Burroughs
MEDIUM SYSTEMS
B 2500/B 3500/B 2700
CENTRAL PROCESSOR and MEMORY

REFERENCE MANUAL

Burroughs Corporation
Detroit, Michigan 48232

$2.00

Printed in U.S. America 12-72 1063781

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

LIST OF TABLES

INTRODUCTION

The Burroughs B 2500, B 3500, and B 2700 memories and central processors offer a totally integrated design that incorporates monolitic, solid-state circuitry.

CORE MEMORY

The core memory, being extremely modular, allows for configurations that easily provide for most user requirements. Base and limit registers are designed within the memory logic to allow the system to utilize dynamic storage allocation and, perhaps of greatest importance, is that all memory addresses are digit addressable and the memory itself is easily incrementable.

CENTRAL PROCESSOR

The central processor contains the circuitry to perform the instruction set. The instructions are capable of performing operations on digit, byte, or word formatted fields. The internal 8-bit code may be either EBCDIC or USASCII code and is programmatically selectable. In addition to executing the instruction set, the processor has an automatic interrupt system.

SECTION 1
CORE MEMORY

GENERAL

The main memory utilized is a four-wire coincident current core memory with a minimum cycle time of one microsecond. Depending on the system storage, memory is variable in size from 10,000 characters up to a maximum of 500,000 characters.

A typical memory cabinet is shown in figure 1-1. Two types of core memory cabinets are available: Central Control and Memory Base A cabinet and the Memory Base B cabinet.

Figure 1-4 is a block diagram of the entire memory system.

Individual core modules, referred to as stacks, are available in 10,000 (10K) character, 20K character, and 30K character sizes. Each of these stacks is completely independent, containing all necessary drivers, sense amplifiers, and core.

CENTRAL CONTROL AND MEMORY BASE A CABINET

The Central Control and Memory Base A cabinet is used in the Two Cabinet System. Within this cabinet there is space allocated for up to two memory stacks and since the largest stack contains 30,000 characters, the Two Cabinet System is limited in core storage to 60,000 characters. Figure 1-2 shows a layout diagram of the Central Control and Memory Base A cabinet, and figure 1-3 shows the internal hardware of the cabinet.

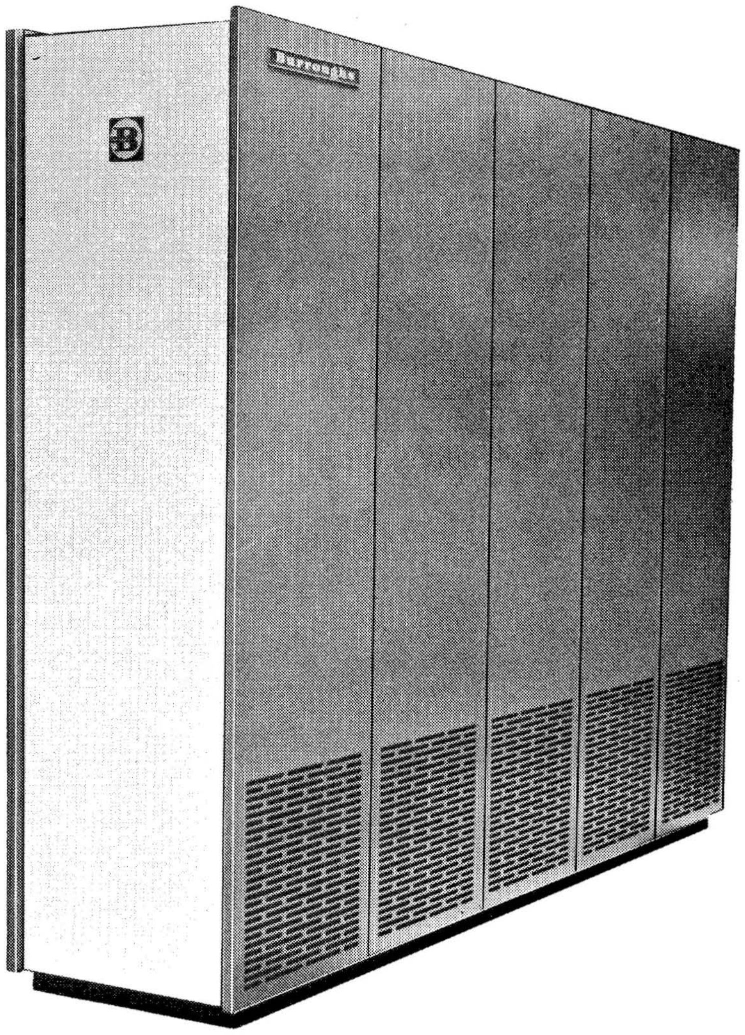

Figure 1-1. Memory Cabinet

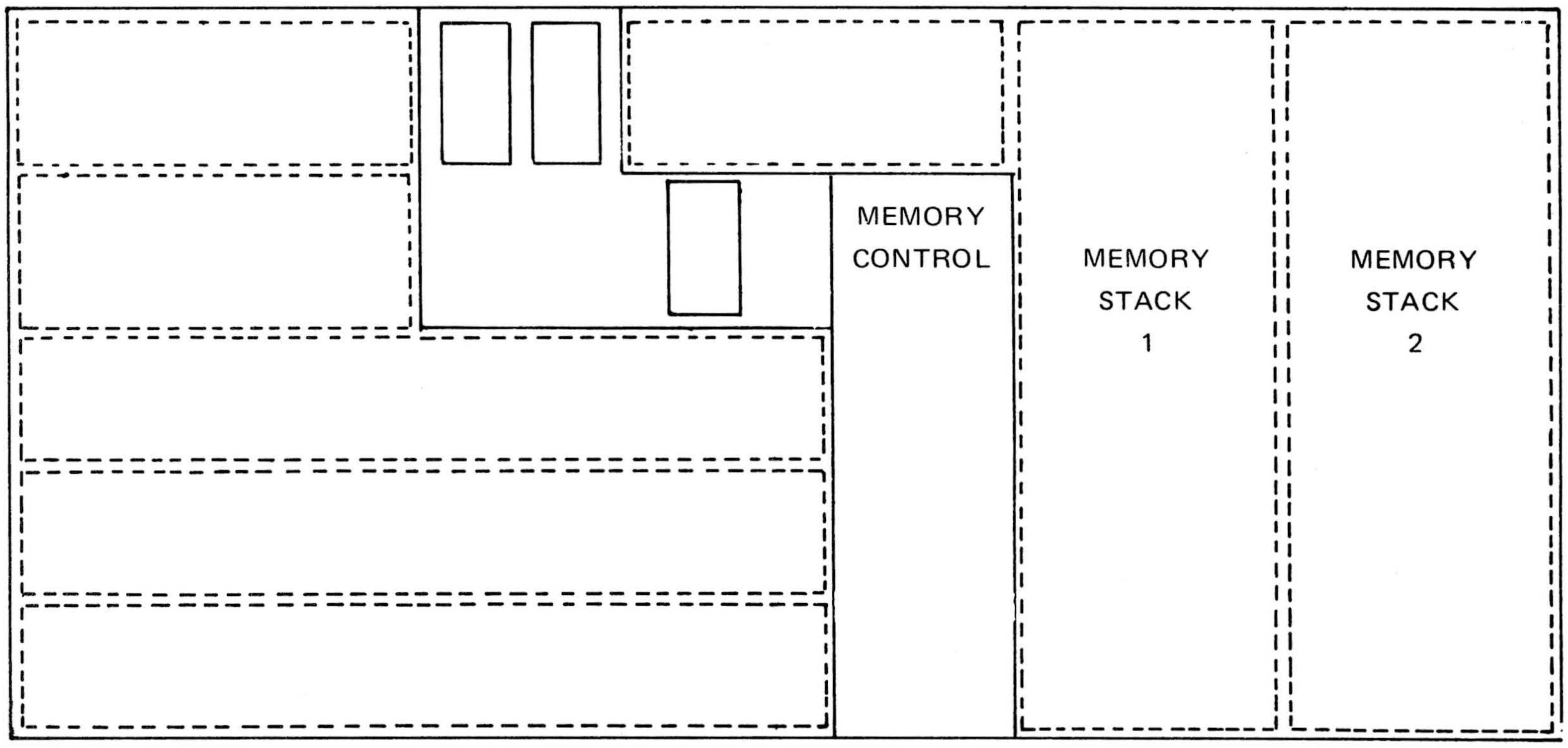

Figure 1-2. Central Control and Memory Base A Cabinet Layout

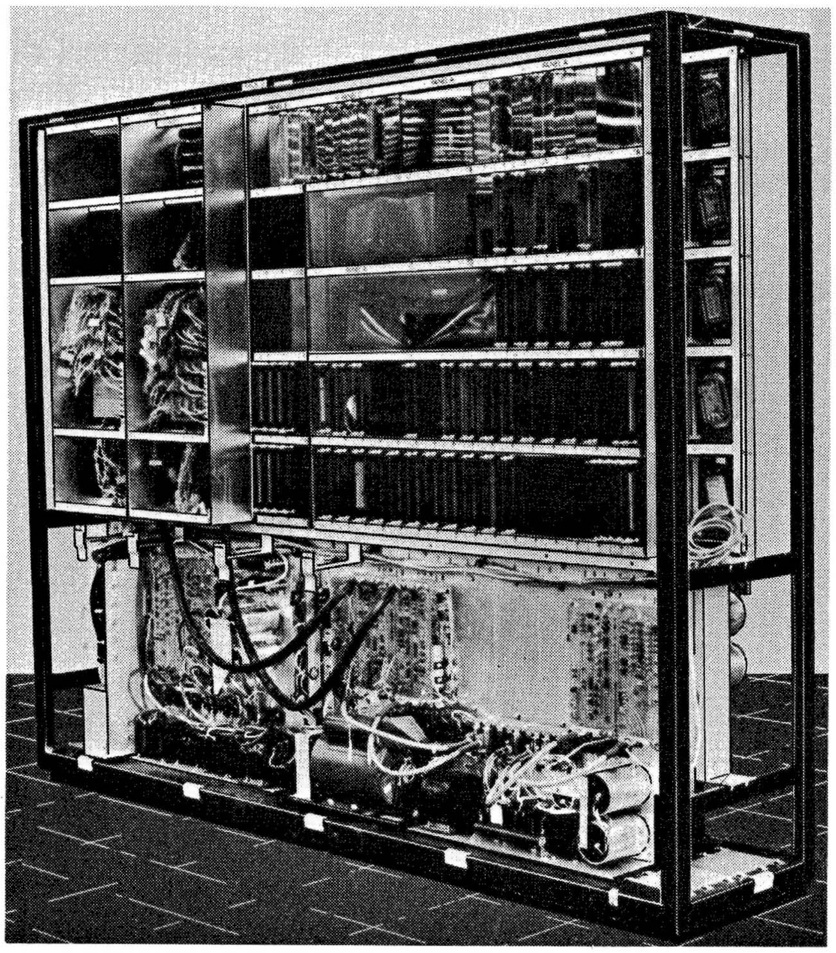

Figure 1-3. Central Control and Memory Base A Cabinet

MEMORY BASE B CABINET

The Memory Base B cabinet houses up to five memory stacks and provides these systems with measurably more storage. Figure 1-5 is a layout diagram of the Memory Base B cabinet and figure 1-6 shows the internal hardware of the cabinet.

Certain Models of the B 2500 and B 2700 are configured into three cabinet systems (CPU, Peripheral Controls, and Memory Base). These Systems operate with one Memory Base B cabinet, with a maximum core storage of 120K characters (four 30K stacks).

Additional models of the B 2700 and B 3500 Systems can each utilize up to four Memory Base B cabinets. The first three cabinets, each containing five 30K stacks, provide these systems with storage for up to 450K characters. The fourth cabinet can house only up to 50K characters, giving the system a total memory storage capacity of 500K characters. The memory stacks are placed in specific locations within the cabinet as shown in the layout diagram, figure 1-5. Each consecutive stack area must contain a full 30K characters before the next location can be utilized.

MEMORY STACKS

Figure 1-7 is a detailed block diagram of a core memory stack with associated control circuits. Each stack is self-sufficient in that it contains all necessary timing circuits, drivers, and sense amplifiers to be totally independent of all other stacks. However, if the storage size of the core memory is changed, the memory-size card, located in the central processor, must be changed.

Stack Configuration

Each core stack is 17 planes deep (figure 1-8), with each word containing one bit from each plane. This results in a 16-bit (four digit) word with one parity bit.

Each plane has its own sense and inhibits lines. They share addressing lines for associated locations on the other planes.

Array positioning provides for coordinate addressing of the desired core position. Figure 1-9 is a block diagram showing the three types of arrays: the 50 x 100 array for the 10,000 character stack, the 100 x 100 array for the 20,000 character stack, and the 100 x 150 array for the 30,000 character stack.

For addressing purposes, each stack is divided into 5,000 character sections referred to as sextants. The 30,000 character stack (figure 1-9) contains six sextants, the 20,000 character stack contains four sextants, and the 10,000 character stack contains two sextants.

MEMORY CONTROL

The interface between memory control, central processor and central control is shown in figure 1-10. Figure 1-11 is a detailed block diagram of the memory control logic and the interfacing of this logic to the memory stacks.

The memory cycle begins when either the central processor or an I/O channel request is granted access to memory. When the access is

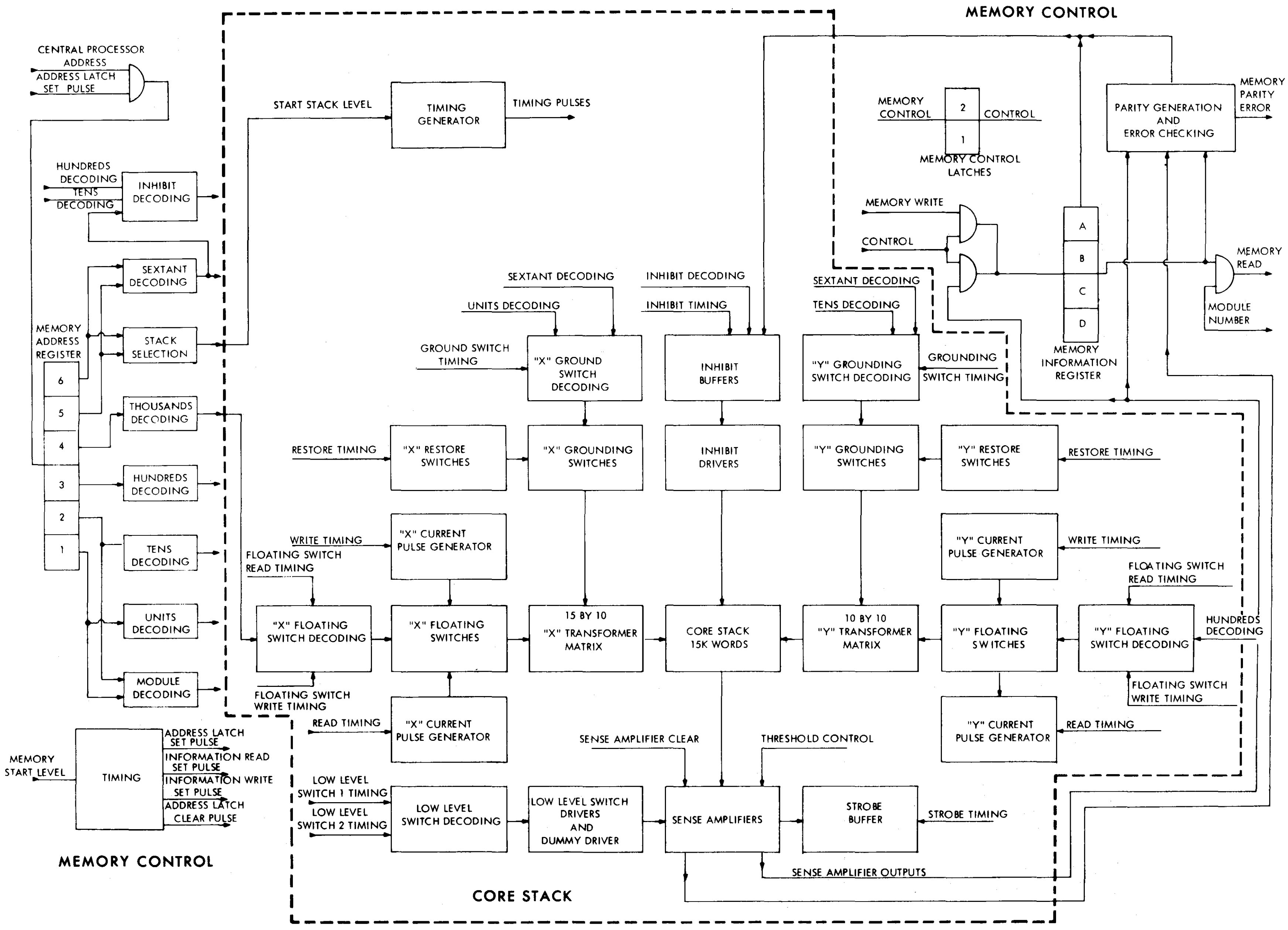

Figure 1-4. Memory Block Diagram

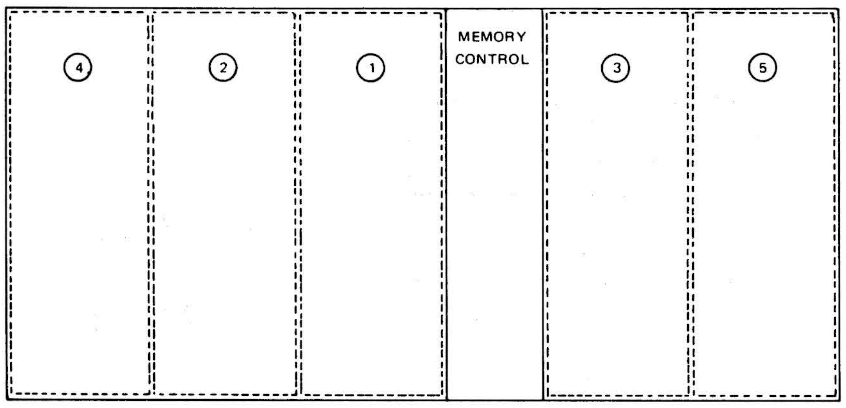

Figure 1-5. Memory Base B Cabinet Layout

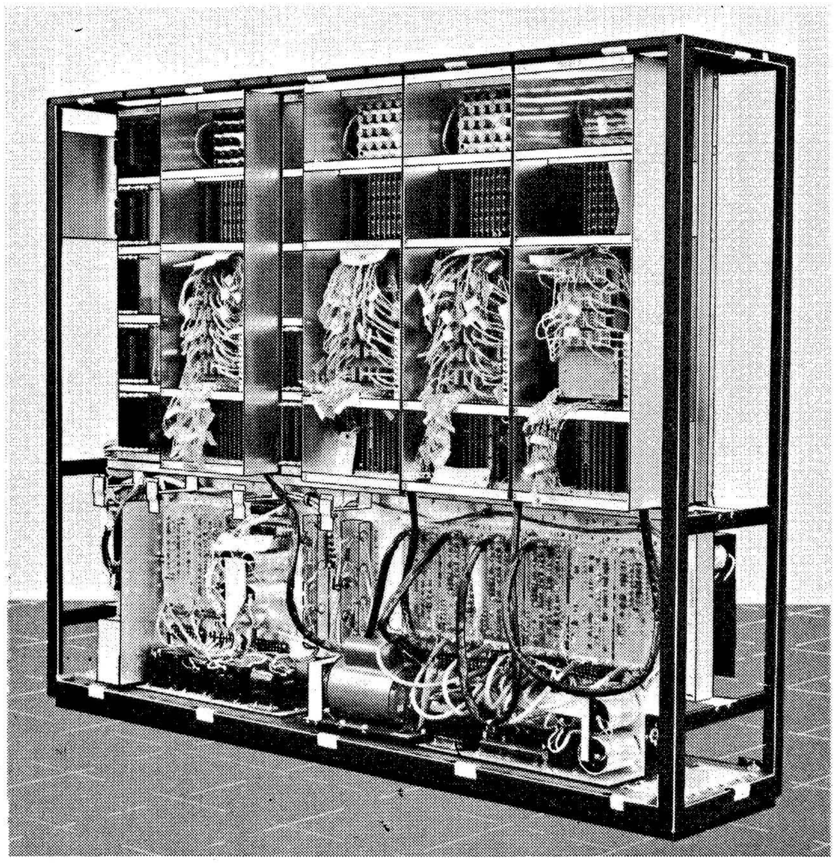

Figure 1-6. Memory Base B Cabinet

granted, a series of address decoding procedures are initiated. This addressing is performed in the following sequence:

a. Cabinet addressing.

b. Stack addressing.

c. Sextant addressing.

d. Location.

