processor 1 Divide by Zero	PkI05F.PkI07F.PkI08F	75
Processor 2 Memory Parity Error	PkI01F	40
Processor 2 Invalid Address	PkI02F	41
Processor 2 Stack Overflow	PkI03F	42
Processor 2 Communication Operator	PkI07F	44
Processor 2 Program Release Operator	PkI05F.PkI07F	45
Processor 2 Continuity Bit	PkI06F.PkI07F	46
Processor 2 Presence Bit	PkI05F.PkI06F.PkI07F	47
Processor 2 Flag Bit	PkI08F	50
Processor 2 Invalid Index	PkI05F.PkI08F	51
Processor 2 Exponent Underflow	PkI06F.PkI08F	52
Processor 2 Exponent Overflow	PkI05F.PkI06F.PkI08F	53
Processor 2 Integer Overflow	PkI07F.PkI08F	54
Processor 2 Divide by Zero	PkI05F.PkI07F.PkI08F	55

Each cell assigned to an interrupt contains a transfer of control. The transfer of control is simply to a specific portion of the MCP elsewhere in core memory. All possible interrupts are sampled continuously and simultaneously by the hardware.

6-211. Processor 2 cannot operate in the control state. When an interrupt occurs that is associated with Processor 2, the processor performs the action described in the store for interrupt operator, and then idles. When processor 1 is operating in the control state, all interrupts remain set until an interrogate interrupt operator is executed by processor 1.

6-212. The action of the store for interrupt operator is to adjust the stack so that any information that happens to be in the A and B registers, if valid, is pushed down into core stack, and on top of these words place an interrupt control word (ICW), an interrupt return control word (IRCW), and to store the core address of the IRCW in R+10 (octal) of the PRT (of the program running at the time this syllable was executed) as an initiate control word (INCW). Within the ICW and IRCW, all registers within the processor will be stored except the state of the CWMF and the S register setting for the address of the top word of the stack (IRCW). The latter two elements are stored in the INCW which was placed in R+10 (octal) of the PRT.

6-213. This syllable is available for programmatic usage or may be forced into the T register by an interrupt being present and the processor operating in normal state. In the case of the interrupt, the syllable code is placed in the T register through hardware functions. When this syllable is executed, the following registers and flip flops will be reset after the store of the control words: R, PROF, MSFF, SALF, CWMF, BROF, AROF and NCSF. The F register will be left pointing at the last MSCW or RCW placed in the stack if this syllable was executed in word mode. If executed in character mode, the F register would be left pointing at the IRCW. The S register will be left pointing at the IRCW unless an Interrogate Interrupt syllable was executed after this syllable. (Refer to ITI syllable.)

6-214. FORCED STORE FOR INTERRUPT. A processor that is designated as number 1 through the display and distribution unit, and that is operating in normal state, will check the IAR in central control (through hardware logic) for the presence of an interrupt in the system after the completion of each syllable executed. If an interrupt is present (IAR not equal to zero), this syllable, through hardware functions, will be placed in the T register. The subsequent action will be to generate the ICW, IRCW, and INCW (ILCW if in character mode). After completing the generation of these control words, the NCSF and CWMF are reset, placing the processor in control state. An Interrogate Interrupt syllable (ITI) code will then be placed in the T register for execution. For subsequent action, refer to the ITI.

6-215. A processor that is designated as number 2, and is operating in normal state, will check its own interrupt register (UIxF'S) for an interrupt after the completion of each syllable executed. If an interrupt is present (UIxF's not equal to zero), the SFI syllable code, through hardware functions, will be placed in the T register. The subsequent action will be to generate the ICW, IRCW and INCW (ILCW if in character mode). After completing the generation of these control words, the NCSF and CWMF are reset. However, since this processor is designated as number 2, TROF is reset, causing it to idle. This idle condition will remain until the processor is reinitiated by an Initiate P2 syllable being executed by the processor designated as number 1.

6-216. PROGRAMMATIC USE OF STORE FOR INTERRUPT SYLLABLE. If this syllable is used programmatically, either in control or normal state, word mode or character mode, the ICW, IRCW and INCW will be generated and stored. The state of the NCSF and CWMF will be unaltered. If at the time this syllable is executed the processor is in the normal state (NCSF=1), the syllable following the SFI syllable will be executed. The S register will be left pointing at the IRCW and the R register equal to zero. If at the time this syllable is executed the processor is in control state, an IINL will be forced as the next instruction. In either case, the F register will be pointing as previously described.

Sequence of the Store for Interrupt Syllable (See Paragraph 5-39 through 5-41)

6-217. WORD MODE. If the A and B registers are full, they are pushed into the core portion of the stack. The B register is cleared. An ICW is generated by transferring the contents of the R register to bits 14 through 16 of the B register; the status of MSFF is bit 16; the present status of SALF is transferred to bit 17; the contents of the N register, to bits 29 through 32; and the M register, to bits 33 through 47. The word is marked as a control word by setting bits 0 and 1 and resetting bit 3, then the B register is stored in the core portion of the stack with the use of the S register.

6-218. The IRCW is then generated in the B register by transferring the contents of the C register to bits 33 through 47; the F register, to bits 18 through 32; the K register to bits 15 through 17; the G register, to bits 12 through 14; the L register, to bits 10 and 11; the V register, to bits 7 through 9; the H register, to bits 4 through 6; and the status of the BROF to bit 2. The word is then marked as a control word and stored in the core portion of the stack on top of the ICW using the S register.

6-219. The INCW is then constructed by transferring the contents of the S register, which is pointing at the IRCW, to bits 33 through 47; the state of the CWMF is transferred to bit 32; and the remaining bits will contain the same configuration as was stored in the IRCW. The word is then stored in R+10 (octal) using the M register.

6-220. As a result of the SFI syllable, the stack would have, from top to bottom, the following words: IRCW, ICW, A register (if valid), B register (if valid), top of stack before interrupt.

6-221. CHARACTER MODE. If the A register is valid, it is stored in the core stack of memory. If the B register is valid, it is then stored on top of the A register word. The contents of the X register is then stored on top of the B register word as an ILCW. On top of the ILCW, the ICW and the IRCW are stored as described in paragraphs 6-217 through 6-220.

6-222. At the completion of this syllable, the stack would have the following words from top to bottom: IRCW, ICW, ILCW, B register (if valid), A register (if valid), loop control words (LCW), if any loops have been entered, and RCW.

ZPI Conditional Halt (CHPL) 2411

6-223. If the STOP OPERATOR switch on the maintenance panel is in the "stop" position, the processor is halted by stopping the processor clock; otherwise, this operator is a no-op.

6-224. The conditional halt operator is for maintenance and/or debugging purposes only. It is operative in either word or character mode.

XRT Set Variant (VARL) 0061

6-225. The processor is set to the program level. If the processor is in the sub-program level on entering this operator, the VARF is set to indicate that the level was changed to program level.

6-226. This operator is intended for use immediately prior to the following syllables:

Operand Call syllable
Descriptor Call syllable
Load operator
Store operators

6-227. It is intended to set the processor temporarily to program level, where necessary, to provide R-relative addressing with a span of 1024 (decimal) words.

6-228. The operators listed above automatically set the processor back to sub-level after the relative address is constructed, if the VARF is set.

6-229. This operator is not intended for use prior to other operators, such as program release and I/O release, which also use relative addressing. These operators do not include the facility for returning to sub-program level after constructing the address.

6-230. Interrupts are not inhibited between the set variant operator and the subsequent, associated operator. The store for interrupt and return to normal operations automatically store and automatically reinstate the conditions of the SALF and VARF.

6-231. There are no syllable dependent processor interrupts associated with the set variant operator. Syllable dependent processor interrupts occurring during an operand call, descriptor call, load, or store operator do not terminate the operator until the VARF is reset and the SALF adjusted, as appropriate. For syllable independent processor interrupts occurring during the operand call, descriptor call, load or store following a set variant operator, the conditions of the SALF and VARF are undefined.

SFT Store for Test (STFL) 3411

6-232. The store for test operation is included for test and diagnostic purposes. It is normally hardware initiated and used as an automatic part of the test procedure described under the test initiate operator, but may also be used as a programed operator. The operation can take place in either word mode or character mode and in either control state or normal state.

Operation is as follows:

a. The processor is set to the control state.

b. The content of the A register is placed in the stack unconditionally.

c. The content of the B register is placed in the stack unconditionally.

d. An ILCW is built up and placed in the stack.

e. An ICW is built up and placed in the stack.

f. An IRCW is built up and placed in the stack.

6-233. The resulting stack in memory is the same as for a character mode interrupt with the A and B registers full.

a. An initiate test control word is built up and stored in memory location R + 10.

b. The processor is set to word mode and program level. The M and R registers and the MSFF are set to 0.

c. If the operation takes place in processor 2, the processor is left idle. If the operation takes place in processor 1, the operation continues as follows:

   The K, V, G, and H registers are set to zero. A word is read from memory, absolute location 0, and placed in the B register. The low-order 15 bits are transferred to C and the operator is exited. The L register is set to zero. (A normal fetch cycle follows and the instruction, whose address had been stored in memory location 0, is fetched and executed.)

SSF Set or Store S or F Registers (FXSL) 2141

6-234. This operator uses the top two words of the stack. Based on the contents of the top word in the stack, either the F or S register will be set from or stored into the second word in the stack. The top word is deleted from the stack. If the F or S register is set, the second word is also deleted. If the F or S register is stored, the second word is left in the stack. If the F register is set, the processor is set to sub-program level. The following illustration shows how the type of operation is selected, based upon the two low-order bits in the A register.

A47	A46	Operation
0	0	B [18=>32]←F STORE
0	1	F←B[18=>32]; SALF ← 1 SET
1	0	B [13=>47]←S STORE
1	1	S←B[13=>47] SET

FBS Flag Bit Search (SSFL) 7031

6-235. Pushup into A occurs if necessary. The operator uses the low-order 15 bits of the top word in the stack as the base address for a search. The flag bit of the word found in the location specified by this address is examined.

6-236. If the flag bit is one, the address is placed in the A register in the low-order 15 bits. The rest of the word is set to zero, and the word is marked as a present data descriptor. The operator is exited.

6-237. If the flag bit is off, the address is increased by one and the flag bit of the word found in this location is tested. This process continues until a word is found with the flag bit on.

LLL Link List Lookup (LLLL) 2541

6-238. This operator scans a linked list of indefinite length and tests a field in each list word against the corresponding field in the A register.

6-239. Pushup occurs into the A and B registers if required. The top two words in the stack are presumed to contain the following:

	9		32
A Register	XXX	TEST FIELD	OOOOO
B Register	XXX	XXXXXXXX	Link Address

6-240. The complete test field can be used or any portion of the more significant end of it can be used. Bits on the less significant end are effectively eliminated from the test field by setting them to zero in the test word in the top of the stack.

6-241. The word addressed by the initial link address is read from memory to the B register. The test field of the word in the B register is compared with the field in the corresponding position in the A register.

6-242. If B field ≥ A field, the address that was used to access the link is left in the A register as a present data descriptor, with the remainder of the word set to zero. The list word is left in the B register. The operator is exited.

6-243. If the B field < A field, the link address in the B register is used to access the next word from memory. The process continues until a link word meeting the test condition is found.

TUS Interrogate Peripheral Status (IPSL) 2431

6-244. This operator places in the top of the stack a word representing the current ready status of the peripheral equipment. One bit in the word is associated with each peripheral unit. This bit is set to 1 if the associated unit is ready, to 0 if the associated unit is not ready.

6-245. The stack is adjusted so that the A register is empty. The A register is set to zero. The low-order bits are set to reflect the current status of the peripheral units. The A register is marked full and the operation is terminated. Table 6-2 shows the association between the A register bit positions and peripheral units.

NOTE

A magnetic tape transport is reported as ready only when the tape is stationary and it is otherwise ready. It is reported as not ready if the tape is rewinding, or if the tape is still indexing to a stop following an operation.

TABLE 6-2

Relation Between A Register Bit Position and Peripheral Unit

A Register Bit Position	Unit Designate	Peripheral Unit
47	1	Magnetic Tape A
46	3	Magnetic Tape B
45	5	Magnetic Tape C
44	7	Magnetic Tape D
43	9	Magnetic Tape E
42	11	Magnetic Tape F
41	13	Magnetic Tape H
40	15	Magnetic Tape J
39	17	Magnetic Tape K
38	19	Magnetic Tape L
37	21	Magnetic Tape M
36	23	Magnetic Tape N
35	25	Magnetic Tape P
34	27	Magnetic Tape R
33	29	Magnetic Tape S
32	31	Magnetic Tape T
31	4	Drum 1
30	8	Drum 2

TABLE 6-2 (Cont)

Relation Between A Register Bit Position and Peripheral Unit

A Register Bit Position	Unit Designate	Peripheral Unit
29	6	Disk File 1
28	12	Disk File 2
27	22	Printer 1
26	26	Printer 2
25	10	Card Punch
24	10	Card Reader 1
23	14	Card Reader 2
22	30	SPO-Keyboard
21	18	Paper Tape Punch 1
20	18	Paper Tape Reader 1
19	20	Paper Tape Reader 2
18	20	Paper Tape Punch 2
17	16	Data Communication Control

TIO Interrogate I/O Channels (TIOL) 6431

6-246. This operator interrogates the I/O channels to determine which channel is currently in line to be assigned next, that is, which is the lowest numbered currently available I/O control unit. Pushdown occurs if necessary and a literal is placed in the top of the stack. The literal indicates the next assigned channel in the following way:

Literal	Channel
0	All channels busy
1	Channel one due for assignment
2	Channel two due for assignment
3	Channel three due for assignment
4	Channel four due for assignment

Control State Operators

6-247. The following operators are used only when operating in the control state. If these operators are encountered when operating in the normal state, they are treated as no-ops.

ITI Interrogate Interrupt (IINL) 0211

6-248. If any interrupt bit is set, the C register is loaded with the 6-bits address which corresponds to the highest priority interrupt bit that is set. The interrupt bit(s) creating this address is reset. The L register is cleared. The S register is set to 64 (100 octal). If no interrupt bit is set, control continues in sequence.

IOR I/O Release (IORL) 2111

6-249. If the flag bit and the presence bit of the word in the A register are both one, the contents of the cell addressed by the fifteen low-order bits of the word in the A register are placed in the A register. The presence bit of the word in the A register is set to one and the word stored back to the cell addressed. The A register is set to empty.

6-250. If the flag bit of the word in the A register is zero, relative addressing takes place as specified for the program release operator.

6-251. If the flag bit of the word in the A register is one and the presence bit zero, the operation is terminated. The presence bit interrupt is not set.

IIO Initiate I/O (IOOL) 4411

6-252. The word in the A register is stored in location 10 (octal) and A register is set to empty. An initiate I/O signal is sent to central control for selection of an I/O channel. The processor proceeds to the next syllable.

6-253. The I/O busy bit is set in the interrupt register if all I/O channels are busy.

IP1 Initiate P1 (INIL) 4111

6-254. The 15 low-order bits in the initiate control word in the A register are transferred to the S register. Bit 32 is transferred to the CWMF. Processor 1 is set to the normal state.

IP2 Initiate P2 (PTOL) 4211

6-255. The initiate control word in the A register is stored in memory location 10 (octal) and the A register is set to empty. An initiate P2 signal is sent to the central control unit and processor 1 proceeds to the next syllable.

6-256. The central control unit sends a control signal to processor 2. Under control of this signal, the initiate control word is transferred from memory location 10 (octal) to the A register in processor 2. Processor 2 then performs the initiate operations as described in paragraph 6-254, initiate P1.

6-257. If processor 2 is not idle or not available, the P2 busy bit is set in the interrupt register by the central control unit.

HP2 Halt P2 (HP2L) 2211

6-258. This operator causes processor 2 to store its registers just as if a P2 interrupt had occurred. If processor 2 is busy, the halt operator in processor 1 is held up. The operator in processor 1 is completed after all appropriate processor 2 registers are stored for interrupt. Processor 2 is left idle.

6-259. If processor 2 is not ready or is absent, the halt P2 operation is immediately terminated.

RTR Read Timer (RDTL) 0411

6-260. The 6-bit timer setting along with the timer interval interrupt setting as the 7th (most significant) bit is placed in the A register as an integer (bits 41 => 47).

IFT Test Initiate (IFTL) 5111

6-261. This operator is intended for test and diagnostic purposes only. It may be used only in the control state; otherwise, it is a no-op. Certain registers, involved in this operator, are used only for internal control and are not described elsewhere in this specification.

6-262. The 15 low-order bits of the initiate test control word in the A register are transferred to the S register.

6-263. The S register, mode bit, and the Q, Y, and Z fields are transferred to the corresponding register and flip flops from the initiate test control word in the A register. The J field and the NCSF, CCCF, MROF, and MWOF bits are transferred from the initiate test control word to temporary storage flip flops (TM 6 =>1).

6-264. The IRCW whose address is specified by the S register, is read. Its contents are distributed to the C, F, K, G, L, V and H registers. The state of bit 2 is unconditionally transferred to the BROF.

6-265. The S register is reduced by 1. The ICW addressed by the S register, is read. The R, mark stack, program level, variant, and M and N fields, are transferred to the corresponding registers and flip flops.

6-266. The S register is reduced by 1. Bits 19 to 47 of the ILCW, addressed by the S register, are transferred to the X register. Bit 2 of the ILCW is transferred to the AROF.

6-267. The S register is reduced by 1 and the word addressed by the S register, is transferred to the B register.

6-268. The S register is reduced by 1 and the word addressed by the S register, is transferred to the A register.

6-269. If the CWMF has been set (from the initiate test control word), the contents of the S register and the S field in the X register are interchanged.

6-270. The contents of the temporary storage flip flops (set from the initiate test control word) are transferred to the four J flip flops. and the NCSF, CCCF, MROF, and MWOF flip flops. If the processor is being initiated to word mode with CCCF=1, the S register is counted down one extra count to address the second cell below that from which the contents of the A register was accessed. The operator is terminated without a normal exit by resetting the TROF flip flop.

6-271. The operator specified by C and L (set by the RCW) is fetched The operator is entered at the J level specified in the initiate test control word.

6-272. If CCCF=0, the operator continues to its normal exit, no store-for-test operation takes place, and the next syllable is executed as usual. If CCCF=1, operation is as follows: After the fetch, a pulse time of operation at the specified J level is performed.

6-273. If no access has been initiated, the CCCF terminates the operation after one pulse time. If a memory access has been initiated, the CCCF terminates the operator after the memory access is complete or after a related set of uninterruptable memory accesses is complete. The operator is terminated without normal exit by resetting the TROF.

6-274. If the CCCF equals 1, it causes the following actions:

a. stores the centents of the J flip flops and the NCSF in the temporary storage flip flops

b. jams the store-for-test operator into the T register

c. enables the T register

The remainder of the operation is an described under the store for test operator. If an interrupt has occurred, it is not answered at this point. The store for test operator has priority over the store for interrupt operation.

SECTION 7

CHARACTER MODE OPERATION

GENERAL

7-1. Each processor of the B 5500 can operate in two modes; word mode or character mode. In word mode, the operators mainly function with individual words. In character mode, operations usually occur with individual characters or parts of characters (bits). Since the functions of various registers are different in the two modes, there is a completely different list of operators. Each processor contains a flip flop to control the two modes. This flip flop is the character-word mode flip flop (CWMF). It is on when operation is in character mode. Character mode operation is in sub-program level.

FUNCTION

7-2. In character mode, the primary areas of memory used are the source string and the destination string. Basically, both of these strings can be thought of as continuous strings of characters or character positions. The function of any character mode operation is to take information (characters) from the source string, process them, and transfer them to the destination string. It is possible to skip characters in either string, compare an equal length field of the two strings, add two fields (one in each string), place the result in the destination string, etc. Unless programed otherwise, characters are always handled sequentially from the most significant character to the least significant character within the word.

CHARACTER MODE DATA REPRESENTATION

Alphanumeric

7-3. Alphanumeric information is represented by characters consisting of six bits; B, A, 8, 4, 2, and 1. With these 6 bits, there are 64 combinations of bits which can form 64 characters. These 64 characters are the alphabetic characters A through Z, numeric characters 0 through 9, and 28 special characters (see Appendix A). Alphanumeric information is stored internally as words of 8 characters. Characters are grouped together in successive words to form a string. The first character of a string is in the word with the lowest memory location. The character following the last character of a word is the first character in the word with the next higher memory location (figure 7-1). The string usually consists of fields of various types of information. A field is just a grouping of information within a string. For instance, a sales evaluation record might be as follows:

String Character Position	Information Contained in Field
1 through 12	Stock number
13 through 30	Item name
31 through 45	Year-to-date sales
46 through 60	Previous year sales, same period
61 through 75	Difference in sales totals

7-4. In this example, the first field (stock number) might contain mixed alphanumeric characters. The second field (item name) would probably contain only alphabetic characters, and the last three fields would be numeric information. The last field (difference in sales totals) points to another consideration. Since a number may be either positive or negative, some provision must be made for the sign of a numeric field. The sign of a numeric field is contained in the least significant character of a particular field. This is shown in figure 7-1. In the above example, the signs for the respective field would be in character positions 45, 60, and 75.

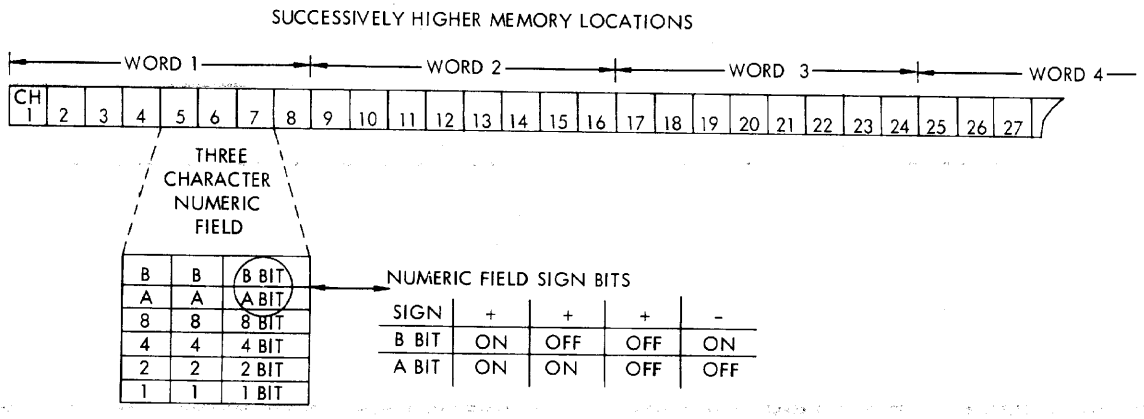


Figure 7-1. Alphanumeric Character String

Numeric

7-5. A numeric field has a negative sign only if the least significant character has the B bit on and A bit off. All other combinations result in a positive sign. The A and B bits of all the other characters in a numeric field are 0 (off) and are not checked for signs. With 6 bits per character, each character will occupy two octades. The characters are numbered from left to right, as 0 through 7 (figure 7-2).

7-6. Within a given character, the bits of that character are numbered from top to bottom and then from left to right. The B bit then is bit 0 of a character, while the 1 bit is bit 5. With the character number and bit number within the character, any one bit can be referenced. Therefore, character 0 encompasses the register bit positions 0 through 5. Accordingly, the 3rd bit of the 7th character is bit 44 of the register, and would be bit 2 within the character.

CHARACTER MODE ADDRESSING

7-7. In character mode, most addressing pertains to the source string and the destination string. The source string is associated with the A register. The destination string is associated with the B register All processing between the two strings is accomplished by use of the serial adder and the Y and Z registers. Character mode addressing is accomplished using the G, H, K, and V registers along with the S and M registers.

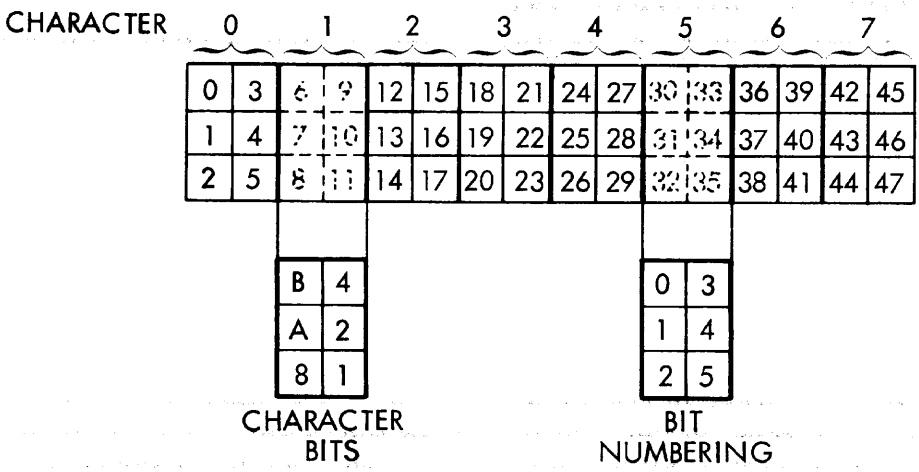


Figure 7-2. Alphanumeric Word

Source String Addressing

7-8. Normally, one word of the source string is contained in the A register. The address of this word is contained in the M register. When a new source string word is needed, the M register is counted up by one. The memory access is requested using the M register for the address while the A register receives the word accessed. The word in the A register contains 8 characters. Each character must be addressed in order to be transferred to the Y register. Gating circuits are included so that any character position of the A register can be shifted to the Y register. This gating network is controlled by the G register. The H register is used to select a particular bit position from the character contained in the Y register. The bit selected is either shifted to the Z register or compared to the bit already in the Z register. This allows for a bit-by-bit comparison of characters.

Destination String Addressing

7-9. Normally, one word of the destination string is contained in the B register. The core address of this particular word is contained in the S register. When all characters of a particular word in the destination string are processed, the word in the B register must be stored back in memory. Thus, the S register is used for both reading and writing operations. After a word is stored, the S register is counted up by one; and the next word is read out of memory and placed in the B register for processing. There are no gating circuits to allow every character position of the B register to be shifted to the Z register, or vice versa. However, there are two character positions for shifting to the Z register, and two for shifting from the Z register to the B register Thus, the word in the B register must be shifted so that the required character is in one of these positions. The B register contains circuitry for shifting by octade, either to the right or to the left. A character shifted out of one end is placed in the other end; thus, a circulation of the contents of the B register is accomplished with no loss of information. The N register counts the shifts of the B register. When the B register is shifted to the right, the N register is counted down one for each octal shift. Under normal conditions when the word in the B register is in the normal position, the N register is equal to 0. The N register may be pre-set to enable proper character alignment when the character positions differ from normal positioning.

7-10. Two octal shifts are required for a complete character position shift. A character may be shifted to or from the B register, from or to the Z register, only via certain positions. These positions are called alignment stations. The normal alignment station for shifting from the B register to the Z register is octades 16 and 15. This is character position 0 when the word is in the normal position. If the B register is shifted left two octades, then character 1 of the word is in the alignment station; and the N register has been counted up to 2. The three high-order bits of the N register designate which character of the word is in the normal outgoing alignment station. This is possible since each two shifts (one octade per shift) of the B register correspond to one full character shift. The three high order bits of the N register can be considered a character counter when the 1 bit of the N register if off. Thus, it is possible to determine which character is in the normal outgoing alignment station. This is shown graphically in figure 7-3. The K register is used to select a specific character of the word that is to be shifted to the Z register, and the N register reflects the number of the character which is in the alignment station.

7-11. By comparing the K register to the high-order bit of the N register and by shifting the B register until an equal condition exists, with the 1 bit of the N register off, the selected character can be placed in the alignment station. When a character in the Z register is to be shifted back to the B register, it will be placed in the normal position designated as octades 2 and 1. When a character is placed back in the B register, the next character is in position to be shifted to the Z register. When adding, the serial adder adds the contents of the Y register to the contents of the Z register and places the result in the Z register. For bit operations, the contents of the V register select the bit contained by the character in the Z register. This is used when bits are to be placed in the Z register from the Y register, or when setting or resetting bits in the destination strings.

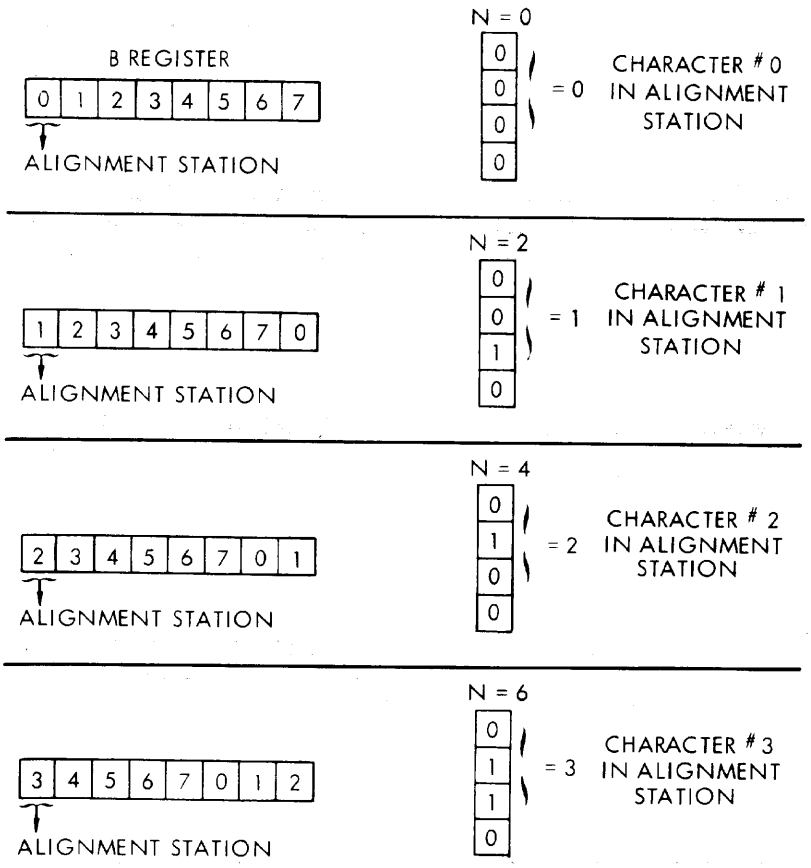


Figure 7-3. Outgoing Alignment Station

7-12. When whole characters are to be shifted from the Y register to the destination string, they are transferred directly to the B register alignment station from the Y register. This eliminates passing through the Z register. When comparing one character of the source string against a given character, the 6 high-order bits of the T register provide for shifting to the Z register. Information may be shifted to the destination string from the program syllable string. Characters which had been placed in the program syllable string are shifted to the Y register, which is then placed into the B register.

7-13. In addition to addressing destination string words, the S register is also used to store loop control words in the stack. The contents of the S register are saved for restoration and the S register is set to the address of the last control word in the stack It is counted up by one, and the control word is stored. The S register is then restored.

7-14. When the processor entered character mode operation, an RCW was generated and stored in the stack. The S register is also used to address the stack below the RCW. This is to obtain a destination string address. The address of the RCW is in the F register. The contents of the F register are placed into the S register, then the S register is counted down by an amount specified in the program syllable. This word is accessed, and the address is transferred to the S register. The M register also addresses the stack below the RCW, in the same manner as the S register. This is for a new source string address.

ENTRANCE TO CHARACTER MODE

7-15. Character mode may be entered through an Operand Call syllable or a Descriptor Call syllable referencing a program descriptor that is marked character mode. In these cases, the program descriptor would have

both the mode bit and the argument bit on. When the program descriptor is marked for character mode entry, and the MSFF is on, an RCW will be placed in the stack (per subroutine action), and the CWMF set. The program segment string is initiated in character mode at the address pointed to by the program descriptor. Another method to enter character mode is with the enter character mode in line (CMN) operator (see section 6 for complete discussion). It results in a change to sub-program level of operation and generates the RCW.

7-16. Consider the first method of entry and assume an MSCW has already been placed in the stack (MSFF will be on). With these conditions, an RCW is generated and placed in the stack on top of the parameter entered last. The C register is set from the address contained in the lower 15 bits of the program descriptor. The address in the S register (address of the RCW in core stack) is placed in the F register. The F register now points to the RCW. The previous setting of the F register was stored in bits 18 through 32 of the Return Control Word. The address of the RCW, contained in the S register, is also placed into the X register. Prior to this transfer, the X register would have been cleared. The X register will be used to locate the top control word of the stack (not always an RCW). This is necessary because the S register is not used to address the stack when operating in character mode, but instead to address the destination string.