Figure 1-12 shows an exploded view of the complete memory scheme. The first breakdown is from the address in the address register (ADR), located within the central processor, into cabinet selection. Once the cabinet is located, the stack within the cabinet is determined from the address in the memory address register (MAR). Next, the proper sextant is selected and, with the last four digits in the MAR, two transformers lines are selected: one "X" and one "Y."

Cabinet Addressing

Cabinet addressing is determined by the decoding of the most significant digit (MSD) of the address within the address register of the processor (ADR) when a memory access has been granted. As the access is granted to either an I/O channel or the central processor, a term called memory start (MST) is gated with the MSD in the ADR, which then sends the start level to the proper cabinet.

Stack Addressing

Stack addressing is accomplished by gating digits 5 and 6 of the memory address register with the start stack timing level (SSTL). Since a stack can contain up to 60,000 digits, stack number one is selected when the MAR is from 000,000 up to 059,999; stack number two is selected when the MAR is from 060,000 up to 119,999, etc. The MAR values and address of stacks are as follows:

Value of MAR	Addressed Stack
000,000 - 059,999	1
060,000 - 119,999	2
120,000 - 179,999	3
180,000 - 239,999	4
240,000 - 299,999	5

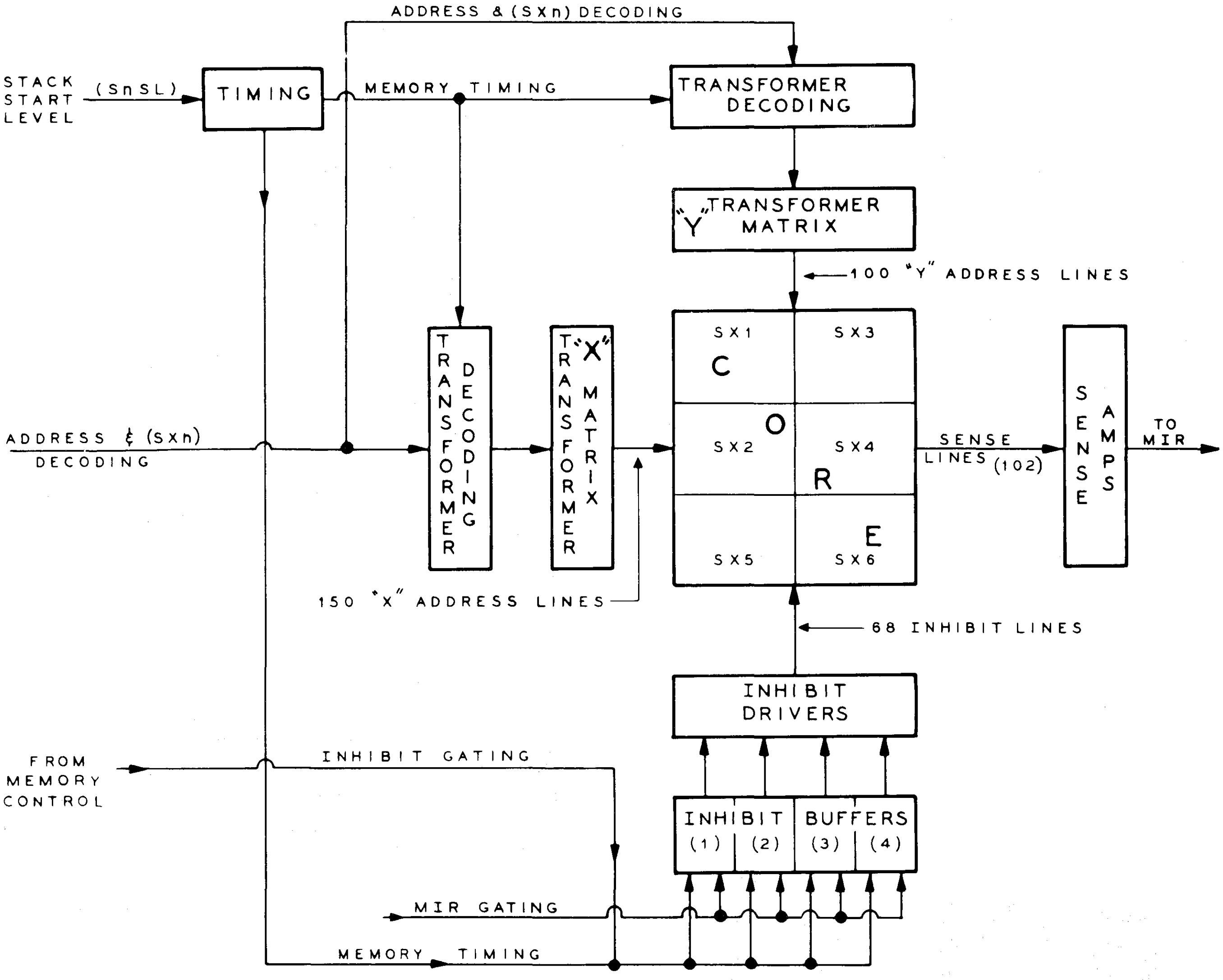

Figure 1-7. Stack Block Diagram

Sextant Addressing

As previously mentioned, a sextant is a 5,000 character section of a stack with six sextants in a 30,000 character stack, four sextants in a 20,000 character stack, and two sextants in a 10,000 character stack. The sextants are designated 1 through 6, depending on the stack size, as shown in figure 1-9. Since each of the five stacks have sextants 1 through 6, there are five groups of stack addresses which could select each sextant term. The proper sextant is determined by digits 5 and 6 of the memory address register.

Sextant one is selected by the term sextant 1 (SX1). This term is used to select the group of transformers that drives current through the selected sextant. Sextants 2 through 6 are selected in a similar manner by terms SX2 through SX6.

Location

As shown in figure 1-9, a full stack contains six sextants, with each sextant occupying a specific location within the stack. The stack is addressed by 150 "X" address lines and 100 "Y" address lines with each sextant having 50 intersecting "X" and "Y" lines.

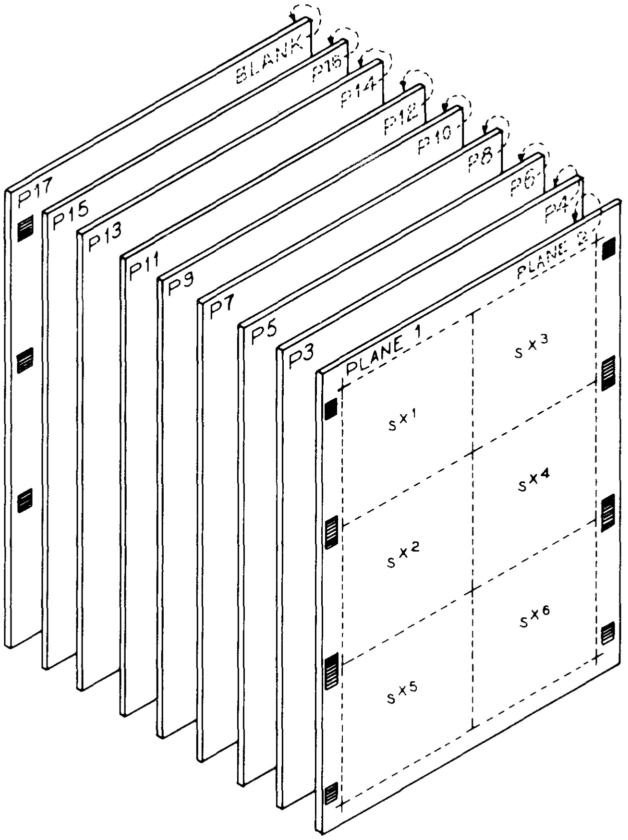

Figure 1-8. Memory Planes

Each of these addressing lines is connected to the secondary of a transformer with a single transformer for each line. The selection of one "X" transformer and one "Y" transformer completes the coordinate addressing of a specific location.

The selection of the proper transformer is accomplished with the last four digits of the MAR and the sextant decoding described earlier. The sextant information divides the transformers into groups of 50. However, the "X" address line that intersects sextant 1 (SX1) also passes through SX3, as shown in figure 1-9. Similarly, the "X" address lines that intersect sextants 2 and 3 also pass through sextants 4 and 6, respectively. The "Y" address that intersects sextants 1 and 3 also passes through sextants 2 and 5, and 4 and 6, respectively.

Figure 1-13 shows that if the address in the MAR selects SX1, only the group of transformers whose output lines pass through SX1 and SX3 for the "X" transformer matrix and through SX1, SX2 and SX5 for the "Y" transformer matrix are enabled. However, only SX1 experiences coordinate intersection, which means that only sextant 1 is provided with both "X" and "Y" half-current.

The summation of these two half-currents provides full current (2 times one-half), which is required to fully change the state of the core. The other sextants included with the transformers that intersect SX1 only receive half of the required current, either from the "X" or "Y" transformer matrix.

The transformers are selected by the intersection of two address decoding lines. The "X" address lines are decoded from sextant MAR thousands and MAR units gating, whereas the "Y" address lines are decoded from the sextant, MAR hundreds, and MAR tens. Refer to figure 1-14. One group of address decoding lines is composed of the thousands position, digit 4 of MAR. Since digit 4 can contain ten different configurations (0 through 9), it is decoded into ten different address decoding lines directly from the binary count in the MAR digit 4 position. These lines are designated K0, K1, K2 through K9.

The other group of 15 lines is composed of the unit sextant gating. The "X" transformers must address through different pairs of sextants.

These five unit terms U0 through U5, combined with the three sextant terms, gates the group of 15 lines to the transformer matrix. A complete block diagram of these lines and associated gating is shown in figure 1-15. At the point of intersection of the decoding lines, a transformer is selected. Only one horizontal line and one vertical line can be enabled during a memory access; therefore, only one transformer is enabled. This drives current through one "X" address line of the core stack.

The selection of the "Y" transformer is accomplished in the same identical manner as the selection of the "X" transformer. The "Y" transformers are arranged in a 10 x 10 array, as shown in figure 1-16. This array is addressed by two groups of ten lines. One group of ten lines is decoded from the hundreds position (digit 3 of the MAR), while the other group of ten lines is decoded from the even tens position (digit 2 of the MAR) and the sextant gating.

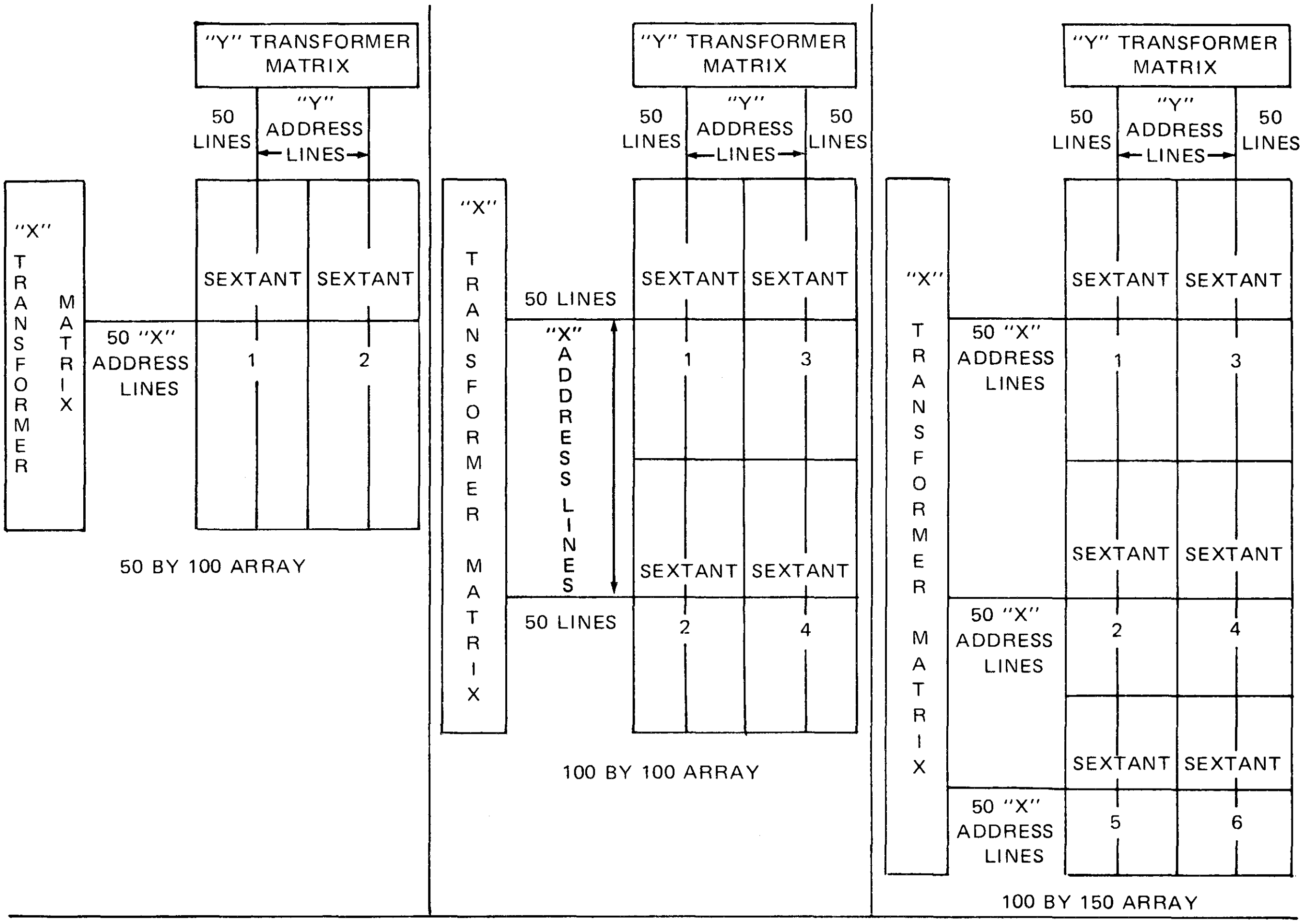

Figure 1-9. Address Lines Through Sextants

The tens and units positions of MAR can describe 100 different digit addresses. There are five word addresses in each 20 consecutive digit addresses starting with 000,000.

The first word of each 20 addresses contains an even digit (MAR tens equal to 0, 2, 4, 6 and 8). The odd MAR digits have a value of 0 through 3 for any digit address in the first word. The address of any digit in the first word is designated by the term U0.

The second word of each 20 addresses contains an even (MAR 2, and MAR 1) digit value of 4 through 7. The term for this decoding is U1.

The three remaining unit terms (2 through 5) are decoded in the same manner, generating the terms U2 through U4.

The hundreds lines are decoded from the binary count in the hundreds position of the MAR. This count can be from 0 through 9, with each count being designated H0 through H9.

The output of the "Y" transformers passes through two groups of three sextants, as shown in figure 1-9. These groups form the terms SX1, SX2, or SX5 and SX3, SX4, or SX6, with each gate having five even tens terms to achieve the ten gating lines to the matrix. The even tens terms are decoded directly from the ten positions of MAR minus the one bit used in the units gating. The terms T0, T2, T4, T6 and T8 are derived from this count. Refer to figure 1-9. Addressing is now complete. (The transformers drive the address lines which select one specific location which contains one word or four digits, plus a parity bit. Examples of address breakdown are given in table 1-1.