7-17. The "begin loop" operator in character mode will cause a loop control word (LCW) to be stored in the stack each time it is encountered in the program segment string. Through the address found in the X register, it is possible to determine where to store the LCW in the stack. The F register always points to the RCW while operating in character mode. The R and S registers are cleared so that they may be used for the tally register and the destination string address registers respectively. Both the CWMF and the SALF are set to one. The Operand/Descriptor call syllable which caused operation to enter the character mode is then terminated with the MSFF reset (off). The next syllable to be executed will be the first syllable at the address specified by the program descriptor. The program segment string will be decoded as character mode operators since the CWMF is on.

Character Mode Syllable Decoding

7-18. Character mode syllables use the lower bits to determine the type of syllable. The upper 6 bits are referred to as a repeat field. The repeat field of a character mode syllable determines the number of characters, or bits, with which a particular syllable will work. With 6 bits in the repeat field, the maximum number of characters or bits in one string that any one syllable can work with is 63 (decimal). See figure 7-4.

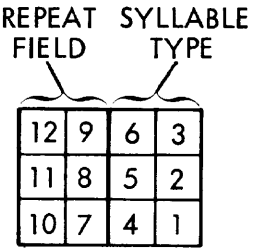


Figure 7-4. Character Mode Syllable

Character Mode Loops

7-19. In character mode, when a group of syllables are to be repeated more than once, a loop is formed using two syllables. These syllables are the begin loop operator and the end loop operator. Consider the following character mode program segments strings where BNS indicates begin loop operator; BIS, set bit operator; and ENS, end loop operator:

05:BNS - 01:BIS - ENS

The resulting action will be to set 5 bits in the destination string starting with the bit address by the S, K, and V registers. As this program string is executed, the following actions will occur.

7-20. When the begin loop operator (05:BNS) is executed, an LCW containing the present contents of the X register will be generated and placed in the stack. If no other begin loop operators were encountered before this begin loop operator, the present contents of the X register will only be the address of the RCW. This address was stored when

character mode was entered (due to an Operand/Descriptor Call syllable executed on a program descriptor marked for character mode or an enter character mode in-line syllable). If a begin loop operator had been executed prior to this begin loop operator, the word stored in the stack would have the format of an LCW. In either case, the word placed in the stack will be called an LCW and is used only in character mode operation.

7-21. After the contents of the X register are stored, the X register will be replaced by the following information: the repeat field of this operator (05:BNS) minus 1; the L count and the C count of the syllable following the begin loop operator; and the address of the top control word of the stack, either the RCW or an LCW. This completes the execution of the Begin Loop Operator syllable (X now contains the most current Loop Control Word.)

7-22. One bit is then set in the destination string due to the Set Bit Operator syllable 01:BIS. The End Loop syllable is then encountered. The function of this syllable is to determine if the loop has been repeated the desired number of times specified by the begin loop operator. If not, control returns to the syllable that follows the corresponding begin loop operator. The return point has been stored in the L and C register field of the X register. To determine if the proper number of loops have been executed, the end loop operator will check the repeat count field bits of the X register for being equal to 0. If they are not equal to 0, they are reduced by 1 and the C and L registers are set to their respective fields from the X registers. If the repeat count bits are equal to 0, the contents of the X register are replaced by the word addressed by the X register. This is the address of an LCW or RCW. The syllable following the end loop operator is then executed.

7-23. LOOP CONTROL WORD (LCW) DESCRIPTION. In character mode, whenever a loop is formed, several settings must be retained such as the beginning point of the loop. This beginning point is the point in the program to which return occurs when the loop is to be reiterated. This means that the contents of the C and L registers must be retained. Another value to be retained is the number of times the loop is to be repeated. This latter value must be corrected with every pass through the loop. This is called the repeat field. If loops are nested within loops, information pertaining to the previous loops are stored as LCW's in the stack above the RCW.

7-24. The X register is a 39 bit register used to contain the information on the current loop. Thus, the X register contains the return point, repeat field, and the address of the previous LCW. When a new loop is formed, the contents of the X register are used to create a new LCW. The information concerning the new loop is stored in the X register, along with the address of the LCW just formed. When a loop is ended, the address in the X register is used to access the last LCW. The X register contents are then replaced by the information stored in the LCW just accessed. This way, conditions are restored for the previous loop.

7-25. LOOP CONTROL WORD FORMAT. The format of the LCW is as follows:

Bit 0	On marks this word as not being an operand.
Bit 1	On marks this word as being a control word or a program descriptor.
Bit 2	Not used.
Bit 3	Off marks this word as a control word as opposed to a program descriptor.
Bits 4 - 9	Not used.
Bits 10 - 11	Used to store the contents of the L register.
Bits 12 - 17	These bits are used to store the repeat field which had been in the X register.
Bits 18 - 32	These bits are used to store the address of the preceding control word. This address will have been in the X register. The field is labeled as F, but will contain the setting of the F register only when the first LCW is stored after entering character mode.

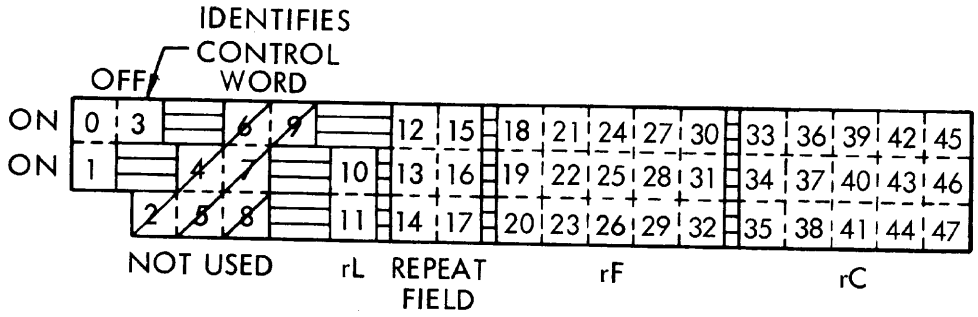


Figure 7-5. Loop Control Word Exlpoded

Bits 33-47 These bits are used to store the contents of the C register. This will be the address of the program word that contains the return point of the associated loop (following the begin loop syllable).

7-26. INTERRUPT LOOP CONTROL WORD (ILCW) DESCRIPTION. If an interrupt condition requires that the contents of the registers be stored, then the information in the X register must also be stored. This will create the interrupt loop control word, ILCW.

7-27. INTERRUPT LOOP CONTROL WORD FORMAT. With the exception of one field, the ILCW is identical to the LCW in format. In character mode, when an interrupt condition causes the register of a processor to be stored, there may be a word of the destination string in the B register. This destination string word must be stored so that it will not be lost. It is stored in the program's stack just above the last LCW. The A register contents, if valid, are stored prior to the storing of the B register. Bit 2 is set in the ILCW to indicate if the A register was valid and stored. Bit 2 of the IRCW is set if the B register contents were valid and stored in the stack. To do this, the contents of the S register (address of the current destination string word) and the address of the previous control word (in the X register) are interchanged. The S register is then counted up and the destination string word is stored. The contents of the X register is then used to form the ILCW. Bits 18 through 32 will contain the address of the destination string word that had been in the B register. For this reason, this field is labeled as S (figure 7-6).

EXIT FROM CHARACTER MODE

7-28. Character mode may be exited by either of two operators: the exit character mode (EXC) or the exit character mode in line (CMX) operators. A complete discussion of these operators is given in section 6. If entry to the character mode was made by an operand/descriptor call executed on a program descriptor, the EXC operator must be used to exit character mode. However, if entry was made by the enter character mode in line operator (CMN), the CMX operator must be used.

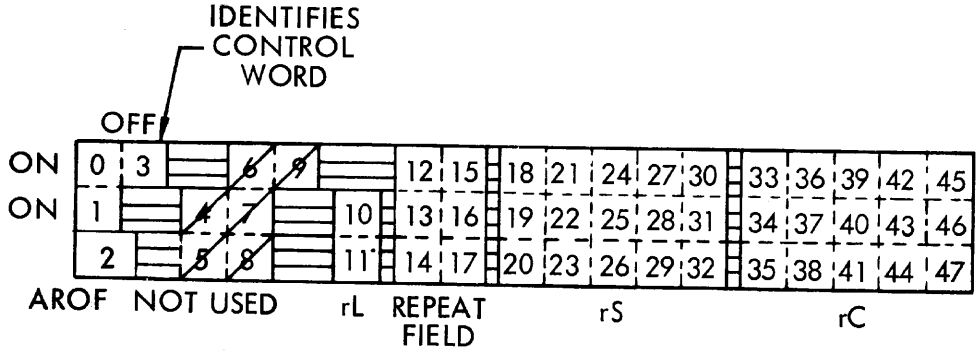


Figure 7-6. Interrupt Loop Control Word Exploded

SECTION 8

CHARACTER MODE OPERATORS

GENERAL

8-1. The Character Mode Operator syllables apply to Stream Procedures in ALGOL and most of COBOL. The registers used to address the source and destination strings are explained in the following paragraphs.

CHARACTER MODE ADDRESSING

Source String

8-2. The M register will be used to select the word from the source string that is to be placed in the A register. The G register will be used to point at the desired character within the word pointed to by M. The H register is used to select a bit within the character pointed at by G. The bit pointer (H) will count from 0 through 5 and back to zero. If the register is caused to overflow (5 to 0), the G register will be incremented by one automatically. The G register is capable of counting from 0 through 7, then back to 0. If the register is cycled from 7 to 0, this will cause the word pointer (M) to be increased by one automatically.

Destination String

8-3. The S register will be used to point at the destination string area and the word accessed will be placed in B register. The K register will be used to select the desired character within the word pointed to by S. The counting capabilities of the K register are the same as those described for the G register of the source string. The V register is used to select a bit within the character pointed at by K. The V register also has counting capabilities like those of the H register in the source string.

Character Movement

8-4. Successive characters proceed from left to right within a word (character 0, (bit 0-5) to character 7, (bits 42-47)) and to consecutively higher memory addresses by word. At the completion of the operation, the next bit, character, or word in sequence will be pointed at unless the syllable executed was only a test operator. A test operates on a single bit or character within a given string and does not alter the setting of the registers.

Address Adjustment

8-5. Prior to the execution of any operator which operates on whole characters or words of the source string (in A register) or destination string (in B register), the corresponding lower-order pointers will be tested for zero. If the respective registers are or are not zero, they will be set to zero, with a possible overflow to the next higher-order register. For example, if a Transfer Word syllable is executed, the source string registers associated with bits and characters will be tested for zero. If the H register is not zero, it will be set to zero, and the G register increased by one. The G register is then tested for zero, and if not zero, it is set to zero, and the M register is increased by one. This same action will occur in the destination string except the registers tested will be the V and K registers. When the syllable finds that the lower order registers are equal to zero, it will then allow the transferring of the words from the source string to the destination string.

OPERATOR SYLLABLES

8-6. The following general limitations apply to character mode operators. Operation in violation of these limitations may produce undefined results.

Operations Involving Memory Accesses to Source and Destination Areas

8-7. At no time during the operation may the source address and the destination address refer to the same word in memory. Access should not be made to the location specified by the current destination address unless that destination address is at a word boundary (K = V = 0). These general limitations apply in all cases.

8-8. Prior to the execution of some operators the H register is tested for zero. If the H register is not equal to zero, it is set to zero and the G register is increased by one. Overflow into the M register can occur.

8-9. Prior to the execution of still other operators the V register is tested for zero. If the V register is not equal to zero, it is set to zero and the K register increased by one. Overflow into the S register can occur.

8-10. If the repeat field is equal to zero, the operator is terminated after any adjustment of address registers as described above.

8-11. Successive characters proceed from left to right within a word and to consecutively higher memory addresses by word. At the completion of the operation, the M, G, S and K registers specify the next source and destination character in sequence. For those operators that operate only on the M and G registers, the S and K registers remain unchanged. For those operators that operate only on the S and K registers, the M and G registers remain unchanged.

TRANSFER OPERATORS

8-12. Transfer operators transfer information from one area in memory to another. In general, transfer operators proceed from high-order to low-order within a word and to successively higher addresses by word. Transfers may occur from any character position within a word to any character position in another word, subject to the limitations of paragraphs 8-6 thru 8-11.

TRS Transfer Source Characters (TSDL) XX77

8-13. This operator transfers characters from the source string starting at the position specified by the M and G registers to the destination string starting at the position specified by the S and K registers. The number of characters transferred is specified by the repeat field (XX).

TRP Transfer Program Characters (TPDL) XX74

8-14. This operator transfers characters from the program string starting at the position specified by the C and L registers to the destination string starting at the position specified by the S and K registers. The number of characters transferred is specified by the repeat field. When the repeat field is odd, the first character in the program string is skipped. The next program syllable fetched is the syllable following the one containing the last character transferred.

TRZ Transfer Zones (TZDL) XX76

8-15. This operator transfers the zone portion of the characters in the source string at the position specified by the M and G registers to the zone portion of the characters in the destination string starting at the position specified by the S and K registers. The numeric portions of the destination string characters are retained. The number of zones transferred is specified by the repeat field.

TRN Transfer Numeric (TNDL) XX75

8-16. This operator transfers the numeric portion of the character in the source string starting at the position specified by the M and G registers to the numeric portion of the character in the destination string starting at the position specified by the S and K registers. The zone bits of the characters in the destination string are set to zero. If the zone portion of the last source character transferred is minus (BA = 10), the true/false indicator is set to true; otherwise, the true/false indicator is set to false. The number of numeric characters transferred is specified by the repeat field.

TRW Transfer Words (TWDL) XX05

8-17. This operator transfers words from the source string starting at the position specified by the M register to the destination string starting at the position specified by the S register. The number of words transferred is

specified by the repeat field. Prior to the execution of the operator, the G, H, K and V registers are tested for zero. If the G or H registers are not equal to zero, they are set to zero and the M register increased by one. If either the K or V registers are not equal to zero they are set to zero and the S register increased by one. At the completion of the operation, the M and S register specify the next word in sequence. Refer to paragraph 8-10.

TBN Transfer Blanks for Non-Numerics (TBZL) XX12

8-18. This operator tests characters specified by the S and K registers, for less than or equal to zero. If the character is less than or equal to zero, it is replaced by a blank character, and the next character is tested. If the character is greater than zero, the operation is terminated. The maximum number of characters tested is specified by the repeat field. If the operator is terminated because a character tests greater than zero, the S and K registers are left pointing at this first non-zero character. If the operator is terminated because the repeat field has been counted to zero, S and K will point at the character following the field tested. If the field contains all blanks, or if the field length is zero, the True/False Flip Flop (TFFF) is set to 1; otherwise it is set to 0.

TEST OPERATORS

8-19. Test operators provide the ability of testing a character or bit in the source area against a predetermined character or bit. These operations do not cause an advancement in the source area enabling repeated tests of a character. Tests of characters are made on the basis of the collating sequence of characters. The limitations are specified in paragraph 8-8.

TGR Test for Greater (TGTL) XX27

8-20. This operator compares the repeat field and the character in the source string specified by the M and G registers for a "greater than" condition. If the source character is greater than the character in the repeat field, the TFFF is set to true which turns the True/False indicator on (true); otherwise, the indicator is off which indicates that the TFFF is reset (false). The M and G registers are not advanced.

TEG Test for Greater or Equal (TGEL) XX26

8-21. This operator compares the repeat field against the character in the source string specified by the M and G registers for a "greater than" or "equal" condition. If the source character is greater or equal to the repeat field character, the true/false indicator is true; otherwise, the true/false indicator is false. The M and G registers are not advanced.

TEQ Test for Equal (TEQL) XX24

8-22. This operator compares the repeat field with the character in the source string specified by the M and G registers for an "equal" condition. If the condition is met, the true/false indicator is true; otherwise, the true/false indicator is false. The M and G registers are not advanced.

TEL Test for Equal or Less (TLEL) XX34

8-23. This operator compares the repeat field with the character in the source string specified by the M and G registers for a "less than or equal" condition. If the source character is equal or less than the repeat field character, the true/false indicator is true; otherwise, the true/false indicator is false. The M and G registers are not advanced.

TLS Test for Less (TLTL) XX35

8-24. This operator compares the repeat field against the character in the source string specified by the M and G registers for a "less than" condition. If the source character is less than the repeat field character, the true/false indicator is true; otherwise, the true/false indicator is false. The M and G registers are not advanced.

TNE Test for Not Equal (TNEL) XX25

8-25. This operator compares the repeat field with the character in the source string specified by the M and G registers for a "not equal" condition. If the source character is not equal to the repeat field character, the true/false indicator is true, otherwise the indicator is false. The M and G registers are not advanced.

TAN Test for Alphanumeric (TANL) XX36

8-26. This operator compares the repeat field and the character in the source string specified by the M and G registers for a "greater than or equal" condition. If the source character is greater than or equal to the repeat field character and the source character is not the "multiply" character (external code 10 1010) or "not equal" character (external code 01 1010), the true/false indicator is true; otherwise the true/false indicator is false. The M and G registers are not advanced. A "not equal" character in the repeat field compared with a "not equal" character in the source string will result in setting the TFFF to true. A "multiply" character in the repeat field compared with a "multiply" character in the source string will result in setting the TFFF to true.

BIT Test Bit (TEBL) XX37

8-27. This operator tests one bit in the source string at the position specified by the M, G and H registers against the low-order bit of the repeat field. If they are equal, the true/false indicator is true; otherwise, the true/false indicator is false. The M, G and H registers are not advanced.

COMPARISON OPERATORS

8-28. The comparison operators are used for comparing two identical length fields of alphanumeric characters. Comparisons are made on the basis of the collating sequence of characters. Fields may start at any position within a word; word boundaries are ignored. Although the result of the comparison may be known before all characters of the fields have been compared, the address registers are advanced the full amount. The limitations are specified in paragraphs 8-6 thru 8-11.

CGR Compare for Greater (SGTL) XX63

8-29. This operator compares a character field in the source string starting at the position specified by the M and G registers with an identical length character field in the destination string starting at the position specified by the S and K registers for a "greater than" condition. If the field in the source string is greater than the field in the destination string, the true/false indicator is true; otherwise, the true/false indicator is false. The number of characters in each of the fields to be compared is specified by the repeat field.

CEG Compare for Greater or Equal (SGEL) XX62

8-30. This operator compares a character field in the source string starting at the position specified by the M and G registers against an identical length character field in the destination string starting at the position specified by the S and K registers for a "greater than or equal" condition. If the field in the source string is greater than or equal to the field in the destination string, the true/false indicator is true; otherwise, the true/false indicator is false. The number of characters in each of the fields to be compared is specified by the repeat field.

CEQ Compare for Equal (SEQL) XX60

8-31. This operator compares a character field in the source string starting at the position specified by the M and G registers against an identical length character field in the destination string starting at the position specified by the S and K registers for an "equal" condition. If the fields are equal the true/false indicator is true; otherwise, the true/false indicator is false.

CEL Compare for Equal or Less (SLEL) XX70

8-32. This operator compares a character field in the source string starting at the position specified by the M and G registers against an identical length character field in the destination string starting at the position specified by the S and K registers for a "less than or equal" condition. If the field in the source string is equal or less than the field in the destination string, the true/false indicator is true; otherwise, the true/false indicator is false. The number of characters in each of the fields to be compared is specified by the repeat field.

CLS Compare for Less (SLTL) XX71

8-33. This operator compares a character field in the source string starting at the position specified by the M and G registers

against an identical length character field in the destination string starting at the position specified by the S and K registers for a "less than" condition. If the field in the source string is less than the field in the destination string, the true/false indicator is true; otherwise, the true/false indicator is false.

CNE Compare for Not Equal (SNEL) XX61

8-34. This operator compares a field of characters in the source string starting at the position specified by the M and G registers against an identical length character field in the destination string starting at the position specified by the S and K registers for a "not equal" condition. If the fields are not equal, the true/false indicator is true; otherwise, the true/false indicator is false. The number of characters in each of the fields to be compared is specified by the repeat field.

JUMP OPERATORS

8-35. These operators are used to adjust the C and L registers to provide branching in the program string and executing repeated program strings. Limitations are specified in paragraph 8-10.

JFW Jump Forward Unconditional (FWJL) XX47

8-36. This operator initiates an unconditional forward jump. The C and L registers are increased by the contents of the repeat field. Prior to the addition of the repeat field of this operator, the C and L registers contain the address of the next syllable in sequence.

JRV Jump Reverse Unconditional (REJL) XX57

8-37. This operator initiates an unconditional reverse jump. The C and L registers are decreased by the contents of the repeat field. Prior to the subtraction of the repeat field of this operator, the C and L registers contain the address of the next syllable in sequence. Operation with repeat field equal to 1 is a closed loop.

JFC Jump Forward Conditional (CFJL) XX45

8-38. This operator initiates a jump forward unconditional if the true/false indicator is false. If the true/false indicator is true, the next syllable in sequence is fetched. The true/false indicator remains unchanged.

JRC Jump Reverse Conditional (CRJL) XX55

8-39. This operator initiates a jump reverse unconditional if the true/false indicator is false. If the true/false indicator is true, the next syllable in sequence is fetched. The true/false indicator remains unchanged.

BNS Begin Loop (BELL) XX52

8-40. This operator begins a string of program syllables which are to be repeated. The end of the string is identified by the end loop operator. The number of times this string of syllables is to be repeated is specified by the repeat field of the begin loop operator. If the repeat field is zero or one, the program string is executed once. Any repeated program string may contain within it repeated program strings. This operator builds a new loop control word in the X register and stores the previous one in the stack. The repeat field minus one is part of the loop control word.

ENS End Loop (ENLL) XX51

8-41. This operator marks the end of a repeated program string. A test is made of the number of times the program string has been executed. If the repeat field in the loop control word is not zero, it is decreased by one and an unconditional branch occurs to the syllable following the corresponding begin loop operator. Otherwise, control continues in sequence and the loop control word in the X register is replaced by a previous LCW or RCW from the stack.

JNS Jump Out-of-Loop Unconditional (JOLL) XX46

8-42. This operator terminates the repetition of a program string bracketed by a begin loop and an end loop operator, and jumps out of the repeated string. The operator deletes the associated loop control word and causes an unconditional forward jump over the number of syllables specified by the repeat field of the operator. If the repeat field of the operator is 00, the loop control word is deleted as above, but no jump occurs. Control passes to the next syllable in sequence. In either case, the current loop control word in the X register is replaced by a previous LCW or RCW from the stack.

JNC Jump Out-of-Loop Conditional (CJOL) XX44

8-43. If the true/false indicator is false, a jump out loop operator is performed. If the true/false indicator is on indicating that the TFFF is set to true, control continues in sequence.

SKIP OPERATORS

8-44. Skip operators are used to set the address registers associated with the source and/or destination character strings.

SFS Skip Forward Source (FSSL) XX31

8-45. The contents of the G and M registers are increased by the contents of the repeat field. See paragraphs 8-8 and 8-10.

SRS Skip Reverse Source (RSSL) XX30

8-46. The contents of the G and M registers are decreased by the contents of the repeat fields. See paragraphs 8-8 and 8-10.

SFD Skip Forward Destination (FSDL) XX16

8-47. The contents of the K and S registers are increased by the contents of the repeat field. See paragraphs 8-9 and 8-10.

SRD Skip Reverse Destination (RSDL) XX17

8-48. The contents of the K and S registers are decreased by the contents of the repeat field. See paragraphs 8-9 and 8-10.

BSD Skip Bit Destination (SBDL) XX02

8-49. The contents of the V, K and S registers are increased by the contents of the repeat field. See paragraph 8-10.

BSS Skip Bit Source (SBSL) XX03

8-50. The contents of the H, G and M registers are increased by the contents of the repeat field. See paragraph 8-10.

ADDRESS OPERATORS

8-51. These operators are used for storing addresses in the stack, calling addresses from the stack, and addressing locations in the stack. In addition, it is possible to obtain source and destination addresses from the source and destination character strings.

SSA Store Source Address (STSL) XX15

8-52. The contents of the G and M registers are stored in the word at the address formed by reducing the address of the RCW by the repeat field. The contents of the M register are stored in bit positions 33 through 47 and the contents of the G register in bit positions 30 through 32 of the word address. The flag bit of the word address is set to zero. The address of the RCW remains unchanged. See paragraph 8-8.

SDA Store Destination Address (STDL) XX14

8-53. The contents of the K and S registers are stored in the word at the address formed by reducing the address of the RCW by the repeat field. The contents of the S register are stored in bit positions 33 through 47 and the contents of the K register in bit positions 30 through 32 of the word address. The flag bit of the word address is set to zero. The address of the RCW remains unchanged. See paragraph 8-9.

SCA Store Control Address (STPL) XX54

8-54. The contents of the C and L registers are stored in the word addressed formed by reducing the address of the RCW by the repeat field. The contents of the C register are stored in bit positions 33 through 47 and the contents of the L register in bit positions 10 and 11 of the word addressed. The flag bit of the word address is set to zero. The address of the RCW remains unchanged.

RSA Recall Source Address (RSAL) XX53

8-55. The word at the address formed by reducing the address of the RCW by the repeat field is read from memory. If the flag bit of the word is a one and the presence bit is zero, the presence bit is set in the interrupt register and the operation terminated. If the flag bit and the presence bit of the word are both one, the bit positions 33 through 47 of the word are transferred to the M register and the G and H registers are set to zero. If the flag bit is zero, bit positions 33 through 47 are transferred to the M register and bit positions 30 through 32 of the word are transferred to the G register and the H register is set to zero.

RDA Recall Destination Address (RDAL) XX04

8-56. The word at the address formed by reducing the address of the Return Control Word

by the repeat field is read from memory. If the flag bit of the word is a one and the presence bit is zero, the presence bit is set in the interrupt register and the operation terminated. If the flag bit and the presence bit of the word are both one, bit positions 33 through 47 of the word are transferred to the S register and the K and V registers are set to zero. If the flag bit of the word is a zero, bit positions 33 through 47 of the word are transferred to the S register. Bit positions 30 through 32 are transferred to the K register and the V register is set to zero. The address of the RCW remains unchanged.

RCA Recall Control Address (RPAL) XX50

8-57. The word at the address that is formed by reducing the address of the RCW by the repeat field is read from memory. If the flag bit and the presence bit of the word are both one, bit positions 33 through 47 of the word are transferred to the C register and the L register is set to zero. If the flag bit of the word is one and the presence bit is zero, the presence bit is set in the interrupt register, and the C and L registers are not advanced. If the flag bit of the word is zero, the bit positions 33 through 47 of the word are transferred to the C register; bit positions 10 and 11 are transferred to the L register, and the C and L registers are advanced by one to specify the next syllable in sequence. The address of the RCW remains unchanged.

SES Set Source Address (SSPL) XX22

8-58. The M register is set to the address formed by reducing the address of the RCW by the repeat field. The address of the RCW remains unchanged. Registers G and H are set to zero (M is pointing at the stack).

SED Set Destination Address (SDPL) XX06

8-59. The S register is set to the address formed by reducing the address of the RCW by the repeat field. The address of the RCW remains unchanged. Registers K and V are set to zero (S is pointing at the stack).

TSA Transfer Source Address (SSAL) XX56

8-60. The eighteen bits of the three characters in the source string, starting at the position specified by the M and G registers, are transferred to the M and G registers. The three most significant bits are transferred to the G register and the remaining 15 bits are transferred to the M register. The H register is set to zero. See paragraph 8-8.

TDA Transfer Destination Address (SDAL) XX07

8-61. The eighteen bits of the three characters in the destination string starting at the position specified by the S and K registers are transferred to the S and K registers. The three most significant bits are transferred to the K register and the remaining 15 bits are transferred to the S register. The V register is set to zero. See paragraph 8-9.

ARITHMETIC OPERATORS

FAD Field Add (FADL) XX73

8-62. This operator algebraically adds a source field whose most significant position is specified by the M and G registers to a destination field whose most significant position is specified by the S and K registers. The lengths of the two fields are equal and are specified by the repeat field. The result is stored in the destination field. The zone positions of the least significant character of a field contain the sign of the field. (BA = 10 is minus; all other combinations are plus). In the result, all zone positions other than the sign are set to 00; a plus sign is BA=00 and a minus sign is BA=10. If the result of the addition overflows the destination field, the overflow is lost and the TFFF set to true; otherwise, the TFFF is set to false. If the operation is an arithmetic addition, and both fields are minus zero, the result is minus zero. If the operation is an arithmetic subtraction, the source field is plus zero and the destination field is minus zero, the result is minus zero. In all other instances a result of zero is a plus zero. See paragraphs 8-6 thru 8-11.

FSU Field Subtract (FSUL) XX72

8-63. This operator algebraically subtracts a source field from a destination field in the manner described in the field add operator.

CONVERSION OPERATORS

ICV Input Convert (ICOL) XX67

8-64. This operator converts decimal characters in the source string starting at the position specified by the M and G registers to an octal word (binary integer) at the position specified by the S register. The number of decimal characters converted is specified by the repeat field; the maximum value for the repeat field is 8. If the value of the field converted is not zero, the sign of the field is obtained from the zone bits of the least significant character of the field (BA=10 is minus; all other combinations are plus) and is stored in the sign position of the word. All other zone bits of the field are ignored. If the value of the field converted is zero, the sign of the word is set to plus. If prior to the execution of this operator the K or V registers are not zero, they are set to zero and the S register advanced by one. If the repeat field is zero, no conversion takes place, but the register manipulation does take place. The conversion process treats the decimal field as an integer. The flag bit, exponent sign and exponent of the octal word are set to zero. See paragraph 8-8 and 8-10.

OCV Output Convert (OCOL) XX66

8-65. This operator converts an octal word in the source string at the position specified by the M register to decimal characters starting at the position specified by the S and K registers. The number of decimal characters resulting from the conversion is specified by the repeat field; the maximum value for the repeat field is 8. If the octal word does not have a value of zero, the sign of the word is stored in the zone bits of the least significant character of the field (for minus BA is set to 10, for plus BA is set to 00). All other zone bits are zero.

8-66. If the octal word has a value of zero, the sign of the word is stored in the zone bits of the least significant character as a plus. If prior to the execution of this operator the G or H registers are not zero, they are set to zero and the M register advanced by one. In the conversion process, the flag bit, exponent sign and exponent are ignored; the mantissa of the octal word is treated as an integer. If the value of the mantissa is larger than can be converted to the field size specified by the repeat field, the characters converted will be the least significant characters of the decimal integer equivalent of the octal integer mantissa. If the number being converted is negative and non-zero, the sign of the result is set to negative even if all of the least significant digits converted are zero. The characters in excess of the number specified by the repeat field are lost. The true/false indicator is set to false. If the word can be converted to the field size specified, or if the repeat field is zero, the true/false indicator is set to true. See paragraphs 8-9 and 8-10.

MISCELLANEOUS OPERATORS

SEC Set Tally (SETL) XX42

8-67. This operator sets the tally register (lower 6 bits of rR) to the value contained in the repeat field.

INC Increase Tally (INTL) XX40

8-68. This operator increases the tally register by the amount of the repeat field. The tally is modulo 64; overflows are lost.

STC Store Tally (STAL) XX41

8-69. This operator stores the 6 bit tally register at the address formed by reducing the address of the RCW by the repeat field. The value contained in the tally register is stored as an integer. The contents of the tally register remain unchanged.

BIR Reset Bit (REBL) XX65

8-70. This operator sets bits to zero in the destination string starting at the position specified by the S, K and V registers. Successive bits proceed from left to right. The number of bits set to zero is specified by the repeat field. The S, K and V registers, upon the completion of this operator, address the next bit in sequence. See paragraph 8-10

BIS Set Bit (SEBL) XX64

8-71. This operator sets bits to one in the destination string starting at the position specified by the S, K and V registers. Successive bits proceed from left to right. The number of bits set to one is specified by the

repeat field. The S, K and V registers, upon the completion of this operator, address the next bit in sequence. See paragraph 8-10.

CRF Call Repeat Field (CLRL) XX43

8-72. The six low-order bits of the word at the address formed by reducing the address of the RCW by the repeat field are transferred to the repeat field of the T register. If this field is not zero, the transfer of the repeat field of the next syllable to the T register is suppressed and the present contents of the repeat field of the T register is used. If the field is zero, a jump forward unconditional operator is executed using the repeat field in the next syllable. No interrupt may occur between the execution of this operator and the execution of the subsequent operator.

EXC Exit Character Mode (RECL) XX00

8-73. Registers A and B are marked empty. The RCW is read from memory. If the flag bit of the word in the B register is 1, the operation is continued. If the flag bit is 0 and the processor is in the normal state, the flag bit interrupt is set and the operator exited with the RCW left at the top of the stack. If the flag bit is 0 and the processor is in the control state, the operator is terminated but the interrupt is not set. The C, L, G, H, K and V registers are set to the contents of the respective field of the RCW. The S register is set to the contents of the F register field of the RCW. The word addressed by the S register, the MSCW, is read from memory. The R and F registers are set to the contents of the respective fields of the MSCW. The MSFF and the SLAF are set to the contents of the respective positions of the MSCW. The S register is decreased by one. The A and B registers are set to empty. The CWMF is set to word mode.

CMX Exit Character Mode In Line (ILEL) 0100

8-74. This operator is the same as the exit character mode operator except that the C and L registers are not set.

SECTION 9

PERIPHERAL UNITS

GENERAL

9-1. The peripheral units are those units that provide the input and output facilities for the B 5500 system. They operate independently of the processor, but always under control of the MCP through the I/O control unit. This section will describe the peripheral units that may be used on a B 5500 system.

B 122 CARD READER

9-2. The B 122 Card Reader is designed for use as a compact general purpose card reader capable of reading 80 column punched cards at a maximum rate of 200 cards-per-minute (CPM) under control of the I/O control unit. Buffered operation, through the I/O control unit, permits computations to proceed while the card data is being read. The card reader is capable of reading binary cards. The card reader can handle cards that are cut in any four corners, and cards that are notched during verification. Cards may also be scribed for ease of folding and tearing.

Functional Characteristics

9-3. A single path mechanism transports cards from the picking mechanism, through the read station, and into the stacker. A failure to feed, or feed jams, causes a "not ready" signal to be relayed to the associated I/O control unit. A jam will halt the card read operation with no more than two cards in a jammed condition. Information punched in the card is read and transferred into the one word input buffer, parallel by bit, serially by column. By use of the switch on the control panel, the validity of each character in the card can be checked. For proper MCP operation, the VALIDITY switch must be turned on. A demand-type card picking mechanism permits the complete reading of an 80 column card in a total time of 350 ms. or less after a start feed signal is received. The card hopper has a capacity of 450 cards. Cards may be placed into the hopper while the unit is operating as long as approximately 150 cards are still in the hopper. During loading, the cards in the hopper remain in proper position for the continuous feeding, without manual support from the operator. A single, one-column data reading station, reads the cards column-by-column serially for the entire 80 columns. The card data may be in tabulating card code or binary code, and is transferred to the input buffer of the associated I/O control unit in 6 bit binary code. The cards are stacked in the stacker in the same sequence as they are fed, and cannot be removed from the stacker while the unit is operating.

Control Panel

9-4. A B 122 Card Reader control panel (figure 9-1) contains the switches and indicators for operation of the unit and indicates error conditions. The function of each of these elements is provided in table 9-1.

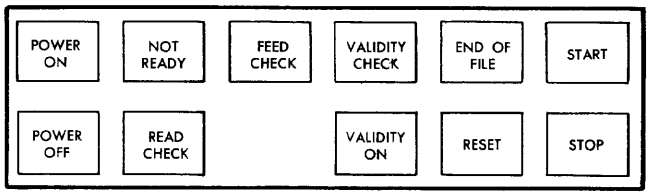


Figure 9-1. B 122 Card Reader Control Panel

TABLE 9-1

B 122 Card Reader Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This is a combination switch-indicator that applies power to the card reader and lights when pressed.
NOT READY	Ths indicator lights when any of the following conditions exist: card jam, stacker full, cover not in place, empty hopper, or STOP switch is pressed. The condition causing the NOT READY indicator to light must be corrected before reading can be resumed.
FEED CHECK	This indicator will light as a result of a card jam or a failure to feed or stack a card properly.
VALIDITY CHECK	This indicator lights when an invalid character is read by the card reader, and the MCP is notified of this by flagging the I/O result descriptor. The VALIDITY CHECK indicator and its associated circuitry are only operative when the VALIDITY ON switch-indicator is lit.
END OF FILE	This switch-indicator is not used for B 5500 operations. An end-of-file is accomplished with control cards.
START	This switch will ready the card reader (turn the NOT READY indicator off) to allow the card reader to read cards under control of the B 5500.
STOP	This switch is used to stop the card reader from feeding cards. When the switch is pressed, the card reader will go "NOT READY."
RESET	This switch clears all error indicators on the card reader. However, the NOT READY indicator is not turned off by pressing this switch.
VALIDITY ON	This switch-indicator provides the means of performing a validity check by the card reader. Validity checking is performed when the switch is pressed and the indicator lights. Validity checking is disabled when the switch is pressed and the indicator goes out.
READ CHECK	This indicator lights when the read check circuitry detects an operational failure.
POWER OFF	This switch removes power from the unit.

B 123 CARD READER

9-5. The B 123 Card Reader is functionally the same as the B 124 Card Reader, except that the B 123 reads cards at the maximum rate of 475 CPM; therefore, separate discussion is not given of the B 123.

B 124 CARD READER

9-6. The B 124 Card Reader is used to process punched cards of 51, 60, 66, or 80 columns of either standard or post card thickness, under control of an I/O control unit, at the rate of 800 CPM. An immediate access clutch provides demand feeding. Read data is transferred to the one word buffer of the I/O control unit and then to memory. Cards cut on any four corners, and cards that have been verified (notched on the right edge) may be used. However, the card stock thickness and the length must be consistent during any one run. The B 124, B 123, and B 122 Card Readers are interchangeable. A maximum of two card readers may be used with any B 5500 system. The card reader is capable of reading binary cards.

Functional Characteristics

9-7. A single one column reading station reads the cards column-by column. Column 1 is read first. The tabulating card code is translated into 6 bit binary coded decimal and transferred to the input buffer of the associated I/O control unit. A demand-type card picking mechanism picks the cards from the card hopper, and if an initial pick fails, a second pick would be attempted automatically. The card hopper has a capacity of 2400 cards and can be loaded by the operator while the unit is operating. The operator does not have to hold the cards already in the hopper in position while loading additional cards. Cards are conveyed from the hopper to the card stacker by means of a card transport mechanism. The cards are then stacked into the card stacker in the same sequence and manner in which they were fed. The stacker will also hold a maximum of 2400 cards. Cards may be removed from the stacker during operation. Failure to feed a card will cause a missing card condition and the card reader will be placed in a "not ready" state. A card jam will not cause mechanical damage, but the unit will stop operating when two cards are jammed.

Control Panel

9-8. The B 124 Card Reader contains a control panel (figure 9-2) for communication with the I/O control unit and to indicate error conditions. The functions of each switch and indicator on the control panel is given in table 9-2.

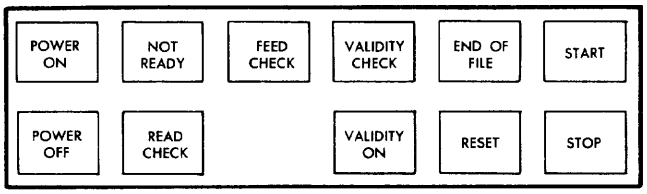


Figure 9-2. B 123/B 124 Card Reader Control Panel

TABLE 9-2

B 123/B 124 Card Reader Control Panel Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This is a combination switch-indicator that applies power to the card reader and lights when pressed.
NOT READY	This indicator lights when any one of the following conditions exists: card jam, stacker full, card line mechanism not locked, empty hopper, or STOP switch is pressed. The condition causing the NOT READY indicator to light must be corrected before reading can be resumed.

TABLE 9-2 (Cont)

B 123/B 124 Card Reader Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
FEED CHECK	This indicator will light as a result of a card jam or a failure to feed or stack a card properly.
VALIDITY CHECK	This indicator lights when an invalid character is read by the card reader. The MCP is notified of this condition by flagging the I/O result descriptor. The VALIDITY CHECK indicator and its associated circuitry are only operative when the VALIDITY ON switch-indicator is lit.
END OF FILE	This switch-indicator is not used for B 5500 operations. An end-of-file is accomplished with control cards.
START	This switch will ready the card reader (turn the NOT READY indicator off) to allow the card reader to read cards under control of the B 5500.
STOP	This switch is used to stop the card reader from feeding cards. When the switch is pressed, the card reader will go "NOT READY."
RESET	This switch clears all error indicators on the card reader. However, the NOT READY indicator is not turned off by pressing this switch.
VALIDITY ON	This switch-indicator provides the means of performing a validity check by the card reader. Validity checking is performed when the switch is pressed and the indicator lights. Validity checking is disabled when the switch is pressed and the indicator goes out.
READ CHECK	This indicator lights when the read check circuitry detects an operational failure.
POWER OFF	This switch removes power from the unit.

B 129 CARD READER

9-9. This high speed input unit provides buffered reading of 1400 cards per minute. Appearance and physical characteristics are the same as the B 124. An empty hopper condition causes the transport to shut off. When cards are placed in the empty hopper, the transport restarts without additional operator action.

B 303 CARD PUNCH

9-10. The B 303 Card Punch feeds, punches, checks, and stacks 80 column cards in both standard and post card thickness at the maximum rate of 100 CPM. The cards may be cut on any of four corners and may also be scribed for either tearing or folding. However, certain types of scribe cards may generate error signals if used with the

PUNCH CHECK switch on (table 9-3). A plugboard is not required in the B 303 Card Punch since all formatting is under control of the program. The B 303 operation is completely buffered through the I/O control unit, thus allowing processing to continue during the card punch operation.

Functional Characteristics

9-11. Cards that are to be punched are placed in the hopper face down, 12-edge first. Card stock thickness must be consistent during any one run and cards can be loaded into the hopper while the unit is operating without disturbing the cards that are already loaded into the hopper. Entry of cards into the feed rollers is accomplished by feed knives which select cards sequentially when activated by a feed signal. Cards are under positive control of pairs of feed rolls during their travel from hopper to stacker (figure 9-3). The B 303 punch unit is capable of punching up to 80 columns simultaneously in any one row of a standard card without overloading. Up to 60 columns can be punched in post card stock cards. Card jams will not cause any damage to the punch mechanism. The stacker holds 800 cards and can be unloaded while the unit is punching. The B 303 is capable of idling with cards in the feed mechanism. Card movement is controlled by the I/O control unit.

Control Panel

9-12. The B 303 Card Punch control panel (figure 9-4) contains switches and indicators for operation of the unit and indication of error conditions. The function of each of these elements is provided in table 9-3.

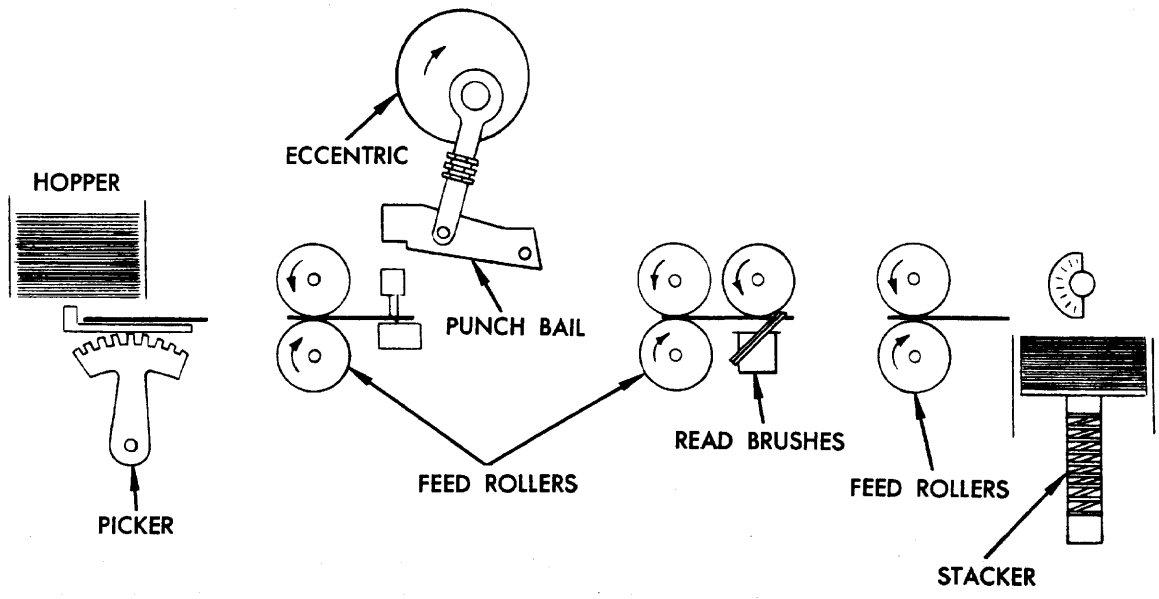


Figure 9-3. B 303 Card Punch Feed Mechanism

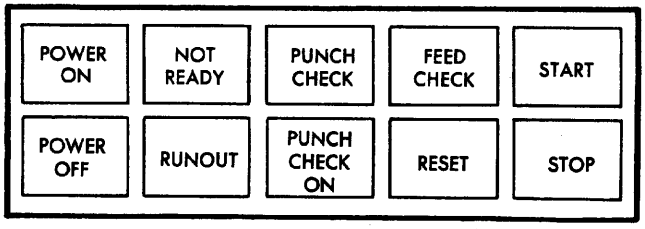


Figure 9-4. B 303 Card Punch Control Panel

TABLE 9-3

B 303 Card Punch Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This is a combination switch-indicator that applies power to the unit when pressed. The indicator lights when power is on.
NOT READY	This indicator will light when any one of the following conditions exists: STOP switch is pressed, empty hopper, improperly registered card, punch die not in place, card line mechanism not locked, stacker full, chip box not in place, and punching error. The condition causing the "not ready" state must be corrected, and the start switch pressed, before operation can be resumed.
PUNCH CHECK	This indicator will light if fewer than 80 data bits are received for each row, or if more or fewer than 12 row cycles are counted (punch station check). It will also light if the number of punched holes does not agree with the number of bits in the original data received from the I/O control unit (post-punch read station check).
FEED CHECK	This indicator will light when either a failure to feed or a jammed condition exists.
START	Pressing this switch causes one card to move from the hopper to the ready station, provided that all "not ready" conditions listed above have been corrected. Pressing this switch does not clear PUNCH CHECK or FEED CHECK conditions.
STOP	Pressing this switch will stop card feeding, light the NOT READY indicator, and set the unit to a "not ready" state. When the switch is pressed, cards that are in motion will be processed completely through the duration of the cycle.
RESET	Pressing this switch clears the FEED CHECK and PUNCH CHECK conditions.
PUNCH CHECK ON	This is a switch-indicator that selects between full punch checking and partial punch checking. The switch includes a mechanical toggle which reverses its choice each time it is pressed. When the switch is pressed and the indicator lights, a check is made of both punch station error conditions and post-punch read station error conditions. When the indicator is not lit, a check is only made on punch station error conditions. This feature allows the use of pre-punched and certain pre-scribed cards.
RUNOUT	As long as this switch is pressed, cards will pass through the unit without being punched. The switch is only effective when the unit is in a "not ready" state. Error conditions, if any, are not cleared.
POWER OFF	Pressing this switch removes power from the unit.

B 304 CARD PUNCH

9-13. The B 304 Card Punch has a maximum card punching capacity of 300 cards per minute (CPM). The format of the output cards is under program control; therefore, no plugboard is used. Buffering through the I/O control unit allows processing to continue during the card punch operation. Cards can remain in the punch station for as much as five minutes while awaiting a punch command without any damage to the card. After five minutes, the card is released to the error stacker. Cards can remain in the punch station for as long as eight hours, with the unit turned off, without damage to the card. Cards can be cut on any of four corners, or scribed for ease of tearing or folding. Certain types of scribing may generate error signals if used with the PUNCH CHECK switch on (table 9-4). Cards of varying thicknesses cannot be used during any one run.

Functional Characteristics

9-14. The B 304 card hopper hold approximately 500 eighty column cards of either standard or post card thickness. The cards must be placed in the hopper face down, 12-edge first (figure 9-5). A removable ramp can be placed on the hopper to increase its capacity by an additional 3000 cards. The ramp automatically feeds cards into the hopper as they are required. As cards are fed from the ramp into the hopper, they are automatically "joggled." Cards can be loaded into the ramp while the unit is operating without holding the previous cards in position. When there are no cards in the ramp, or if the ramp is not used, a follow block is required for proper feeding from the hopper. Cards are punched one row at a time by a single row of 80 punch dies. A punch station holds the card until it is punched. The same or random alphanumeric characters can be punched in all 80 columns of every card. Punching of all 960 holes in several successive cards due to punch or system malfunction does not result in equipment damage. A post-punch read station is used for punch checking. The reading is done by a row of 80 brushes. The B 304 includes three card stackers; primary, error, and auxiliary. The primary stacker is a ramp type with a follow block that keeps the cards stacked neatly. Cards can be unloaded from the primary stacker while punching takes place. A full primary stacker will cause a "not ready" condition. Error cards, ejected, and runout cards are stacked in the error stacker. A full error stacker causes a "not ready" condition. A full auxiliary stacker causes a "not ready" condition. The capacity of this stacker is 850 cards.

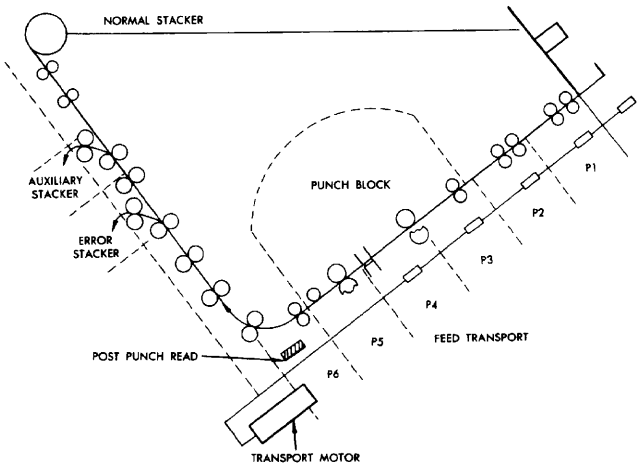


Figure 9-5. B 304 Card Punch Feed Mechanism

Control Panel

9-15. The B 304 Card Punch control panel (figure 9-6) is located to the right of the card hopper and contains the switches and indicators for operation of the unit and for error indication. The function of these elements is contained in table 9-4.

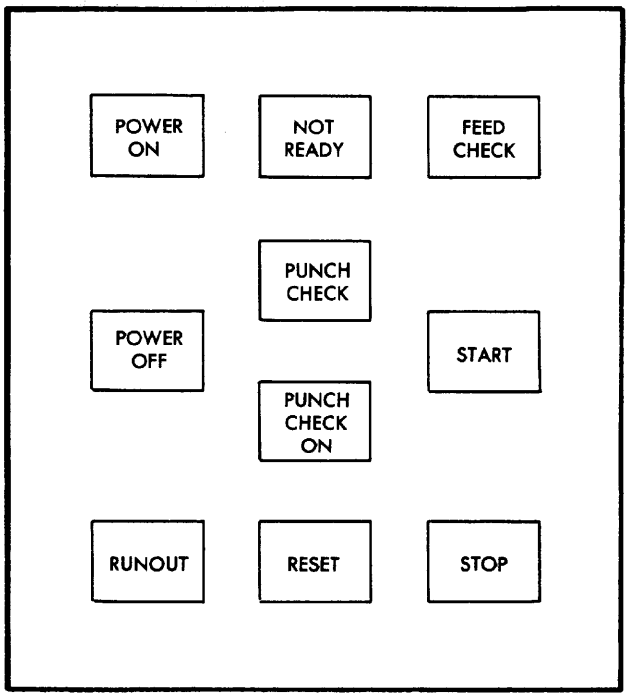


Figure 9-6. B 304 Card Punch Control Panel

TABLE 9-4

B 304 Card Punch Control Panel Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This is a switch-indicator that applies power to the unit and lights when pressed.
NOT READY	This indicator lights when one of the following conditions exists: no card at the punch station; feed check condition; card transport mechanism open; punch die not in place; covers not in place; punch error; and primary, auxiliary, or error stacker full. The error conditions must be cleared before processing can begin.
FEED CHECK	This indicator lights when there is no card present at the punch station because of either a failure to feed or a card jam (except when automatically ejected because of delayed punching).
PUNCH CHECK	This indicator will light if fewer than 80 bits of data are received for each row, or if more or less than 12 row cycles are counted (punch station check). It will also light if the number of punched holes does not agree with the number of bits in the original data received from the central processor (post-punch read station check).

TABLE 9-4 (Cont)

B 304 Card Punch Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER OFF	This switch removes power from the unit.
START	This switch readys the punch for operation under control of the B 5500. The switch does not reset FEED CHECK or PUNCH CHECK error conditions.
PUNCH CHECK ON	This is a switch-indicator that selects between full punch checking and partial punch checking. The switch includes a mechanical toggle that reverses its choice each time it is pressed. When the switch is pressed and the indicator lights, a check is made of both punch station error conditions and post-punch read station error conditions. When the indicator is not lit, a check is only made on punch station error conditions. This feature allows the use of pre-punched and certain prescribed cards.
RUNOUT	This switch causes cards in the feed line to pass through the machine without being punched. No additional cards are fed from the hopper. The switch is only effective when the unit is in the "not ready" state. Runout cards are directed to the error stacker. Error conditions, if any, are not reset.
RESET	This switch clears the FEED CHECK and PUNCH CHECK error conditions.
STOP	This switch causes the punch operation to stop after completing the punching of the card in the dies, and then places the unit in the "not ready" state.

B 141 PAPER TAPE READER

9-16. The B 141 Paper Tape Reader is capable of reading punched paper tape at speeds of 1000 or 500 character-per-second (CPS). If metalized Mylar or fanfold tape is to be read, the speed must be 500 CPS. The B 141 can accommodate 5, 6, 7, or 8 channel tape, as selected by the operator. Optional code translation facilities are available if required. Tape guides provide positive detent action to handle 11/16, 11/16 teletype; 7/8, and 1-inch tape interchangeably. Beginning and end of tape are sensed via adhesive opaque strips. Tape reels can be either 5.5 or 7 inches in diameter. Two paper tape readers with one paper tape punch can be used on one system. The paper tape reader is buffered through the I/O control unit. The B 141 is also capable of checking tape for parity errors as an off-line operation. In the off-line mode, the B 141 will stop upon detection of a parity error. It will also check for parity on line but does not stop. An error signal is sent to the I/O unit.

Functional Characteristics

9-17. Start time for the paper tape reader is 5 ms or less. Start time (when using 10 characters per inch tape) is defined as the duration from the moment a start signal is re-

ceived until the next character is read. Stop time for the B 141 is one ms or less (for all tape). The paper tape reader requires 20 ms stop stabilization time prior to executing another instruction. When the tape drive is reversed, the stabilization time is 300 ms. A minimum of four feet of tape leader is required with reeling. For strip reading, a one-foot tape leader is required. If a broken tape condition occurs, the tape reel motors are shut off automatically. Rewind can be initiated by the B 5500.

Channel Select Plugboard

9-18. A channel select plugboard is provided for interchanging channels. This action changes the bit configuration from paper tape to an interchanged bit configuration in memory. Paper tape with even parity can be accommodated by inverting the desired channel. All unused channels must be connected to the corresponding C channel. Figure 9-7 illustrates the channel select plugboard BCL and teletype wiring configuration.

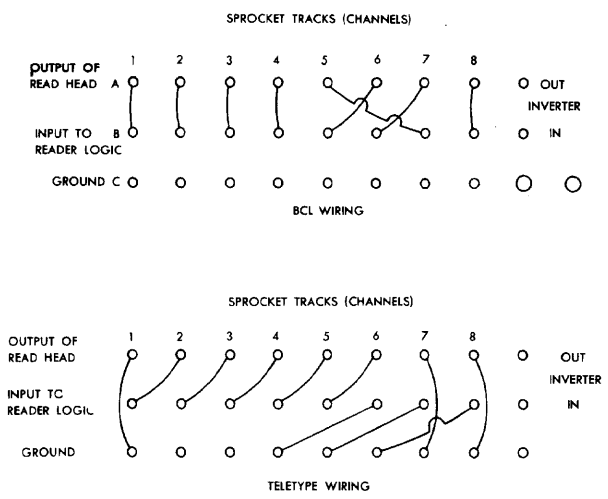


Figure 9-7. Channel Select Plugboard

Code Translator

9-19. The B 142 Code Translator, which is an optional feature, permits translation of 5, 6, 7, or 8 level tape. Up to 64 of the possible 256 information levels may be translated. The code translator is located in the paper tape reader cabinet. The following describes the plugboard layout (figure 9-8).

a. Exits: The exit hubs represent data as received from the paper tape channel select plugboard and consists of 256 possible configurations that can be generated by eight level code. Column numbers are the decimal equivalents of the binary numbers represented by the "input to reader logic" (B) hubs 1 to 4 of the channel select plugboard. For example;

Binary Equivalent	(1)	(2)	(4)	(8)	
Channel B	1	2	3	4	
	0	1	0	1	goes to column 10

Row numbers are the decimal equivalents of the binary numbers represented by the "input to reader logic" (B) hubs 5 to 8 of the channel select plugboard. For example;

Binary Equivalent	(1)	(2)	(4)	(8)	
Channel	5	6	7	8	
B	0	1	1	1	goes to row 14

b. Entries: The entry hubs represent data sent to the I/O control unit consisting of the possible 64 BCL combinations. Column numbers are the decimal equivalents of the binary numbers represented by the B and A bits of the BCL code (BA = 0, 1, 2, and 3). Row numbers are the decimal equivalents of the binary numbers represented by the 8, 4, 2, 1 bits of the BCL code. For example, if row 7 of column 3 is connected, the bit configuration is represented as B A 8 4 2 1 or the BCL character G.

c. Stop Controls: There are 8 sets of stop control hubs. To designate a stop code, an exit hub is wired to the input of a stop control. Only one exit can be wired directly to a single stop control input. If the stop code is to be stored, the output of a stop control is wired to an entry hub. Any exit code not wired is deleted and is not transferred to the I/O control unit.

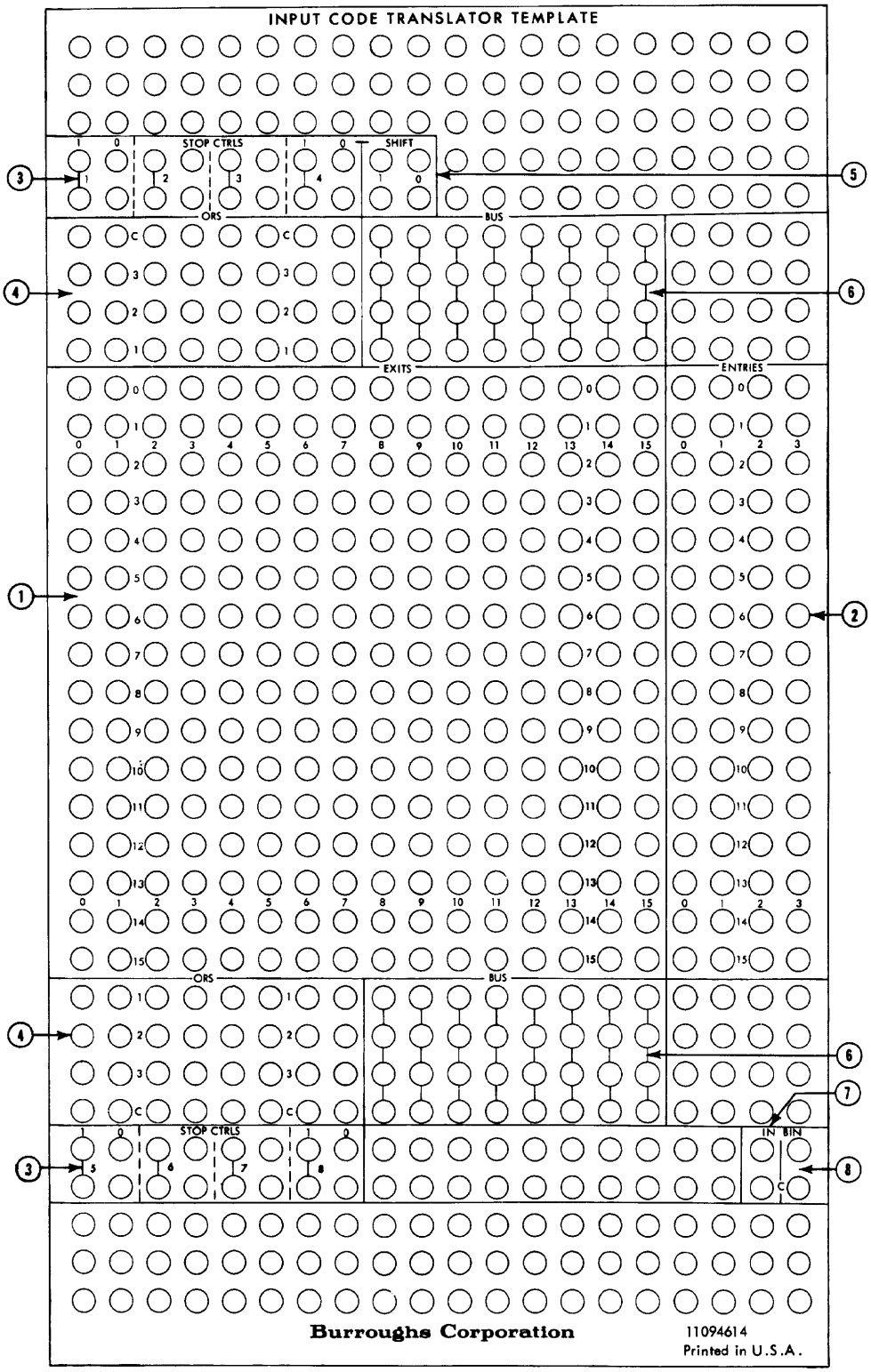


Figure 9-8. Plugboard Layout

d. Shift Codes: The shift code is designated by wiring an exit to the upper shift code input. An un-shift code is designated by wiring an exit hub to the lower shift code input. The shift code is made functional by connecting two shift output hubs together. When in the shift case, channel 8 (channel selector plugboard) is set to 1. When in the un-shift case, channel 8 is set to zero.

NOTE

Teletype code can be converted to a single case code via the teletype switch.

e. BCL/Binary Input: To enable the translator, the two enable hubs must be connected together. If they are not connected the translator is bypassed. To obtain an output which is the direct image of the channel selector plugboard (6 channels only), the binary hubs must be connected together. The I/O control unit will perform a BCL to internal code translation on this input.

f. OR Hubs and BUS Hubs: The following "3 input OR hubs" and "BUS hubs" usage are permitted:

(1) Up to 9 exits can be connected to a single entry by using 3 OR elements and a bus.

(2) Up to 9 exits can be connected to a single stop control by using 3 OR elements and a bus.

(3) Up to 6 exits can be connected to a single stop control by using 2 OR elements (no bus required).

Control Panel

9-20. The B 141 Paper Tape Reader control panel (figure 9-9), contains switches and indicators for operation of the unit and for the detection of errors. The function of each of these elements is contained in table 9-5.