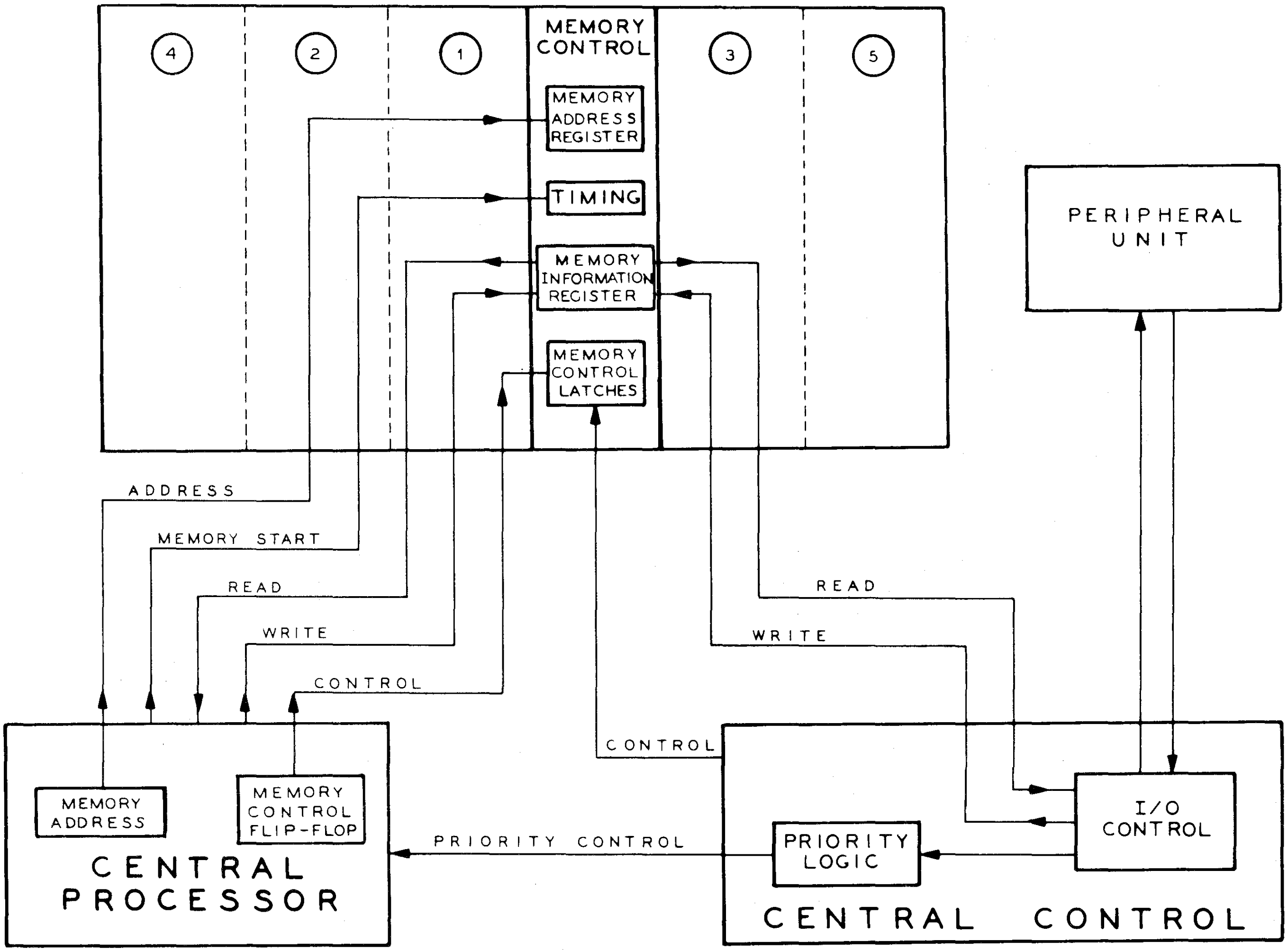

Figure 1-10. Memory Control Interface

Table 1-1. Examples of Address Breakdown

Address in ADR	Cabinet Location	Address in MAR	Stack Location	Sextant Location	MAR Digit 4 (Thousands Term)	MAR Digit 3 (Hundreds Term)	MAR Digit 2 (Tens Terms)	Unit Term
183620	1	183620	4	1	K3	H6	T2	U0
382195	2	082195	2	3	K2	H1	T8	U3
491835	2	191835	4	2	K1	H8	T2	U3
604171	3	004171	1	1	K4	H1	T6	U2
895952	3	295952	5	6	K5	H9	T4	U3
962111	4	062111	2	1	K2	H1	T0	U2

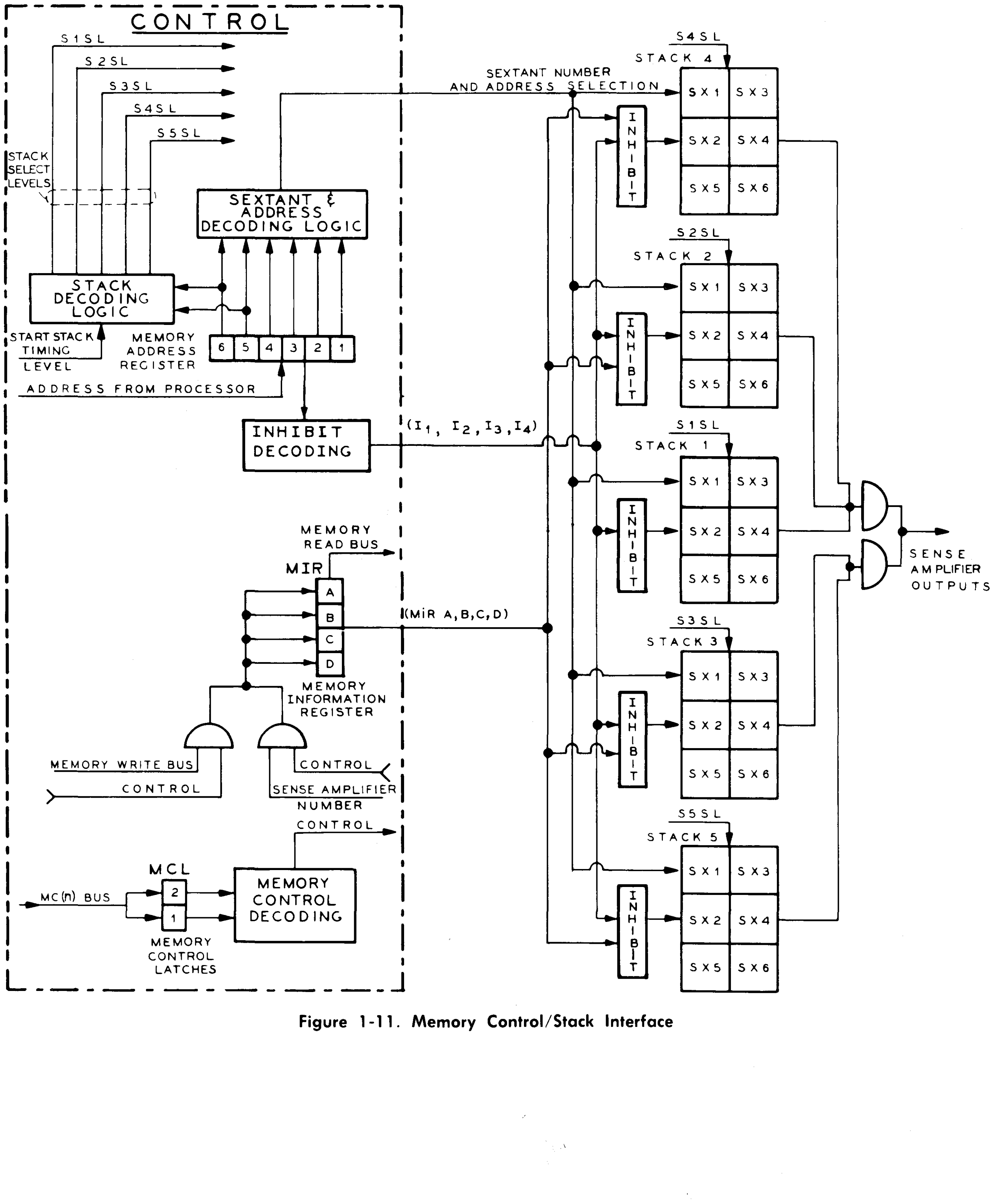

Figure 1-11. Memory Control/Stack Interface

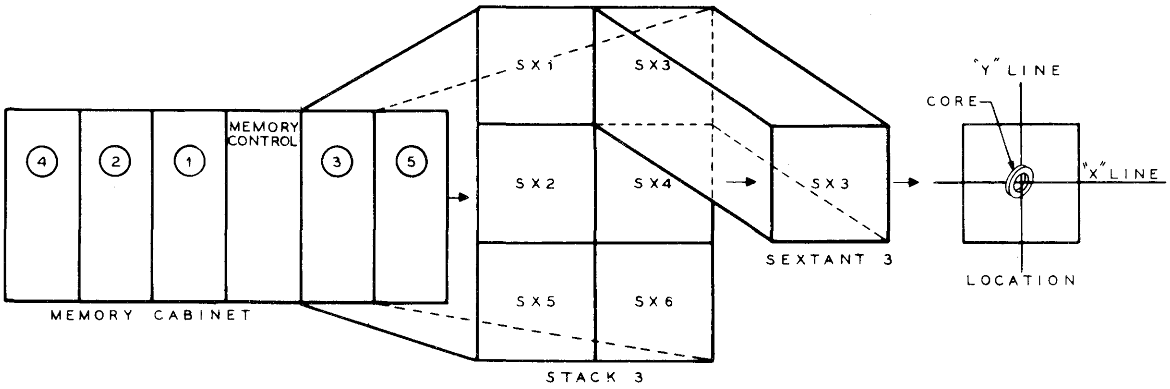

Figure 1-12. Addressing Sequence

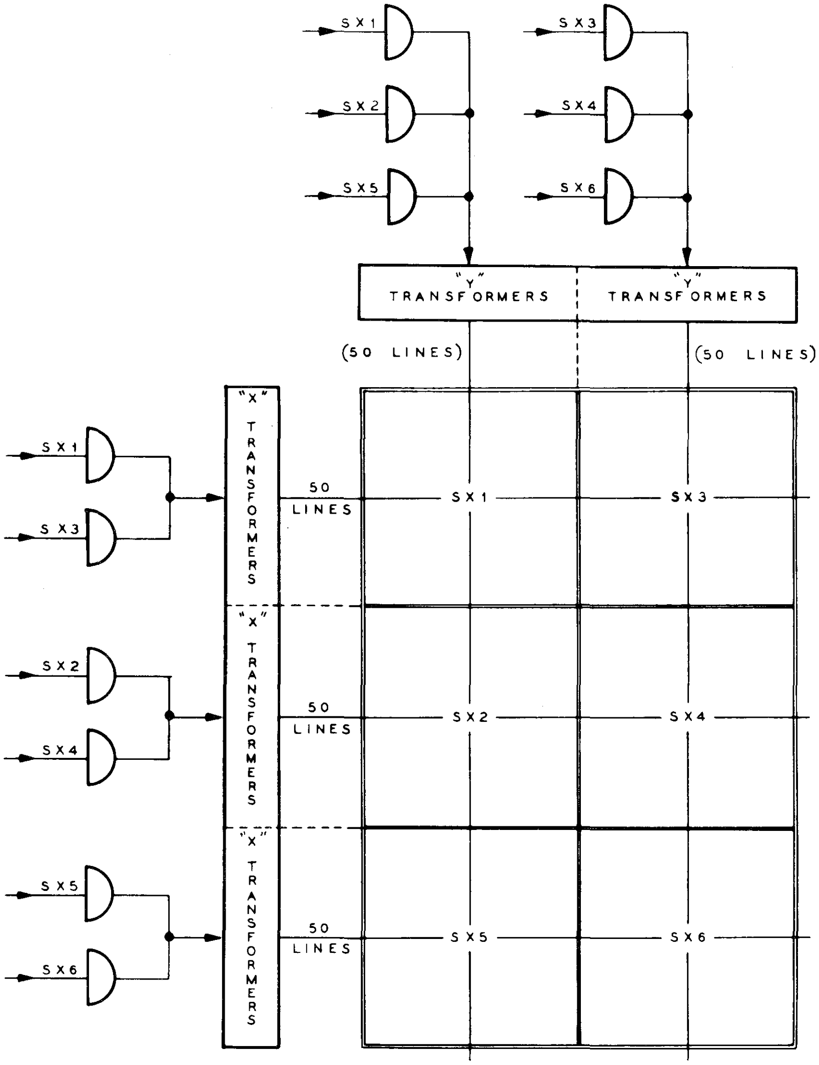

Figure 1-13. Sextant Gating of Transformers

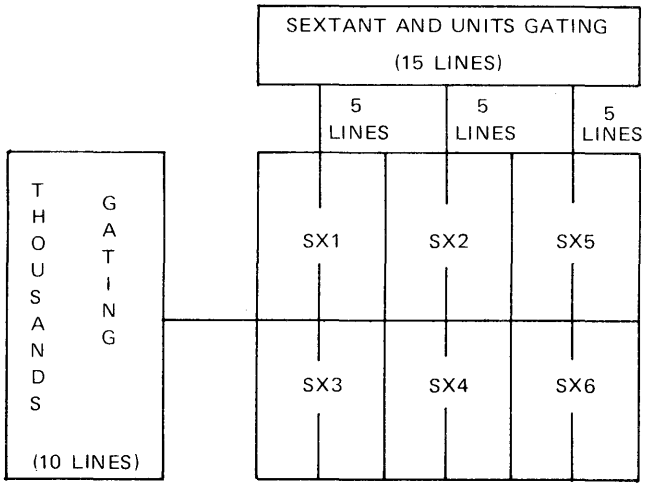

Figure 1-14. 15 x 10 "X" Transfer Array

PARITY

The memory system operates with odd parity. This stipulates that an odd number of bits is set for each word location. That is, if a word is placed in the memory information register (MIR), which contains an even number of bits set, the parity bit must also be set at the time the word is written. If the word within the MIR contains an odd number of bits, the parity bit need not be utilized.

MEMORY TIMING

The memory cycle is one microsecond in duration. Logic generated within the central control, and which is then called to memory, synchronizes timing pulses between memory, the central control, and the central processor. The time duration between clock pulses is 560 nanoseconds.

Memory timing is generated in both memory control and the stack. When a memory cycle is required by the central processor or an I/O channel, the central processor triggers the timing circuits located within memory control. The appropriate stack is selected by address decoding and its timing circuits enabled by a Start Stack Level (SnSL). This logic produces timing pulses required to control the stack function.

STACK TIMING

The timing generated by the stack timing logic is used to control the sequence of operations that causes read and write current to flow within the stack.

READ OPERATION

The address to be used during the read operation is received from the address register within the central processor and is stored in the memory address register (MAR). The MAR is decoded by the stack decoding logic and gated to produce one of five SnSL levels. This level, in turn, determines the stack to be used. The MAR is also utilized by the address and sextant decoding logic to determine the location within the stack that is to be utilized. This information is gated to all stacks within the memory cabinet, but it is usable only in the stack receiving the SnSL level. After the stack is triggered by SnSL, the output from the sense amplifiers is sent to the memory information register (MIR). The sense amplifier senses the status of the cores within the stack. The type of operation to be performed is stored in the memory control latches (MCL), which is then decoded and controls the input to the MIR. In the case of a read operation, the outputs from the sense amplifiers are gated to the MIR where they are then sent to the central control or central processor for usage.

WRITE OPERATION

For a write operation, the logic utilized for the read operation is valid, except that the MCL now contains codes for a write operation. The outputs of the sense amplifiers are inhibited and only the new information from the central control or central processor is latched in the MIR. The inhibit gating, in turn, controls the writing of the information within the MIR into a memory location addressed by the MAR.