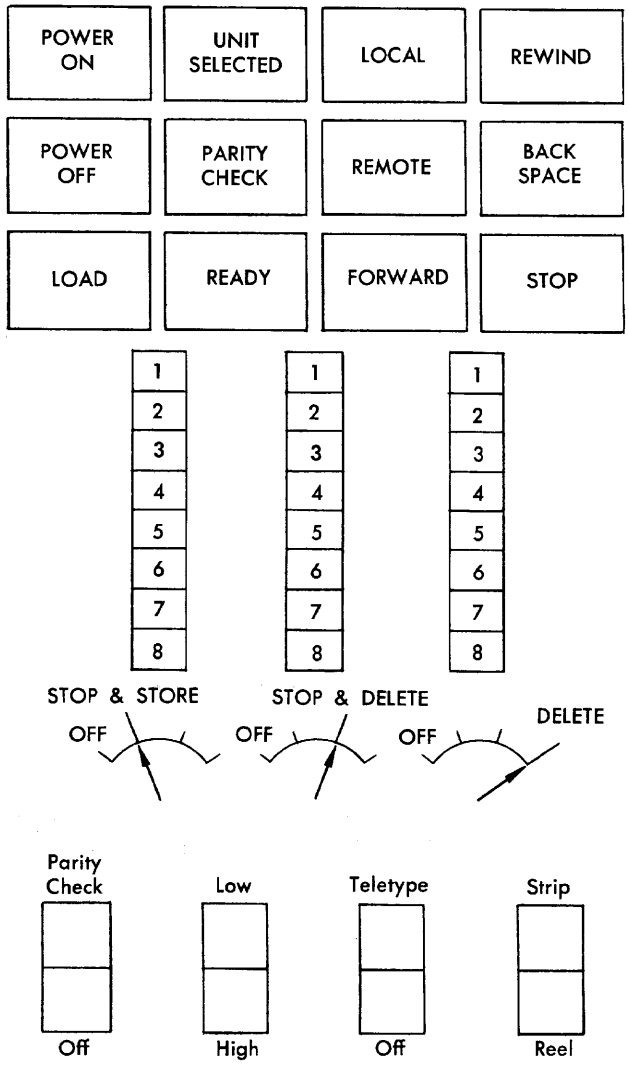


Figure 9-9. B 141 Paper Tape Reader Control Panel

TABLE 9-5

B 141 Paper Tape Reader Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This switch/indicator lights when pressed, indicating that power is applied to the unit.
NOT READY	This indicator lights when a not ready condition exists.
LOCAL	This switch places the B 141 in a local condition and the unit is not available to the B 5500 system. The LOCAL indicator will also light.
REWIND	When pressed, the paper tape moves in the reverse direction until a beginning-of-tape condition is detected. The tape will then stop. This switch is active only when the unit is in a LOCAL state and the STRIP/REEL switch is in the REEL position.
POWER OFF	When pressed, removes power from the unit.
PARITY CHECK	This indicator lights when a parity error is detected. This is activated only when the PARITY switch is ON.
REMOTE	This switch/indicator lights when pressed indicating that the unit is under control of the B 5500 system.
BACK SPACE	Moves the tape in a reverse direction to the next control code, or beginning-of-tape. This switch is active only when the unit is in a local condition. The switch may also be used to check parity offline.
LOAD	This switch releases the brake, allowing loading of the paper tape. This switch is active only when the unit is in the local condition.
READY	When pressed, this switch sets the brake and starts the capstan rollers. The servos are also activated when the STRIP/REEL is in the REEL position and the tape is properly positioned in the storage arms.
FORWARD	This switch moves the tape forward to the next control code, or to the end-of-tape. This is only active when the paper tape reader is in a local status. The switch may also be used to check parity offline.
STOP	The operation of the B 141 will stop when this switch is pressed. This is only active when the paper tape reader is in a local status.

TABLE 9-5 (Cont)

B 141 Paper Tape Reader Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
CONTROL CODE	A set of three switches that provide manual selection of three different control codes. Any combination of control codes may be used concurrently. The control code characters may be stored or deleted, as selected. A four position switch for each code set determines the action taken when the control code is detected. The CONTROL CODE switches are active in either the local or remote condition. The four positions of the switch are: OFF, STOP AND STORE, STOP AND DELETE, and DELETE.
PARITY ON-OFF	When in the ON position, parity checking is enabled. The parity error level is reset when in the OFF position.
HIGH-LOW	In the HIGH position, high speed operation is selected (1000 CPS); in the LOW position, low speed operation is selected (500 CPS).
TELETYPE ON-OFF	When in the ON position, 5 levels of teletype are converted to a 6-level single case code.
STRIP-REEL	In the STRIP position, the reel motors are deactivated and the NO TAPE switches are by-passed. In REEL position, the reel motors are activated and the NO TAPE switches are activated.
NO TAPE	These switches are activated when the STRIP REEL switch is in the REEL position and there is no tape loaded or the tape breaks. Activation of these switches deactivates the reel motors.
GUIDE SELECTION SWITCH	This switch is located to the right of the read mechanism. The switch adjusts the paper guiding to the width of the tape being used.

B 341 PAPER TAPE PUNCH

9-21. The B 341 Paper Tape Punch is basically a teletype paper tape punch which is capable of punching standard paper tape format in BCL code. The B 341 will punch 5, 6, 7, or 8 level tape at a minimum rate of 100 CPS, ten characters-per-inch. Standard tape widths of 11/16, 7/8 and 1-inch may be punched, as selected by the operator. Either oiled paper tape, laminated fiber, dry paper tape, metalized or laminated Mylar paper tape may be punched on the B 341. The maximum size supply reel that can be placed on the B 341 is eight inches in diameter. The reel hub measures two inches in diameter. The punched tape is wound onto a five and one half or seven inch diameter take-up reel. It is not necessary to use the take-up reel when punching tape. The end-of-tape is indicated by the LOW TAPE indicator when approximately 35 feet of tape remain on the supply reel. The B 341 is buffered through the I/O control unit. This allows processing to continue while punching paper tape.

Functional Characteristics

9-22. A method is provided for the operator, through the channel select plug-

board wiring, to interchange any of the channels that might be desired. Undesignated channels in the channel select plugboard are not punched or sensed as controls for the B 341. Up to 64 different alphanumeric characters and special characters can be punched. When the B 341 receives a paper tape write instruction from the I/O control unit, paper tape will be punched in a forward direction. The output record length is determined by a specific word count in the I/O descriptor or a control code in the data stream which is manually designated by a switch setting on the B 341 control panel or translator. The code is punched or suppressed as indicated by the control code switch or translator plugboard wiring. BCL codes are transferred from the I/O control unit one character at a time to the paper tape punch. The code translator permits the translation of BCL to a single frame code by means of a removable plugboard. Also, teletype codes can be translated. Teletype is a double case code (figures/letters shift) with several special requirements. To accomodate the shift used by teletype code, each of the allowable characters is designated as a figures or a letters code. Whenever a character is of a different case (figure/letter) than its predecessor, the appropriate shift code must be punched prior to the character. The two shift codes used for teletype tape can be designated by code translator plugboard wiring. The special requirements used for teletype codes are:

a. Automatic generation of codes for the figures shift after SPACE, TAB, LINE FEED, and CARRIAGE RETURN.

b. Automatic generation of codes for the carriage return, and line feed only must be generated immediately following all end-of-line codes.

Channel Select Plugboard

9-23. This plugboard is provided mainly for purposes not requiring a translator. It is possible for the operator to select any of the 7 BCL levels and interchange them to any of the 8 possible paper tape channels. Paper tape with even parity can be punched by inverting the desired channel. All unused channels must be connected to the corresponding C channel. Figure 9-10 illustrates the channel select plugboard BCL and Teletype wiring configuration.

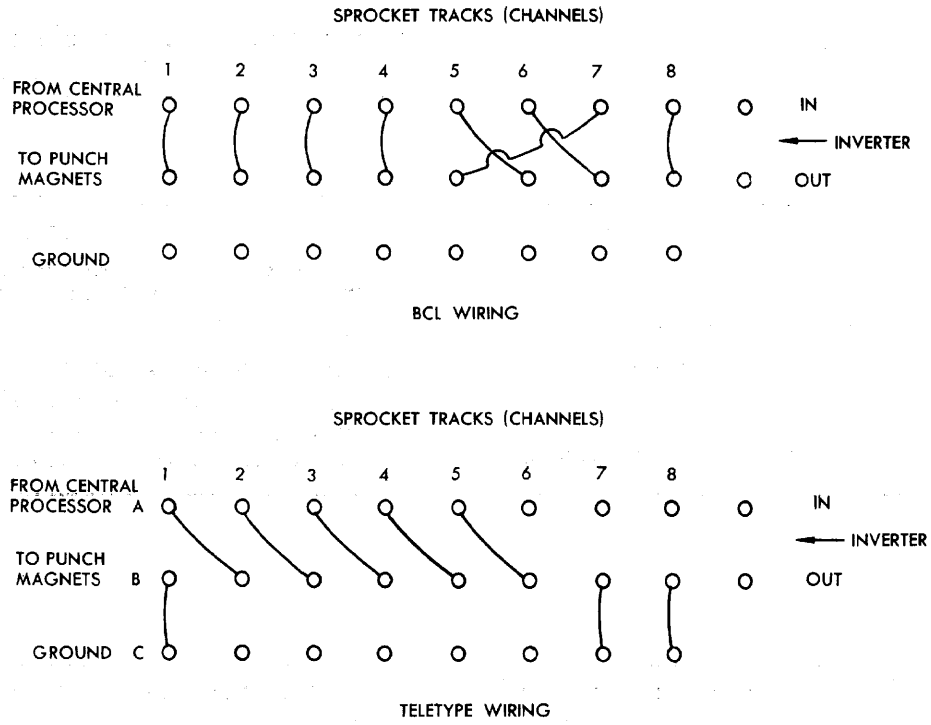


Figure 9-10. Channel Select Plugboard

Code Translator

9-24. The B 342 Code Translator, which is an optional feature, permits translation of BCL code to any 5, 6, 7, or 8 channel code. Up to 64 codes can be translated. The code translator is located in the paper tape punch cabinet. Character (code) flow is from the I/O control unit to the channel select plugboard to the translator to the paper tape punch. The following describes the plugboard layout (figure 9-11).

a. Exits: The exit hubs represent data sent from the I/O Control Unit via the channel select plugboard. Assume the following bit configuration.

	1	2	4	8	A	B	P
		•	•		•	•	•

Corresponding channel hubs (B) on channel select plugboard	1	2	3	4	5	6	7

Input to punch logic (B) hubs 5 and 6 of the channel select plugboard identify the translator column (binary value).

Binary Equivalent	(1)	(2)	
(B) Channel hubs (A and B bits)	5	6	
	•	•	or column 3

Input to punch logic (B) hubs 1 to 4 of the channel select plugboard identify the translator row (binary value).

Binary Equivalent	(1)	(2)	(4)	(8)	
(B) Channel hubs (bits from 1 to 4)	1	2	3	4	
		•	•		or row 6

Therefore, the assumed bit configuration identifies column 3, and row 6 of the exit hubs.

b. Entries: The entry hubs, when impulsed, generate the selected 5, 6, 7, or 8-channel output character to be punched. There are 256 possible combinations using right level code. The code punched is determined by column and row. The decimal value of the column and row are converted to a binary value as follows:

The binary value of the column identifies which channels 1 to 4 to be punched. Example: column 10 would punch

Binary Equivalent	(1)	(2)	(4)	(8)
Channel	1	2	3	4
Punches		•		•

The binary value of the row identifies which channels of 5 to 8 are to be punched. Example: row 7 would punch

Binary Equivalent	(1)	(2)	(4)	(8)
Channel	5	6	7	8
Punches	•	•	•	

Therefore, if column 10 of row 7 was impulsed, the following code would be punched in paper tape.

Channel	1	2	3	4	5	6	7	8
Punches		•		•	•	•	•	

c. Stop Controls: There are 4 sets of stop control hubs. To designate a stop code, an exit hub is wired to a stop control hub. If the stop code is to be stored, a stop control hub, impulsed by a connected exit hub, is connected to the desired entry hub. If an entry hub is not corrected to the stop control hub, the stop code will not be punched.

d. Shift Codes: These hubs are required when the output data requires shift and unshift codes. These hubs are connected to exit hubs to determine which codes require a shift code (maximum 32). Any codes not connected to these hubs are considered as requiring an unshift code. The associated hub is connected to an entry hub for the required code translation. Whenever a change is required from an unshift code to a shift code or visa-versa, as selected on these hubs, the appropriate shift or unshift code is punched.

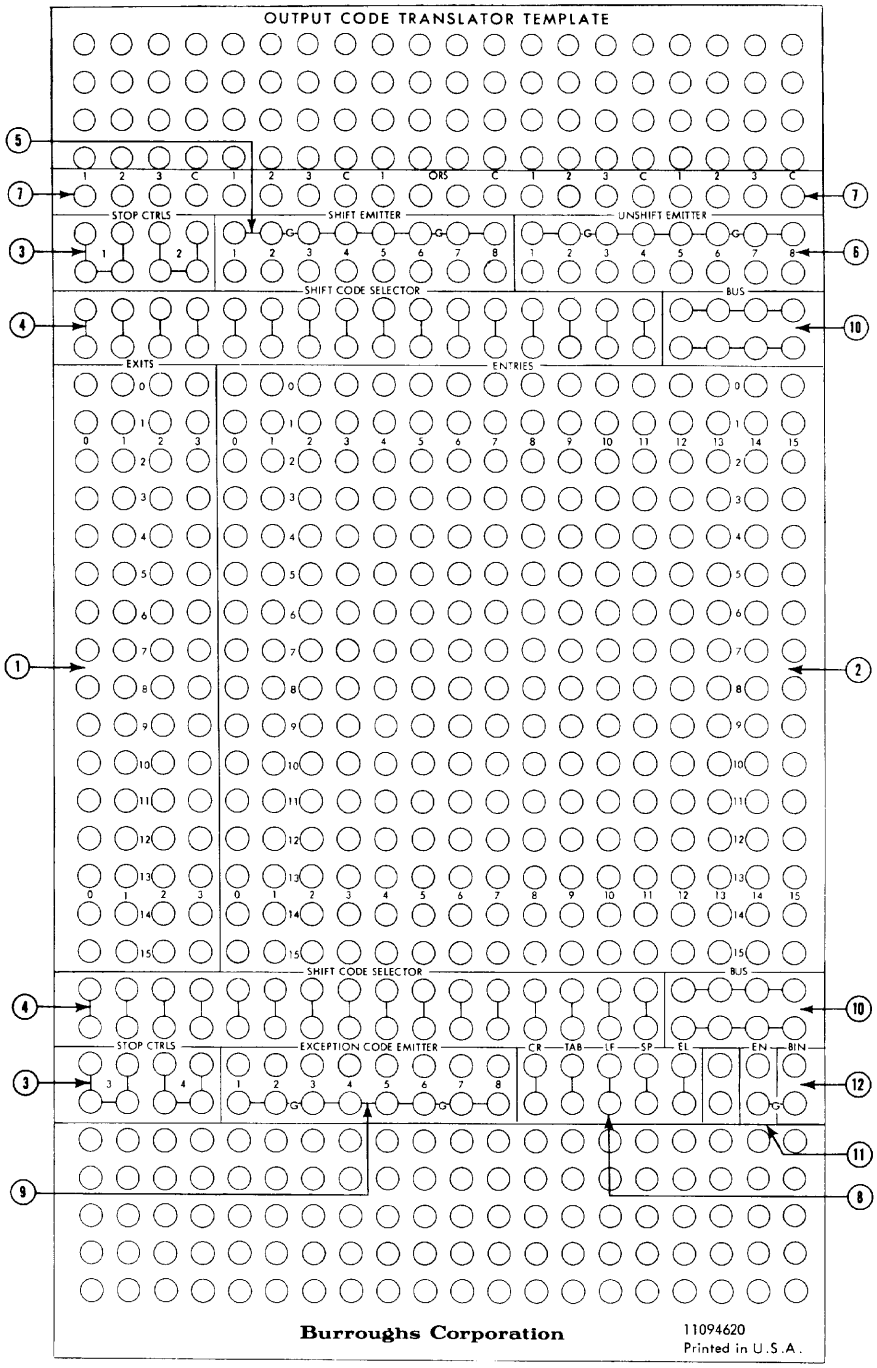


Figure 9-11. Plugboard Layout

e. Shift Emitter: Any 8-bit code can be selected as the shift code by connecting the channel requiring a bit to the hub located directly above the designated channel. All channels unconnected will be considered as a zero (no bit). This code will be punched when required as designated by the shift code selection.

Unshift Emitter: Any 8-bit code can be selected as the unshift code by selecting the channel requiring a bit to the hub located directly above the designated channel. All channels unconnected will be considered as a zero (no bit). This code will be punched when required as designated by the unselected codes, that

is, those not connected to the shift code selector hubs.

"OR" Hubs: The "OR" hubs permit up to three different codes, designated by the exit hubs, to initiate one common code or action.

f. Exception Codes: These hubs are provided to handle special Teletype code set problems. These codes are CR, TAB, LF, SP and EL. These codes are connected from the exit hubs and to the selected entry hubs. Since these codes will not be selected as shift codes, they will be considered as unshift codes. The EL or end-of-line will initiate the punching of the exception code before the actual code is punched. The exception codes are set up in the exception code emitter.

g. Exception Code Emitter: Any 8-bit code can be selected as this code by connecting the channel requiring a bit to the hub located directly above the designated channel. All channels unconnected will be considered as a zero (no bit). This code will be punched when required by the designated EL code.

h. Bus Hubs: There are two sets of bus hubs. Each set permits multiple connections to a single hub.

i. Enable Hubs: These hubs must be connected to activate the translator. If unconnected the normal translation of BCL internal code to BCL paper tape code will take place.

Control Panel

9-25. The B 341 Paper Tape Punch control panel (figure 9-12) contains the switches and indicators for operation and error indication. The function of each element is provided in table 9-6.

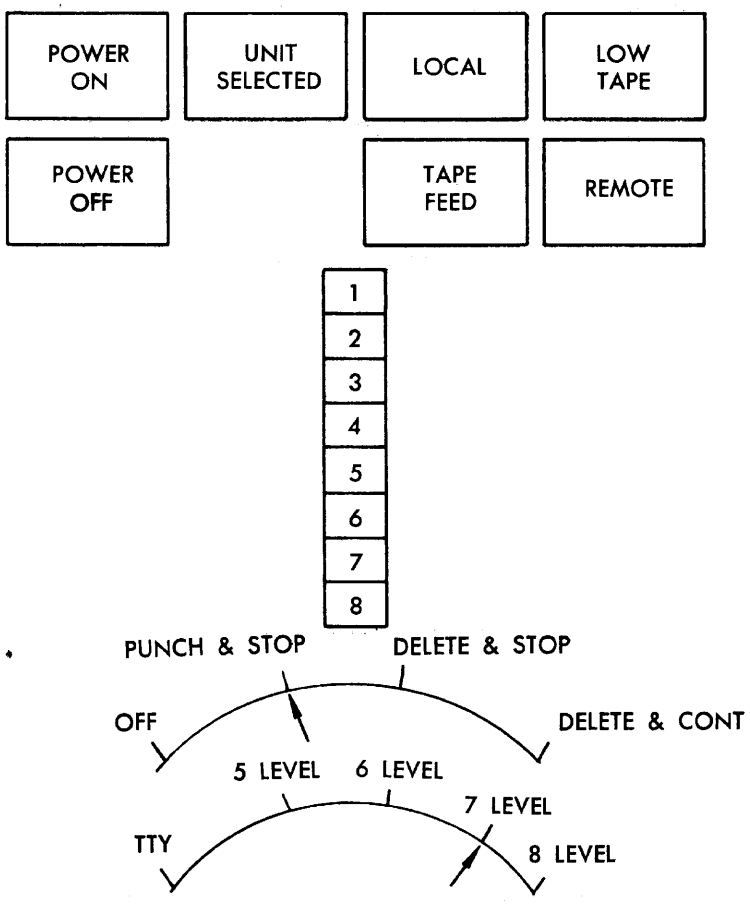


Figure 9-12. B 341 Paper Tape Punch Control Panel

TABLE 9-6

B 341 Paper Tape Punch Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This switch-indicator lights when pressed indicating that power is applied to the unit.
NOT READY	This indicator will light when a not-ready condition exists.
LOCAL	This switch-indicator places the B 341 in a local condition and the unit is not available to the B 5500 system.
LOW TAPE	This indicator will light when 35 feet of tape, or less, remains on the supply reel.
POWER OFF	This switch removes power from the unit.
TAPE FEED	This switch feeds tape with all holes punched. The switch is active when the LOCAL switch is activated. Tape feed rate is 100 characters-per-second.
REMOTE	This switch-indicator lights when pressed, indicating that the unit is under control of the B 5500 system.
CONTROL CODE	This switch allows the operator to designate a control code. The code may or may not be punched. The switch is active in REMOTE or LOCAL and has four positions which determine the action taken when a control code is detected. The four positions of the switch are: OFF, PUNCH AND STOP, DELETE AND STOP, and DELETE AND CONTINUE.
LEVEL DESIGNATION	This switch is used to select the number of channels and type of paper tape to be used.

B 320 LINE PRINTER

9-26. The B 320 Line Printer is functionally the same as the B 321 Line Printer, except that the B 320 prints at a maximum rate of 475 lines per minute (lpm); therefore, a separate discussion is not given.

B 321 LINE PRINTER

9-27. The B 321 Line Printer is a drum-type printer capable of printing 700 lines per minute when single spacing, or 650 lines per minute when double spacing. Formatting, editing, skipping, and spacing are under program control. The B 321 receives BCL coded information from the I/O control unit in parallel by bit, serial by character transfer and stores this data in a 120-position buffer. Upon completion of loading the buffer, the I/O control unit is released and the printer will complete the print cycle independent of the I/O control unit. The 120-character buffer is included making the print cycle independent of the system once the information has been transferred to the printer buffer. Continuous forms are used, and skipping and spacing operations are controlled by the program through use of the I/O control unit. Vertical spacing may be 0,

1, or 2 spaces per line. Each line consists of 120 print positions, 10 positions per inch. Vertical spacing is 6 or 8 lines per inch. There are 63 alphanumeric and special characters, plus a blank for each print position.

Functional Characteristics

9-28. The 64 characters contained in each print position consists of 26 alphabetic, 10 numeric, and 28 special characters (figure 9-13). Printing is done on continuous paper forms which may be from 5 to 20 inches in width including marginal punch strips. Length can be 22 inches (at 6 lines per inch) or 16-1/2 inches (at 8 lines per inch). The forms are loaded in the cabinet below the printing mechanism. The forms are transported through the unit, by means of pin-fed tractors, to the stacker. As many as five carbons plus the original may be printed. In general, the printer can process legible copy forms up to 0.02 inches in over-all thickness. Thinnest form that can be processed is 0.0025 inches. The optimum number of copies can be legibly printed by using premium paper and carbon. A clearance adjustment, required when changing from one form thickness to another, is available to the operator and can be accomplished within 30 seconds without the aid of tools.

Blank	$	—	#
.	*	/	@
[	)	'	:
(	;	%	>
<	≤	=	≥
←	≠	]	+
&	×	"	?

Figure 9-13. Special Character Set

9-29. TAPE CONTROLLED CARRIAGE. The B 321 Line Printer does not directly control the feeding and spacing of the forms. This is performed by the tape controlled carriage of the printer in conjunction with instructions from the I/O Control Unit.

9-30. CONTROL TAPE. The carriage control tape (figure 9-14) has column positions (1-12) called channels. Horizontal lines can be skipped up to 132 lines for control of a form. For ease in preparation, the tape is somewhat longer than required. Pre-punched holes located in the center of the tape are used by a pin feed mechanism to move the tape past the sensing device. Movement of the carriage control tape is synchronized with the movement of the form through the carriage. Skipping the form to any predetermined position is accomplished by addressing any one of 11 holes in the carriage control tape. A twelfth channel in the control tape is used to signal the last print line on the form. When a hole in this position is sensed, the printer sends a signal to the I/O control unit flagging the result descriptor for the MCP. The twelve carriage control tape channels are usually punched to control the following functions:

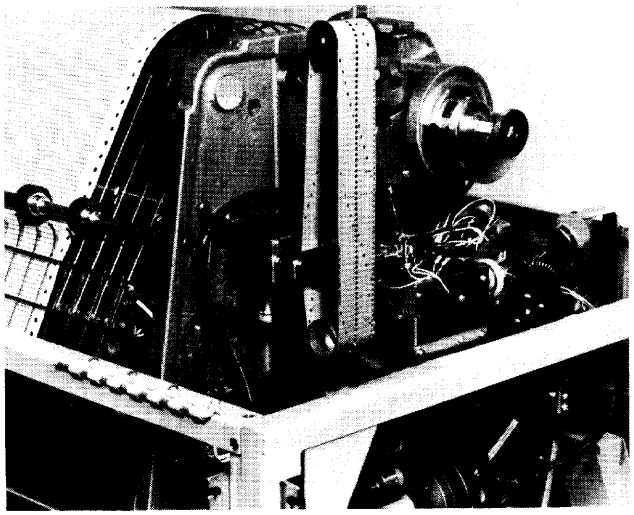


Figure 9-14. Carriage Control Tape

a. Channel 1 will normally be used to identify the first print line (heading) of a form.

b. Channel 2 will usually be used to indicate the first body line of a form on which detail information appears. In an invoicing operation, where the first printing line and first body line are not the same, channel 1 would be used to indicate the heading on the form, in this case a name and address.

Channel 2 would correspond to the first printed line of detail information.

c. Channel 3-11 will normally be used to identify any one of 9 user determined print positions. These channels may be used in any desired sequence.

d. Channel 12 is reserved for punching the hole indicative of the last printing line in the body of a form.

Control Panel

9-31. The B 321 Line Printer control panel (figure 9-15), contains switches and indicators for operation of the equipment and for error indications. The control panel is located at the front of the unit, to the right of the print section. The function of the switches and indicators is provided in table 9-7.

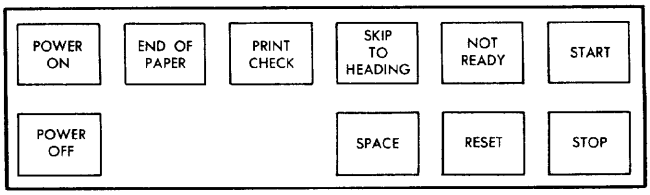


Figure 9-15. B 320/B 321 Line Printer Control Panel

TABLE 9-7

B 320/B 321 Line Printer Control Panel Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This is a switch-indicator that applies power to the unit and lights when pressed.
END OF PAPER	This switch-indicator signals nearing an out-of-paper condition. Pressing this switch removes the end-of-paper condition and extinguishes the light. One line can then be printed; thereafter, the unit returns to end-of-paper condition. Successive depressions of this switch enable printing successive lines, to complete the last line on a form.
PRINT CHECK	This indicator lights when a print check error has been sensed, or when the print drum is not properly synchronized.
SKIP TO HEADING	Pressing this switch causes the carriage to skip to the first punch in channel 1 of the carriage control tape.
NOT READY	This indicator will light when anyone of the following conditions exist: the END OF PAPER indicator is lit, the 6/8 lines-per-inch switch is in position N, the unit "slews" paper for more than one second, or the START switch has not been pressed.
START	This switch is used to signal the B 5500 system that the printer is ready for use. It is also used to restart printer operations caused by a line printer "not ready" condition.

TABLE 9-7 (Cont)

B 320/B 321 Line Printer Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
STOP	Pressing this switch will stop the line printer prior to the execution of the next pring instruction. The print buffer will not be loaded after the switch has been depressed, and the printer will go "not ready".
RESET	Pressing this switch resets the PRINT CHECK indicator.
SPACE	Pressing this switch causes the forms to be single spaced.
POWER OFF	This switch removes power from the unit.

B 325 LINE PRINTER

9-32. The B 325 Line Printer is essentially the same as the B 321 except that the B 325 provides 132 character positions. To accomodate 132 print positions, a factory installed adapter is required for each I/O channel.

B 328 LINE PRINTER

9-33. The B 328 is functionally the same as the other printers. It will print 120 character positions at a maximum rate of 1040 lpm.

Functional Characteristics

9-34. In order to achieve the higher print speed, the entire range of 64 characters has been relocated around the drum. The 37 most frequently used characters (26 alpha, 10 numeric, and the period) have been arranged in consecutive locations. The remaining 27 have been positioned to reflect their frequency of use and the most commonly used are clustered near the 37. As soon as a character is printed, its buffer position is set to blank. When all characters in the buffer are blank, the printer is released for spacing. This allows paper to be advanced while unused character positions on the drum are passed. The drum revolves at 1040 rpm, or one revolution every 57.6 ms. The character time for each of the 64 positions is .9 ms. To advance the form one space requires 24.3 ms, with additional spaces taking .7 ms. If 27 characters are unused during the cycle, single spacing time can be masked and the 1040 rpm rate maintained. The average attainable lines per minute while printing within various consecutive character sets and while spacing a given number of lines are as follows:

Number of Lines Spaced	LPM With Various Consecutive Character Sets		
	10	37	64
1	1040	1040	734
2	1040	780	680
3	1040	715	625
4	1040	660	584
5	700	610	546

B 329 LINE PRINTER

9-35. Functionally, the B 329 is the same as the B 328 except that the B 329 provides 132 character print positions. This capability necessitates a factory installed adapter for each I/O channel.

Control Panel—B 328/B 329

9-36. As standard features, both the B 328 and the B 329 have a ribbon tracking device sense and limit mistracking of the ribbon. Also, duplicate central panels are located on both the front and at the rear of the printers for increased operator efficiency. The con-

trols and indicators are the same as the other line printer control panels.

B 423 MAGNETIC TAPE UNIT

9-37. The B 423 Magnetic Tape Unit is functionally the same as the B 422 Magnetic Tape Unit, except that the B 423 allows only a packing density of 200 (24 KC) frames per inch. Therefore, a separate discussion is not given of the B 423.

B 422/B 423 MAGNETIC TAPE UNIT TRANSPORT

9-38. The B 422/B 423 Magnetic Tape Unit was designed to read, write, erase, backspace, and rewind magnetic tape under control of the central processor. Any valid six-bit character may be written or read by the magnetic tape unit. The B 422/B 423 Magnetic Tape Unit is capable of reading or writing either in BCL or binary mode. Writing and erasing is done in the forward direction only. Reading is done forward or backward. Up to 16 B 422 or B 423 Magnetic Tape Units can be used with a B 5500 Information Processing System. Only one tape unit may receive instructions from an I/O control unit at any one time.

9-39. The magnetic tape unit can accommodate 2400 feet of 1/2-inch tape on 10-1/2 inch reels. The transport moves the tape, for reading and writing, under a dual-gap magnetic read/write head at 90 inches per second or 120 inches per second on the B 422 or B 423 respectively. A full reel will rewind in less than 90 seconds. When used with the B 5500, there are two switch-selected packing densities that may be read or written. The high-packing density is 555.5 characters per inch while the low-packing density is 200 characters per inch. The clock pulse repetition rate used with the B 422 is 50 KC/sec. for high density and 18 KC/sec. at low density. The clock rate with the B 423 is 24 KC/sec. at low density. At high-packing density, over 15,000,000 characters may be stored on a reel of tape. The minimum number of characters that may be stored in any one group is seven. The maximum number is limited only by the input information or the physical end of tape.

9-40. Start time for the B 422 transport system is 4.5 ms. Start time is defined as the time that the instruction is executed and the time until the first character is read or written. On a read instruction, the stop time is 4.2 ms. On a write or erase instruction, the stop time is 4.2 ms. Stop time is the time from completion of the read or write operation until the tape unit is ready to accept another instruction.

Functional Characteristics

9-41. Figure 9-16 illustrates the position of the tape reels in relation to the read-write head and feed rollers. When reading or writing, tape is passed from the file reel past the read-write head to the take-up reel. During rewind, the motion is reversed. To prevent tape breakage and to minimize start time, two vacuum columns are used. As the tape is drawn from one vacuum column, it is replenished from the reel above. As it is fed into the other vacuum, the associated reel takes up the slack tape. Movement of the tape past the read head is at 120 inches per second. Rewind is at 300 inches per second; therefore, approximately 2400 feet of tape can be re-wound in 90 seconds. Start time is the time lapse between issuing a tape order (read, write, erase, backspace) until the first character is read, written, erased, or bypassed. Start distance is not relevant.

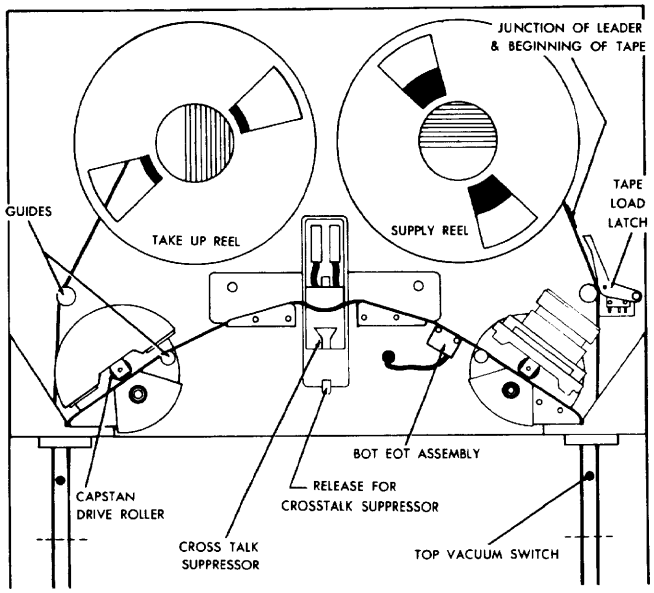


Figure 9-16. B 422/B 423 Magnetic Tape Unit Transport Mechanism

Magnetic Tape

9-42. The magnetic tape used with the magnetic tape units is 0.5 inches wide, 2.0 mils thick, and approximately 2400 (±20) feet in length. The base material of the tape is Mylar with a heavy duty binder for longer wearing characteristics. The tape units feature devices called "latch leaders" (figure 9-17). The purpose of the latch leader is to minimize tape load-unload time. The leaders are two-part: male and female. The male leader is opaque-black Mylar and is non-magnetic. The length of this leader is approximately 6 inches from the end of the leader to the magnetic tape splice which is on the file reel. The female leader is opaque Mylar and is manually wound onto the take-up reel by the operator, one time only. The female leader is approximately 14 feet in length. The tape unit contains a latching device which holds the female leader in place during loading and unloading. When the reels rewind for unloading, the transport will move the leaders back until their connection is on the file reel-side of the tape latch. Then the latch is closed to grip the female leader while the male leader is disconnected. After a new file reel has been placed on the unit and leaders are connected, the tape latch is opened.

Control Panel

9-43. The B 422/B 423 Magnetic Tape Unit control panel (figure 9-18) contains switches and indicators for operation of the equipment. The function of these switches and indicators is contained in table 9-8.

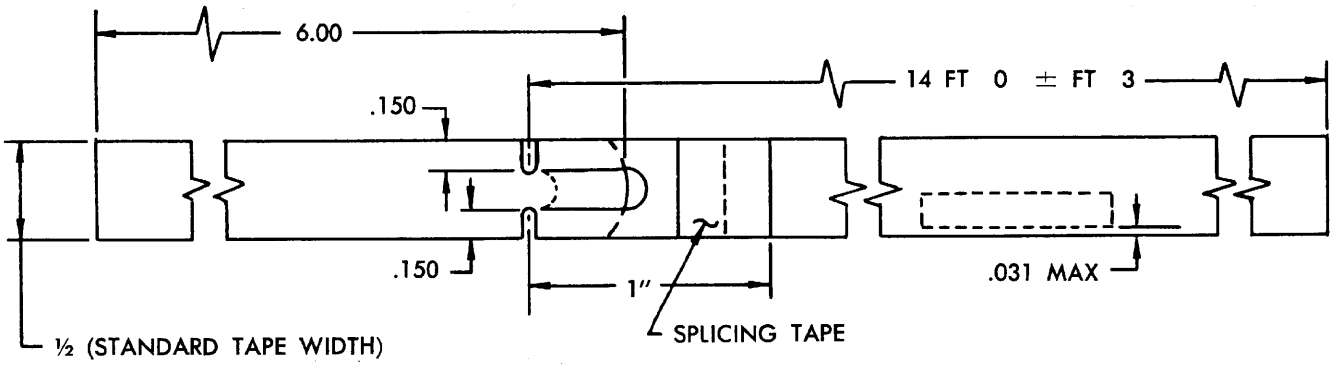


Figure 9-17. Magnetic Tape "Latch Leaders"

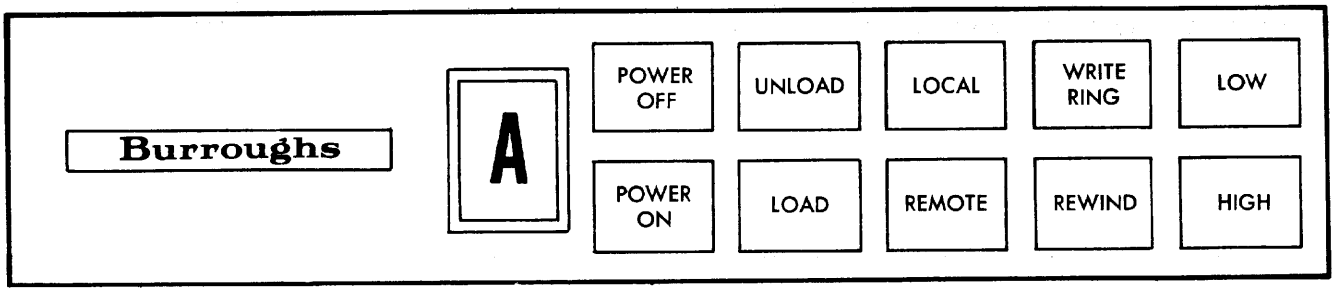


Figure 9-18. B 422/B 423 Magnetic Tape Unit Control Panel

TABLE 9-8

B 422/B 423 Magnetic Tape Unit Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER OFF	Removes power from the tape unit.
UNLOAD	Positions tape to the point where the latch leader is on the file reel-side of the tape latch, thereby, permitting the operator to unload the tape. This is only active in the local status.
LOCAL	Removes the tape unit from control of the B 5500 system. The switch lights when pressed.
WRITE RING	Signals that the file reel has a write ring installed and that writing can be performed on the tape.
LOW	When this indicator is ON, the unit is in the 200 character per inch recording density.
HIGH	This switch-indicator selects the high recording density. When this is ON, the unit is in the 555.5 character per inch recording density.
REWIND	Rewinds the tape to the beginning-of-tape mark. Rewind speed is 320 inches per second. This switch is active only when the unit is in a local condition.
REMOTE	Places the tape unit under control of the B 5500 system. The switch lights when pressed and the LOCAL indicator goes off.
LOAD	Causes tape to be drawn into the vacuum columns and moves the tape so that the beginning-of-tape is at the read-write head. This is only active in local status.
POWER ON	Applies power to the unit. The switch lights when pressed.
UNIT DESIGNATOR	This will be an alphabetic character placed there by the field engineer. The designation is dependent on the jack into which the cable has been plugged in the display and distribution unit.

B 424 MAGNETIC TAPE UNIT TRANSPORT

9-44. This unit operates at 83 inches per second at 800 bits per inch density only, giving a transfer rate of 66 KC. Appearance and physical characteristics are the same as the B 422. The B 424 may be used separately or, if used on combination, it may only be combined with B 422 drives operating at 120 inches per second in a density of either 200 or 555.5 BPI.

B 450 DISK FILE/DATA COMMUNICATIONS BASIC CONTROL UNIT

9-45. The B 450 consists of a modified B 5500 cabinet with power supply and hardware facilities for mounting two full size logic gates. It serves as a housing for a disk file control unit or a data communications control unit or a combination of these units.

B 5470 DISK FILE CONTROL UNIT

9-46. The B 5470 contains the control and checking circuitry to accommodate from 1 to 10 B 471 Disk File Electronics Units. Each electronics unit contains the circuitry necessary to control from 1 to 5 B 475 Disk File Storage Modules. Each storage module contains four magnetic disks. Each B 475 Disk Storage Module contains 9.6 million alphanumeric characters. Since there is a maximum of two B 5470 units per system, one B 5500 system may contain up to 100 B 475 Disk Storage Module Units with a maximum storage capacity of 960 million alpha characters. All information transfer and addressing from these storage units is controlled by the B 5470. The B 5470 is under control of the I/O control units. Because independent checking features are incorporated in the unit, the processor is free to execute other operations or input or I/O operations when the control unit is performing a check. Checking of each disk file address as it is transferred is provided by using the least significant digit position of the B register for address comparison. If an error occurs, the transfer operation stops and no data will be transferred. Also, an address parity indicator will be set. For each word of data written during a write operation, a "check character" is developed and written. This code is regenerated and pared against the written check character during a read operation. If the comparison is unequal, an information error indicator is set. When a non-existent address is referenced, the operation is terminated and an invalid indicator is set. Attempting to write on a disk which is locked out will set a write lock out indicator. Reading and writing is prevented while the control unit is in a busy or not ready status. The control unit is in a not ready status if either of the power switches (AC-DC) are off, or if the REMOTE/LOCAL switch is set to the local position.

B 471 DISK FILE ELECTRONICS UNIT

9-47. The B 471 Disk File Electronics Unit incorporates all of the disk file system electronics for controlling a maximum of 48 million alphanumeric characters of information (five B 475 disk file storage units), in addressable segments of 240 characters. The unit contains the main air pressure system starting controls, basic head switching logic, and power supplies for a maximum of 5 storage units. Lock-out switches for the unit and for individual disks are provided on the lock-out panel. The unit LOCK-OUT switch prevents writing on the entire unit. Individual DISK LOCK-OUT switches allows disks to be individually locked out. When the unit or disk is placed in a write lock-out state, it is still possible to read from the unit and/or disk. The unit is made ready for operation when the POWER ON switch is on and the REMOTE/LOCAL switch is set to the remote position.

Control Panel

9-48. The switches and indicators located on the control panel (figure 9-19) used for the operation of the B 471 Disk File Electronics Unit are described in table 9-9.

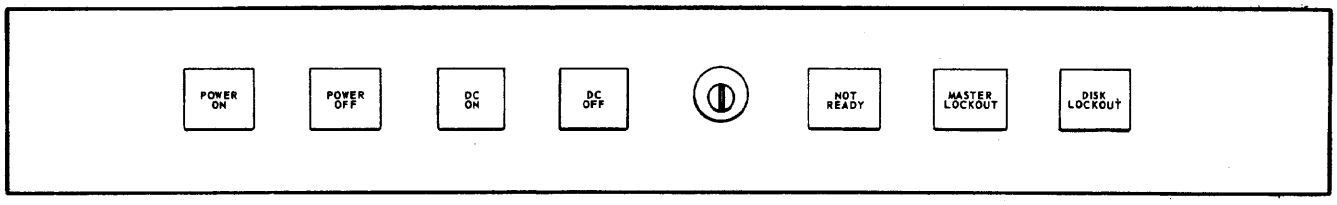


Figure 9-19. B 471 Disk File Electronics Unit Control Panel

TABLE 9-9

B 471 Disk File Electronics Unit Control Panel
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER ON	This switch applies AC power to the electronics unit and the storage modules connected to it.
POWER OFF	This switch removes AC power from the electronics unit and from the storage modules connected to it.
NOT READY	This indicator will light when one of the following conditions exists: a. REMOTE/LOCAL switch is in the LOCAL position. b. All disks in the storage modules are not up to speed. c. Air pressure is low.
DC ON	This switch applies DC power to the electronics unit and to the storage modules attached to it.
DC OFF	This switch removes DC power from the electronics unit and the storage modules connected to it.
MASTER LOCKOUT	This indicator lights when the MASTER LOCKOUT switch (located under a hinged cover) is pressed to lock out all of the disks connected to the particular disk file electronics unit.
DISK LOCKOUT	This indicator lights when one or more of the disk lockout switches (located under the hinged cover) are pressed to lock out the disks connected to the particular disk file control unit.
DISK LOCKOUT SWITCHES	Above the front control panel of the disk file electronics unit is a hinged cover that conceals the DISK LOCKOUT switches (see figure 9-9). This cover is provided with a lock to prevent anyone but the Field Engineer and the Data Processing Manager to gain access to these switches. There are 20 individual DISK LOCKOUT switches (one for each of the possible 20 disks that can be connected to the electronics unit) and one MASTER LOCKOUT switch for locking out all of the disks that are connected to the storage unit. When one or more of the individual lockout switches is set to the ON position, the DISK LOCKOUT indicator will light. When the master switch is set to the ON position, the MASTER LOCKOUT indicator will light.

B 475 DISK FILE STORAGE MODULE

9-49. The B 475 Disk File Storage Module consists of four vertically mounted magnetic disks comprising a storage capacity of 9.6 million alphanumeric characters. Each disk surface has 50 data tracks which are divided into addressable segment sizes of 240 alphanumeric characters. One data track has a 24,000 character capacity. Every track is equipped with its own read/write head and by means of electronic switching, the heads can rapidly access data from the tracks. A fail safe mechanism within each storage module prevents the read/write heads from contacting or damaging the magnetic disk surface.

Disks

9-50. The magnetic disks rotate at 1500 revolutions per minute. With the head-per-track design incorporated for reading and writing data, the average time required to access data from disk storage is one-half disk revolution or 20 ms. regardless of file size or organization of records. All data recorded onto the disk will remain on the disks until replaced with new information. Transfer of information from and to the file is at an average rate of 100,000 characters per second.

9-51. The B 475 Disk File Storage Module must be used in conjunction with the B 471 Disk File Electronics Unit. One B 472 Disk File Storage Unit is composed of one B 471 Disk File Electronics Unit plus one B 475 Disk File Storage Module. Each B 5470 Disk File Control unit may control a maximum of 10 B 471 Disk File Electronics Units for a maximum of 480 million characters per control unit. Since a B 5500 system may contain two B 5470 Disk File Control Units, the maximum system storage capacity is 960 million characters.

Control Panel

9-52. The controls for operating the B 475 Disk File Storage Module are located on the B 471 Disk File Electronics Unit. This unit provides individual DISK LOCK-OUT switches for the disks within the storage modules. These switches allow individual disks to be locked out by the operator. For a complete functional description of the switches and the indicators, refer to table 9-9.

B 451 DISK FILE EXPANDED CONTROL UNIT

9-53. This unit contains the circuitry necessary to control disk storage systems requiring more than one B 471 Electronics Unit. One B 451 is required for systems with two to five B 471 Electronics Units and would be added to the B 5470 Disk File Control Unit. An additional B 451 is required as expansion occurs from six to ten B 471 Electronics Units. A system will require a B 451 for each additional five B 471 Electronics Units, and the B 451 would be added to the second B 5470 Disk File Control Unit in a second B 450 Basic Control Unit.

B 5480 DATA COMMUNICATION CONTROL UNIT

9-54. The B 5480 Data Communication Control Unit provides the interface between the I/O control unit and the various terminal units. Only one B 5480 unit may be connected to a B 5500 system. It can serve from one to 15 terminal units of any combination. A B 5500 system may have two B 5470 Disk File Control Units and one B 5480 or one B 5470 and one B 5480. The B 5480 may have a cable length of up to 50 feet from the associated I/O control unit and is under control of the I/O control unit only when loading or unloading the terminal unit buffer to or from core memory.

Functional Characteristics

9-55. The B 5480 provides the code translation facility for conversion between Burroughs Common Language (BCL) and Baudot code. In a system where different types of terminal units are used, BCL to Baudot conversion takes place only when the scanner in the B 5480 is addressing a teletype terminal unit.

9-56. The B 5480 can recognize that any terminal unit is in one of four possible states: busy, input ready, output ready, or not ready.

Busy State

9-57. When a designated terminal unit is in the busy state, the associated B 5480 cannot communicate with that terminal unit. A busy state occurs when one of the following situations is present: there is a non-related call on the net of a teletype terminal unit or when loading or unloading a terminal unit buffer to or from the I/O control unit.

Input Ready State

9-58. A terminal unit that has received a complete message from an inquiry station is in the input ready state. The completion of an input message is recognized by an end-of-message character which is transmitted to the associated I/O control unit as a group mark (BCL-011 1111).

Output Ready State

9-59. The terminal unit is in the output ready state after it has completed transferring the contents of its buffer to an inquiry station and has not detected an end-of-reply (group mark) character.

9-60. The B 5480 also provides an interrupt to the I/O control unit. This interrupt is set when any terminal unit is in the input ready or the output ready state, and the terminal unit is being addressed by the scanner.

9-61. The character transfer rate through the B 5480 is a maximum of 30,000 characters-per-second. Transfer is serial-by-character, parallel-by-bit, and in all cases of an inquiry reply, the message must be terminated by a group mark character. The scanner in the B 5480 has the facility to connect any of the terminal units to the I/O control unit. The time required for the scanner to examine adjacent channels for ready status is a maximum of 220 microseconds and the B 5480 gives priority, in undirectional sequence, to the terminal units that are in the output ready state.

B 481 TELETYPE TERMINAL UNIT

9-62. The B 481 Teletype Terminal unit provides the interface between the B 5480 and the teletype stations on a net. A single B 481 Teletype Terminal can service up to 400 teletype station sets, allowing a possible 6000 teletype stations in 15 networks if only teletype terminals are used (1 to 15 terminal units per B 5480). The B 481 may have a cable length of up to 50 feet from the B 5480 and, as an optional device, may have a teletype page printer included as part of the terminal unit.

Functional Characteristics

9-63. The B 481 Teletype Terminal Unit provides serial-parallel code conversion, special teletype character detection and insertion, and buffer storage capability. Control and timing levels are generated and sensed so that the B 481 is compatible with the B 5480.

9-64. Character control is provided for the insertion and deletion of special teletype characters such as "line feed." The character control also provides the end-of-reply, and the station disconnect signals for the teletype net. Character control further inserts change of print mode signals in the data being sent to the teletype stations when there is a change from either figures (figs) to letters (ltrs).

9-65. The B 481 Teletype Terminal Unit incorporates a buffer which stores 6 bit characters. Buffer size may be 120 or 240 characters and the access time for the terminal buffer is 20 microseconds. An inquiry message may be entered via the keyboard of any station on the net by selectively calling the B 481. The B 481 Teletype Terminal Unit can service only one call at a time. A teletype page printer may be included as part of the B 481 Teletype Terminal Unit. This printer can be used for monitoring all messages transmitted through the network.

B 483 TYPEWRITER TERMINAL UNIT

9-66. The B 483 Typewriter Terminal Unit provides the interface between the B 5480 and typewriter inquiry stations. The B 483 provides facilities for 1 to 8 typewriter inquiry stations and includes the input station selection circuitry and a buffer of 60 characters

for each of 8 typewriter stations. (The 480-character buffer is divisable by the number of typewriter stations. Therefore one station would have 480 characters.) Control and timing levels are generated and sensed so that it will be compatible with the B 5480. The typewriter terminal unit may have a cable length of up to 50 feet from the B 5480.

Functional Characteristics

9-67. The unit provides an input buffer to store simultaneous input of 60 characters (including end-of-message character) for each of 8 typewriter inquiry stations. The buffer has 8 segments, each of which is reserved for a specific typewriter inquiry station.

9-68. The B 483 Typewriter Terminal Unit also provides input scanning facilities to accept data from any of the 8 possible typewriter inquiry stations. This data is picked off and stored as it is available, a character at a time, and is directed to the proper buffer.

9-69. In addition, the unit provides an input latch facility which interrupts the scanner and holds a station buffer while data is transferred to the B 5480 and to the I/O control unit. The latch is initiated when the end-of-message input character is stored.

9-70. The same 60 character buffer is used for output. This buffer will store a reply message from the I/O control unit. The flow of data from the buffer is governed by timing levels generated in the typewriter inquiry station.

9-71. As a part of the typewriter terminal unit, a counter (60 characters including end-of-message character) is furnished to direct input from a typewriter inquiry station to the proper address in the buffer. The counter advances with each input character and is reset by pressing the station reset key.

B 493 TYPEWRITER INQUIRY STATION

9-72. The B 493 Typewriter Inquiry Station is a Model 33 Teletype machine.

Functional Characteristics

9-73. The B 493 Typewriter Inquiry Station communicates with the B 483 Typewriter Terminal Unit via a twisted pair cable. The inquiry stations may be up to 2000 feet from the B 483. The station set operates at a standard rate of 10 characters per second by selectively pressing the keys and space bar of the keyboard in the same manner as typing.

DATA TRANSMISSION SYSTEM

9-74. The Burroughs Data Transmission System is even more powerful and versatile than the Communication System. The Communication System which could be more appropriately called an "Inquiry" system because an inquiry must be "Asked" before a reply can be given; i.e., the computer system is unable to transmit a message without first receiving an inquiry, with the exception of the teletype network.

9-75. All operations between any of the various types of stations and the computer system are completely buffered and independent. The transfer rate between the buffers and the computer system is 100KC. The Data Transmission System has a much greater capacity to handle various types of transmission devices in any combination:

a. typewriter (240 stations)

b. TWX networks (240)

c. teletype (95,760 stations)

d. 801 automatic calling units

e. Data Speed II

f. UNIVAC 1004

g. IBM 1050

Functional Description

9-76. The Data Transmission System is composed of the following components:

a. B 452 Disk File Data Transmission Basic Control (power supply)

b. B 249 Data Transmission Control Unit (DTCU)

c. B 487 Data Transmission Terminal Unit (DTTU)

d. Line Adaptors

9-77. The B 452 houses the B 249 and also houses the Disk File Control Unit if it is used. The Terminal Units and Adaptors are not free standing boxes, but are in fact, gates which are mounted inside the B 452 cabinet. For this reason, the B 452 has increased power requirements over its predecessor, the B 450.

B 249 DATA TRANSMISSION CONTROL UNIT

9-78. The DTCU serves as a multiplexing device which allows the computing system to handle up to 15 Transmission Terminal Units. If only one terminal unit is on the system, the DTCU is not required.

9-79. One of the major functions of the DTCU is to provide code translation between the computing system and the transmission device. For example, typewriters and TWX networks both function in ASCII code; and, therefore, the DTCU must provide translation both ways between BCL and ASCII or ASCII and BCL. In the case of teletype networks, they function in Baudot code; and, therefore, the DTCU must provide translation between this code and BCL again, both on input and output.

NOTE

If a DTCU is not used, then it is the object program's function to provide the correct translation on both input and output.

B 487 DATA TRANSMISSION TERMINAL UNIT (DTTU)

9-80. A DTTU contains a message area of 448 characters of core memory which is divided into sixteen 28-character sub-sections numbered 0 through 15. An "adaptor" is assigned a number of these sub-sections according to user requirements. The division of the storage area into "buffers" is accomplished by placing line adaptors at line adaptor connection points. Each buffer area is then defined to start at the point where its line adaptor is connected and to extend to the next line adaptor or the end of storage, whichever occurs first. The "buffer address" (sometimes called "station address") is actually composed of two quantities: the terminal unit number followed by the section number. When communicating with any transmission device, both of these quantities are required. The combination of both quantities shall henceforth be referred to as the "buffer address".

9-81. For example, assume a terminal unit were assigned a number 2 (this is accomplished by its physical connection), and it had three adaptors. If the line adaptors were attached to the connection points at sub-sections 0, 3 and 7, the entire storage area would be divided into three message or information areas. The first area would have a "buffer address" of 20 and would be 84 characters in length. The second area would have a "buffer address" of 23 and would be 112 characters in length. The third area would have a "buffer address" of 27 and would be 252 characters in length.

9-82. When a message or group of characters is received from a transmission device, the information passes through a line adaptor into the adaptor's terminal unit information area. When the end-of-information is sensed, an interrupt level is produced by the terminal unit and sent either through the control unit, if it is present, to the computer system or directly to the computer system without a control unit, which subsequently causes the computer system to be interrupted. At this time, an "Interrogate" is performed in order to determine the "buffer address" of the information.

9-83. At the completion of a data transmission operation, either input or output, a buffer address can be flagged either "Normal" or "Abnormal". This flag of normal or abnormal in conjunction with an input or output operation can indicate a large number of combinations reflecting errors, special attention, etc.

Buffer Conditions

9-84. A buffer may be in any one of a number of conditions, depending on its adaptor. These conditions are:

a. Idle. This condition indicates that the buffer is not presently receiving a message from a transmission device nor is it transmitting information to that device.

b. Read Ready. This condition indicates that an entire message has been received from a transmission device and the computer system should now process that information.

NOTE

When the computer system does read that information out of the buffer, the buffer is then returned to an Idle condition.

It should also be noted that a computer-initiated message to a buffer is allowed any time a buffer is in an idle condition. Thus, at the completion of a read, the transmission device may continue sending information; or the computer may send a reply concerning the information just read in from the buffer. In most cases, if a transmission device like a typewriter or teletype is being used, it would be advisable to send a reply, if only a carriage return and line feed, to indicate that the operator may continue with the input. When a message is transmitted to an idle buffer, complete with group mark, an interrupt is produced when the transmission device has fully accepted the output message; and an Idle Interrupt is produced to the computer system.

9-85. If, however, the output message sent to the buffer does not contain a group mark or end-of-message character, a Write Ready Interrupt is produced when the buffer has fully transmitted the information to the transmission device. A Write Ready condition indicates that the buffer is now ready for the next portion of the total message. This Write Ready condition in previous terminal units caused a temporary lock-out of all other transmission devices during the time that multiple groups of information were being transmitted to the transmission device; however, in the B 487 this condition does not effect any device, other than itself.

NOTE

Whenever any interrupt of any type has been interrogated, that interrupt has been removed and will not cause another interrupt under any circumstances.

9-86. All of the above conditions can be flagged either "normal" or "abnormal". For example, an operator typing the left arrow (end of message) key causes a read-ready normal interrupt. If, however, for some reason the operator keys in the exclamation point (end of transmission) character, a Read-Ready Abnormal Interrupt is produced. On output, when the left arrow is encountered upon retrieval from the buffer to be sent to the adaptor, the buffer assumes an Idle Normal condition with an interrupt. If, during the time that a message is being sent out of the buffer to the adaptor and this adaptor is TWX, the operator should press "BREAK" key, the buffer terminates the output of the message and immediately assumes a Read-Ready Abnormal condition with an interrupt. These are but a few of the numerous combinations and conditions of input and output with errors, end of transmission, loss of carrier, buffer overflow, etc., that can occur and subsequently flag the operation as normal or abnormal. The handling of these conditions is left totally up to the object program.

NOTE

There is hardware priority available that causes selection of the lowest numbered buffer within a terminal unit for system attention.

Line Adaptors

9-87. There is a specially designed line adaptor for each type of transmission device, which serves to interface that device with the DTTU. With the appropriate line adaptor, it is possible to interface any device to a DTTU and thus achieve information exchange with the computer system.

Typewriter

9-88. A single typewriter station connects with a typewriter line adaptor. A maximum of 16 typewriters can be connected with the

16 adaptors to a single DTTU. However, the buffer size of each would then be restricted to 28 characters. The buffer may optionally consist of two 28-character buffers which are alternately loaded with input. This allows the system to empty one buffer while the other is being filled. Each typewriter is entirely independent of all other adaptors regardless of what type they might be. If the buffer associated with a typewriter station is not "busy", the computer system can initiate, at any time, a message to be typed out on that station just as if it were a Supervisory Printer.

9-89. If, while typing a message in from the remote station, the operator notes a mistake, the operator can perform a "backspace" with the " < " character and thereby erase one character. This backspace may be performed as often as required. Another function is the ability to handle paper tape from a model 33 or 35 ASR typewriter station. Messages may be stacked on paper tape as shown in figure 9-20.

9-90. The computer system can exercise control over the paper tape by performing a Start Paper Tape function (this is a reserved character "#"), and the paper tape will continue sending until a character on the paper tape indicates "stop paper tape." After an X-OFF code is read, the paper tape reader will read one or more characters before stopping, thereby leaving the buffer in a busy status. The paper tape will stop, the computer system will read the message, process it, and either send a reply back to the station, which will be typed out, or simply send a control character to resume reading from the paper tape.

TWX Networks

9-91. The following illustration (figure 9-21) describes the terminal unit adaptor data set connection required for a TWX network.

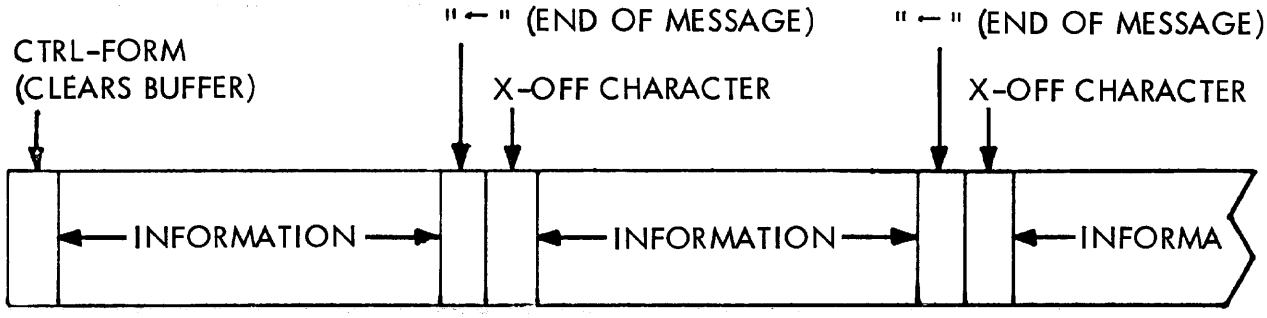


Figure 9-20. Possible Paper Tape Format

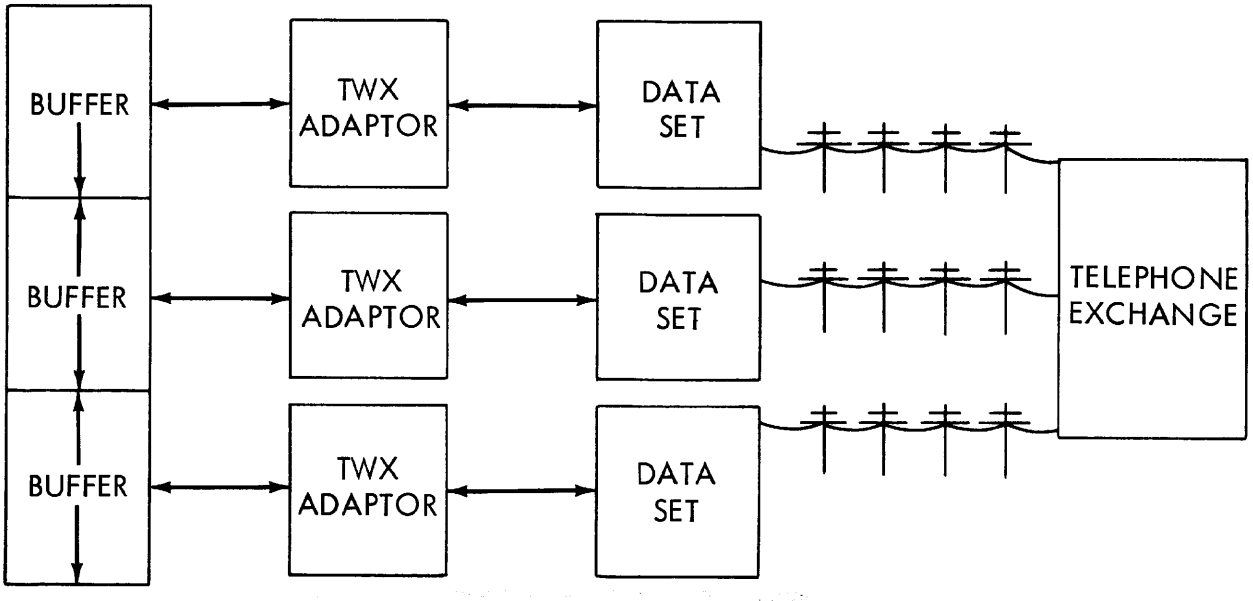


Figure 9-21. TWX Network

9-92. The dataset is, for all practical purposes, a telephone and has a dialable number. The operator at a remote TWX typewriter station dials the number of the dataset connected to the adaptor; and, when the connection is made, the TWX station functions exactly like a typewriter station as previously described, with this exception: the data set is continuously monitoring the quality of the data carrier; and, if the quality drops below a certain threshold value, the dataset performs a "loss of carrier disconnect". This can occur on either input or output. In both cases, however, an abnormal condition is flagged and an interrupt produced.

Teletype Networks

9-93. Figure 9-22 illustrates the connection of an adaptor with a teletype network.

9-94. Each teletype circuit is connected with a teletype line adaptor. The buffer size would be a minimum of 28 characters. If two buffers are used, they may be alternately loaded with input to allow the processor time to empty the buffer before the information is lost. Each of the stations and the adaptor has a "2-character call." If the net, as it is called, is idle, the computer system can initiate a call sequence and/or message to any or all of the stations on the net. In the same manner, any station, if the net is idle, can call the adaptor and/or any other station and transmit information to that station. Teletype networks have a standard calling and message discipline that is followed by all who use the net.

801 Automatic Calling Unit (ACU)

9-95. Figure 9-23 illustrates the usage of an 801 ACU.

9-96. The 801 adaptor is first used to transmit a dial sequence through the 801 dataset. The 801 dataset is, in fact, connected to the telephone exchange through the normal dataset that will be used once the telephone connection has been established. After the connection has been made, the information is processed to and from the remote station through the TWX adaptor. The 801 is not used during this period of time. The 801 line adaptor uses one 28-character buffer.