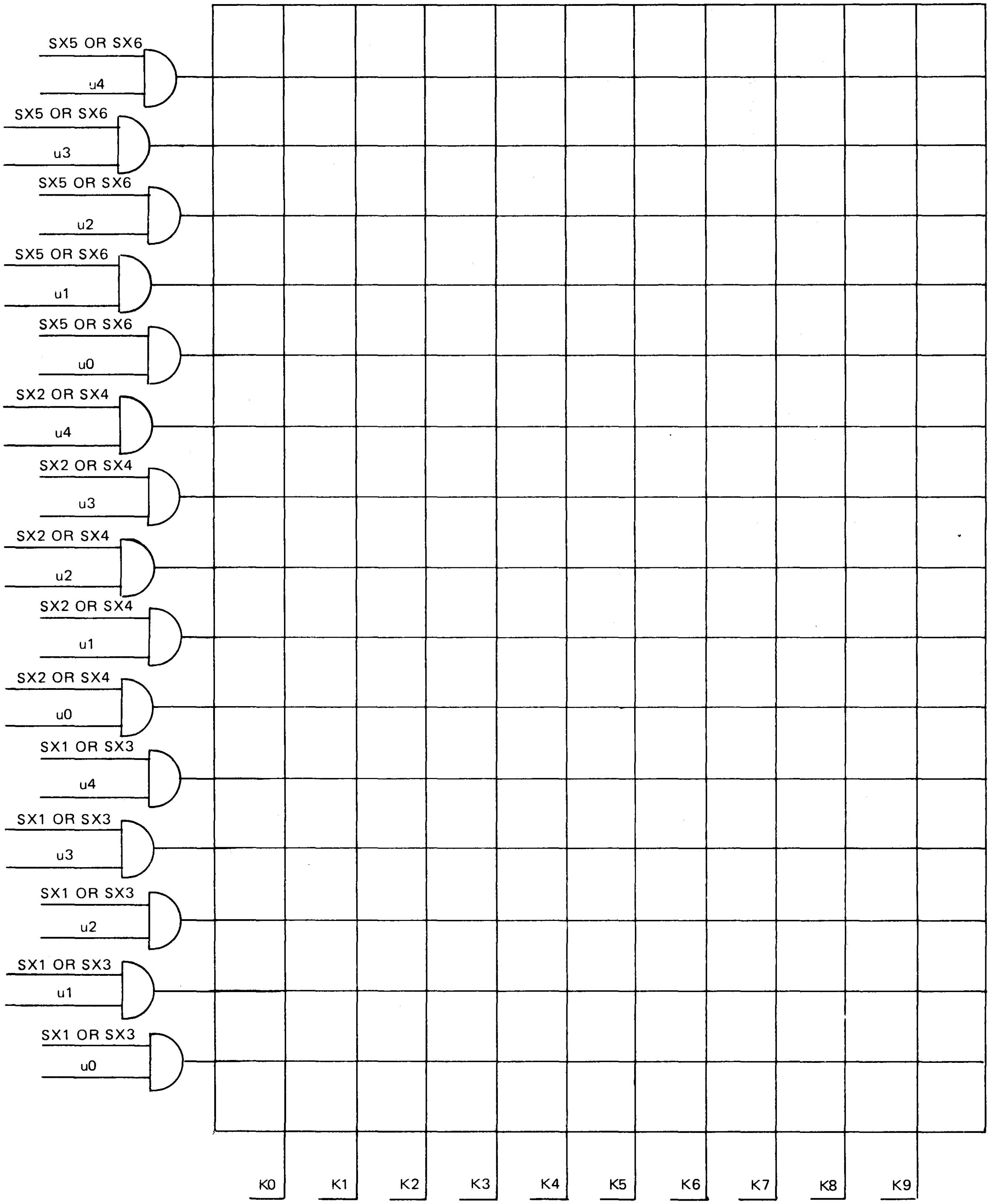

Figure 1-15. "X" Transformer Decoding Lines

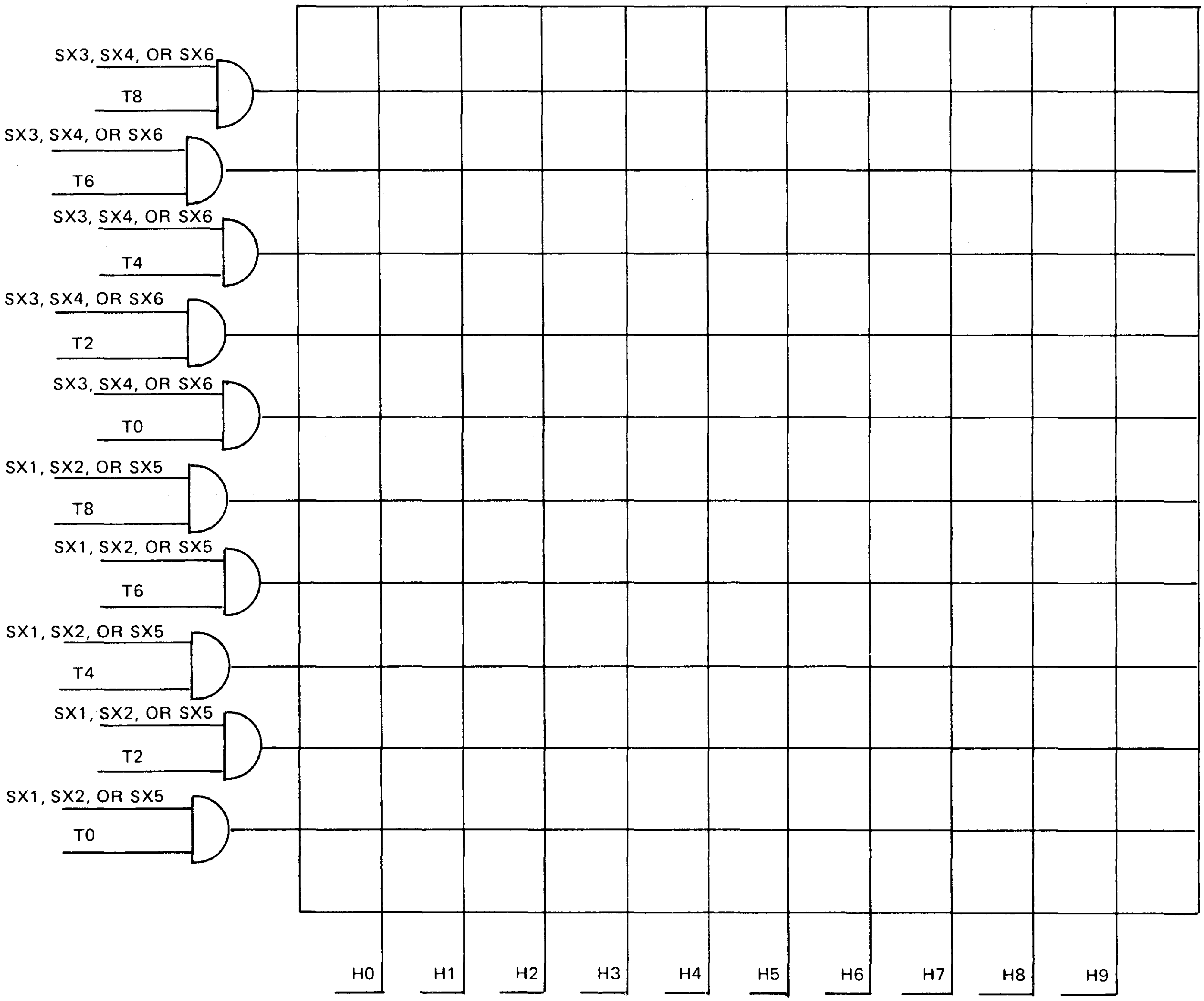

Figure 1-16. "Y" Transformer Decoding Lines

SECTION 2
CENTRAL PROCESSOR

GENERAL

The Central Processors contain the circuitry and logic required to perform the instruction set utilized by their associated systems. These instruction sets are capable of performing operations on digit-, byte-, or word-formatted data fields. The internal working code utilized by the central processors is 8-bit EBCDIC or 8-bit USASCII, both of which are programmatically selectable. Integrated into the hardware is an automatic interrupt system capable of informing the processor of any condition experienced by any portion of the system.

The B 2500/B 3500 and the B 2700 Central Processor speeds are as follows:

Model	Processor Speed
B 2510/B 2520	0.5 MHz
B 2731	0.67 MHz
B 2501/B 2502/B 2741/B 2761/ B 2540/B 2765	1.0 MHz
B 2751/B 2771/B 2772/B 3501/ B 3506/B 3508/B 3510/B 3514	2.0 MHz

PROCESSOR OPERATING MODES

The Central Processors can operate in one of four modes: control state, zero base; control state, non-zero base; normal state, zero base; and normal state, non-zero base.

Control State, Zero Base

In this mode, the interrupt flip-flop can be set, but the interrupt branch is not executed until the return to normal state. Any privileged instructions are allowed.

Control State, Non-Zero Base

In this mode, the interrupt flip-flop can be set, but the interrupt branch is not executed until the return to normal state. Any privileged instructions are disallowed.

Normal State, Zero Base

In this mode, the interrupt branch is allowed to be executed and privileged instructions are allowed.

Normal State, Non-Zero Base

In this mode, the interrupt branch is allowed to be executed and privileged instructions are disallowed.

Object program, assemblers, compilers, and generators are executed in the normal state, non-zero base, whereas the greater majority of the Master Control Program (MCP) is executed in the control state, zero base. A small number of privileged instructions, used exclusively by the MCP, can be executed only in zero base. These priviledged instructions include operations such as initiate output and read timer.

LOGICAL UNITS

There are several logical units contained within the central processor that directly or indirectly affect its operation. These logical units control the execution of specific instructions and are, themselves, set or changed by other instructions.

OVERFLOW Flip-Flop

The OVERFLOW flip-flop is a hardware logical unit that indicates the data field of a move or arithmetic instruction is exceeded. If an overflow condition is detected, the command is executed but the data is not affected. The OVERFLOW flip-flop is not cleared at the beginning of an arithmetic operation but is preserved; therefore, it indicates overflow that has occurred any time before or during a series of arithmetic operations or other interjected non-arithmetic operations. Instructions that can create an overflow condition are:

a. Arithmetic instructions except multiply.

b. Floating point arithmetic instructions.

c. Move Numeric.

d. Move Alphanumeric.

Overflow cannot occur during a multiply instruction, since the receiving field is always large enough to contain the product. In all cases except the floating point instructions, overflow results when the receiving field cannot contain the sending field. With floating point instructions, the overflow can also be caused by an out-of-range exponent. The OVERFLOW flip-flop is reset by the Branch-On-Overflow instruction. Once cleared, it can be set if the conditions arise while executing those instructions that may cause it to be set. There are two ways in which the present setting of the OVERFLOW flip-flop may be stored in the four-bits of reserved address memory and then cleared, as shown below.

a. Branch Communicate.

b. Automatic interrupt feature.

The OVERFLOW flip-flop is restored from the reserved memory location by the Branch Reinstate instruction. In addition, the object program may branch to some subroutine and the OVERFLOW flip-flop setting must be retained and restored at the conclusion of the subroutine execution, prior to continuing in the object program. The Enter instruction stores the flip-flop setting into the four bits of the reserved character location in the memory stack, and then clears the OVERFLOW flip-flop. The Exit instruction restores the OVERFLOW flip-flop setting from this reserved location.

COMPARISON Flip-Flops

Two hardware flip-flops make up the comparison logical unit. These two flip-flops have the following four combinations (0 = reset, 1 = set):

a. 00 — cleared.

b. 01 — greater or high comparison result.

c. 10 — less or low comparison result.

d. 11 — zero or equal comparison result.

These four states represent the result of executing an instruction that affects these logical units. The cleared state actually indicates that there is no comparison result existing. The instructions that set a result into the COMPARISON flip-flops and are reflected on the appropriate console indicators are as follows:

a. All arithmetic instructions.

b. All floating point instructions.

c. Compare instructions.

d. Move Numeric instruction.

e. Move Alphanumeric instruction.

f. The bit test instruction.

g. The logical instructions.

h. The scan instructions.

i. The Edit instruction.

j. The Scan Result Descriptor instruction.

k. Search instruction.

Any branching that is done on the basis of the comparison will not change the status of the flip-flops. Only another instruction that affects them can change their status. When entering control state by means of the Branch Communicate instruction or the automatic interrupt system, the status of the comparison flip-flops is stored. The 2-bit and 1-bit of the same character that stores the OVERFLOW flip-flop status will contain the status of the COMPARISON flip-flops. The flip-flops are then cleared before branching to control state. When return is made to the normal state, the Branch Reinstate instruction will restore the comparison flip-flops from the character in reserved address memory. Similarly, when entering a subroutine, the Enter instruction stores the COMPARISON flip-flops in the 2-bit and 1-bit of the specified character in the memory stack. This same character contains the OVERFLOW flip-flops status. When leaving the subroutine with the Exit instruction, the COMPARISON flip-flops are restored from this character in the stack.

EBCDIC/USASCII Mode Flip-Flop

This flip-flop determines whether the processor is operating on EBCDIC or USASCII internally coded data. The reset state of the flip-flop indicates that the processor is using the EBCDIC code. The flip-flop can be programmatically set to indicate that the processor is to use USASCII code. The Set Mode instruction sets or resets this flip-flop, as desired by the programmer. The following instructions are sensitive to the setting of the EBCDIC/USASCII mode flip-flop:

a. All Arithmetic instructions.

b. All Floating Point instructions.

c. Move Numeric instruction.

d. Move Alphanumeric instruction.

e. Move Repeat instruction.

f. Translate instruction.

g. Scan Delimiter Equal instruction.

h. Scan Delimiter Unequal instruction.

i. Edit instruction.

When branching to the control state, the setting of the EBCDIC/USASCII MODE flip-flop is also stored in the 8-bit of the same character containing the settings of the OVERFLOW and COMPARISON flip-flops. It is then cleared prior to entering the control state. This is done by either of the following instructions:

a. Branch Communicate.

b. Automatic interrupt branch.

When returning to normal state, the Branch Reinstate instruction will restore the EBCDIC/USASCII mode flip-flop from the character in reserved address memory. When entering a subroutine, the Enter instruction will store the status of the MODE flip-flop in the 8-bit of the specific character in the stack which also contains the OVERFLOW and COMPARISON flip-flop settings. The Exit instruction will restore the EBCDIC/USASCII MODE flip-flop from the character in the stack when leaving the subroutine. This is necessary in the event that the flip-flop is changed by the subroutine. The special character in reserved memory or the stack contains the setting of the EBCDIC/USASCII mode flip-flop in the 8-bit, the setting of the OVERFLOW flip-flop in the 4-bit, and the settings of the COMPARISON flip-flops in the 2-bit and 1-bit.

INTERRUPT Flip-Flop

The INTERRUPT flip-flop is a hardware logical unit that indicates the presence of an interrupt. This flip-flop controls the automatic interrupt circuitry when the processor is operating in normal state. If the flip-flop is reset (no interrupt condition), execution continues in sequence from instruction to instruction. If an interrupt occurs, the flip-flop is set. If the flip-flop is set at the completion of an instruction execution cycle, the automatic interrupt branch to control state, base zero occurs. The flip-flop is set by the detection of any I/O complete interrupt. The Scan Result Descriptor instruction will reset the INTERRUPT flip-flop while executing the interrupt handling routine.

Normal/Control State Flip-Flop

This flip-flop can be set only by the Branch Reinstate instruction which is executed by the Operating System. It is reset by either the Branch Communicate instruction or the automatic interrupt branch. When the flip-flop is set, the processor is operating in the control state which inhibits the automatic interrupt circuitry.

The setting of this flip-flop dictates the state of operation of the processor. When it is set, the processor is operating in normal state and allows the execution of the automatic interrupt branching.

Timer

The timer consists of two words in address memory. The first timer word is counted at a one KiloHertz rate. This word is compared with the value of the second word which is placed there by the Operating System. When the first word is equal to or greater than the second word, a clock interrupt occurs. The timer will continue to count after the interrupt and requires the Operating System to read and clear the first word of the timer. The value read from the timer word is used by the Operating System for logging functions and updating the real-time clock.

Base Register

The base register in the processor is a three-digit register. These three digits are always added to the two high-order digits of the five-digit program and instruction address to form a six-digit absolute address. This makes the base register modulo 1000, that is, a base register value of 137 is actually 137,000. This base register value is always added to the base relative address contained in the instructions. This computation takes place during the fetch cycle, prior to storing the absolute address generated into address memory. For any program, the base register value is created by the Operating System and stored in reserved memory location 000070 (table 2-1) prior to initiating the program. When the Branch Reinstate instruction is executed, the base register is set to the value contained in reserved memory, establishing the base register of the normal state program. If a Branch Communicate instruction or an automatic interrupt branch is executed, the base register value is stored in the reserved address memory location prior to entering control state and the base register is also cleared to zero.

Limit Register

The limit register within the processor is a three-digit register. The purpose of the limit register is to provide memory protection. Memory protection is accomplished by comparing the high-order three digits of the absolute address generated for instructions with the base and limit registers. The base register provides the lower limit and the limit register provides the upper limit. The value of the limit register is provided by the Operating System and stored in reserved memory location 000073. The Branch Reinstate instruction sets the limit register to the value contained in reserved memory. When the Branch Communicate instruction or automatic interrupt branch is executed, the limit register value is stored in reserved memory prior to entering control state and the limit register is set to the system memory size. This allows the Operating System to access all memory.

Index Registers

The index registers used in the processors are not hardware registers as are the base and limit registers. Three index registers are contained in reserved locations of an object program and are always assigned the same base relative address. The format of the index register is shown in figure 2-1.

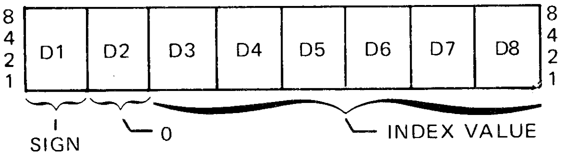

Figure 2-1. Index Register Format

Each index register consists of eight digit positions. The most significant digit (D1) is the sign of the index value. The second digit position is not used when indexing. The value of the index is the decimal value contained in the remaining six digit positions (D3 through D8). If an instruction address specifies indexing, the value of the specified index register is added algebraically to the base register, plus the base relative address prior to storing the generated address in address memory. The index register value can be changed by any instruction that addresses memory. In addition, the Enter and Exit instructions use and alter the setting of Index Register 3 (IX3); however, the settings are saved and restored. In the zero-base state, the Scan Result Descriptor instruction and the Search instruction use IX1 in their execution.

DATA MODES

Two data modes are utilized in the Processors to represent internal data: the 4-bit mode and the 8-bit mode.

Four-Bit Mode

In the 4-bit mode, data is interpreted in units of 4-bits and, where a sign is expected, the sign is interpreted as a separate and leading 4-bit unit. The internal code in 4-bit mode is interpreted by the arithmetic unit in the processor as follows:

Binary Code	Sign Code	Decimal Equivalent
0000	+	0
0001	+	1
0010	+	2
0011	+	3
0100	+	4
0101	+	5
0110	+	6
0111	+	7
1000	+	8
1001	+	9
1010	+	~~0~~ (undigit 0) *
1011	+	~~1~~ (undigit 1) *
1100	+	~~2~~ (undigit 2) *
1101	-	~~3~~ (undigit 3) *
1110	+	~~4~~ (undigit 4) *
1111	+	~~5~~ (undigit 5) *

* These units are accepted by the arithmetic unit but they will result in unspecified results.

When a signal 4-bit format is specified in the receiving field for any operation, a plus sign compares higher than a minus sign in the system's collating sequence, as shown below:

USASCII Code	EBCDIC Code
+ = 1011	+ = 1100
- = 1101	- = 1101

Eight-Bit Mode

In an 8-bit mode, data is interpreted in units of 8-bits (one byte). Any required conversion between 4-bit and 8-bit mode is accomplished automatically during the execution of instructions with no timing costs absorbed by the program being operated.

For code sensitive instructions including the manipulation of 4-bit numeric data, the most significant fours bits of a 2-byte receiving field are automatically set to the code indicating the numeric subset of the selected 8-bit code. The four bits are 1111 (undigit 5) in EBCDIC code and 0101 (binary 5) is USASCII code.

Except in the move alphanumeric, move numeric and edit instructions, 8-bit data is considered unsigned.

Alphanumeric comparisons are performed in binary code. The collating sequence for EBCDIC code is special characters, alphabetical characters, and then digits. The collating sequence for USASCII is special characters, digits, and then alphabetical characters.

INSTRUCTION FORMAT

The two instruction formats used by the processors are format A, which is variable in length from one to four syllables, or format B, which is a fixed length of eight digits. Most instructions consist of from one to four syllables, depending on the particular instruction. An instruction syllable consists of six digits as shown in figure 2-2.

The first syllable of every instruction contains the operation code in digits D1 and D2. Digit positions D3 through D6 have various functions as specified by each instruction. Generally, D3 and D4 specify the length of the A field, whereas D5 and D6 specify the length of the B field. Several instructions consist of a single syllable. In multiple syllable instructions, the second, third, and fourth syllables are the A field, B field, and C field addresses, respectively. The addresses always address the most significant digit of the field. Digits D7, D13, and D19 specify various control functions with respect to the A, B, and C field addresses, respectively. The two address-index bits determine which index register, if specified, to use in generating the absolute address. The two address-controller bits specify whether signed or unsigned 4-bit or 8-bit formatted information is in the data field. The address controller bits may also specify indirect addressing. Digits D3 and D4 of the first syllable can specify that the second syllable (D7 through D12) is a literal value and not an address. The format of the literal is given in digits D3 and D4.