Dataspeed II

9-97. The Dataspeed II Paper Tape Reader or Paper Tape Punch may be attached to the system through a line adaptor. The buffer size is two 56-character or two 112-character buffers which are alternately loaded on input. The code used on the paper tape may be either ASCII or BCL paper tape code.

IBM 1050

9-98. The IBM 1050 can be attached to the B 487 through its special line adaptor. The buffer size is any multiple of 28 characters. Since the IBM 1050 may be polled, an object program may poll the device with the special characters necessary for it.

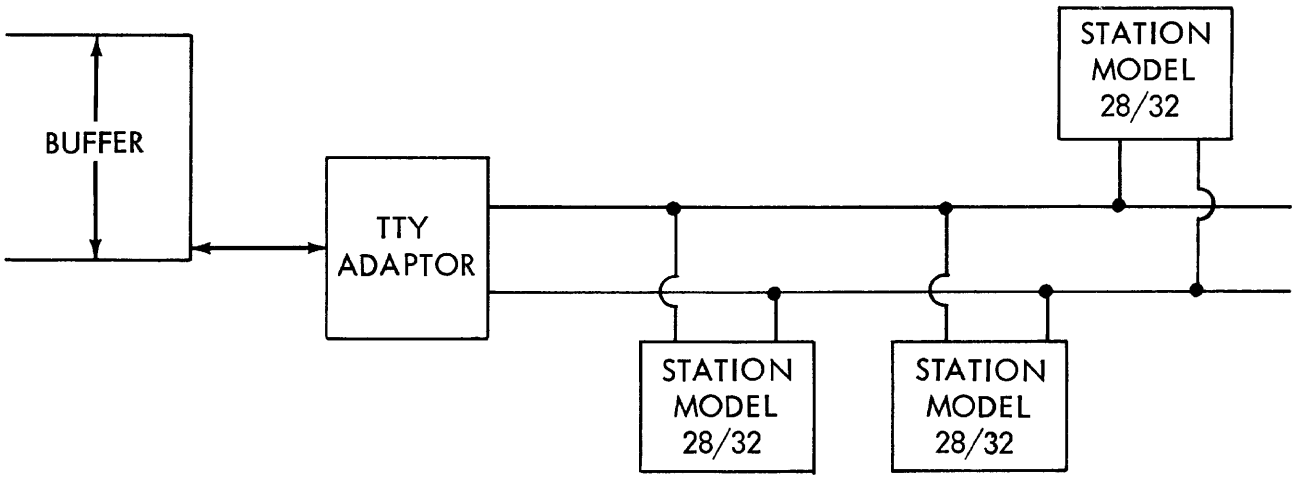


Figure 9-22. Teletype Network

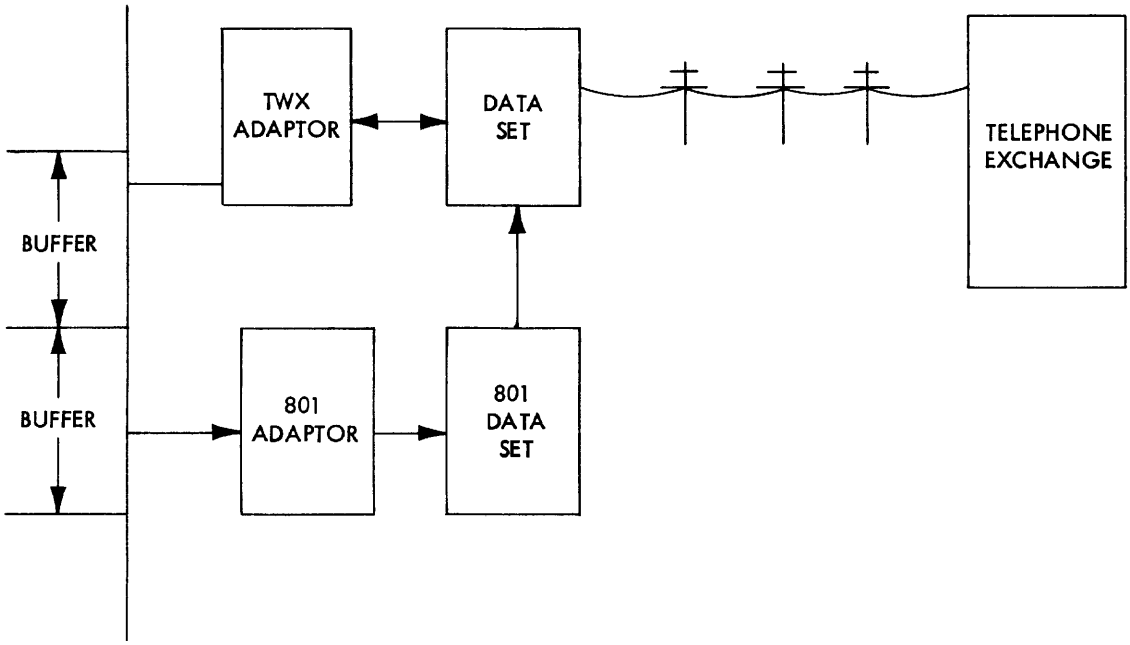


Figure 9-23. 801 ACU Connection

B 300, UNIVAC 1004

9-99. The B 300 or the UNIVAC 1004 can be connected with the system with their corresponding line adaptors. They both require two buffers of 56 characters each which will alternate on input. The positive or negative acknowledgment of messages between the B 5500 and the other computers is accomplished by the object programs in each computer.

Console

9-100. With the exception of the peripheral devices, all operator control will be from the console and the keyboard message printer. The indicators and switches on the console are described in table 9-10.

TABLE 9-10

Console Control Panel Switches and Indicators

SWITCH/INDICATOR	FUNCTION
HALT BUTTON	When pressed, this switch halts processor #1 and processor #2 after completion of syllables currently in process, and lights the indicator. The indicator remains lit, until the LOAD button is pressed and a load operation is initiated.
NOT READY	This is an indicator which notifies the operator that one or more units that are installed as part of the system, and that do not have local "Not Ready" indicators have become unavailable for normal use by the system. The following units are included in this check: both processors, core memory modules, I/O control units, magnetic drum storage unit #1, and message printer/keyboard. Other input/output peripheral units are not included. The indicator is not lit by signals indicating non-availability because of current use with the system.

TABLE 9-10 (Cont)

Console Control Table
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
MEMORY CHECK	The MEMORY CHECK indicator is lit when either a processor #1 memory parity interrupt or a processor #2 memory parity interrupt is set.
LOAD AND LOAD SELECT	The LOAD switch is pressed to initially load part of the Master Control Program into core memory and to start processor #1 processing it. If, at the time the switch is pressed, processor #1 is not idle, pressing this switch has no effect. If processor #1 is idle and the LOAD SELECT switch on the console is in the drum position, pressing the LOAD Switch loads 512 words (decimal) - 1000 octal - from the band 0 of the drum unit #1 into core memory, starting with cell 20 (octal). Control of processor #1 is then transferred to the program word in cell 20 (octal). If processor #1 is idle and the LOAD SELECT switch is in the card position, one binary card is read into core memory locations 20 (octal) through 43 (octal). Processor #1 then accesses cell 20 (octal) and transfers control to this program word. Card reader #1 must be used.
A NORMAL	This indicator is lit when processor #1 is in the normal state (NCSF = 1).
A CONTROL	This indicator is lit when processor #1 is in the control state (NCSF = 0).
B NORMAL	This indicator is lit when processor #2 is in the normal state (NCSF = 1).
B CONTROL	This indicator is lit when processor #2 is in the control state (NCSF = 0).
POWER-ON	This switch, when pressed, initiates the process of cycling power-on. Power is applied to the main frame and drum. Both processors are left idle. The indicator is lit and remains lit as long as power remains on.
POWER-OFF	This switch, when pressed, initiates the process of cycling power-off and cycles down the entire system (including all connected I/O equipment).

SUPERVISORY PRINTER

9-101. The supervisory printer and keyboard (referred to as the SPO) is the means by which instructions and certain information is printed out to the operator. The operator also may signal a reply or request information about programs in process or operational conditions from the MCP. The SPO is a character at a time printer, using BCL code as input. The output of the keyboard is also in BCL code. The printer provides for communication between the operator and this system. The printer is used to inform the operator of conditions within the system that require operator action and to give the operator instructions. The keyboard is used to key in an acknowledgment to a message that has been printed out. The keyboard is also used to key in special instructions for the MCP. The REMOTE switch must be pressed when the printer/keyboard is to be used with the system. If it is to be used locally, the LOCAL switch must be pressed. Other indicators include the power lights.

Functional Characteristics

9-102. When the operator wishes to key in a message by use of the keyboard, the INPUT REQUEST key (table 9-11) must first be pressed, setting a keyboard request interrupt (CC 105F) in central control. When an I/O control unit has been selected and initiated on a keyboard input operation, the READY light is turned on. The operator can then key in the message. At the completion of the message, the END-OF-INPUT key must be depressed. This signals the end of the message by inserting a group mark in the input string and releasing the I/O channel. The program or MCP controls subsequent action. If a keyboard I/O descriptor is initiated, and the operator is pressing the END-OF-INPUT button, the I/O control unit is released immediately even though information was not entered on the keyboard. Therefore, before consecutive keyboard operation, time should be allowed by the operator after removing his finger from the END-OF-INPUT key. The ERROR switch is used when a message being keyed is in error. When pressed, the ERROR causes the operation to be terminated by placing a group mark (octal 37) in the input string. This group mark is set with a parity error so that the result descriptor reflects the error condition. The operator would then retry the operation. Since this unit is an I/O device, operation is by way of an I/O control unit and no processing time is lost during its use. The rest of the controls on the keyboard are for normal typewriter operation.

TABLE 9-11

Supervisory Printer
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
INPUT REQUEST	A momentary contact switch that begins feeding the input message stored in memory when pressed.
READY	This indicator lights when the computer reaches a Read Supervisory Print instruction.
REMOTE	A self-indicating switch that places the printer under control of the associated central processor.
LOCAL	A self-indicating switch that removes the printer from control of the central processor. Operator can type comments when this switch is on.

TABLE 9-11 (cont)

Supervisory Printer
Switches and Indicators

SWITCH/INDICATOR	FUNCTION
POWER	This indicator lights when power is applied to the unit.
ERROR	The operator momentarily presses this switch in the event of an operator input error (i.e., keystroke error). The program control will continue in sequence.
END-OF-MESSAGE	Momentarily pressing this switch terminates the message. A group mark is stored in memory and the READY indicator is extinguished.

GENERAL DESCRIPTION OF I/O DESCRIPTORS

9-103. The following is a brief description of the control bits that are used in the I/O descriptors (bits 18 through 24) and their individual usage.

Bit 18 — Memory inhibit bit - When on, it will inhibit communication with memory. When off, this bit will allow communication with memory.

Bit 21 — Binary/alpha bit - When on, this indicates the operation is to be in binary. When off, it indicates the operation is in alpha.

Bit 22 — Direction bit - This bit is used in conjunction with magnetic tape. When on, it indicates that the tape is to be read in a backward direction. When off, it indicates that the tape is to be read forward.

Bit 23 — Word counter bit - When this bit is on, the operation will use the word counter, bits 8 through 17, to control the number of words read or written. When off, the word counter is not used.

Bit 24 — Input/output bit - When this bit is on, the operation is an input operation. When off, the operation is an output operation (in respect to the respective input/output unit).

9-104. All I/O descriptors are marked as a descriptor because bit 0 is set. The status of this bit, or bit 2 (except drum) is not checked by the I/O control unit when the I/O descriptor is brought into the W register and; therefore, allows these two bits to be reset in the I/O descriptor. The I/O operation will be unaltered if these two bits are left reset. The unit designate bits, 3 through 7, have the following binary weight (see Appendix B for list of peripheral units and corresponding unit designate numbers):

Bit 3 = 16
Bit 4 = 8
Bit 5 = 4
Bit 6 = 2
Bit 7 = 1

GENERAL DESCRIPTION OF RESULT DESCRIPTOR

9-105. A result descriptor is constructed for each piece of peripheral equipment at the time the I/O operation is terminated. This descriptor will have all the corresponding bits set in it that are set into the initiating I/O

descriptor except for bits 0, 1, 2 which will always be reset. The word count bit should always be at 0. Bit 30 through 32 will have the following meanings for each piece of peripheral equipment.

Bit 26	If set, indicates that a memory overflow occurred or an attempt was made to access a non-existent memory module.
Bits 27, 28, 29	These bits have individual error meanings and depend upon the piece of peripheral equipment referenced by the result descriptor.
Bit 30	If set, the designated unit was not ready.
Bit 31	If set, indicates that a parity error occurred while accessing the address of the I/O descriptor from cell 10 (octal) or accessing the I/O descriptor itself.
Bit 32	If set, indicates that an identical unit designate number is already present on another I/O control unit. Bits 26 through 32 (error bits) will be equal to 0 if the I/O operation was initiated without error.
Bits 33 - 47	These are discussed relative to each unit.

9-106. If the designated unit was in operation, and the unit designation was not present in another I/O control unit, the initiating I/O control unit will wait until the previous operation has been completed. This is valid for all units except magnetic tape units. The difference is that if a read or write operation is to be initiated, and the tape unit is in a rewind status, the initiating I/O control unit will terminate and bit 32 will be set in the result descriptor. The result descriptor will be stored in the following octal core address for the respective I/O control unit:

I/O 1 in 14
I/O 2 in 15
I/O 3 in 16
I/O 4 in 17

Supervisory Printer I/O-Result Descriptors

9-107. The purpose of the I/O descriptor is to print information from memory via the SPO. The lower 15 bits of the descriptor specify the core starting address. Character 0 of the designated address is the first character transferred to the printer. The descriptor format is as follows:

SUPERVISORY PRINTER

0	3	6					24			33				
1	4	7												
	5													47

I/O Descriptor:

The following bits are on:

0 Flag bit.
2 Presence bit.
3, 4, 5, 6 Unit designate equal 30.
All control bits off.
33-47 Starting core address.
All other bits must be off.

Seventy characters may be printed on one line by the SPO before the carriage is automatically returned. The print operation is terminated when a group mark (octal 37) is encountered in memory. The group mark is not printed, but will cause automatic carriage return and release the I/O control unit.

Result Descriptor:

The following bits are on:

Bits 0 - 24 These bits have the same meaning as bits 30 - 32 as stated in paragraph 9-79.

Bit 27 - 28	Not used.
Bit 29	Memory to SPO parity error.
Bits 33 - 47	The address within these bits will be the last word address in the area to be printed, plus 1.

Keyboard I/O-Result Descriptors

9-108. The purpose of this descriptor is to read information from the keyboard on the SPO, and transfer this information to the core address specified by the lower 15 bits of the I/O descriptor. The format of the descriptor is as follows:

KEYBOARD

0	3	6					24			33				
1	4	7												
	5													47

I/O Descriptors:

The following bits are on:

0	Flag bit.
2	Presence bit.
3, 4, 5, 6, 7	Unit designate equal 30.
24	Control bit for reading.
33-47	Starting core address.

The first character transferred from the keyboard will be relative to character position 0 of the word pointed to by the core address field. The transfer will then be to successive higher word positions.

Result Descriptor:

The following bits are on:

Bits 0-24, Bits 30-32	Have the same meaning as stated in paragraph 9-79.
Bit 27	Not used.
Bit 28	Parity error received from keyboard. Error button pressed.
Bit 29	Not used.
Bits 33 - 47	The address of the last word that information was entered in (including the group mark) plus 1.

Magnetic Drum

9-109. A magnetic drum-type memory is used as a fast access auxiliary storage device for the B 5500. This is the B 430 Magnetic Drum Memory. The B 430 is considered an I/O device; and as such, all access is through the I/O control units. There may be one or two drum units. Both drums are contained in one cabinet. Each drum unit is independent of the other in operation. Each drum has a capacity of 32,768 words segmented into 64 channels of 512 words each. The 512 words in a channel are interlaced to provide core memory access time. The format of a channel is shown in figure 9-24. Starting from the blank space, or zero point on the drum, the first word to pass the read/write head is the word 0, then word 256, then word 1, then 257, etc. During a drum operation, every other word in a particular section of a channel is either read or written. If 512 words are to be read from one channel, then the first 256 words are read on the first revolution of the drum and the second 256 words are read on the second revolution. The maximum number of words that may be read or written with any one operation is 1023. In this case, successive channels are used. Each word contains 10 frames of 6 bits each. These 10 frames consist of 8 characters, 1 parity digit, and a space. Each of the 8 characters is made up of 6 bits representing either an alphanumeric character or two octal digits. The information is written on the drum exactly as it appeared in core memory. No encoding or decoding is used. Average access time is about 8.5 ms and maximum access time is about 17 ms. Drum rotation is 3600 RPM.

Magnetic Drum I/O-Result Descriptors

9-110. The purpose of these descriptors is to read from or write onto the drum.

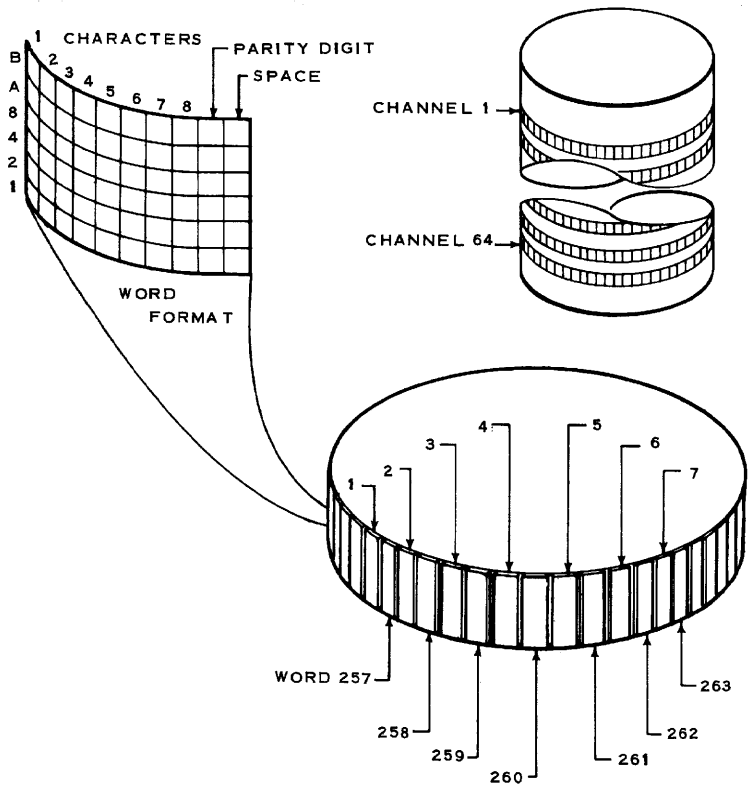


Figure 9-24. Magnetic Drum Format

DRUM

0	3	6				18					33				
1	4	7													
2	5	8			17					32					47

I/O Descriptor:

Bit 0	Flag bit.
Bit 2	Read/write indication bit.
Bits 5-6	Indicates the desired drum unit.
Bit 5	Allows addressing of drum unit #1.
Bit 6	Allows addressing of drum unit #2.
Bits 8-17	Indicates the number of words that are to be read from or written into the drum.
Bits 18-32	The starting drum address.
Bits 33-47	The starting core address.

The following is an example of the usage of this descriptor for reading from the drum. The number of words indicated by bits 8 through 17 will be transferred from the drum to core memory. The starting drum address is in bits 18-32 and the starting core address is in bits 33 through 47. Both addresses are counted up sequentially as the words are transferred. If the drum address is caused to count from 77777 to zero, a drum address overflow results and the drum will be placed in a not-ready status. To place the drum back in a ready status requires manual intervention.

Drum Read Result Descriptor:

Bits 0-24, 30-32	The same as previously discussed.
Bits 27-28	Not used.
Bit 29	Parity error from drum to I/O.
Bits 33-47	Address of the last core word read into plus one.

Drum Write Result Descriptor:

Bits 0-24, 30-32 } The same as previously discussed.

Bit 27 — Tried to write on a locked out portion of the drum.

Bit 28 — Not used.

Bit 29 — Parity error from memory to I/O.

Magnetic Tape Unit

9-111. There may be up to 16 tape units connected to the B 5500. Each unit is separate in operation, since all system control is by way of the I/O control units and I/O exchange in central control. Simultaneous operation of several units is possible, providing there is more than one I/O control unit. The tape has seven channels. Six of these are information channels and one is lateral parity. In figure 9-25, a frame refers to a one bit space in each of the seven channels. Each frame constitutes either an alphanumeric character or two octal digits (6 binary bits). Alphanumeric characters are coded in BCL code, and even parity is used. Odd parity is used when information is coded in binary. Information is recorded in variable length groups with blank spaces of tape between the groups. These groups of information are called records. Each record is written entirely in either alphanumeric or binary code. The length of a binary coded record is from 1 to 1023 words. Alphanumeric records are limited to length by the core memory capacity only and is controlled, when written, by a group mark in memory. At the end of every record is a longitudinal parity bit. This longitudinal parity digit has a bit in every channel that contains an odd number of bits. In this way, every channel in a record has an even number of bits. A tape read operation may be conducted with forward or backward tape movement. A read operation is terminated in one of two ways. A number of words may be specified to be read and upon completion of transferring this specified number of words to memory, the transfer of information to memory is terminated; however, the operation is not complete until the longitudinal parity bit is encountered. If the end of the record is encountered before the specified number of words has been read, the operation is terminated The end of the record may be used to terminate a read operation; in which case, the whole record is read into memory. However, the tape movement is not terminated until an inter-record gap is encountered. The result descriptor address will reflect the actual number of words read from the tape unit by subtracting the starting address from the address found in the result descriptor. Only forward tape motion is permitted with a tape write operation. A specified number of words (from 1 to 1023) may be written in binary. An alphanumeric write is terminated by a group mark (octal 37) in core memory. This permits any number of characters to be written, up to a maximum of the memory capacity. The dual gap heads permit a simultaneous write/read operation to check the parity of the information written during a tape write operation. In addition to read and write operations, tape can be spaced forward or backward, rewound, or erased. The length of tape erased is controlled in the same way as the record length in a write operation. The tape is magnetized in one direction so that tape is blank in this area.

Magnetic Tape I/O-Result Descriptors:

9-112. The purpose of these descriptors is to control the reading and writing on the tape unit designated with the unit designate field. The tape unit selected by the unit designate field is as follows for:

Unit Designate Number in Decimal	Tape Unit Letter
1	A
3	B
5	C
7	D
9	E
11	F
13	H
15	J
17	K
19	L
21	M
23	N

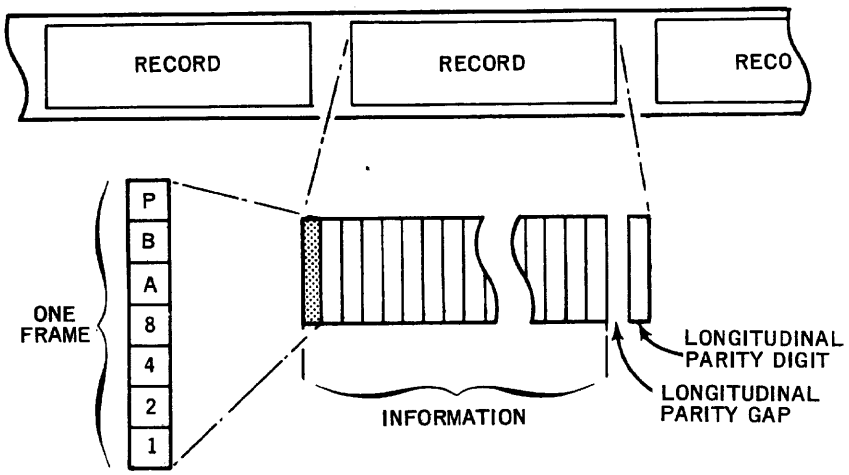


Figure 9-25. Tape Record Format

Unit Designate Number in Decimal	Tape Unit Letter
25	P
27	R
29	S
31	T

Binary tapes are read and written with the control of the word counter bits. When writing binary tapes, the number of words specified by the word count field is transferred from memory to the tape unit. When reading a binary tape, the number of words specified by the word count field is transferred from the tape unit to memory.

I/O descriptors for reading tape have the following format:

TAPE READ - LONGITUDINAL PARITY GAP

0	3	6				18	21	24			33				
1	4	7					22								
	5						23								47

Alphanumeric Tape Read:

Bit 0	Mark the word as a descriptor.
Bit 2	Mark the word as present.
Bits 3-7	Unit designate bits.
Bits 8-17	Not used.

Control Bits:

Bit 18	Off because tape is to be read into memory.
Bit 21	Reset because reading alpha tape.
Bit 22	May be set or reset. If reset, tape is read forward. If set, tape is read backward.
Bit 23	Reset, if word count field is not used. Set, if word count field is to be used.
Bit 24	Must be on to read from tape unit.
Bits 33-47	Starting address to read tape into. If read forward, address will be incremented. If read backward, address will be decremented.

The read operation is terminated when an inter record gap is reached on the tape. The word count field may be used to terminate the reading operation if bit 25 is set.

TAPE READ - WORD COUNTER

0	3	6				18	21	24			33				
1	4	7					22								
	5	8			17		23								47

Binary Tape Read:

Bit 0 — Mark the word as a descriptor.

Bit 2 — Mark the word as present.

Bits 3-7 — Unit designate bits.

Bits 8-17 — Number of words to be used.

Control Bits:

Bit 18 — Reset because tape is to be read into memory.

Bit 21 — Set to read a binary tape.

Bit 22 — Reset to read forward, set to read backward.

Bit 23 — Must be set because word counter must be used.

Bit 24 — Must be set to allow reading from tape.

Bits 33-47 — Same as alpha read.

The transfer of information to memory is terminated when the specified number of words have been read from the tape.

Magnetic Tape Read Result Descriptor (Binary or Alpha Mode):

Bits 3-24 — The same as the Initiation I/O Descriptor.

Bits 30-32 — Discussed under general description of result descriptors.

Bit 27 — End of file (tape mark (15) was encountered on tape).

Bit 28 — Character parity error from tape to I/O, either latitudinal or longitudinal.

Bit 29 — Not used.

Bits 33-47 — If the tape is read forward, will contain the address of the last core read into plus one word. If the tape was read backward, these bits will contain the address in the original I/O descriptor minus the number of words read, minus one more word.

I/O descriptors for writing tape have the following format:

TAPE WRITE

0	3	6									33				
1	4	7													
	5	8			17										47

Alphanumeric Tape Write:

Bit 0 — Mark the word as a descriptor.

Bit 2 — Mark the word as present.

Bits 3-7 — Unit designate bits.

Bits 8-17 — Not used.

Control Bits:

Bit 18 — Reset because information is to be read from memory.

Bit 21 — Reset because this is to be written in alpha.

Bit 22 — Must be reset due to writing only allowed in the forward direction.

Bit 23 — Reset the word count bit, not used in alpha mode.

Bit 24 — Reset to allow writing on the tape.

Bits 33-47 — Starting core address.

The write operation is terminated when a group mark is encountered in memory.

Binary Tape Write:

The following bits are to be set within the descriptor:

Bit 0	Mark the word as a descriptor.
Bit 2	Mark the word as present.
Bits 3-7	Unit designate bits.
Bits 8-17	Number of words to be used.

Control Bits:

Bit 18	Reset to allow reading from memory.
Bit 21	Set due to binary writing desired.
Bit 22	Must be reset due to only forward writing allowed.
Bit 23	Must be set because word counter is to be used.
Bit 24	Must be reset to allow writing on tape.
Bits 33-47	Starting core address.

The write operation will be terminated when the designated number of words have been written. When the designated number of words have been written, an inter record gap is placed on the tape.

Magnetic Tape Write Result Descriptors:

Bits 3-24	Same as the initiation I/O descriptor.
Bit 27	End-of-file (tape mark (15) was encountered on tape).
Bit 28	Character parity error from tape to I/O, either latitudinal or longitudinal.
Bit 29	Not used.
Bits 30-32	Discussed under general description of result descriptors.
Bits 33-47	Address of last cell written, plus one.

Printer I/O-Result Descriptors:

9-113. The descriptor for printer operations will print one line of print (15 computer words) with the most significant character in print position 1, least significant character in print position 120.

PRINTER

0	3	6				18		24	27	30	33			
1	4	7							28	31				
	5	8							29	32				47

I/O Descriptors:

Bit 0, 1	1, 0 (flag, I.D. bits).
Bit 2	On (presence bit).
Bits 3-7	Unit designate u, u = 22 or 26 (binary).
Bits 8-17	Not used.
Bit 18	0 print, 1 inhibit print. Space paper as specified by bits 21-16.
Bit 19, 20	XX (integer, continuity bits).
Bit 21, 22	00
Bit 23	0
Bit 24	0 Output operation.
Bit 25, 26	Not used.
Bit 27, 28	Space bits; 00 no space, 01 double space, 10 single space, 11 double space.
Bits 29-32	For control F. When F = 0, space paper as specified in bits 21, 20. When F $\neq$ 0, skip to the stop specified by printer loop control, channel F.
Bits 33-47	Starting memory address.

Result Descriptors

Bits 3-24	Same as initiation I/O descriptor.
Bit 27	End-of-page. A channel 12 punch was detected in format tape. The condition will be reset in the printer when a skip descriptor is initiated. If the skip descriptor is not initiated, the printer will continue to print on the same line.
Bit 28	Print check on previous line of print. Reset next time printer is initiated.
Bit 29	Parity error from memory to I/O.
Bits 30-32	Same as standard I/O error bits.
Bits 33-47	Starting core address minus one.

Card Reader I/O Result Descriptors:

9-114. This descriptor will read one punched card.

CARD READ

0	3	6					21	24			33				
1	4	7													
	5														47

I/O Descriptors:

Bit 0, 1	1, 0 (flag, I.D. bits).
Bit 2	On (presence bit).
Bits 3-7	Unit designate u, u = 10 or 14 (binary).
Bits 8-17	Not used.
Bit 18	0
Bit 21	0 alphanumeric input, 1 binary input.
Bit 22, 23	0
Bit 24	1 input operation.
Bits 25-32	Not used.
Bits 33-47	Starting memory address.

Result Descriptors:

Bits 3-24	Same as initiating I/O descriptor.
Bit 27	End-of-file. Hopper was empty and operator pressed end-of-file button on panel.
Bit 28	Read check error. Will be reset at next initiation of card reader.
Bit 29	Invalid character. Could not be set if reading in binary.
Bits 30-32	Same as standard configuration.
Bits 33-47	Last core address read into plus one.

Punch I/O Result Descriptors:

9-115. The descriptor for card punch operations will punch one card.

CARD PUNCH

0	3	6						24			33				
1	4	7													
	5														47

I/O Result Descriptors:

Bit 0, 1	1, 0 (flag, I.D. bits).
Bit 2	On (presence bit).
Bits 3-7	Unit designate u, u = 18 (binary).
Bits 8-17	Not used.
Bit 18	0
Bit 21,22	00
Bit 23	0
Bit 24	0, output operation.
Bits 25-32	Not used.
Bits 33-47	Starting memory address.

Result Descriptors:

Bits 3-24	Same as initiating I/O descriptor.
Bit 28	Punch error would not be detected until third card is initiated for punching.
Bit 29	Parity error from memory to I/O.
Bits 30-32	Same as standard error field.
Bits 33-47	Last core address read, plus one.

Paper Tape Reader I/O Result Descriptors:

9-116. The I/O descriptor for the paper tape reader is basically a read descriptor but has space and rewind variants.

PAPER TAPE READ OR REWIND

0	3	6				18	21	24			33				
1	4	7					22								
	5	8			17		23								47

Paper Tape I/O Read Descriptors:

Bit 0	Always for descriptor.
Bit 2	On (presence bit).
Bits 3-7	Unit designate u, u = 18 or 20.
Bits 8-17	Word counter (0 = no op).
Bit 18	0 paper tape read, 1 paper tape space.
Bits 19-20	Integer and continuity bits.
Bit 21	0 alphanumeric read, 1 binary read.
Bit 22	0 read or space as indicated by bit 30, 1 rewind.
Bit 23	Always on (must use word counter).
Bit 24	Always on (input operation).
Bits 33-47	Starting memory address.