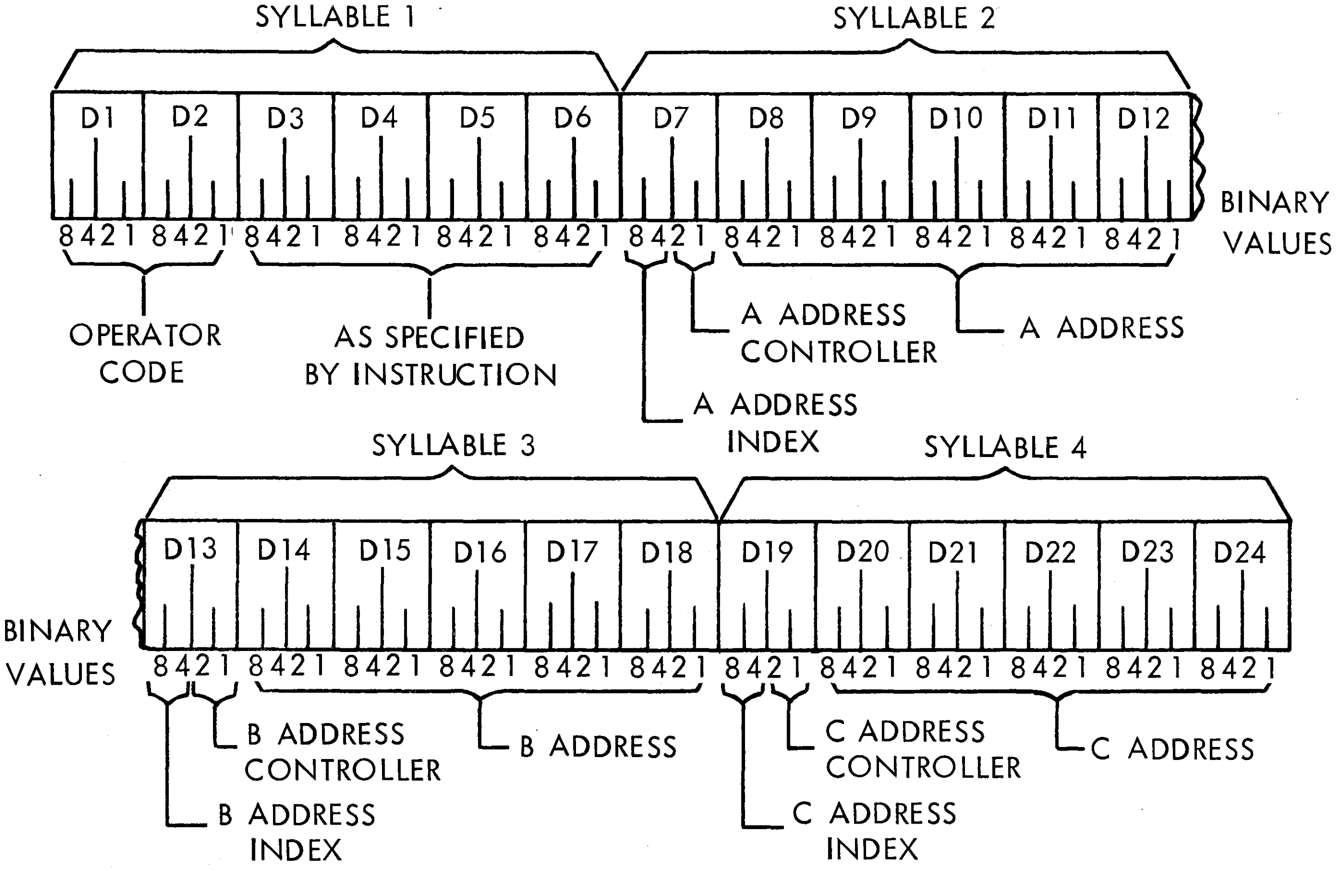

Figure 2-2. Six-Digit Instruction Format

The 8-digit format is used only when an operation code and address are necessary, such as in a Branch instruction. The 8-digit instruction format is shown in figure 2-3.

Address Index

The two most significant bits (8 and 4) of the first digit in the address field is the address index control. The bit configuration of the address index signifies which of three index registers is to be used while generating an absolute address. The eight-bit and four-bit configuration is as follows:

a. 00 — no indexing.

b. 01 — index register 1 (IX1).

c. 10 — index register 2 (IX2).

d. 11 — index register 3 (IX3).

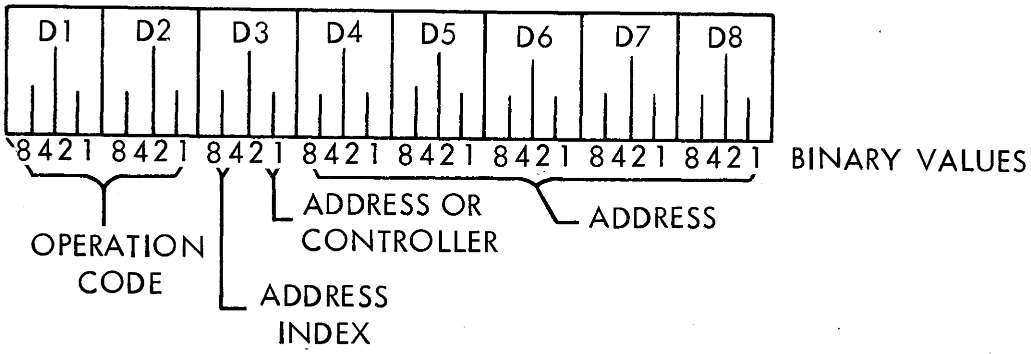

Figure 2-3. Eight-Digit Instruction Format

There is an address index controller for the A, B, and C address fields, and also for the address field of the eight-digit instruction. If indexing is specified, the designated index register value contained in reserved locations within an object program is added algebraically to the address field and base register during the fetch cycle, prior to storage in its reserved address memory location.

Address Controller

The two low-order bits (2 and 1) of the first digit in the address field specify the data field format. The address controller bits and address index bits comprise the entire first-digit position of the address field. Four different two-bit and one-bit combinations can be specified by the two address controller bits as follows:

a. 00 — unsigned four-bit format.

b. 01 — signed four-bit format.

c. 10 — unsigned eight-bit format.

Most significant address digit for the branch instructions

d. 11 — indirect address.

Any combination may be used except where prohibited by a specific instruction. The unsigned 4-bit format means that the data field consists of digits of information without a sign digit. The signed four-bit format specifies a digit data field with a sign digit preceding the most significant digit of the field. The field length does not include the sign digit position when a signed data field is specified. A bit configuration of 1101 is always treated as a minus sign (-), whereas any other bit configuration is considered as a plus sign (+). The unsigned eight-bit (character/byte) format specifies that the data field is in an internal eight-bit alphanumeric code. The indirect address bit configuration specifies that the data field address is located at the specified base relative location. During the fetch cycle, another access must be made to obtain the specified field address.

An indirect address must be even and is checked after indexing, if any. An odd indirect address is considered to be a non-synchronized address contained in an instruction and causes a processor interrupt.

Address Digits

Since the first digit of the address field comprises the address index and controller bits, five digits remain for use as the data field address; however, addresses may generate beyond this seemingly maximum address of 099,999 by utilizing the address controller bits at binary values up to a 2, thus creating an actual maximum address range per program of 000000 to 299,999. Programs exceeding the address of 099,999 cannot contain referenced data areas in excess of the 099,999 address. This address is always base relative and is added to the base register value during the fetch cycle. The absolute address stored in address memory is comprised of the value in the address field added to the base register. It can also be indexed by an index register when specified. The address digits are limited to the decimal digits zero through nine, thus creating a completely decimal addressing scheme for a beneficial system/programmer/operator interface.

Operator Code

The first two digits of the first syllable represent the operator code. This operator code is a decimal value ranging from 00 through 99, but all values are not used. All unassigned codes are reserved for expansion. If the processor receives any operator code that has not been assigned, an invalid instruction interrupt is generated. The operator code itself determines the number of syllables that must be fetched.

Field Length

For a majority of the instructions, the third, fourth, fifth, and sixth digits (D3 through D6) of the first syllable control the field lengths of the data fields (see figure 2-4). Digits D3 and D4 determine the A field length (AF), whereas digits D5 and D6 determine the B field length (BF).

The field length values can range between 00 and 99, inclusively. If a field length of 00 is specified, the field length is 100. The digits D3 and D5 may also specify an indirect field length. This is accomplished by making the D3/D5 8-bit and 4-bit both equal to a 1. The base relative indirect address of the field length is obtained from the two low-order bits of D3/D5 and the even values of the D4/D6 digit. Thus, the indirect field-length base relative address can be any even value from 00 through 38 for a total of 20 base relative addresses.

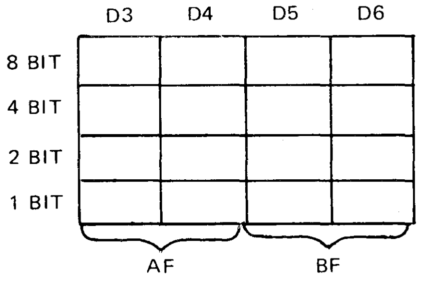

Figure 2-4. Field Length Digits

Literal Specifications

Some of the instructions allow the A field length digits to specify that the A address syllble contains a literal value and not the address of a data field. Figure 2-5 shows the bit configuration for the various literals that the A address syllable can contain.

When D3 has the 8-bit and 2-bit ON with the 4-bit OFF, the A address is a literal. The D3 1-bit and D4 8-bit specify the format of the literal. When the literal specification is an unsigned four-bit format, the literal length can be one through six — the total number of digits in the address field. When the signed four-bit format is specified, the literal length can be one through five since the most significant digit is the sign digit.'

When the eight-bit format is specified, the literal length can have a value of one through three. If the literal length is less than the maximum length for the specified format, the literal is always assumed to be left-justified.

Read-Only Memory

The read-only memory (ROM) within the processor is a resistive-type of storage. It contains a set of microprograms that controls most of the actions taken by the processor. The microprograms are initiated by the operation codes of program instructions after they are fetched from memory. The microprograms utilize the addresses stored in address memory during the execution of the instruction.

HARDWARE INSTRUCTIONS

The following list is a composite of all mnemonic operation codes used by the systems processor. Assemblage of this list is only by general classification of the individual codes.

Each operation code will be broken down into its associated descriptor.

Data Movement Operation Codes

The following operator codes are utilized to manipulate data to and from different areas within the system.

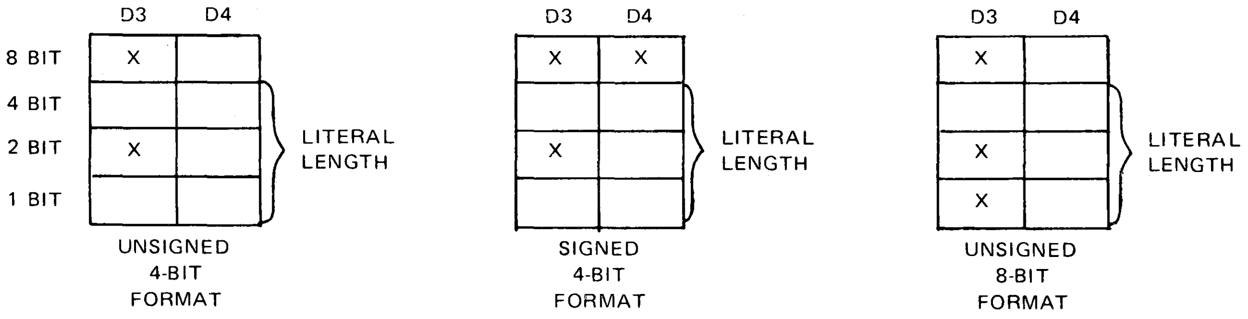

Figure 2-5. Literal Specification Control

MOVE ALPHANUMERIC (MVA)

The number of syllables is equal to two. The descriptor for the MVA operation is shown below:

OP Code	AF	BF	A Address	B Address
10	Length of data field	Length of destination field	Address of data field	Address of destination field

This instruction moves digits or characters from the A field (AF) and stores them at the location specified by the B address. The count in AF and BF specifies the length of the A and B fields, respectively. Up to 100 digits or characters may be moved, with the one-hundredth character specified by the count 00 in either length field.

MOVE NUMERIC (MVN)

The number of syllables is equal to two. The descriptor for the MVN operation is shown below:

OP Code	AF	BF	A Address	B Address
11	Length of data field	Length of destination field	Address of data field	Address of destination field

This instruction moves digits or the numeric portion of characters from a location specified by the A address to a location specified by the B address. The value within AF specifies the number of digits to be moved. The length of the destination field is contained in BF.

MOVE REPEAT (MVR)

The number of syllables is equal to two. The descriptor for the MVR operation is shown below:

OP Code	AF	BF	A Address	B Address
14	Length of A field data	Number of times that data is to be moved to the B field	Most significant digit (MSD) of the data to be moved	Most significant digit (MSD) of the result field

This instruction moves data from a location specified by the A address to a location specified by the B address. AF contains the number of digits or characters in the A field. BF specifies the number of times the entire A field is to be moved to the B field. The length of the B field is the product of AF times BF.

MOVE WORDS (MVW)

The number of syllables is equal to two. The descriptor for the MVW operation is shown below:

OP Code	AF	BF	A Address	B Address
12	Number of words to be moved		Address of data to be moved	Address of designation field

This instruction moves words from a location specified by the A address and stores them in a location specified by the B address. At the completion of the instruction, both the A and B fields (AF and BF) contain identical information.

MOVE WORDS AND CLEAR (MVC)

The number of syllables is equal to two. This descriptor is identical to that of the MVW operation, except for the following:

a. The OP code is equal to 13.

b. Zeros are written into the A location after each word has been moved.

MOVE LINKS (MVL)

The number of syllables is equal to three. The descriptor for the MVL operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
09	Length of all fields (1 through 100)	Unused	Instruction Addressed		

This instruction is comprised of four syllables: the OP code, the AF variant, the BF variants, and three addresses. The AF variant specifies allowable data field lengths. These lengths are from 1 to 100 digits or characters. The final address controllers for all three addresses must be alike and can indicate unsigned numeric, signed numeric, or unsigned alpha information.

During the execution of this instruction, the processor ignores the A and B controllers and uses the C controller to control all of the information fields.

Arithmetic Operation Codes

The following operator codes are utilized to perform arithmetic calculations.

TWO ADDRESS ADD (INC)

The number of syllables is equal to two. The descriptor for the INC operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
01	"A" field length	"B" field length	Address of addend or subtrahend	Address of augend or minuend	Address of result field

This instruction algebraically adds the contents of the A field to the contents of the B field. The sum is stored in the B address.

THREE ADDRESS ADD (ADD)

The number of syllables equals three. The ADD instruction is identical to the two address add instruction except for the following:

a. The OP code is equal to 02.

b. The sum is stored in the C address.

TWO ADDRESS SUBTRACT (DEC)

The number of syllables equals two. The DEC instruction is identical to the two address add instruction except for the following:

a. The OP code is equal to 03.

b. The difference is stored in the B address.

THREE ADDRESS SUBTRACT (SUB)

The number of syllables equals three. The SUB instruction is identical to the two address add instruction except for the following:

a. The OP code is equal to 04.

b. The difference is stored in the C address.

MULTIPLY (MPY)

The number of syllables equals three. The descriptor for the MPY operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
05	"A" field length	"B" field length	Address of multiplier	Address of multiplicand	Address of product

This instruction algebraically multiplies the multiplicand specified by the B address by the multiplier specified by the A address and places the product into the address specified by C.

AF and BF in this instruction specify the length of the A field and B field, respectively. These fields may specify from 1 to 100 digits, with the value of 00 representing a length of 100. The length of the C field will be the sum of AF and BF.

DIVIDE (DIV)

The number of syllables equals three. The descriptor for the DIV operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
06	A field length	B field length	Address of divisor	Address of dividend or remainder	Address of quotient

This instruction algebraically divides the contents of the B field by the contents of the A field and places the quotient in the C field. Any remainder as a result of the division will be found in the B field.

Floating Point Arithmetic Operation Codes

The following operation codes enhance the arithmetics of the system by allowing floating-point operation.

FLOATING POINT ADD (FAD)

The number of syllables is equal to three. The descriptor for the FAD operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
80	Length of A mantissa (not including sign)	Length of B mantissa (not including sign)	Address of A field	Address of B field	Address of C field

In floating point addition, the exponents of A and B must be made equal before algebraically adding the mantissa. Equalizing the exponents is accomplished by incrementing or decrementing the exponents of A or B addresses, while effectively shifting the decimal point in the mantissa.

FLOATING POINT SUBTRACT (FSU)

The number of syllables equals three. The descriptor for the FSU operation is identical to that of the floating point add operation, except that the OP code is equal to 81.

FLOATING POINT MULTIPLY (FMP)

The number of syllables equals three. The descriptor for the FMP operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
82	Length of A mantissa (not including sign)	Length of B mantissa (not including sign)	Address of multiplier	Address of multiplicand	Address of product

This instruction algebraically multiplies the contents of the B address by the contents of the A address and places the product into the location specified by the C address.

FLOATING POINT DIVIDE (FDV)

The number of syllables equals three. The descriptor for the FDV operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
83	Length of A mantissa	Length of B mantissa	Address of divisor	Address of dividend or remainder	Address of quotient

This instruction algebraically divides the contents of the B address by the contents of the A address and places the quotient into the location specified by the C address. Any remainder is found at the location specified by the B address and has the same exponent as the original B field (BF). The length of the C mantissa is the difference between AF and BF.

Logical Operation Codes

The following operation codes affect certain logic areas within the system.

TRANSLATE (TRN)

The number of syllables equals three. The descriptor for the translate operation is shown below:

OP Code	AF	BF	A Address	B Address
15	Number of characters to be translated	Address of data to be translated	Address of translate table	Address of result field

The translate instruction translates characters from the A field by the table in the B field and stores them in the C field. The number of characters to be translated is contained in the four digit number combined in the AF and BF variants. The maximum number of characters is 10,000, represented by 0000 in AF and BF. The A character is used to determine the address in the translate table of the new character.

SCAN TO DELIMITER - EQUAL (SDE)

The number of syllables equals two. The descriptor for the SDE operation is shown below:

OP Code	AF	BF	A Address	B Address
16	Number of delimiter(s)	Number of digits or characters in the data field	Address of the delimiter(s)	Address of data

This instruction informs the programmer of the presence or absence of any delimiter(s) and the location of the delimiter(s). The delimiter is a character, or group of characters used to terminate the instruction. The SDE instruction counts the number of digits or characters prior to a delimiter. The AF and BF variants can specify up to 100 digits or characters with the value of 00 being equal to 100.

SCAN TO DELIMITER - UNEQUAL (SDU)

The number of syllables equals two. This operation is identical to the scan to delimiter — equal operation except for the following:

a. The OP code is equal to 17.

b. The SDU counts the number of delimiters before a character is found that is not a delimiter.

SCAN TO DELIMITER - ZONE EQUAL (SZE)

The number of syllables equals two. This operation is identical to the scan to delimiter — equal operation except for the following:

a. The OP code is equal to 18.

b. The SZE operation utilizes the zone portion of the code being used (ASCII or EBCDIC).

SCAN TO DELIMITER - ZONE UNEQUAL (SZU)

The number of syllables equals two. The SZU operation is identical to the scan to delimiter — unequal operation except for the following:

a. The OP code is equal to 19.

b. The SZU operation utilizes the zone portion of the code being used (ASCII or EBCDIC).