Paper Tape Read Result Descriptors:

Bits 0-24	Same as initiating I/O descriptor.
Bit 28	Beginning of tape marker.
Bit 29	Parity error from reader to I/O.
Bits 30-32	Same as standard error field.
Bits 33-47	Last core address read into, plus one.

Paper Tape Punch I/O Result Descriptors:

PAPER TAPE PUNCH

0	3	6				18	21	24			33				
1	4	7													
	5	8			17		23								47

Bit 0, 1	1, 0 (flag, I.D. Bits).
Bit 2	On (presence bit).
Bits 3-7	Unit designate.
Bits 8-17	Word counter (max. = 1023; min. 1; 0 = no op). Word count n specifies the number of words to be punched in binary mode and the maximum number of words in alpha mode. In alpha mode, an end-of-file character may terminate the punch operation before the word counter is equal to zero.
Bit 18	1 - Used with D27F on to indicate a tape feed operation (no information transferred).
Bit 21	0 alphanumeric, 1 binary.
Bit 22	Not used.

Bit 23 — 1 - inhibit memory transfer after word count n has been satisfied. In alphanumeric, the record may be terminated by an end-of-message character or WC = 0 (whichever occurs first). Bit 25 is always on regardless of type (binary alpha) of record being punched.

Bit 24 — 0 - Output operation.

Bits 33-47 — Starting memory address.

Paper Tape Punch Result Descriptors:

Bits 3-24 — Same as initiating I/O descriptor.

Bit 27 — Physical end-of-tape marker sensed. This indicates that approximately 35 feet of paper tape are left.

Bit 28 — Not used.

Bit 29 — Parity error from core memory.

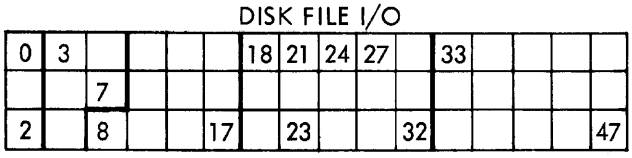
Bits 30-32 — Same as standard error field.

Bits 33-47 — Last word of core memory read, plus one.

Disk File I/O-Result Descriptors:

9-117. The descriptor for disk file will read or write up to 63 segments. If less than one segment is to be read or written, then word count indicates the number of words. The last seven characters of the first word addressed by bits 33-47 contain the disk file address in BCD format.

DISK FILE I/O



Disk File I/O Descriptors:

Bit 0 — 1 (flag, 1 = descriptor).

Bit 2 — X (presence bit, 1 = core memory assigned).

Bits 3-7 — Unit designate.
BCD 6 = Disk File Control Unit #1 (or 14, octal)
BCD 12 = Disk File Control Unit #2 (or 30, octal)

Bits 8-17 — Word count (max. = 1023 decimal).

Bit 18 — 1 read check - inhibit data transfer.

Bit 21 — 1 binary, 0 alpha (BCL).

Bit 23 — 1 word counter override (when 0, less than one segment is to be read).

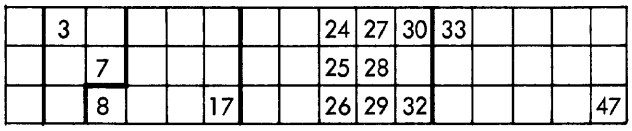
Bit 24 — 1 disk file read, 0 disk file write.

Bits 27-32 — Number of segments (max. = 63 decimal).

Bits 33-47 — Starting core memory address.

In table 9-12, the disk descriptor combinations are related to the operation to be performed.

DISK FILE RESULT



Disk File Result Descriptor:

Bits 0-2 — 0.

Bits 3-7 — Unit designate.

Bits 8-17 — Remaining word count.

Bit 24 — 1 if operation was read; 0 if write.

Bit 25 — 1 if read check error on prior operation.

Bit 26 — 1 if core memory address error.

Bit 27 — 1 if disk file electronics unit is NOT READY, or an attempt to access non-existent disk address.

Bit 28 — 1 if PARITY ERROR on transfer of data from disk to I/O during read operation.

Bit 29 — 1 if core memory parity error; parity error during: disk file address transfer, or data transfer during write operation, to DF control unit.

Bit 30 — 1 if DF control unit NOT READY.

Bit 32 — 1 if DF control unit is busy with another I/O channel.

Bits 33-47 — 1 = last address accessed +1 for all read/write operations or, initial address +1 for read check and interrogate operations.

TABLE 9-12

Descriptor Combinations

8 - 17 Word Count	18	21	23	24	27 - 32 Segment Count "n"	Operation
	1				$1 \leq n \leq 77\phi$	READ CHECK
		0	0	1	$1 \leq n \leq 77\phi$	Read with BCL translation; ignore Word Count
		1	0	1	$1 \leq n \leq 77\phi$	Read without translation (binary); ignore Word Count
$1 \leq WC \leq 1777\phi$		0	1	1	$1 \leq n \leq 77\phi$	Read with BCL translation; Word Count Override
$1 \leq WC \leq 1777\phi$		1	1	1	$1 \leq n \leq 77\phi$	Read without translation; Word Count Override
		0	0	0	$1 \leq n \leq 77\phi$	Write with BCL translation; ignore Word Count
		1	0	0	$1 \leq n \leq 77\phi$	Write without translation; ignore Word Count
$1 \leq WC \leq 1777\phi$		0	1	0	$1 \leq n \leq 77\phi$	Write with BCL translation; Word Count Override
$1 \leq WC \leq 1777\phi$		1	1	0	$1 \leq n \leq 77\phi$	Write without translation; Word Count Override
WC - 0			1			Interrogate

NOTE: The "0" and "1" are required where shown, and blanks are irrelevant. ϕ means octal.

9-118. One B 5480 Data Communication Control Unit (DCCU) may be attached to the B 5500 Processor and I/O control unit in addition to the other I/O devices available to the system. One control unit controls up to 15 terminal units. It is possible to connect any combination of teletype and typewriter terminal units to a maximum of 15.

9-119. INPUT OPERATIONS. After an inquiry station has entered a complete message, the related terminal unit is in the "Input Ready" state. When the DCCU scanner addresses the terminal unit, the scanner stops and an interrupt bit is set in the interrupt register provided no other terminals are "Output Ready". An I/O control unit is latched to the DCCU and the terminal unit buffer is then loaded into a designated area of core memory. At the completion of the data transfer, the result descriptor generated contains the address of the terminal unit to which the scanner is latched. The I/O control unit is then unlatched from the DCCU and the scanner advances to the next ready terminal unit in sequence.

9-120. OUTPUT OPERATIONS. As indicated in the general systems description, under normal operation, two basic buffer conditions are encountered: one in which the inquiry or reply is shorter than the selected terminal unit buffer capacity; the second occurs when a reply or inquiry, because of its length, exceeds the terminal unit buffer capacity and would be broken up into several full buffer capacity units. The detailed methods for handling these two conditions are outlined below.

9-121. NOT EXCEEDING TERMINAL UNIT BUFFER CAPACITY. After the appropriate inquiry routine has processed the inquiry message and has assembled the reply message, an inquiry write operation is initiated using the address of the terminal unit which was contained in the read result descriptor. An I/O control unit is latched to the DCCU and the designated terminal unit buffer is loaded from core memory. When the end of reply character (group mark) is detected by the I/O control unit, the operation is terminated. The I/O control unit and designated terminal unit are unlatched from the DCCU. The scanner advances to the next ready terminal unit in sequence.

9-122. EXCEEDING TERMINAL UNIT BUFFER CAPACITY. After the appropriate inquiry routine has processed the inquiry message and has assembled the reply message, an inquiry write operation is initiated using the address of the terminal unit which was contained in the read result descriptor. An I/O control unit is latched to the DCCU and the designated terminal unit buffer is loaded from core memory. When the buffer is fully loaded, detecting an end-of-reply character (group mark), the I/O control unit is unlatched from the DCCU. The scanner advances to the next ready terminal unit while the buffer of the original terminal unit is unloaded to the inquiry station. During the time that the first terminal unit is unloading its buffer to the inquiry station, the DCCU can service other terminal units as required. When the original terminal unit addressed has completed unloading its buffer, it is set to the "Output Ready" state. When the scanner addresses this unit again, it stops and an interrupt bit is set in the interrupt register. The inquiry control program then initiates an inquiry read operation. An I/O control unit is latched to the DCCU. No operation takes place, but the result descriptor generated provides the ICP with the terminal unit address and status. An inquiry write operation is initiated and if this part of the message contains an end-of-reply character (group mark), the operation is terminated and the I/O control unit and designated terminal unit are unlatched from the DCCU. If no end-of-reply character is detected, the operations outlined above are repeated until the complete message has been sent to the inquiry station.

9-123. OUTPUT OPERATION (COMPUTER INITIATED MESSAGES - TELETYPE TERMINAL UNITS ONLY). This output operation allows for B 5500 programs to independently initiate messages to teletype stations on a teletype terminal network. The B 5500 program initiates an inquiry write operation and an I/O control unit is latched to the DCCU. If the DCCU is not busy, the scanner

is set to the terminal unit designated by the inquiry write descriptor. If the terminal unit is "Not Ready" or "Busy" (message from an inquiry station being loaded, unrelated call on the net), or "Input Ready", the write operation is terminated and the result descriptor contains the "Busy" bit or the "Input Ready" bit. If the designated terminal unit is "Ready", the operation continues as described in the previous output operations.

Data Communication I/O-Result Descriptors

9-124. The following are the data descriptors used with the data communication system when attached to the B 5500.

Bit 0	1 (flag bit).
Bit 1	1 (identification bit).
Bit 2	Presence bit.
Bits 3-7	Unit designation (DCCU 16).
Bits 8-17	Not used.
Bit 18	0 (Zero).
Bit 19	Integer bit.
Bit 20	Continuity bit.
Bit 21	0 (Zero) alphanumeric mode.
Bits 22-23	00.
Bit 24	1 indicates read operation.
Bits 25-28	Not used.
Bits 29-32	0 (Zero).
Bits 33-47	Starting memory address.

INQUIRY READ

0	3					18	21	24			33				
1		7				19	22	25	28						
2		8			17	20	23		29	32					47

Inquiry Read Result Descriptor:

The following positions only are used:

Bits 10-13	Address of terminal unit.
Bit 26	Memory address error. Either memory overflow or attempt to access a nonexistent memory address.
Bit 27	Terminal unit is "Output Ready."
Bit 28	Terminal unit is "Busy."
Bit 29	Memory parity error in transfer from DCCU to I/O control unit.
Bit 30	DCCU or terminal unit addressed is "Not Ready."
Bit 31	Descriptor parity error, either on descriptor address access (memory location 8) or I/O descriptor access.
Bit 32	DCCU is "Busy."

INQUIRY READ RESULT

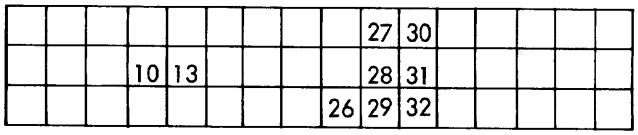
									27	30					
			10	13					28	31					
								26	29	32					

Inquiry Write Descriptor:

Bit 0	1 (flag bit).
Bit 1	1 (identification bit).
Bit 2	Presence bit.
Bits 3-7	Unit designation (DCCU = 16).
Bits 8-17	Not used.
Bit 18	0 (Zero).
Bit 19	Integer bit.

Bit 20	Continuity bit.
Bit 21	0 (Zero) indicates alphanumeric mode.
Bits 22-23	00.
Bit 24	0 (Zero) indicates write or output operations.
Bits 25-28	Not used.
Bits 29-32	Address of terminal unit.
Bits 33-47	Starting memory address.

INQUIRY WRITE



Inquiry Write Result Descriptor:

In addition to the basic inquiry write descriptor, the following bit positions are used:

Bits 10-13	Address of terminal unit.
Bit 26	Memory address error. Either memory overflow or attempt to access a nonexistent memory.
Bit 27	Terminal unit addressed is "Input Ready."
Bit 28	Terminal unit addressed is "Busy."
Bit 29	Memory parity error on transfer from I/O control unit to DCCU.
Bit 30	DCCU or terminal unit addressed is "Not Ready."
Bit 31	Descriptor parity error, either a descriptor address access (memory location 8) or I/O descriptor access.
Bit 32	DCCU is "Busy."

Data Transmission

9-125. The Data Transmission Terminal Unit (DTTU) provides the facilities for up to 16 simultaneous transmissions. For use of more than one terminal (or if translation is required between ASCII and BCL or BAUDOT and BCL) the Data Transmission Control Unit must be used.

9-126. The Data Transmission Terminal Unit provides a buffer memory to store simultaneous messages from up to 16 different lines. The basic buffer size is 28 characters per line or a total of 448 characters per DTTU. Multiples of 28 characters may also be used for one line.

9-127. TYPEWRITER/TWX LINE ADAPTOR. In operation, the operator, if required, completes the connection following normal dialing procedures. The operator then enters a message via the station keyboard or paper tape reader. The message is terminated by the End-of-Message character (←). The buffer is set to the Read Ready Interrupt state.

9-128. By use of the Interrogate command, the processor detects this condition and obtains the buffer number. The processor then executes a read command which transfers the information in the terminal unit buffer to the processor memory. The buffer is then idle and an output message can then be sent or another input message entered.

9-129. On input, if the message is longer than the buffer length, the buffer will be filled before the End-of-Message (EOM) character is received. When the buffer is filled, it is set to the Read Ready Interrupt state. If the buffer is emptied before the next character is accumulated, no information is lost. However, if the buffer is not emptied when the next character has been accumulated, information is lost. The buffer is set to the Busy Interrupt state at the completion of the processor read operation. If two 28-character buffers are alternatively loaded with input, more time is allowed for the processor to empty the buffers.

9-130. On output, if a reply exceeds buffer capacity, the buffer is set to the Write Ready Interrupt state after the contents of the buffer

are transmitted to the associated station (without detection of an EOM code). After the final output (containing an EOM code), the buffer will be in an idle state.

9-131. A line loss or a disconnect is detected by the adapter resulting in the buffer being set to the idle interrupt state with the abnormal condition flag set in the result descriptor. A break (operator-generated during typing of output) on output results in the buffer being set to the read ready interrupt state and the abnormal condition flag set in the result descriptor.

9-132. On input, the "?" character causes the abnormal condition flag to be set. (This is used by the MCP for system messages.)

9-133. TELETYPE LINE ADAPTORS. The teletype system works in the same manner as the typewriter system except for the operational characteristics (e.g., the need to first call out the stations to be talked to) and the code. The teletype code is BAUDOT. (The special EOM code is blank blank.)

9-134. DATASPEED II LINE ADAPTOR. The Dataspeed II line adaptor works like the typewriter line adaptor with alternating buffers. On input, parity errors are detected and the abnormal condition flag is set in the result descriptor. When a parity error is detected, that character is replaced by the "?" character.

9-135. 801 AUTOMATIC CALLING LINE ADAPTOR. Each Automatic Calling adaptor provides automatic calling for one other line. (The Bell 801A1 Automatic Calling Unit with the Z-option is used.) The number to be dialed is loaded into the buffer by a write command. If a connection is made, the buffer provides an interrupt with an idle state. If a connection is not made, the buffer goes to the Write Ready Interrupt state.

9-136. IBM 1050 LINE ADAPTOR. This adaptor controls an IBM 1050 Data Communications System. Any multiple number of 28 characters may be used for this device. A station addressing sequence can be initiated by a Write command. The response from the remote station is stored in the first character of the buffer followed by a group mark. The buffer is then set to the Busy Interrupt state. (Response codes with a parity error are converted to a BCL minus code.) When there is no response from the remote station after two seconds have elapsed, a B code is stored in the first character of the buffer followed by a group mark. The buffer is then set to the Read Ready Interrupt state.

9-137. A station polling sequence is initiated by a Write command. When the response from the remote station is negative, the response code is stored in the first character of the buffer followed by a group mark code, and the buffer is set to the Read Ready Interrupt state. A character with vertical parity error is converted to a BCL minus code. When a parity error (vertical or longitudinal) is detected, a Bust Interrupt is set at the end of transmission. When the input message is interrupted for 20 seconds and the input message has not been completed, a B code is stored in the first character of the buffer followed by a group mark; and the buffer is set to the Read Ready Interrupt state.

9-138. The Write operation functions in the normal manner with the following exceptions:

a. After transmission to the remote station is completed, the response from the remote station is stored in the first character of the buffer followed by a group mark; and the buffer is set to the Read Ready Interrupt state.

b. Response codes with a parity error are converted to a BCL minus code and a Busy Interrupt is set.

c. When there is no response from the remote station after two seconds have elapsed, a B code is stored in the first character of the buffer followed by a group mark; and the buffer is set to the Read Ready Interrupt state.

9-139. When an input message block exceeds the assigned buffer size, the block is discarded. A B character is placed in the first buffer location and a group mark in the next location. The buffer is set to the Read Ready Interrupt state.

9-140. B 300, UNIVAC 1004 LINE ADAPTORS. The line adaptors for these two computers work in a fashion similar to the adaptors for the other devices. The adaptor uses a total of four 28-character buffers of the DTTU. On input, the two 56-character parts of the buffer are loaded alternately. On output, it is treated as one 112-character buffer.

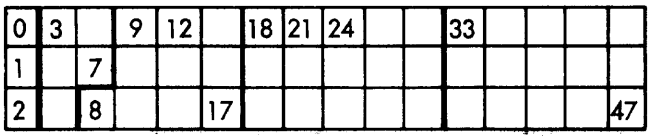
9-141. If a lateral parity error is sensed in the line adaptor, the character on which the error has been sensed is replaced with a "?" code and the abnormal condition flag is set. The buffer overflow condition is sensed in the adaptor and the buffer is set to the Busy Interrupt state. (This should be a rare occurrence, and the handling of this problem is a function of the object program.)

9-142. On output, if the object program did not precede the data of a message with four or more synchronizing characters, a Write Ready Interrupt occurs with the abnormal condition flag set.

Data Transmission I/O Result Descriptors

9-143. The following are Data Transmission I/O Result Descriptors:

Data Transmission I/O Descriptor



Bit 0 — 1 (flag bit).

Bit 1

Bit 2 — Presence bit.

Bit 3-7 — Unit designate.

Bit 8 — 0 (zero).

Bit 9-12 — Terminal Unit Number.

Bit 13 = 0 — Terminate buffer unloading or loading when a group mark is detected or buffer exhausted.

= 1 — Ignore group marks (sometimes called "transparent read/write").

Bit 14-17 — Buffer Number.

Bit 18 = 1 — Interrogate (bit 24 must be zero).

= 0 — Read or Write Operation.

Bit 19-20 — Not used.

Bit 21 = 1 — Bypass BCL Code to Internal Code Translator.

= 0 — Use the BCL Code to Internal Code Translator.

Bit 22-23 — Not used.

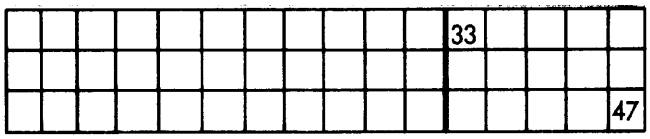
Bit 24 = 1 — Read.

= 0 — Write or Interrogate (depending on bit 18).

Bit 25-32 — Not used.

Bit 33-47 — Core Memory Address.

Data Transmission I/O Result Descriptor



Bit 3-7 — Unit Designate.

Bit 8 = 1 — DTCU not used.

= 0 — DTCU used.

Bit 9-12 — Terminal Unit Number.

Bit 13 = 1 — Ignore Group Marks.

= 0 — Group Mark Ending.

Bit 14-17 — Buffer number.

Bit 18 = 1 — Interrogate.

= 0 — Read or Write.

Bit 21 = 1 — Binary.

= 0 — Alpha.

Bit 23 — Adapter sensed abnormal condition.

Bit 25 = 1 — Buffer exhausted ending or buffer "filled" ending.

= 0 — Group mark ended.

Bit 26 — Memory Overflow.

Bit 24,27,28 (Interrogate Result Bits)

24	27	28	
0	0	0	Idle
0	0	1	Busy
0	1	0	Write Ready
1	0	0	Read Ready

Bit 27,28 (Read/Write Results Bits)

27	28	
0	0	Normal
0	1	Buffer Busy
1	0	Attempt to read "write ready" buffer or attempt to write "read ready" buffer.

Bit 28,30 (Any operation Result Bits)

28	30	
0	0	Normal
0	1	DTCU not ready
1	0	Buffer Busy
1	1	Buffer not ready (includes adapter not ready).

Bit 29 = 1 — Memory parity error.

Bit 31 = 1 — Descriptor parity error, either on descriptor address access (memory location 10) or I/O descriptor access.

Bit 32 = 1 — DTCU busy with another I/O channel.

Bit 33-47 — Core memory address.

APPENDIX A

CODES

CHAR.	INTERNAL CODE			BCL CODE		CARD CODE	
	BA	8421	OCTAL CODE	BA	8421	ZONE	NUM.
Blank	11	0000	60	01	0000	-	-
.	01	1010	32	11	1011	12	8 - 3
[	01	1011	33	11	1100	12	8 - 4
(	01	1101	35	11	1101	12	8 - 5
<	01	1110	36	11	1110	12	8 - 6
←	01	1111	37	11	1111	12	8 - 7
&	01	1100	34	11	0000	12	-
$	10	1010	52	10	1011	11	8 - 3
*	10	1011	53	10	1100	11	8 - 4
)	10	1101	55	10	1101	11	8 - 5
;	10	1110	56	10	1110	11	8 - 6
≤	10	1111	57	10	1111	11	8 - 7
-	10	1100	54	10	0000	11	-
/	11	0001	61	01	0001	0	1
,	11	1010	72	01	1011	0	8 - 3
%	11	1011	73	01	1100	0	8 - 4
=	11	1101	75	01	1101	0	8 - 5
]	11	1110	76	01	1110	0	8 - 6
"	11	1111	77	01	1111	0	8 - 7
#	00	1010	12	00	1011	-	8 - 3
@	00	1011	13	00	1100	-	8 - 4
:	00	1101	15	00	1101	-	8 - 5
>	00	1110	16	00	1110	-	8 - 6
≥	00	1111	17	00	1111	-	8 - 7
+	01	0000	20	11	1010	12	0
A	01	0001	21	11	0001	12	1
B	01	0010	22	11	0010	12	2
C	01	0011	23	11	0011	12	3
D	01	0100	24	11	0100	12	4
E	01	0101	25	11	0101	12	5
F	01	0110	26	11	0110	12	6
G	01	0111	27	11	0111	12	7

LOW ↑ COLLATING SEQUENCE ↓ HIGH

APPENDIX A (Cont)

CHAR.	INTERNAL CODE			BCL CODE		CARD CODE	
	BA	8421	OCTAL CODE	BA	8421	ZONE	NUM.
H	01	1000	30	11	1000	12	8
I	01	1001	31	11	1001	12	9
×	10	0000	40	10	1010	11	0
J	10	0001	41	10	0001	11	1
K	10	0010	42	10	0010	11	2
L	10	0011	43	10	0011	11	3
M	10	0100	44	10	0100	11	4
N	10	0101	45	10	0101	11	5
O	10	0110	46	10	0110	11	6
P	10	0111	47	10	0111	11	7
Q	10	1000	50	10	1000	11	8
R	10	1001	51	10	1001	11	9
≠	11	1100	74	01	1010	0	8 - 2
S	11	0010	62	01	0010	0	2
T	11	0011	63	01	0011	0	3
U	11	0100	64	01	0100	0	4
V	11	0101	65	01	0101	0	5
W	11	0110	66	01	0110	0	6
X	11	0111	67	01	0111	0	7
Y	11	1000	70	01	1000	0	8
Z	11	1001	71	01	1001	0	9
0	00	0000	00	00	1010	-	0
1	00	0001	01	00	0001	-	1
2	00	0010	02	00	0010	-	2
3	00	0011	03	00	0011	-	3
4	00	0100	04	00	0100	-	4
5	00	0101	05	00	0101	-	5
6	00	0110	06	00	0110	-	6
7	00	0111	07	00	0111	-	7
8	00	1000	10	00	1000	-	8
9	00	1001	11	00	1001	-	9
?	00	1100	14	00	0000	ALL OTHER CARD CODES	

LOW ↑ HIGH ↓

APPENDIX B

DESIGNATION NUMBER of PERIPHERAL UNITS

Type of Peripheral Unit	A Register Bit Position	Unit Designation
Magnetic Tape Unit A	1	1
Reserved For Expansion		2
Magnetic Tape Unit B	2	3
Drum Memory #1	17	4
Magnetic Tape Unit C	3	5
Disk File Control #1	19	6
Magnetic Tape Unit D	4	7
Drum Memory #2	18	8
Magnetic Tape Unit F	5	9
Card Reader #1	24	10
Card Punch	23	10
Magnetic Tape Unit G	6	11
Disk File Control #2	20	12
Magnetic Tape Unit H	7	13
Card Reader #2	25	14
Magnetic Tape Unit J	8	15
Data Communication Control	31	16
Magnetic Tape Unit K	9	17
Paper Tape Reader #1	28	18
Paper Tape Punch #1	27	18
Magnetic Tape Unit L	10	19
Paper Tape Reader #2	29	20
Paper Tape Punch #2	30	20
Magnetic Tape Unit M	11	21
Printer #1	21	22
Magnetic Tape Unit N	12	23
Reserved For Expansion		24
Magnetic Tape Unit P	13	25
Printer #2	22	26
Magnetic Tape Unit R	14	27
Reserved For Expansion		28
Magnetic Tape Unit S	15	29
Supervisory Printer And Keyboard	26	30
Magnetic Tape Unit T	16	31

APPENDIX C

OPERATORS, ALPHABETICAL LIST

WORD MODE

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
B EQUAL TO A	EQL	4425	BEQL
B GREATER THAN A	GTR	0225	BGAL
B GREATER THAN OR EQUAL TO A	GEQ	0125	BGEL
B LESS THAN A	LSS	4225	BLAL
B LESS THAN OR EQUAL TO A	LEQ	4125	BLEL
B NOT EQUAL TO A	NEQ	0425	BNEL
"B" STORE DESTRUCTIVE	STD	0421	BSDL
"B" STORE NON-DESTRUCTIVE	SND	1021	BSNL
BRANCH BACKWARD CONDITIONAL	BBC	0131	BBCL
BRANCH BACKWARD UNCONDITIONAL	BBW	4131	BBUL
BRANCH FORWARD CONDITIONAL	BFC	0231	BFCL
BRANCH FORWARD UNCONDITIONAL	BFW	4231	BFUL
BRANCH RETURN	BRT	0135	RJPL
CHANGE SIGN BIT	CHS	1031	CSSL
COMMUNICATION OPERATOR	COM	1011	COML
COMPARE FIELD EQUAL	FCE	XX75	CFEL
COMPARE FIELD LOW	FCL	XX71	CFLL
COMPARE FOR EQUAL	CEQ	XX60	SEQL
COMPARE FOR EQUAL OR LESS	CEL	XX70	SLEL
COMPARE FOR GREATER	CGR	XX63	SGTL
COMPARE FOR GREATER OR EQUAL	CEG	XX62	SGEL
COMPARE FOR LESS	CLS	XX71	SLTL
COMPARE FOR NOT EQUAL	CNE	XX61	SNEL
CONDITIONAL HALT	ZP1	2411	CHPL
CONDITIONAL INTEGER STORE DESTRUCTIVE	CID	0121	CSDL

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
CONDITIONAL INTEGER STORE NON-DESTRUCTIVE	CIN	0221	CSNL
CONSTRUCT DESCRIPTOR CALL	CDC	1241	MDAL
CONSTRUCT OPERAND CALL	COC	0241	MDVL
DELETE TOP OF STACK	DEL	0051	DELL
DESCRIPTOR CALL SYLLABLE	DESC	XXXX	DCSL
DIAL A	DIA	XX55	DIAL
DIAL B	DIB	XX61	DIBL
DOUBLE PRECISION ADD	DLA	0105	AD2L
DOUBLE PRECISION DIVIDE	DLD	1005	DV2L
DOUBLE PRECISION MULTIPLY	DLM	0405	MU2L
DOUBLE PRECISION SUBTRACT	DLS	0305	SU2L
DUPLICATE	DUP	2025	DUPL
ENTER CHARACTER MODE IN LINE	CMN	4441	ECML
EXCHANGE	XCH	1025	EXCL
EXIT	XIT	0435	REWL
FLAG BIT SEARCH	FBS	7031	SSFL
HALT P2	HP2	2211	HP2L
INDEX	INX	0141	INDL
INITIATE P1	IP1	4111	INIL
INITIATE P2	IP2	4211	PTOL
INITIATE I/O	IIO	4411	IOOL
INTEGER DIVIDE	IDV	3001	DV3L
INTEGER STORE DESTRUCTIVE	ISD	4121	ISDL
INTEGER STORE NON-DESTRUCTIVE	ISN	4221	ISNL

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
INTERROGATE INTERRUPT	ITI	0211	IINL
INTERROGATE I/O CHANNELS	TIO	6431	TIOL
INTERROGATE PERIPHERAL STATUS	TUS	2431	IPSL
I/O RELEASE	IOR	2111	IORL
LINK LIST LOOK-UP	LLL	2541	LLLL
LITERAL SYLLABLE	LITC	XXXX	LTSL
LOAD OPERATOR	LOD	2021	LODL
LOGICAL AND	LND	0415	LOAL
LOGICAL EQUIVALENCE	LQV	1015	LOEL
LOGICAL NEGATE	LNG	0115	LUNL
LOGICAL OR	LOR	0215	LOOL
NON-ZERO FIELD BRANCH BACKWARD DESTRUCTIVE	CBD	XX51	ZBDL
NON-ZERO FIELD BRANCH BACKWARD NON-DESTRUCTIVE	CBN	XX51	ZBNL
NON-ZERO FIELD BRANCH FORWARD DESTRUCTIVE	CFD	XX51	ZFDL
NON-ZERO FIELD BRANCH FORWARD NON-DESTRUCTIVE	CFN	XX51	ZFNL
OPERAND CALL SYLLABLE	OPDC	XXXX	OCSL
READ TIMER	RTR	0411	RDTL
REMAINDER DIVIDE	RDV	7001	DV4L
RESET FLAG BIT	MOP	2015	FBL
RESET SIGN BIT	SSP	4431	MSPL
RETURN NORMAL	RTN	0235	RNML
RETURN SPECIAL	RTS	1235	RSPL

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
SET FLAG BIT	MDS	4015	SFBL
STORE FOR INTERRUPT		3011	SFIL
STORE FOR TEST	SFT	3411	STFL
SET OR STORE S OR F REGISTERS	SSF	2141	FXSL
SET SIGN BIT	SSN	0431	MSNL
SET VARIANT	XRT	0061	
SINGLE PRECISION ADD	ADD	0101	AD1L
SINGLE PRECISION DIVIDE	DIV	1001	DV1L
SINGLE PRECISION MULTIPLY	MUL	0401	MU1L
SINGLE PRECISION SUBTRACT	SUB	0301	SU1L
TEST FLAG BIT	TOP	2031	TFBL
TEST INITIATE	IFT	5111	IFTL
TRANSFER BITS	TRB	XX65	TRFL
TRANSFER CORE FIELD TO CORE FIELD	CTC	5425	CCXL
TRANSFER CORE FIELD TO F FIELD	CTF	7425	CFXL
TRANSFER F FIELD TO CORE FIELD	FTC	1425	FCXL
TRANSFER F FIELD TO F FIELD	FTF	3425	FFXL
WORD BRANCH BACKWARD CONDITIONAL	LBC	2131	JBCL
WORD BRANCH BACKWARD UNCONDITIONAL	LBU	6131	JBUL
WORD BRANCH FORWARD CONDITIONAL	LFC	2231	JFCL
WORD BRANCH FORWARD UNCONDITIONAL	LFU	6231	JFUL
WORD MODE NO-OP	NOP	0055	WMNP