BIT ZERO TEST (BZT)

The number of syllables equals one. The descriptor for the BZT operation is shown below:

OP Code	AF	BF	A Address
40	Length of A field	Mask of bits to be tested	Address of data to be tested

This instruction is used to test for a bit or bits in the zero state of a string of digits or characters. The bits of a digit or character to be tested are indicated by the two-digit mask located in the BF position of the instruction. The AF position indicates the length of the information tested in digits or characters.

BIT ONE TEST (BOT)

The number of syllables equals one. The BOT operation is identical to the bit zero test operation except for the following:

a. The OP code is equal to 41.

b. The operation is used to test for a bit or bits in the one state.

AND (AND)

The number of syllables equals three. The descriptor for the AND operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
42	Length of the A field	Length of the B field	Address of the A field	Address of the B field	Address of the logical result

This instruction will provide a logical result of two fields: the two fields of data are addressed by the A and B addresses. The logical result is stored at the location specified by the C address.

The field lengths of A and B are indicated by the AF and BF variants, respectively. The length of the C Field is equivalent to the length of the longer field: A or B. The field lengths can be from 1 to 100 digits or characters, with a value of 00 indicating a field length of 100.

The AND instruction will produce a bit-by-bit logical product of the A and B fields. The result bit will be set if the corresponding bits are both set in the A and B fields. Figure 2-6 contains a truth table for the AND instruction in which A and B are corresponding bits of the data fields and C is the corresponding result bit.

A ADDRESS	B ADDRESS	C RESULT
0	0	0
1	0	0
0	1	0
1	1	1

Figure 2-6. "AND" Truth Table

OR (ORR)

The number of syllables equals three. The ORR instruction is identical to the AND instruction except for the following:

a. The OP code is equal to 43.

b. The ORR instruction will set a bit in the result field if either corresponding bit in the A or B fields is equal to one. Refer to figure 2-7. If the A and B fields are of unequal length, then the shorter field is made equal to the longer by assuming trailing zeros.

A ADDRESS	B ADDRESS	C RESULT
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2-7. "ORR" Truth Table

NOT (NOT)

The number of syllables equals three. The NOT instruction is identical to the AND instruction except for the following:

a. The OP code is equal to 44.

b. The NOT instruction will set a bit in the result field if a corresponding bit is on in either the A or B fields, but not on in both (exclusive OR).

Refer to figure 2-8 for the NOT truth table. If the A and B fields are of unequal length, the shorter field is made equal to the longer by assuming trailing ones.

A ADDRESS	B ADDRESS	C RESULT
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2-8. "NOT" Truth Table

SEARCH (SEA)

The number of syllables equals three. The descriptor for the SEA operation is shown below:

OP Code	AF	BF
34	Length of A field	B field increment

The A field (AF) contains the length of the mask and the search is from the B field to the C address in accordance with the C controller. The C controller specifies the type of search, as follows: Equal, Low, or Lowest.

COMPARE ALPHANUMERIC (CPA)

The number of syllables equals two. The descriptor for the CPA operation is shown below:

OP Code	AF	BF	A Address	B Address
45	Number of characters in the data field	Number of characters in the reference field	Address of the data field	Address of the reference field

This instruction compares the characters in the A field to the characters in the B field. The comparison indicators are set to indicate whether the A field is less than the B field (low), equal to the B field (equal), or greater than the B field (high). The count in AF and BF specifies the number of characters in their respective fields. The fields may be of unequal length.

COMPARE NUMERIC (CPN)

The number of syllables equals two. The descriptor for the CPN operation is shown below:

OP Code	AF	BF	A Address	B Address
46	Number of digits or characters in the A field.	Number of digits or characters in the B field.	Address of data to be compared.	Address of reference field.

This instruction compares the numeric data of the A field with the numeric data of the B field. The comparison indicators are set according to the result of the comparison.

The AF and BF variants can designate a number of digits from 01 through 99 or 00 which represents 100. If the variants are not equal, the shorter field is assumed to have leading zeros until the field lengths are equal.

EDIT (EDT)

The number of syllables equals three. The descriptor for the EDT operation is shown below:

OP Code	AF	BF	A Address	B Address	C Address
49	Reserved: must be equal to zero.	Number of micro-operations in the B field.	Address of data	Address of micro-operators.	Address of result field.

This instruction moves data from the A field to the C field in a manner specified by micro-operators in the B field. This instruction is used to mask data into a format which would normally be used on output media. That is, for business transactions it would insert the dollar signs, commas and periods in the appropriate places in the data for printing. For other operations, it may insert dashes, negative or positive signs, or whatever else is deemed necessary.

The count in BF may specify up to 100 micro-operators, with 100 indicated by 00.

Address Branching

The following operator codes are utilized for address manipulation within the system.

NO OPERATION (NOP)

The number of syllables equals one. The descriptor for the NOP operation is shown below:

OP Code	AF	BF
20	NOT USED	

This instruction has no logical function.

BRANCH ON LESS THAN (LSS)

The number of syllables equals one. The descriptor for the LSS operation is identical to the no operation instruction except the OP code is equal to 21.

This instruction results in a branch if the comparison indicators are low as a result of a prior instruction. If the comparison is other than low (either greater or equal), the operation is a NOP.

BRANCH ON EQUAL (EQL)

The number of syllables equals one. The descriptor for the EQL operation is identical to the no operation instruction except the OP code is equal to 22.

This instruction results in a branch if the comparison indicators are equal as a result of a prior instruction. If the comparison is other than equal (either greater or low), the operation is a NOP.

BRANCH ON EQUAL OR LESS THAN EQUAL (LEQ)

The number of syllables equals one. The descriptor for the LEQ operation is identical to the no operation instruction except the OP code is equal to 23.

This instruction results in a branch if the comparison indicators are equal or less as a result of a prior instruction. If the comparison is other than equal or less the operation is a NOP.

BRANCH ON GREATER (GTR)

The number of syllables equals one. The descriptor for the GTR operation is identical to the no operation instruction except the OP code is equal to 24.

This instruction results in a branch if the comparison indicators are greater as a result of a prior instruction. If the comparison is other than greater (either less or equal), the operation is a NOP.

BRANCH ON NOT EQUAL (NEQ)

The number of syllables equals one. The descriptor for the NEQ operation is identical to the no operation instruction except the OP code is equal to 25.

This instruction results in a branch if the comparison indicators are high or low as a result of a prior instruction. If the comparison is other than high or low, the operation is a NOP.

BRANCH ON GREATER THAN OR EQUAL (GEQ)

The number of syllables equals one. The descriptor for the GEQ operation is identical to the no operation instruction except the OP code is equal to 26.

This instruction results in a branch if the comparison indicators are equal or greater as a result of a prior instruction. If the comparison is other than equal or greater, the operation is a NOP.

BRANCH UNCONDITIONALLY (BUN)

The number of syllables equals one. The descriptor for the BUN operation is identical to the no operation instruction except the OP code is equal to 27.

This instruction does not sample the comparison indicators, but rather branches regardless of their value.

BRANCH ON OVERFLOW (OFL)

The number of syllables equals one. OFL operation is identical to the no operation instruction except the OP code is equal to 28.

This instruction branches if the OVERFLOW flip-flop is set.

Branching Operator Codes

The following operating codes are provided for data and operation branching while remaining in the same hardware flow.

ENTER (NTR)

The number of syllables equals one. The descriptor for the NTR operation is shown below:

OP Code	AF	BF	A Address	P —— P
31	Number of parameters in character, if AF and BF equal 00, no parameters are required		Branch address of the instruction	Parameters or data, if required

This instruction is used to allow a branch to a string of program instructions (referred to as a subroutine) followed by a return from the subroutine to the instruction following the NTR instruction. This allows a program to use an identical string of instructions (subroutine) in several locations within a program with the instructions written only once.

EXIT (EXT)

The number of syllables equals one. The instruction for the EXT operation is shown below:

OP Code	A Address
32	Branching address of this instruction

This instruction is basically an address branch instruction with the exception that its primary use is for exiting from subroutines. This instruction will have the opposite effect on the programmatic stack that the enter instruction had; that is, the enter instruction constructed a stack entry and then branched to a subroutine, the exit instruction will delete this stack entry and branch out of the subroutine.

BRANCH COMMUNICATE (BCT)

The number of syllables equals zero. The descriptor for the BCT operation is shown below:

OP Code	AF	BF
30	Communicate address	

This instruction is used to allow a program to branch to the MCP. The instruction can be used programmatically or it can be hardware generated. In either case, the AF and BF portion of this instruction is the address of a communicate address in the control program.

The branch communicate instruction will store sufficient information into the MCP area of memory to allow a program to be reinitiated by the Reinstate instruction. The instruction can be generated in normal state if an interrupt exists during a fetch operation. It can also be generated if an interrupt exists and the Reinstate instruction is executed while in control state.

Privileged Operation Codes

The following operation codes are classified as privileged operators in that they may be utilized only in zero-base mode.

BRANCH REINSTATE (BRE)

The number of syllables equals zero. The descriptor for the BRE operation is shown below:

OP Code	AF	BF
90	Normal and control state indicator	Unused

The Reinstate is a privileged instruction that is used by the MCP to initiate a program on the processor. This instruction consists of the "AF" and "BF" variants and no addresses.

The Reinstate instruction will transfer to the processor registers from absolute addresses 64 through 76 the information needed to enter a program. This information is located in memory in the following manner:

a. 64 - 69 — Program address. The address of the first instruction to be executed in the program being initiated.

b. 70 - 72 — The base address of the program.

c. 73 - 75 — The limit address of the program.

d. 76 — The value of the COMPARISON, OVERFLOW and MODE flip-flops.

The AF variant indicates if the program initiated is to be in normal or control state. If AF = 1, then the NORMAL flip-flop is set to put the processor into normal state. The BF variant is not used by this instruction.

If the Reinstate instruction is entered with the INTERRUPT flip-flop set, the instruction is changed to the branch communicate (OP code = 30) with the address of 0094. This will prevent an entry to normal state when an interrupt condition exists in the processor.

INITIATE (IIO)

The number of syllables equals one. The descriptor for the IIO operation is shown below:

OP Code	AF	BF	A Address
94	Not Used	Channel number of the I/O control that is to receive the I/O descriptor	Address of the I/O descriptor.

This instruction is a privileged instruction and, as such, the base address register must be equal to zero in order for it to be validly executed.

The purpose of the Initiate instruction is to read an I/O descriptor out of memory and route it to the appropriate places. As such, it serves as a fetch for I/O instructions (descriptors).

The address of the descriptor must specify unsigned numeric or indirect addresses. If an indirect address is used, the final address must specify unsigned numeric.

READ ADDRESS (RAD)

The number of syllables equals one. The descriptor for the RAD operation is shown below:

OP Code	AF	BF	A Address
92	Refer to text	I/O channel designate	Storage Address

Read address is a privileged instruction and, as such, is only valid when the base register is equal to zero. The A storage address may specify indirect addressing. Whether it is direct or indirect addressing it must ultimately be an even address and specify unsigned numeric.

The action taken by the instruction is determined by the least significant digit (LSD) of AF. If AF is equal to 0, the begin word of the designated channel is moved into the A field. If AF = 01, the end word is moved into the A field. If AF = 09, six digits in the A field are written into the begin and end words of the designated channel. I/O Channels "0" through 19 may be designated by BF.

In order for the Read Address instruction to be valid, it must be executed with the designated I/O channel not-busy.

READ AND CLEAR TIMER (RCT)

The number of syllables equals one. The descriptor for the RCT operation is shown below:

OP Code	AF	BF	A Address
95	not used		Storage Address

The Medium Data Processing Systems have a built-in one millisecond timer for logging purposes. Address memory words G and H are reserved for timer operations. Word G is referred to as the first timer word and it is used to count one millisecond time intervals. The maximum count that may be stored in word G is 999,999 milliseconds or approximately 16.67 minutes.

Word H, the second timer word, may be programmatically set to any value of 0 through 999,999. Each time word G is incremented, it is compared to word H.

The A address must be even and may specify unsigned numeric or Indirect Addresses. If indirect addressing is indicated, the final address must specify unsigned numeric.

The RCT operation causes the contents of the first timer word (G) to be placed into core memory starting at the location given by the A address and set the contents of the first timer word to zero.

READ TIMER (RDT)

The number of syllables equals one. The RDT instruction is identical to the read and clear timer instruction except for the following:

a. The OP code is equal to 96.

b. The RDT operation causes the contents of the first timer word to be placed into core memory starting at the location given by the A address without disturbing the word G contents.

SET TIMER (STT)

The number of syllables equals one. The STT instruction is identical to the read and clear timer instruction except for the following:

a. The OP code is equal to 97.

b. The STT operation causes the second timer word (H) to set the value contained at the location specified by the A address.

SCAN RESULT DESCRIPTOR (SRD)

The number of syllables equals one. The descriptor for the SRD operation is shown below:

OP Code	AF	BF
91	Address of the first result descriptor to be scanned.	

This privileged instruction is used by the Operating System to test for the presence of a Result Descriptor Completion and Exception in reserved memory. In addition, a test is made to determine if an Exception condition has occurred.

Miscellaneous Operator Codes

The following operator codes are utilized but not categorized in specific locations.

SET MODE (SMF)

The descriptor for the SMF operation is shown below:

OP Code	AF	BF
47	This variant designates the action of setting or resetting the ASCII (ASC) flip-flop.	Unused

This instruction is used to place the processor into the EBCDIC or ASCII mode of operation. This instruction is a no-address instruction that uses only the high order digit of the AF variant. The BF variant is not used for this instruction.

If AF is equal to one, the instruction will place the processor into the ASCII mode of operation by setting the ASC flip-flop. If AF is equal to zero, the instruction will reset the ASC flip-flop and place the processor into the EBCDIC mode of operation. If AF is not equal to 0 or 1, then the instruction will perform no logical function. In effect, it will act as a NOP.

HALT, BRANCH (HBR)

The number of syllables equals one. The descriptor for the HBR operation is shown below:

OP Code	A Address
29	Branch address of this instruction

This instruction is basically an address branch instruction with the ability to stop the operation of the processor. Depending upon the value of this digit and the status of the NORMAL/CONTROL STATE flip-flop, the processor will execute Halt, Ignore Halt, or treat the HBR instruction as an invalid instruction.

HALT, BREAKPOINT (HBK)

The number of syllables equals zero. The descriptor for the HBK operation is shown below:

OP Code	AF	BF
48	Unused	Contains the breakpoint mask digits

This instruction is a halt that is conditioned by a comparison between the mask in the BF field and the contents of base relative 46 and the halt execution digit in absolute address 77. The mask and base relative 46 digits are used to indicate if the program being executed requires this halt. The contents of absolute address 77 indicates the type of action to occur: execute the halt, ignore the halt, or consider the HBR an invalid instruction. If the halt is executed, operator intervention is required to restart the processor by the depression of the STOP/RUN or the SINGLE INSTRUCTION button. The halt will have no affect on any I/O operations that are in progress. Only processor actions will be stopped.

PROCESSOR RESULT DESCRIPTORS

All processor interrupts that do not turn off the processor clock generate a result descriptor. The INTERRUPT flip-flop is not necessarily set. The format of the result descriptor is shown in figure 2-9.

The result descriptor is a complete memory word of 16 bits, with bit 1 being the most significant. The function of each bit is shown in table 2-1.

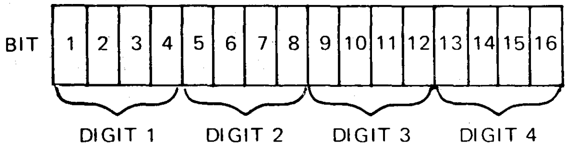

Figure 2-9. Processor Result Descriptor

Table 2-1. Bit Functions of Processor Result Descriptor

Bit	Function
1	This bit is ON whenever a result descriptor is stored.
2	This bit is ON whenever an exceptional condition occurs (always ON whenever a result descriptor is stored).
3	This bit is reserved.
4	This bit is ON if the interrupt was generated by the execution of an invalid I/O operation.
5	This bit is ON if the interrupt was generated by the execution of an invalid instruction.
6	This bit is ON if the interrupt was caused by a memory parity error.
7	This bit is ON if the interrupt was caused by a memory address error.
8	This bit is ON if the interrupt was caused by an instruction time out.
9	This bit is ON if the interrupt was caused by the clock interrupt.
10	This bit is ON if the interrupt was caused by an operator.
11-16	These bits are reserved.

If a processor interrupt is generated, reserved memory will contain a word with bit 1, bit 2, and one or more of the other specified bits ON. When the result descriptor is handled, the first two bits must be turned OFF by the Operating System. The privileged instruction (Scan Result Descriptor) is used by the Operating System to sense the presence of a result descriptor in reserved memory.

NORMAL STATE INTERRUPTS

When certain operational conditions occur within the processor while executing instructions in the normal state, the following logic occurs:

a. If either an operator interrupt, a timer interrupt, or an I/O interrupt occurs, the interrupt will generate a result descriptor and store the descriptor in the reserved address memory location.

b. The interrupt will set the interrupt flip-flop and store the program return point and logical register settings.

c. An automatic branch is taken to the address specified by the contents or reserved address memory location 000094. The processor operational mode is changed from normal state to control state, zero base.

The branch to the control state is a branch to the Operating System — the Master Control Program. It is the function of the Operating System to determine the type of interrupt and the course of action that is to be followed. The interrupt system, as explained, is automatic and is an integral part of the processor hardware system.

Memory Parity Error

When a memory parity error is sensed during a processor access from main memory, an interrupt occurs. (The instruction is continued until completion if the error did not occur during fetch.) The instruction being executed is completed before allowing the automatic interrupt system to become operative.

Address Error

A memory address error can occur under any of the following conditions:

a. Base or limit address error.

b. Non-synchronized address of instructions.

c. Non-synchronized addresses contained in instructions.

d. Non-decimal digits contained in addresses.

Detection of address errors is made on those addresses developed after the addition of the base and index register, if applicable, and on all developed addresses which are advanced during the execution of the instruction. Note that when the base or index register is added to an address containing a non-decimal digit, the resultant address may be changed to an undefined address which no longer contains a non-decimal digit and, therefore, will not be detected. Detection of non-synchronized addresses contained in instructions is limited only to addresses which are used to write into memory.

If any of the above four error conditions occurs, the instruction is completed, but all memory write cycles are inhibited. A memory read cycle is inhibited only when an address contains a non-decimal digit. However, when addresses are advanced during the execution of an instruction, non-decimal digits may be changed to decimal digits, allowing a memory read or write cycle to be performed.

When one of the above error conditions occurs and an instruction is completed, an interrupt is generated and the automatic interrupt system takes over control. A base or limit address error occurs whenever a memory address is out of bounds of the base and limit register settings. This feature provides for memory protection when operating in a multiprogramming mode.

Non-synchronized addresses of instructions is an error when a branch occurs to an address that is not an even value (modulo 2). An error that results from non-synchronized addresses contained in an instruction is caused when an address is not modulo 2 or modulo 4 when required by the format control bits or the instruction code. A non-decimal digit in an address is a digit that has a binary value of 10 through 15 and is referred to as an undigit.

Instruction Time-Out

When an instruction is fetched, an adjustable timer of approximately 250 milliseconds is triggered. If execution of the instruction is not completed during the preset time, an interrupt is generated immediately. This does not include actual I/O operations, since the processor only initiates them, whereas the I/O control unit executes them.

The contents of the reserved location for the next instruction address may or may not have meaning.

Invalid Instructions

An invalid instruction is defined as follows:

a. All non-assigned operator codes.

b. Operator codes requiring options which are not present.

c. Invalid Branch Communicate instruction.

d. Invalid halts.

e. Privileged instructions during non-zero base state operation.

If any of these conditions occur, the absolute address of the invalid instruction is stored in the memory location reserved for the program address. Execution of the instruction and all memory write cycles are inhibited.

Privileged Instructions

There is a set of instructions that can only be executed in zero-base state. If one of these privileged instruction codes is detected while operating in non-zero-base state, an interrupt is generated. Memory write cycles are inhibited, no instruction is executed, and the memory location reserved for the program address contains the absolute address of the privileged instruction.

Clock Interrupt

Two words that comprise the timer and its control value are contained within address memory. The first timer word is counted at a one millisecond rate. The second timer word can contain any six-decimal digit value. When the real timer word is counted to the value of the control, the clock interrupt is generated. The real timer value will continue to count and requires the MCP to read and clear this value from the address memory location. The maximum value that the real timer word can achieve is 999,999 milliseconds. In any event, the current instruction is completed before automatic interrupt handling takes place.

CONTROL STATE INTERRUPTS

When the processor is operating in control state, there are interrupt conditions generated, but no automatic interrupt handling takes place at the completion of an instruction. Rather, at the time that the MCP attempts to return to normal state, automatic branching to control location 000094 will occur so that these interrupts will be handled by the MCP.

If the processor is operating in the zero-base state, the following processor conditions will cause the processor clock to be turned off, leaving all conditions static:

a. Memory parity error interrupt.

b. Address error interrupt.

c. Instruction time out interrupt.

d. All non-assigned operator codes.

e. Operator codes requiring options which are not present.

f. Invalid halts.

Since the processor is in the zero-base state, any of the above interrupts indicates that a hardware or an Operating System error exists. A privileged instruction interrupt cannot occur because all of the instructions are executable in the zero-base state. The clock interrupt condition occurs in the same manner as in normal state, but no automatic branching occurs in control state.

Invalid I/O Operation

Since the zero-base state must be entered to execute an initiate I/O operation, it is possible to have an invalid I/O descriptor which will set the interrupt flip-flop ON and store a result descriptor. Automatic interrupt branching occurs in normal state, but not in control state. An invalid I/O operation interrupt can occur under any of the following conditions:

a. The specified I/O channel is not attached to the system.

b. The I/O descriptor for the specified channel is an invalid operation.

c. An attempt is made to initiate an I/O operation on a channel that is busy with a previously initiated operation.

d. The addresses within the I/O descriptor are not valid (not synchronized, non-decimal digits, or the ending address is not greater than the beginning address).

STACK

The memory stack was previously mentioned with respect to storing the mode, overflow, and comparison flip-flop status prior to entering a subroutine. The stack is an area of memory that has been reserved by the programmer in which to store subroutine linkage words. The beginning address of the stack is stored in the base relative reserved memory location 000040. when an Enter instruction is executed, the address of the stack is obtained from base relative location 000040 and into this stack is automatically stored:

a. The six-digit address of the next instruction.

b. The eight-digit contents of Index Register 3.

c. One character containing the status of the mode, overflow, and comparison flip-flops.

d. The parameters as specified by the Enter instruction.

Index Register 3 is set to the address of the beginning of the stack, and reserved memory location 000040 is set to the address following the parameters (beginning of the next stack) in the event that another subroutine is entered from the one that is being executed. As additional nested subroutines are entered, the same storage process of linkage words into the stack takes place. As the subroutines are exited, return to the preceding level occurs by restoring the values of registers from the stack linkage words. These exits occur until the object program is reached at which time IX3, the address in 000040, and the logical flip-flops are restored to the conditions existing at the time of the first Enter instruction execution. This stack concept greatly simplifies the ability to call on subroutines and to make nested calls on other subroutines.

Adder

The processor uses an adder that accumulates two fields from the most significant to the least significant digit positions. Reverse addition has the advantage of detecting an overflow condition prior to altering the receiving field for the result. The principle used in this type of adder is illustrated by the flow chart in figure 2-10 and the five examples in figure 2-11. If the data fields are signed, sign manipulation takes place prior to the addition since they are the most significant digits.

In figure 2-11, there is no manipulation of signs shown. The addition of two data fields is only shown to present the technique of a left-to-right adder.

In example 1, as each set of digits is added, no carry is generated and no nines are produced. As each new result is generated by the adder, the previous result is stored in the result field in core memory. In example 2, an overflow condition is immediately detected because of a carry on the first digit addition. In example 3, the first five nines are not stored until the result equal to eight is generated. The eight is retained until the carry is generated when adding the last digits, at which time the eight is made into a nine and stored. The following nines are then stored as zeros in the receiving field, and then the final digit (8) is stored. Nothing is stored in example 4 until the final digits are added. The receiving field in example 5 remains unchanged even though an overflow condition is not detected until the final digits are added. This is due to the result being contained in the nine's counter of the adder.

Floating Point Representation

In addition to fixed point arithmetic, the processors are capable of performing floating point arithmetic. If floating point arithmetic is used, the numeric data must be in the specific format shown in figure 2-12.

The first digit (D1) of the field is the sign of the exponent and specifies whether the exponent is positive or negative. The actual value of the exponent is contained in the second and third digit (D2 and D3) positions of the field. The first three digits allow the exponent value to range from -99 to +99, a very large range. The fourth digit (D4) always indicates the sign of the mantissa. The mantissa is variable in length from 1 to 100 and is contained in positions D5 through Dn, as required. The mantissa

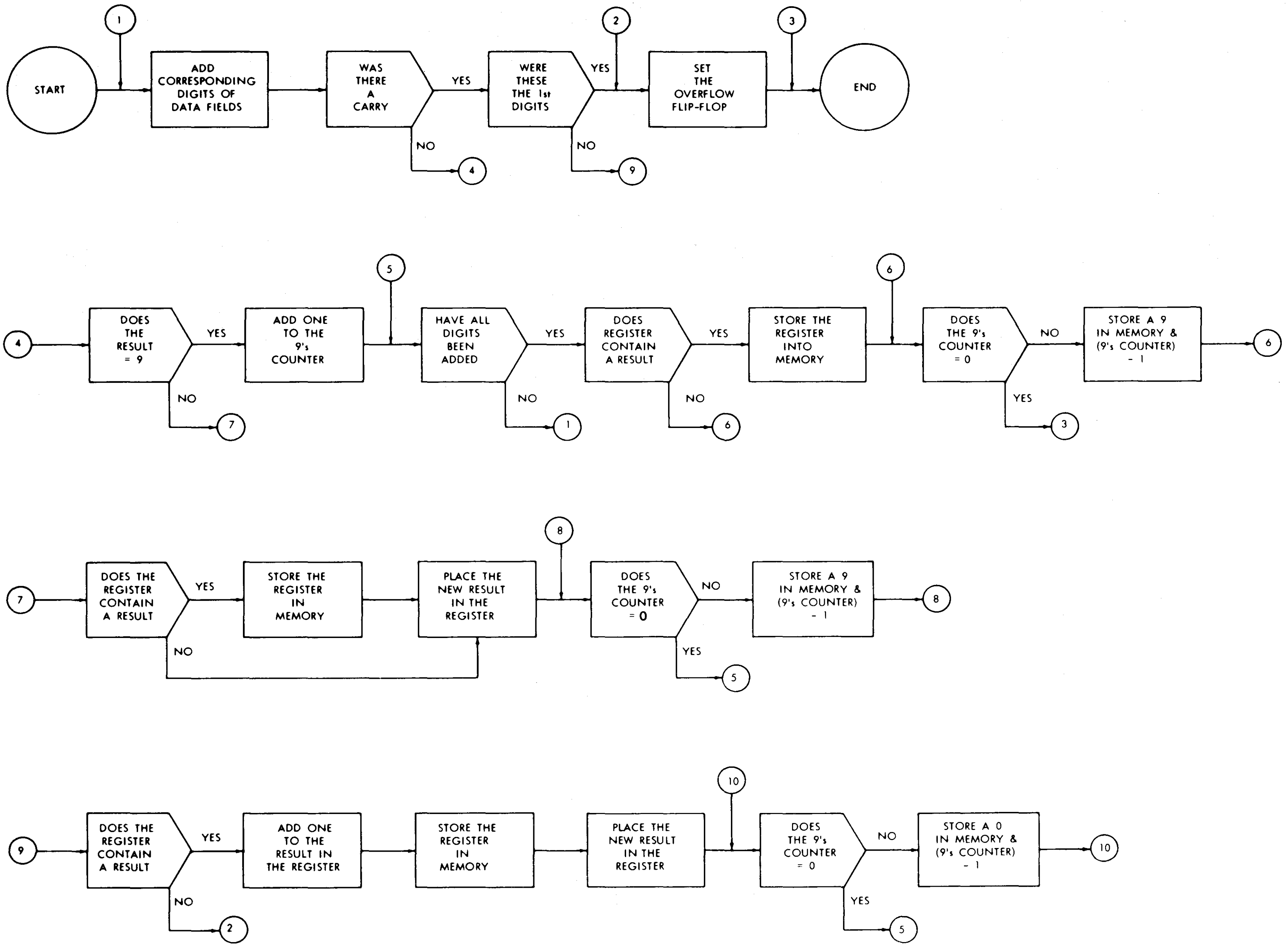
1
START
ADD CORRESPONDING DIGITS OF DATA FIELDS
WAS THERE A CARRY
YES
NO
4
WERE THESE THE 1st DIGITS
YES
NO
9
2
SET THE OVERFLOW FLIP-FLOP
3
END
4
DOES THE RESULT = 9
YES
NO
7
ADD ONE TO THE 9's COUNTER
5
HAVE ALL DIGITS BEEN ADDED
YES
NO
1
DOES REGISTER CONTAIN A RESULT
YES
NO
6
STORE THE REGISTER INTO MEMORY
6
DOES THE 9's COUNTER = 0
NO
YES
3
STORE A 9 IN MEMORY & (9's COUNTER) - 1
6
7
DOES THE REGISTER CONTAIN A RESULT
YES
NO
STORE THE REGISTER IN MEMORY
PLACE THE NEW RESULT IN THE REGISTER
8
DOES THE 9's COUNTER = 0
NO
YES
5
STORE A 9 IN MEMORY & (9's COUNTER) - 1
8
9
DOES THE REGISTER CONTAIN A RESULT
YES
NO
2
ADD ONE TO THE RESULT IN THE REGISTER
STORE THE REGISTER IN MEMORY
PLACE THE NEW RESULT IN THE REGISTER
10
DOES THE 9's COUNTER = 0
NO
YES
5
STORE A 0 IN MEMORY & (9's COUNTER) - 1
10

Figure 2-10. Adder Functional Flow Chart

is always a whole number with the decimal point assumed to the right of the last digit. Several examples of floating point data are shown in table 2-2.

```
     EXAMPLE 1:

                   12343
                   12343
         RESULT    24686

     EXAMPLE 2:

                   12345
                   92345
      RESULT 1:    0 + CARRY
  FINAL RESULT:    OVERFLOW FLIP-FLOP SET

     EXAMPLE 3:

                   876543229
                   123455779
      RESULT 1:    999998998 + CARRY
  FINAL RESULT:    999999008

     EXAMPLE 4:

                   0876543
                   0123457
      RESULT 1:    0999990 + CARRY
  FINAL RESULT:    1000000

     EXAMPLE 5:

                   876543
                   123457
      RESULT 1:    999990 + CARRY
  FINAL RESULT:    OVERFLOW FLIP-FLOP SET
```

Figure 2-11. Adder Examples

Table 2-2. Floating Point Representation

Value	Floating Point Data
+123	+00+123
-1.23	-02-123
+0.0057	-04+57
-9.786 x 10^{-6}	-09-9786
+3.75 x 10^{6}	+04+375

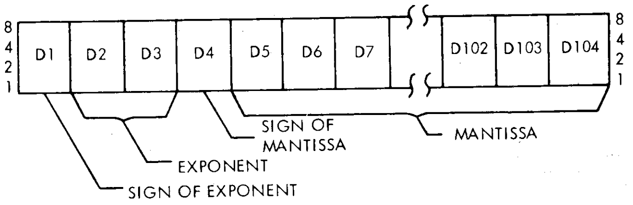

Figure 2-12. Floating Point Data Format

LOAD FUNCTION

Two types of load commands are available. The first type, called Universal Load, permits the operator's selection of the input media; the second type, called Normal Load, restricts the selection of the input media to that of a particular peripheral unit (normally systems memory or disk). The two types of load commands are discussed in the following paragraphs.

Normal Load

The Normal Load operation is initiated by clearing the computer by pressing the CL (clear) key and the LD (load) key. Pressing the LD key writes a 2-digit channel number and the first syllable of an I/O descriptor into reserved memory starting at address 000000. The channel number and I/O descriptor are predesignated by the user and are field engineering modifications. To retrieve a new copy of the MCP by the quickest means available requires that the LD key be wired to call on the appropriate disk channel on which the resident MCP is stored. After writing the channel number and I/O descriptor into core memory, the processor will initiate an I/O operation using the information in address 000000. The hardware automatically sets the beginning address to 001000 and the ending address to 001400. If the I/O unit is systems memory or a disk file, the segment address is set to zero. The processor idles until the interrupt flip-flop is set by the I/O control unit at the completion of the I/O operation. The processor then automatically branches to absolute address 001000 and begins executing the program that was just loaded. If an exception condition occurs, the Load operation must be repeated by the systems operator. It must be noted that if the I/O unit is designated as systems memory or disk, a previously cold-started MCP or MCP loader operation must have been accomplished after a power-off condition.

Universal Load

The Universal Load operation is initiated by the operator in the following manner:

a. Press the STOP/RUN key to stop the system.

b. Press the CL (clear) key to clear the system.

c. Press the AD key and then the WR key.

d. Starting at absolute address 000000, enter the two digits for the channel number and the six digits (first syllable) of the I/O descriptor. This allows the operator to select any channel and the input peripheral unit which contains the cold-start or MCP loader deck (or images).

e. Press the OP key and enter OP Code 66 into the OP register by typing 660000 on the keyboard.

f. Press the STOP/RUN key to start the system.

When the STOP/RUN key is pressed at this point, the processor executes a Load operation that selects the control program from a "SYSTEM" magnetic tape and loads it. The processor will now go into a pseudo initiate I/O cycle in which the channel number and the first syllable of an I/O descriptor are obtained from core memory starting at address 000000. The beginning and ending addresses for the I/O descriptor are set to 001000 and 001400, respectively and the segment number, when systems memory or a disk file is used, is set to zero. Having completed the initiate I/O cycle, the processor idles until the interrupt flip-flop is set, signifying completion of the I/O operation. When the interrupt flip-flop is set, the processor transfers the absolute address of the pertinent result descriptor to IX1 and clears the result descriptor area to zero after transferring the two most significant bits of the result descriptor to the comparison flip-flops. If the comparison is high (operation complete and no exceptions), the processor executes a Move Alphanumeric instruction with a field length of 100 for both the A and B fields. The address controllers for the A and B fields are 8-bit format and unsigned 4-bit format, respectively. The A and B addresses are both 001000. This move strips off the zone bits (most significant digit of a character) and compresses the field to 100 4-bit digits. A branch is taken to location 001000.

If the comparison is equal, indicating that the peripheral control returned an exception bit, operation contained in the above paragraph is automatically retried.

An invalid descriptor causes the processor to halt and the comparison LOW indicator is set.

CONSOLE

The operator console shown in figure 2-13 consists of two panels: a display panel and a control panel. The display panel is the vertical upper panel; the control panel is the curved lower panel that is placed on a slope with the display panel.