CHARACTER MODE

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
BEGIN LOOP	BNS	XX52	BELL
CALL REPEAT FIELD	CRF	XX43	CLRL
END LOOP	ENS	XX51	ENLL
EXIT CHARACTER MODE	EXC	XX00	RECL
EXIT CHARACTER MODE IN LINE	CMX	0100	ILEL
FIELD ADD	FAD	XX73	FADL
FIELD SUBTRACT	FSU	XX72	FSUL
INCREASE TALLY	INC	XX40	INTL
INPUT CONVERT	ICV	XX67	ICOL
JUMP FORWARD CONDITIONAL	JFC	XX45	CFJL
JUMP FORWARD UNCONDITIONAL	JFW	XX47	FWJL
JUMP OUT OF LOOP CONDITIONAL	JNC	XX44	CJOL
JUMP OUT OF LOOP UNCONDITIONAL	JNS	XX46	JOLL
JUMP REVERSE CONDITIONAL	JRC	XX55	CRJL
JUMP REVERSE UNCONDITIONAL	JRV	XX57	REJL
OUTPUT CONVERT	OCV	XX66	OCOL
RECALL CONTROL ADDRESS	RCA	XX50	RPAL
RECALL DESTINATION ADDRESS	RDA	XX04	RDAL
RECALL SOURCE ADDRESS	RSA	XX53	RSAL
RESET BIT	BIR	XX65	REBL
SET BIT	BIS	XX64	SEBL
SET DESTINATION ADDRESS	SED	XX06	SDPL

Name	OSIL/ESPOL Mnemonic	Octal Form	Engineering Mnemonic
SET SOURCE ADDRESS	SES	XX22	SSPL
SET TALLY	SEC	XX42	SETL
SKIP BIT DESTINATION	BSD	XX02	SBDL
SKIP BIT SOURCE	BSS	XX03	SBSL
SKIP FORWARD DESTINATION	SFD	XX16	FSDL
SKIP FORWARD SOURCE	SFS	XX31	FSSL
SKIP REVERSE DESTINATION	SRD	XX17	RSDL
SKIP REVERSE SOURCE	SRS	XX30	RSSL
STORE CONTROL ADDRESS	SCA	XX54	STPL
STORE DESTINATION ADDRESS	SDA	XX14	STDL
STORE SOURCE ADDRESS	SSA	XX15	STSL
STORE TALLY	STC	XX41	STAL
TEST BIT	BIT	XX37	TEBL
TEST FOR ALPHANUMERIC	TAN	XX36	TANL
TEST FOR EQUAL	TEQ	XX24	TEQL
TEST FOR EQUAL OR LESS	TEL	XX34	TLEL
TEST FOR GREATER	TGR	XX27	TGTL
TEST FOR GREATER OR EQUAL	TEG	XX26	TGEL
TEST FOR LESS	TLS	XX35	TLTL
TEST FOR NOT EQUAL	TNE	XX25	TNEL
TRANSFER BLANKS FOR NON-NUMERICS	TBN	XX12	TBZL
TRANSFER DESTINATION ADDRESS	TDA	XX07	SDAL
TRANSFER NUMERIC	TRN	XX75	TNDL
TRANSFER PROGRAM CHARACTERS	TRP	XX74	TPDL
TRANSFER SOURCE ADDRESS	TSA	XX56	SSAL
TRANSFER SOURCE CHARACTERS	TRS	XX77	TSDL
TRANSFER WORDS	TRW	XX05	TWDL
TRANSFER ZONES	TRZ	XX76	TZDL

APPENDIX D

OPERATORS, NUMERICAL LIST

WORD MODE

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
0051	DEL		DELETE TOP OF STACK	DELL
0055	NOP		WORD MODE NO-OP	NOPL
0061	XRT		SET VARIANT	VARL
0101	ADD		SINGLE PRECISION ADD	AD1L
0105	DLA	ADL	DOUBLE PRECISION ADD	AD2L
0111	PRL		PROGRAM RELEASE	PREL
0115	LNG		LOGICAL NEGATE	LUNL
0121	CID		CONDITIONAL INTEGER STORE DESTRUCTIVE	CSDL
0125	GEQ		B GREATER THAN OR EQUAL TO A	BGEL
0131	BBC		BRANCH BACKWARD CONDITIONAL	BBCL
0135	BRT		BRANCH RETURN	RJPL
0141	INX		INDEX	INDL
0211	ITI	INI	INTERROGATE INTERRUPT	IINL
0215	LOR		LOGICAL OR	LOOL
0221	CIN		CONDITIONAL INTEGER STORE NON-DESTRUCTIVE	CSNL
0225	GTR		B GREATER THAN A	BGAL
0231	BFC		BRANCH FORWARD CONDITIONAL	BFCL
0235	RTN	RNO	RETURN NORMAL	RNML
0241	COC		CONSTRUCT OPERAND CALL	MDVL
0301	SUB		SINGLE PRECISION SUBTRACT	SU1L
0305	DLS	SDL	DOUBLE PRECISION SUBTRACT	SU2L
0401	MUL		SINGLE PRECISION MULTIPLY	MU1L
0405	DLM	MDL	DOUBLE PRECISION MULTIPLY	MU2L
0411	RTR	RTM	READ TIMER	RDTL
0415	LND		LOGICAL AND	LOAL
0421	STD		"B" STORE DESTRUCTIVE	BSDL

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
0425	NEQ		B NOT EQUAL TO A	BNEL
0431	SSN	SSB	SET SIGN BIT	MSNL
0435	XIT		EXIT	REWL
0441	MKS		MARK STACK	MSOL
1001	DIV		SINGLE PRECISION DIVIDE	DV1L
1005	DLD	DDL	DOUBLE PRECISION DIVIDE	DV2L
1011	COM		COMMUNICATION OPERATOR	COML
1015	LQV		LOGICAL EQUIVALENCE	LOEL
1021	SND		"B" STORE NON-DESTRUCTIVE	BSNL
1025	XCH		EXCHANGE	EXCL
1031	CHS	CSB	CHANGE SIGN BIT	CSSL
1235	RTS	RSP	RETURN SPECIAL	RSPL
1241	CDC		CONSTRUCT DESCRIPTOR CALL	MDAL
1425	FTC		TRANSFER F FIELD TO CORE FIELD	FCXL
2015	MOP	RFB	RESET FLAG BIT	FBL
2021	LOD		LOAD OPERATOR	LODL
2025	DUP		DUPLICATE	DUPL
2031	TOP	TFB	TEST FLAG BIT	TFBL
2111	IOR		I/O RELEASE	IORL
2131	LBC		WORD BRANCH BACKWARD CONDITIONAL	JBCL
2141	SSF		SET OR STORE S OR F REGISTERS	FXSL
2211	HP2	HPB	HALT P2	HP2L
2231	LFC		WORD BRANCH FORWARD CONDITIONAL	JFCL
2411	ZP1		CONDITIONAL HALT	CHPL
2431	TUS		INTERROGATE PERIPHERAL STATUS	IPSL
2541	LLL		LINK LIST LOOK-UP	LLLL

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
3001	IDV		INTEGER DIVIDE	DV3L
3011			STORE FOR INTERRUPT	SFIL
3411	SFT		STORE FOR TEST	STFL
3425	FTF		TRANSFER F FIELD TO F FIELD	FFXL
4015	MDS	SFB	SET FLAG BIT	SFBL
4111	IPI	INI	INITIATE P1	INIL
4121	ISD		INTEGER STORE DESTRUCTIVE	ISDL
4125	LEQ		B LESS THAN OR EQUAL TO A	BLEL
4131	BBW	BBU	BRANCH BACKWARD UN-CONDITIONAL	BBUL
4211	IP2	INB	INITIATE P2	PTOL
4221	ISN		INTEGER STORE NON-DESTRUCTIVE	ISNL
4225	LSS		B LESS THAN A	BLAL
4231	BFW	BFU	BRANCH FORWARD UNCONDITIONAL	BFUL
4411	IIO		INITIATE I/O	IOOL
4425	EQL		B EQUAL TO A	BEQL
4431	SSP	RSB	RESET SIGN BIT	MSPL
4441	CMN		ENTER CHARACTER MODE IN LINE	ECML
5111	IFT		TEST INITIATE	IFTL
5425	CTC		TRANSFER CORE FIELD TO CORE FIELD	CCXL
6131	LBU		WORD BRANCH BACKWARD UNCONDITIONAL	JBUL
6231	LFU		WORD BRANCH FORWARD UNCONDITIONAL	JFUL
6431	TIO		INTERROGATE I/O CHANNELS	TIOL
7001	RDV		REMAINDER DIVIDE	DV4L
7031	FBS		FLAG BIT SEARCH	SSFL

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
7425	CTF		TRANSFER CORE FIELD TO F FIELD	CFXL
XX45	ISO		VARIABLE FIELD ISOLATE	VFIL
XX51	CBD		NON-ZERO FIELD BRANCH BACKWARD, DESTRUCTIVE	ZBDL
XX51	CBN		NON-ZERO FIELD BRANCH BACKWARD, NON-DESTRUCTIVE	ZBNL
XX51	CFD		NON-ZERO FIELD BRANCH FORWARD, DESTRUCTIVE	ZFDL
XX51	CFN		NON-ZERO FIELD BRANCH FORWARD, NON-DESTRUCTIVE	ZFNL
XX55	DIA		DIAL A	DIAL
XX60	CEQ		COMPARE FOR EQUAL	SEQL
XX61	CNE		COMPARE FOR NOT EQUAL	SNEL
XX61	DIB		DIAL B	DIBL
XX62	CEG	CGE	COMPARE FOR GREATER OR EQUAL	SGEL
XX63	CGR	CMG	COMPARE FOR GREATER	SGTL
XX65	TRB	TFR	TRANSFER BITS	TRFL
XX70	CEL		COMPARE FOR EQUAL OR LESS	SLEL
XX71	CLS		COMPARE FOR LESS	SLTL
XX71	FCL	CFL	COMPARE FIELD LOW	CFLL
XX75	FCE	CFE	COMPARE FIELD EQUAL	CFEL
XXXX	DESC		DESCRIPTOR CALL SYLLABLE	DCSL
XXXX	LITC		LITERAL SYLLABLE	LTSL
XXXX	OPDC		OPERAND CALL SYLLABLE	OCSL

CHARACTER MODE

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
XX00	EXC	ECM	EXIT CHARACTER MODE	RECL
0100	CMX		EXIT CHARACTER MODE IN LINE	ILEL
XX02	BSD	SBD	SKIP BIT DESTINATION	SBDL
XX03	BSS	SBS	SKIP BIT SOURCE	SBSL
XX04	RDA		RECALL DESTINATION ADDRESS	RDAL
XX05	TRW	TWD	TRANSFER WORDS	TWDL
XX06	SED		SET DESTINATION ADDRESS	SDPL
XX07	TDA		TRANSFER DESTINATION ADDRESS	SDAL
XX12	TBN		TRANSFER BLANKS FOR NON NUMERICS	TBZL
XX14	SDA		STORE DESTINATION ADDRESS	STDL
XX15	SSA		STORE SOURCE ADDRESS	STSL
XX16	SFD		SKIP FORWARD DESTINATION	FSDL
XX17	SRD		SKIP REVERSE DESTINATION	RSDL
XX22	SES		SET SOURCE ADDRESS	SSPL
XX24	TEQ		TEST FOR EQUAL	TEQL
XX25	TNE		TEST FOR NOT EQUAL	TNEL
XX26	TEG	TGE	TEST FOR GREATER OR EQUAL	TGEL
XX27	TGR		TEST FOR GREATER	TGTL
XX30	SRS		SKIP REVERSE SOURCE	RSSL
XX31	SFS		SKIP FORWARD SOURCE	FSSL
XX34	TEL		TEST FOR EQUAL OR LESS	TLEL
XX35	TLS		TEST FOR LESS	TLTL
XX36	TAN	TFA	TEST FOR ALPHANUMERIC	TANL
XX37	BIT	TBT	TEST BIT	TEBL
XX40	INC	INT	INCREASE TALLY	INTL
XX41	STC	STT	STORE TALLY	STAL
XX42	SEC	SET	SET TALLY	SETL

Octal Form	OSIL/ESPOL Mnemonic	Alternate OSIL	Name	Engineering Mnemonic
XX43	CRF		CALL REPEAT FIELD	CLRL
XX44	JNC	JLC	JUMP-OUT-OF-LOOP CONDITIONAL	CJOL
XX45	JFC		JUMP FORWARD CONDITIONAL	CFJL
XX46	JNS	JLP	JUMP-OUT-OF-LOOP UNCONDITIONAL	JOLL
XX47	JFW	JFU	JUMP FORWARD UNCONDITIONAL	FWJL
XX50	RCA		RECALL CONTROL ADDRESS	RPAL
XX51	ENS	ELP	END LOOP	ENLL
XX52	BNS	BLP	BEGIN LOOP	BELL
XX53	RSA		RECALL SOURCE ADDRESS	RSAL
XX54	SCA		STORE CONTROL ADDRESS	STPL
XX55	JRC		JUMP REVERSE CONDITIONAL	CRJL
XX56	TSA		TRANSFER SOURCE ADDRESS	SSAL
XX57	JRV	JRU	JUMP REVERSE UNCONDITIONAL	REJL
XX64	BIS	SEB	SET BIT	SEBL
XX65	BIR	REB	RESET BIT	REBL
XX66	OCV		OUTPUT CONVERT	OCOL
XX67	ICV		INPUT CONVERT	ICOL
XX72	FSU		FIELD SUBTRACT	FSUL
XX73	FAD		FIELD ADD	FADL
XX74	TRP		TRANSFER PROGRAM CHARACTERS	TPDL
XX75	TRN	TNU	TRANSFER NUMERIC	TNDL
XX76	TRZ	TZN	TRANSFER ZONES	TZDL
XX77	TRS		TRANSFER SOURCE CHARACTERS	TSDL

APPENDIX E

GLOSSARY

ACCESS CYCLE Period during which a storage device is being referenced.

ACCESS, RANDOM Access to storage under conditions in which the next position from which information is to be obtained is in no way dependent on the previous position or access.

ACCESS TIME The time interval between the instant at which information is called for from storage and the instant at which delivery is completed, i.e., the read time; or, the time interval between the instant at which information is ready for storage and the instant at which storage is completed, i.e., the write time.

ADDER A device capable of forming the sum of two or more quantities.

ADDRESS A label, name, or number which designates a register, location, or device where information is stored.

ADDRESS, ABSOLUTE The label assigned to a specific storage location by the designers of a machine.

ADDRESS, BASE The label identifying the first word in a routine. The base address is added to the relative address (or index) to obtain the absolute address.

ADDRESS, RELATIVE A label to identify a word in a routine with respect to its position in that routine.

ALGOL (Abbr. ALGOrithmic Language) An international problem language designed for the concise, efficient expression of arithmetic and logical processes, and the control (iterative, etc.) of these processes.

ALGORITHM A statement of the steps to be followed in the solution of a problem.

ALLOCATE To assign storage locations to the main routines and subroutines, thereby fixing the absolute values of any symbolic addresses.

ALPHANUMERIC Contraction of alphabetic-numerical; a system including letter, digits, and special symbols.

ARGUMENT Known reference factor necessary to find the desired item in a table or array. In subroutines it refers to the necessary information (obtained from outside of the subroutine) which is an integral part of the execution of the subroutine.

ARRAY An ordered arrangement of items of information; a table.

BAND A group of recording tracks on a magnetic drum.

BASE A number base; a quantity used implicitly to define some system of representing numbers by positional notation. Two is the base of the binary system; eight, the base of the octal system; and ten, the base of the decimal system.

BINARY Involving the integer two. For example, the binary number system uses the base two and contains only two symbols, zero and one.

BIT Contraction of binary digit. Usually represents the status of one flip flop, either off or on (0 or 1).

BOOLEAN Refers to a system dealing with truth values, operating upon logical conditions rather than numbers.

BUFFER Any device which stores information temporarily during data transfers. A facility linked to: (1) an input device in which information is assembled from external storage and stored ready for transfer to internal storage; or (2) an output device into which information is transmitted from internal storage and held for transfer to external storage. Since computation continues while transfers between buffer storage and external devices take place, buffers are used to compensate for differences in the speed of the various components of the system so that the system can operate as an integrated unit.

BURROUGHS COMMON LANGUAGE (BCL) A binary code representation of alphanumeric characters common to Burroughs equipment.

CALL A set of characters or bits which demand an action to occur or some item of information to be obtained; for example, subroutine call, operand call, descriptor call.

CELL Storage for one unit of information, usually one character or one word usually a location specified by whole or part of the address and possessed of the faculty of storage. Specific terms as column, field, location are preferable when appropriate.

CENTRAL CONTROL Maintains information flow, control logic, timing and provides for system coordination.

CHANNEL A path along which information may flow. See Input/Output channel.

CHARACTER One of a set of elementary symbols which may be arranged in ordered groups to express information; these symbols may include the decimal digits 0 through 9, the letters A through Z, punctuation symbols, special input and output symbols, and any other symbols which a computer may accept.

CLOCK A time-increment counting register used for program interrupt and job-time accounting.

COBOL (Abbr. Common Business Oriented Language) A compiler designed for expressing problems of data manipulation and processing in English narrative form. Intended to be used for business applications.

COMPILE Reduce and/or assemble from a source language the necessary subroutines into a main routine (or program) into a machine language.

COMPILER A program making routine, which produces a specific program for a particular problem by determining the intended meaning of an element of information expressed in pseudo-code; selecting or generating the required subroutine and transforming the subroutine into specific coding for the specific problem; assigning specific storage registers, etc., and entering it as an element of the problem program; maintaining a record of the subroutines used and their position in the program, and continuing to the next element of information in pseudo-code.

CONCATENATE Link together by forming a chain or series; a series or order of things depending on each other.

CONFIGURATION Relative arrangement of various components of the system.

CONSOLE The unit of a computer which provides the communication between the computer and the operator. The console contains the controls for starting and stopping, indicators for displaying the status of the system, and a means for manual intervention or operation of the system

CONTROL COUNTER A register which indicates the location of the next syllable to be executed by the processor.

CORE, MAGNETIC A magnetic material capable of assuming and remaining at one of two conditions of magnetization, thus capable of providing storage and gating or switching functions. It is usually of toroidal shape and pulsed or polarized by electric currents carried on wire wound around or through the material.

CORE, STORAGE A form of high speed storage which utilizes magnetic cores.

CYCLE A set of operations repeated as a unit; a non-arithmetic shift in which the digits dropped off at one end of a word are returned at the other end in a circular fashion, cycle right and cycle left.

DEBUG To isolate and remove all malfunctions from a computer or to sense and correct all mistakes in a routine or program.

DESCRIPTOR A computer word used specifically to define characteristics of a program element. For example, descriptors are used for describing a data record, a segment of a program, or an input/output operation.

DESCRIPTOR CALL SYLLABLE A syllable of the B 5500 which will direct the processor to place in the stack a descriptor which points to the location of an indicated piece of information, either data or a program segment.

DIAGNOSTIC ROUTINE Routine designed to locate malfunctions or errors.

DOUBLE PRECISION A quantity having twice as many digits as are normally carried in a specific computer word. Often called double length.

DRUM, MAGNETIC A rapidly rotating cylinder whose surface is coated with a magnetic material upon which information is stored in the form of small magnetized spots.

EDIT To rearrange information. Editing may involve the deletion of unwanted data, the selection of pertinent data, tests for validity and reasonableness, the insertion of invariant symbols such as page numbers and typewriter characters, and the application of standard processes such as zero-suppression.

ESAP Engineering Symbolic Assembly Program.

ESPOL Executive Systems Programing Operating Language.

EXECUTIVE ROUTINE A routine designed to control and process other routines. See Master Control Program.

EXPONENT In word mode, a number may be divided into an exponent and a mantissa. In the B 5500, the exponent is expressed in a 6 bit field.

EXTERNAL STORAGE Storage facilities removable from the computer itself but holding information in a form acceptable to the computer (magnetic tape, punched card, etc.).

FIELD A set of one or more characters (not necessarily all lying in the same word) which is treated as a whole; a set of one or more columns on a punched card consistently used to record similar information.

FILE A collection of records; an organized collection of information directed toward some purpose. (The records in a file may or may not be sequentially filed according to a key contained in each record.)

FIXED-POINT A notation or system of arithmetic in which all numeric quantities are expressed by a predetermined number of digits with the point implicitly located at some predetermined position. Contrasted with floating-point.

FLAG One bit which is used to indicate whether the word is a descriptor or operand.

FLIP-FLOP A bi-stable device which may assume a given stable state depending upon the pulses of one or more input points and which has one or more output points. The device is capable of storing a bit of information, controlling gates, etc.

FLOATING-POINT REPRESENTATION An arithmetic notation in which all numeric quantities have an associated indication of the decimal point location (base 10), octal point location (base 8), or binary point location (base 2). Automatic alignment of numbers and calculation of the location of the point can be provided in arithmetic of floating-point numbers. In the B 5500, a floating-point number consists of two parts: a 13 digit octal integer with sign called the mantissa; and a signed 2 digit octal number called the exponent which indicates the number of places to the right or left that the actual octal point is from the assumed octal point in the mantissa.

FORMAT The predetermined arrangement of characters, fields, lines, page numbers, punctuation marks, etc. in input, output, or records.

FRACTIONAL The portion of a number which is to the right of the decimal, octal, or binary point.

GATE An electronic circuit with two or more inputs and one output, with the property that a pulse goes out on the output line if and only if some specified combination of pulses occurs on the input lines.

HARDWARE The mechanical, magnetic, electronic and electrical devices from which a computer is fabricated; the assembly of physical material forming a computer.

HOUSEKEEPING Operations in a routine which do not directly contribute to a solution of the problem at hand, but which are made necessary by the method of operation of the computer; packing or rearranging data, sub-program linkages, reset addresses, clear units, etc.

INDEX Increment or decrement to a base address.

INDICATOR A light, usually on the operator's console, that is turned on to indicate a particular condition occurring in the computer.

INPUT/OUTPUT CHANNEL A device which allows independent communication between the memory exchange and the input/output exchange. It controls any peripheral device and performs all validity checking on information transfers.

INPUT/OUTPUT EXCHANGE An electronic switch which connects an input/output channel to the designated peripheral device.

INTEGRAL The whole number portion of either a decimal or octal number. Refers to all the digits to the left of the decimal.

INTERRUPT A signal generated when certain conditions arise in the system. The interrupt system provides the Master Control Program with the facility to maintain control of all system functions.

ITERATION Repetitive execution of program steps or loops.

JUMP An operation which may alter the normal sequence of a program. Normally, syllables are executed in sequence; a jump operation causes a termination of the sequence and directs the processor to a specified syllable.

KEYBOARD The portion of the supervisory printer via which the operator can communicate with the system.

LANGUAGE, MACHINE Information recorded in a form which a computer can handle. The coded operations that control information and addresses employed within the processor to express a program.

LIBRARY Collection of fully tested standard programs for repeated use by, or incorporation into, other programs.

LITERAL An element in a program which is itself a quantity to be used by the program rather than being an address of the quantity.

LOCATION A storage position in a main internal storage; one computer word; a storage register.

LOG Summary of scheduling, timing, programs run, etc. In the B 5500, the MCP maintains an internal log.

LOOP A coding technique whereby a group of instructions is repeated with modification and/or with modification of the data being operated upon. It is a series of instructions, the last of which directs the computer to start again at the first instruction of the series.

MAGNETIC DISK A rotating disk with a magnetizable surface on which information may be stored as a pattern of polarized spots along any one of a number of concentric circular recording tracks.

MANTISSA Integral part of a floating-point number (maximum 13 octal digits in a single precision number of the B 5500). See floating-point representation.

MASTER CLOCK The device which controls the basic timing impulses of the computer.

MASTER CONTROL PROGRAM A computer program to control the operation of the system. It is designed to reduce the amount of intervention required of the human operator. The master control program provides the following functions: schedules programs to be processed; initiates segments of programs; controls all input/output operations to insure efficient utilization of each system component; allocates memory dynamically; issues instructions to the human operator and verifies that his actions were correct; performs corrective action on errors in a program or system malfunction.

MEGACYCLE/SECOND A million cycles per second. The basic pulse rate of the B 5500 is one megacycle per second.

MEMORY Internal computer storage. Distinguished from other types of storage in the B 5500 which are part of the peripheral equipment.

MEMORY EXCHANGE An electronic switching device which allows connection between any memory module and input/output channel or processor.

MICROSECOND One millionth of a second.

MILLISECOND One thousandth of a second.

MODE, CHARACTER The method of operation in which the basic unit of information is a character One computer word may contain a maximum of 8 characters composed of 6 bits each. When transferred, a character is transferred usually as 7 bits; 6 bits plus 1 bit for parity check.

MODE, WORD The method of operation in which the basic unit of information is a word composed of 48 bits plus one for parity.

MODULUS (MODULO) The number of distinct integers in a finite system of numbers. For example, in a modulo 5 system, the numbers are 0, 1, 2, 3, and 4. In this system larger numbers are expressed by dividing them by the modulus until a remainder less than the modulus is obtained. For example, 19 is 4 in the modulo 5 system. If a counter is "modulo 5"; when it is set at 5, an increment of 1 will result in a setting of 0.

MULTIPROCESSING Processing several programs or program segments concurrently on a "time-share" basis. Each processor is only active on one program at any one time while operations such as input/output may be performed in parallel on several programs. The processor is directed to switch back and forth among programs under the control of the Master Control Program.

NESTING Enclosing one program element of a particular type within another of the same type.

NORMAL STATE See state.

NORMALIZE To adjust the exponent and mantissa of a floating-point result so that the mantissa lies in the prescribed standard (normal) range; standardized.

OBJECT PROGRAM A set of machine language instructions for the solution of a problem, obtained as the end result of a compilation process. It is generated from the source program.

OCTADE A group of 3 bits used to represent one octal digit. There are 16 octades in one B 5500 word.

OCTAL A number system based on powers of 8 rather than 10 as in the decimal system. Includes only the digits 0, 1, 2, 3, 4, 5, 6, and 7.

OPERAND Any of the quantities entering into an operation. An operand is typically a number for arithmetic operations. For comparison operations, an operand may be an alphanumeric field.

OPERAND CALL SYLLABLE A syllable which specifies that an operand be brought to the stack, either directly from the program reference table or indirectly by means of a descriptor.

OPERATORS Symbols that denote a fixed, predefined set of operations to be performed in a specified sequence. In the B 5500 the operators fall into two classes; word mode operators and character mode operators.

OSIL Operating System Implementation Language.

OVERFLOW In arithmetic operations, the generation of a quantity beyond the capacity of the register or location which is to receive the result; over capacity; the information contained in an item of information which is in excess of a given amount.

OVERLAY A technique for bringing routines into high speed memory from some other form of storage during processing, so that several routines will occupy the same storage location at different times; used when the total memory requirements for a program exceed the available high speed memory.

PACK To combine several brief or minor items of information into one machine item or word by utilizing different sets of digits for the specifications of each brief or minor item.

PARALLEL OPERATIONS Flow of information through the system or any part of it, using two or more communication lines or channels simultaneously.

PARALLEL PROCESSING Processing more than one program at a time on a parallel basis, where more than one processor is active at one time (distinguished from multiprocessing where only one processor is active on one program at a time).

PARAMETER In a subprogram, a quantity which may be given different values when the subprogram is used in different parts of one main program, but which usually remains unchanged throughout any one such use. To use a subprogram successfully in many different programs requires that the subprogram be adaptable by changing its parameters.

PARITY CHECK A parity check makes use of a self-checking code of binary digits in which the total number of zeros, or ones, is always odd or always even.

PRESENCE BIT A single flag bit appearing in descriptors to indicate whether or not the information, which is referenced by the descriptor is in core memory at this time.

PRIORITY A value assigned to programs or program segments to specify the relative processing sequence. The priority of all programs to be run are taken into consideration by the Master Control Program in arriving at a schedule.

PROBLEM LANGUAGE The language used by the programmer to state the definition of a problem. ALGOL, COBOL, and FORTRAN are examples of problem languages. Problem languages are closely related to the type of problem being stated, i.e., algebraic statements from mathematical problems (ALGOL, FORTRAN) and narrative English statements for commercial problems (COBOL). Problem language should not be confused with machine language. A source program is written in a problem language by the programmer. This source program is then translated into the object program (in machine language) by a compiler program.

PROGRAM (NOUN) A plan for the solution of a problem. A B 5500 program may be a statement of the problem in ALGOL, FORTRAN or COBOL or the translated segmented object (compilation result) program.

PROGRAM (VERB) To make a program.

PROGRAM INDEPENDENT MODULARITY Property of the B 5500 to accept changes in system configuration and adjust programs in order to yield maximum utilization of all modules without reprograming or recompilation.

PROGRAM REFERENCE TABLE (PRT) An area in memory for the storage of operands, references to arrays, references to segments of a program, and references to files. Permits programs to be independent of the actual memory locations occupied by data and parts of the program.

RANDOM ACCESS See access.

RECURSIVE Having the characteristic of occurring within itself. For example: An ALGOL procedure could contain a procedure statement in its body calling for the activation of itself.

REGISTER The internal hardware used temporarily to store one or more computer words.

RELOCATABILITY A facility whereby programs or data may be located in a place in memory at different times without requiring modification to the program. In the B 5500, segments of the program and all data are independently relocatable with no loss in efficiency.

RESET To return a device to zero or its initial condition.

RESTORE To return a cycle index, a variable address, or other computer word to its initial or a preselected value.

ROUTINE A set of coded instructions arranged in proper sequence to direct the computer to perform a desired operation or series of operations.

RUN One performance of a program on a computer.

SCHEDULING Designation of time and sequence of projected operations. One of the functions of the B 5500 Master Control Program.

SEGMENT (VERB) To divide a program into sections, each of which performs some part of the total program and is capable of being completely stored in internal memory.

SERIAL Handle one after the other in a single facility or single piece of equipment.

SERIAL TRANSFER A system of data transfer in which elements of information are transferred in succession over a single line.

SET To return a device to one or to the "on" state.

SOFTWARE Programs, routines, and procedures which augment and support a computer system. (e.g., The Master Control Program, compilers, etc.)

STACK A portion of memory and two registers used for temporarily holding information. A stack, as used in the B 5500, operates on the "last-in first-out" principle. That is, the last item of information placed in the stack will be the first item of information used when information is required from the stack. Operators perform their operation on information at the top of the stack. (See operators).

STATE, CONTROL Condition of operation wherein instructions that can be performed in normal state are augmented by additional control operations. Most MCP routines are written to operate in control state.

STATE, NORMAL The condition of operation wherein the instructions are concerned with the conventional aspects of computation (adding, subtracting, information transfer, etc.). The detection of an exceptional condition (interrupt) that occurs while in this state suspends operation in this state and processing continues in control state.

SUBROUTINE A set of instructions necessary to carry out a defined operation, a sub-unit of a program.

SYLLABLE The basic unit of the B 5500 program string. Each computer word contains four syllables.

SYNTAX Connected system or order of symbol arrangement, the rules or grammar of a programing language.

SYSTEM An assembly of components united by some form of regulated interaction to form an organized whole.

TRANSFER To copy, exchange, read, record, store, transmit, transport, or write data; to change control.

TRANSLATE To change information from one form of representation to another without significantly affecting the meaning.

VERIFY To check a data transfer or transcription, especially those involving manual processes such as keypunching.

WORD A set of characters which occupy one storage location and are treated by the computer as a unit and transferred as such. In the B 5500, a word contains 8 alphanumeric characters or 48 bits, plus parity.

APPENDIX F

ABBREVIATIONS

ABBREVIATIONS	MEANING
AROF	A Register Valid Flip Flop
BROF	B Register Valid Flip Flop
CWMF	Character-Word Mode Flip Flop
DESC	Descriptor Call Syllable
IAR	Interrupt Address Register
ICW	Interrupt Control Word
ILCW	Interrupt Loop Control Word
INCW	Initiate Control Word
IRCW	Interrupt Return Control Word
HLTF	Halt Flip Flop
LCW	Loop Control Word
LITC	Literal Call Syllable
MCP	Master Control Program
MRAF	Memory Read Access Flip Flop
MROF	Memory Read Obtained Flip Flop
MSCW	Mark Stack Control Word
MSFF	Mark Stack Flip Flop
MWOF	Memory Write Obtained Flip Flop
NCSF	Normal-Control State Flip Flop
OPDC	Operand Call Syllable
OSIL	Operating System Implementation Language
PROF	P Register Shift Flip Flop
PRT	Program Reference Table
PST	Program Segment String
RCW	Return Control Word
SALF	Sub-Program Level Flip Flop
SFIL	Store For Interrupt Syllable
TFFF	True-False Flip Flop
TROF	T Register Valid Flip Flop
VARF	Variant Flip Flop

INDEX

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)

INDEX (CONT)

Burroughs B 5500 REFERENCE MANUAL

1021326 5-67 Printed in U.S. America