Display Panel

The display panel contains a NIXIE ® tube display and back-lighted fixed message displays. The display panel can be divided into three areas as follows:

a. A NIXIE ® tube display consisting of two groups of tubes representing six contiguous digits of memory data or address information. Each group is appropriately identified

®Registered Burroughs Trademark

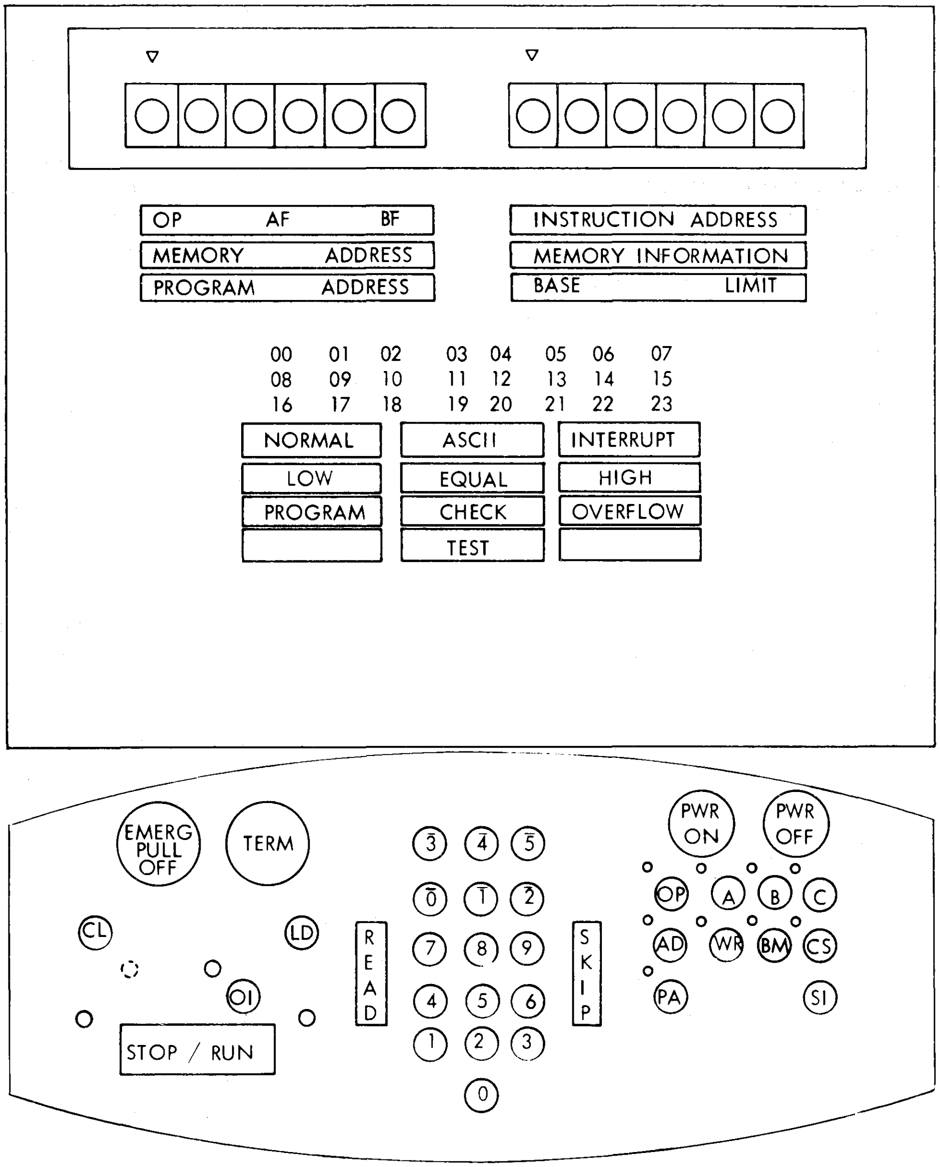

Figure 2-13. Operator's Console

by one of three legends which are illuminated directly below the tube display when the system is operating in that particular mode. Each tube position is capable of displaying all 16 digits of the extended binary set. Undigits will be displayed as their decimal value, less ten, with a slash through the digit. A halt, either programmed or manual, displays the legend OP AF BF and INSTRUCTION ADDRESS. An 8-digit instruction (Address Branch, Exit, Halt Branch) displays its OP Code on the left side of the display panel and the instruction address on the right side.

When indirect field length is specified in the AF or BF fields, the display in the AF or BF positions is the resultant field length and not the bit combination specifying indirect field and its address. When a literal is specified in the AF field, the bit combination specifying the literal operand and the bit combination specifying the controller are cleared to zero before display. Subsequent use of the A key on the control panel displays the address of the operand and not the literal. The left group of six NIXIE ® tubes on the display panel reflects:

(1) OP AF BF (first interruption syllable).

(2) MEMORY ADDRESS.

(3) PROGRAM ADDRESS.

The right group of six NIXIE ® tubes reflects:

(4) INSTRUCTION ADDRESS.

(5) MEMORY INFORMATION.

(6) BASE and LIMIT (registers).

b. Twenty back-lighted channel indicators (00-19) that illuminate when an appropriate channel is busy, and are used for testing purposes. Indicators 20-23 are used by the Maintenance Test Routine (MTR).

c. Back-lighted, fixed-message indicators which reflect the state of the logical flip-flops. Control state operation is indicated by the NORMAL indicator not being lit. The PROGRAM indicator lights when an invalid operation or an invalid address is encountered. The PROGRAM and CHECK indicators light when an instruction time out occurs. The CHECK indicator lights if a parity interrupt occurs. The INTERRUPT indicator lights if a logical interrupt is not completed. A complete list of the indicators is as follows:

(1) NORMAL.

(2) LOW.

(3) PROGRAM.

(4) ASCII.

(5) EQUAL.

(6) CHECK.

(7) TEST.

(8) INTERRUPT.

(9) HIGH.

(10) OVERFLOW.

Control Panel

The control panel consists of keys which allow the operator to enter information into registers or memory, keys to control the display of information, a key to start and stop the processor, one key for emergency power off, one key for terminating operation, and keys for power on and off. The uses of these keys are as follows:

CL (CLEAR) KEY

When this key is pressed, all of the flip-flops in the processor, I/O controls, central control, and memory control are reset to the cleared state and the limit register is set to the system memory size. This key is only active when all I/O operations are completed and the processor is halted.

LD (LOAD) KEY

After the system has been cleared by the CL key, pressing this key initiates the normal load operation. Loading of the Operating System is performed on the I/O channel that has been predesignated (normally the systems memory or a disk unit) by the field engineer.

STOP/RUN KEY

The STOP/RUN key serves two functions. If the processor is running, pressing the key causes the processor to stop. If the processor is stopped, pressing the key causes the processor to continue execution of the stopped program. A stop will only occur at the end of the instruction being executed; the keyboard is inactive until all of the I/O operations are completed. The status of processor operations is indicated by the stop or run indicators located near the STOP/RUN key.

SI (SINGLE INSTRUCTION) KEY

This is a single step key which is active when the processor is in a stopped status. Pressing this key causes the current instruction displayed in the NIXIE ® tube display to be executed and the next to be fetched and displayed in the NIXIE ® tube display. If the current instruction is a Halt instruction, only a fetch and display of the next instruction occurs.

OI (OPERATOR INTERRUPT) KEY

The system may be interrupted temporarily by the operator pressing this switch.

OP KEY

This key is active in the stopped status only. Pressing this key causes the OP AF BF and address of the current instruction to be displayed in the left and right groups of NIXIE ® tubes, respectively. Digits can then be entered, right to left, into the OP AF BF register from the keyboard. The entry of the first digit blanks the left-most NIXIE ® tubes and the first digit will be displayed. The remaining digits will be displayed as they are entered from the keyboard.

A, B AND C (FIELD ADDRESS) KEYS

These three keys all essentially perform the same function as the OP key and are only active when the processor is in a stopped status. Pressing any one of the three keys causes the OP AF BF of the current instruction to be displayed in the left group of NIXIE ® tubes. The right group of NIXIE ® tubes will contain the A, B, or C field address of the instruction, depending on whether the A, B and C key was pressed.

PA (PROGRAM ADDRESS) KEY

This key is only active in the stopped status. Pressing the key causes the program address of the next instruction to be displayed in the left group of NIXIE ® tubes. The three-digit base register and three-digit limit register values will be displayed in the right group of NIXIE® tubes. The program address can be changed by entering digits from the keyboard. Entry of the first digit blanks the display and the first digit of the new program address will be displayed. A limit of six digits can be entered, at which time the keyboard becomes inactive.

AD (ADDRESS) KEY

This key is only active in the stopped status. When the key is pressed, the memory address register is displayed in the left group of NIXIE® tubes and the information contained at that address is displayed in the right group of NIXIE® tubes. A limit of six digits can be entered from the keyboard to change the memory address. Entry of the first address digit blanks the remaining display and will move from right to left. The entered digits are shifted with newly entered digits following, until the register is full.

WR (WRITE) KEY

This key can only be used after pressing the AD key and is used to write new information into core memory, a digit at a time, via the keyboard. The memory address register advances one digit position for each entry. The digit currently contained at the memory address appears left-justified in the right group of NIXIE® tubes, and the keyboard entry digit is right-justified. It is not necessary to change a digit, it can be bypassed by means of the SKIP key. Once the WR key is pressed, the write function is active until one of the other keys is pressed (other than a digit or SKIP key).

SKIP KEY

This key is only active after the WR key has placed the keyboard into a write function status. Pressing this key causes the memory address register to increase by one digit position. It is used to skip over digit positions that are not to be changed in core memory.

READ KEY

This key is active only after the AD key has been pressed. When displaying information from core memory, this key causes the next digit (or groups of digits) to be read from memory and displayed in the right group of NIXIE ® tubes. The memory address is increased by one if the address is not modulo-4. Once the address is modulo-4, words will be read out and the address will increment by four.

BM (BASE MEMORY) KEY

When this key is pressed and used in conjunction with other appropriate keys, the memory address, program address, or instruction address is displayed base relative instead of its absolute systems address.

CS (CONTROL STATE) KEY

This key is used in conjunction with the SI (single instruction) key. In normal state, the SI key causes execution of a single instruction. If the execution of that single instruction causes a return to control state when this key is active, all control state instructions are executed continuously and the processor halts only when normal state is reinstated. Thus, the execution of a Branch Communicate instruction consists of the execution of all control state instructions associated with the Branch Communicate.

TERM (TERMINATE) KEY

This key performs the same function as the CL (clear) key except that this key is always active and provides an absolute means of halting the processor which otherwise may not be able to be halted because of undefined hardware or program difficulties. The halt is immediate and all displays are blank.

DIGIT KEYS

A total of sixteen digit keys on the keyboard are available for manually entering information into the system. The ten keys, labeled 0 through 9, represent the 4-bit binary values of decimal digits $\underline{0}$ through $\underline{9}$, whereas the other six keys, labeled $\bar{0}$ through $\bar{5}$, represent the 4-bit binary values of 10 through 15 for entering the upper 4-bit stack of 8-bit character combinations. The transfer of this information is controlled by the previous pressing of an appropriate control key. The keyboard is active in the stop state and inactive in the run state.

EMERG PULL OFF (EMERGENCY POWER) KEY

This switch provides a signal to an external device that causes power to be removed from all of the units. This signal is not distributed to the system and no system unit has the capability to respond to this signal. It is provided for emergency conditions only and must not be used for normal power shutdown operations.

PWR ON KEY

This key is used to apply power to the system.

PWR OFF KEY

This key is used to remove power from the system.

SECTION 3
SYSTEM POWER CONFIGURATION

GENERAL

In general, the power supplies are modular and each cabinet contains a major portion of the power required to service that cabinet. Figure 3-1 shows a three cabinet system and the basic interfacing between cabinets. The processor supplies a +160 volt DC source voltage to the other two cabinets. The sequence controls and three control voltages, -10 volts, +10 volts, and +19 volts are located in the processor. A special DC supply, which may be located in any cabinet except the processor, supplies special DC voltages to the other two cabinets. Each cabinet contains voltage sensing circuits which are returned to the sequence control in the processor to cycle power off if an over or undervoltage occurs, or if a supply voltage did not come up in the allotted time during a power on cycle.

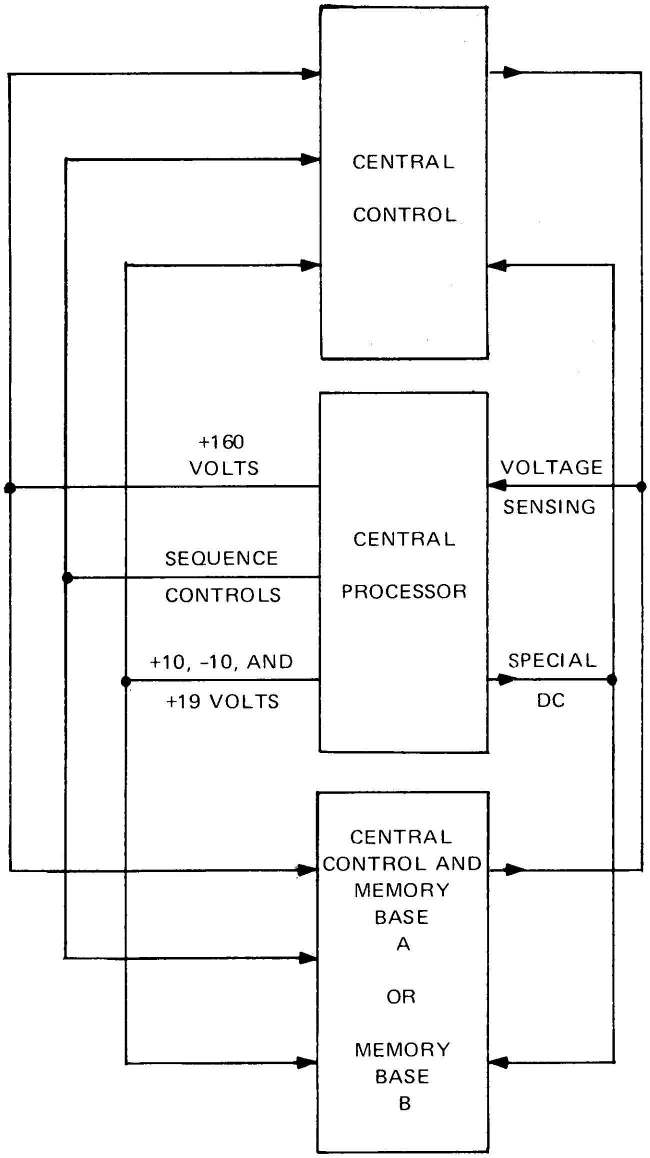

Figure 3-1. General Power Interface Block Diagram

The following units comprise the system power supplies:

a. Sequence Control.

b. AC Control.

c. +170 Volt Supply.

d. AC to DC Converter.

e. Inverter.

f. Special DC Supply.

g. Memory Supply.

h. Memory Regulator.

Figure 3-2 shows the cabinet location of the supplies and controls in a typical minimum system configuration as well as the voltage and current capabilities of each supply.

OVERVOLTAGE AND UNDERVOLTAGE SENSING

The overvoltage and undervoltage (OV/UV) sensing circuits are used on all supplies except the +34 volt supply. The +34 volt supply is sensed by the +47 volt regulator. Voltage failures are indicated in the following manner:

a. A lamp within the power supply which has failed is turned on.

b. One of the two lamps on the Processor Maintenance Panel is turned on to indicate whether the failure encountered was due to an OV or UV condition.

System power is cycled down and, before it can be turned on again, the lamps which are illuminated must be reset by pressing the PWR OFF pushbutton.

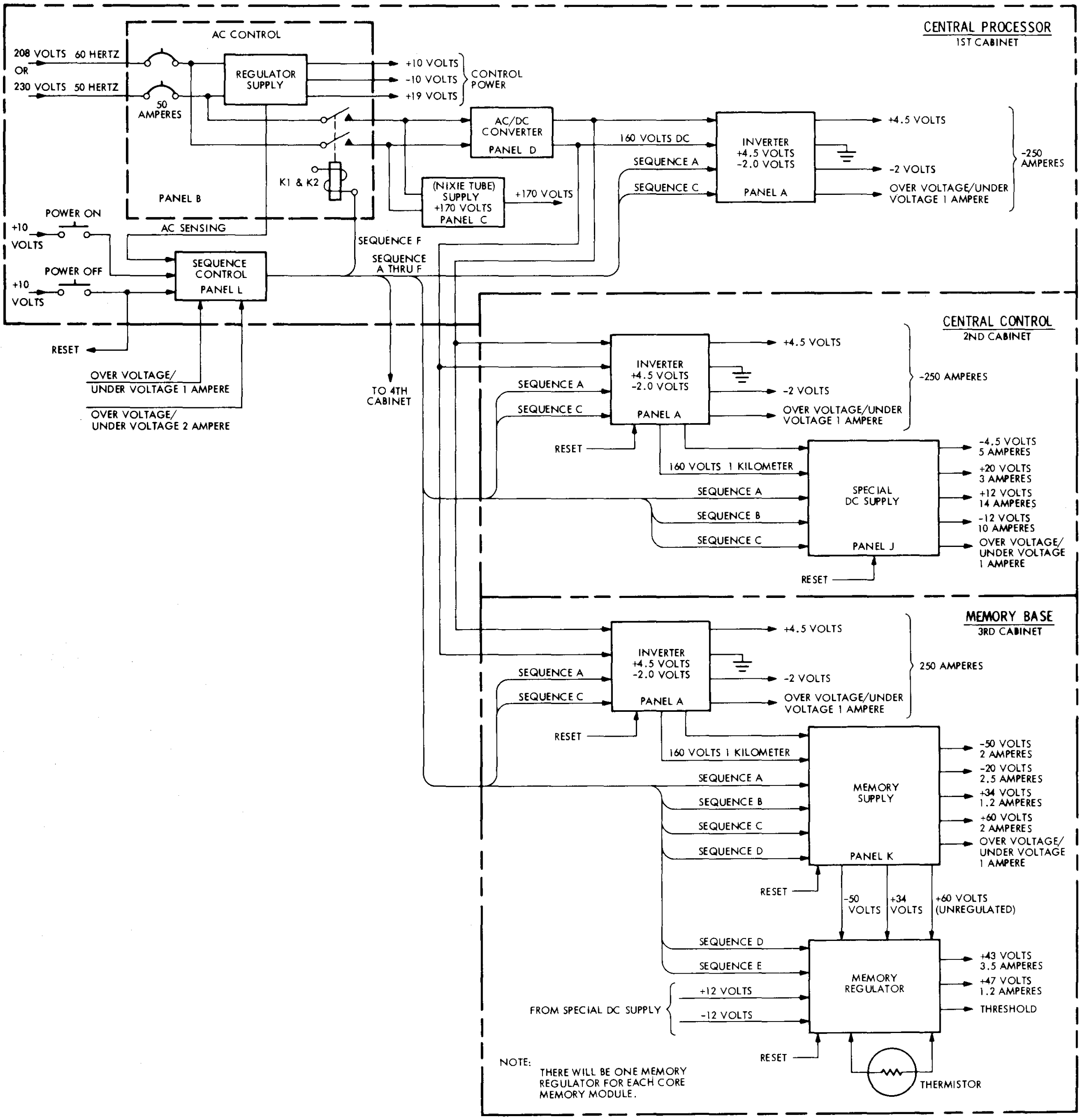

Figure 3-2. Power Distribution Layout

cut along dotted line

BURROUGHS CORPORATION
DATA PROCESSING PUBLICATIONS
REMARKS FORM

TITLE: MEDIUM SYSTEMS
B 2500/B 3500/B 2700
CENTRAL PROCESSOR and MEMORY
Reference Manual

FORM: 1063781
DATE: 12-72

CHECK TYPE OF SUGGESTION:

□ADDITION □DELETION □REVISION □ERROR

GENERAL COMMENTS AND/OR SUGGESTIONS FOR IMPROVEMENT OF PUBLICATION:

FROM: NAME ____________________ DATE ____________
TITLE ____________________
COMPANY ____________________
ADDRESS ____________________
____

STAPLE

FOLD DOWN SECOND FOLD DOWN

Postage Will Be Paid by Addressee

No Postage Stamp Necessary If Mailed in the United States

BUSINESS REPLY MAIL
First Class Permit No. 817, Detroit, Mich. 48232

Burroughs Corporation
6071 Second Avenue
Detroit, Michigan 48232

attn: Sales Technical Services
Systems Documentation

FOLD UP FIRST FOLD UP