SCIENTIFIC DATA SYSTEMS

Reference Manual

SDS 9300 Computer

SDS 9300 BASIC INSTRUCTIONS
(Central Processor)

Mnemonic		Code	Name	Page
LOAD/STORE				
LDA	M, T	16	Load A	2-1
STA	M, T	76	Store A	2-2
LDB	M, T	14	Load B	2-2
STB	M, T	74	Store B	2-2
LDP	M, T	26	Load Double Precision	2-2
STD	M, T	75	Store Double Precision	2-2
LDS	M, T	06	Load Selective	2-2
STS	M, T	70	Store Selective	2-2
LDX	M, T	X-17	Load Index X	2-2
STX	M, T	X-77	Store Index X	2-3
EAX	M, T	15	Copy Effective Address into Index Register 1	2-3
STZ	M	0-77	Store Zero	2-3
XMA	M, T	36	Exchange Memory and A	2-3
XMB	M, T	34	Exchange Memory and B	2-3
XMX	M, T	X-37	Exchange Memory and Index Register	2-3
ARITHMETIC, BINARY				
ADD	M, T	05	Add	2-3
DPA	M, T	25	Double Precision Add	2-4
SUB	M, T	04	Subtract	2-4
DPS	M, T	24	Double Precision Subtract	2-4
MPO	M, T	71	Memory Plus One	2-4
MPT	M, T	72	Memory Plus Two	2-4
ADM	M, T	35	Add to Memory	2-4
MUL	M, T	63	Multiply	2-5
DIV	M, T	62	Divide	2-5
TMU	M, T	61	Twin Multiply	2-5
DPN	M, T	27	Double Precision Negate	2-5
ARITHMETIC, FLOATING POINT (OPTIONAL)				
FLA	M, T	65	Floating Add	2-6
FLS	M, T	64	Floating Subtract	2-7
FLM	M, T	67	Floating Multiply	2-7
FLD	M, T	66	Floating Divide	2-7
LOGICAL				
ETR	M, T	11	Extract	2-7
MRG	M, T	13	Merge	2-7
EOR	M, T	12	Exclusive OR	2-8
REGISTER CHANGE				
RCH	M, T	40	Register Change	2-9
AXB	*M, T	4X-40	Address to Index Base	2-14
COPY		40	Copy	2-11
BRANCH				
BRU	M, T	01	Branch Unconditionally	2-14
BRX	M, T	X-57	Increase Index and Branch	2-14
BRC	M, T	0-57	Branch and Clear Interrupt	2-14
BRM	M, T	03	Mark Place and Branch	2-15
BMA	M, T	43	Branch and Mark Place or Argument Address	2-15
BRR	M, T	41	Return Branch	2-15
TEST/SKIP				
SKE	M, T	45	Skip if A Equals Memory	2-16
SKG	M, T	46	Skip if A Greater than Memory	2-16
SKL	M, T	44	Skip if A Less than Memory	2-16
SKM	M, T	55	Skip if A Equals M on B Mask	2-16
SKU	M, T	47	Skip if A Unequal to Memory	2-16
SKQ	M, T	56	Skip if Masked Quantity in A Greater than M	2-16
SKF	M, T	50	Skip if Floating Exponent in B is Greater than or Equal to Exponent in Memory	2-17
SKA	M, T	54	Skip if A and Memory do not Compare Ones Anyplace	2-17
SKB	M, T	52	Skip if B and Memory do Compare Ones Anyplace	2-17
SKN	M, T	53	Skip if Memory is Negative	2-18
SKR	M, T	73	Reduce M, Skip if Negative	2-18
SKP	M, T	51	Skip if Bit Sum Even	2-18
SKS	M, T	20	Skip if Signal Not Set	4-6
REGISTER SHIFT				
SHIFT	M, T	60	Shift	2-19
ARSA	N, T	60-20	Arithmetic Right Shift A	2-20
ARSB	N, T	60-10	Arithmetic Right Shift B	2-20
ARSD	N, T	60-00	Arithmetic Right Shift Double	2-20
ARST	N, T	60-30	Arithmetic Right Shift Twin	2-20
LRSA	N, T	60-21	Logical Right Shift A	2-20
LRSB	N, T	60-11	Logical Right Shift B	2-21
LRSD	N, T	60-01	Logical Right Shift Double	2-21
LRST	N, T	60-31	Logical Right Shift Twin	2-21
CRSA	N, T	60-22	Circular Right Shift A	2-21
CRSB	N, T	60-12	Circular Right Shift B	2-21
CRSD	N, T	60-02	Circular Right Shift Double	2-21
CRST	N, T	60-32	Circular Right Shift Twin	2-21
ALSA	N, T	60-24	Arithmetic Left Shift A	2-22
ALSB	N, T	60-14	Arithmetic Left Shift B	2-22
ALSD	N, T	60-04	Arithmetic Left Shift Double	2-22
ALST	N, T	60-34	Arithmetic Left Shift Twin	2-22
LLSA	N, T	60-25	Logical Left Shift A	2-22
LLSB	N, T	60-15	Logical Left Shift B	2-22
LLSD	N, T	60-05	Logical Left Shift Double	2-22
LLST	N, T	60-35	Logical Left Shift Twin	2-23
CLSA	N, T	60-26	Circular Left Shift A	2-23
CLSB	N, T	60-16	Circular Left Shift B	2-23
CLSD	N, T	60-06	Circular Left Shift Double	2-23
CLST	N, T	60-36	Circular Left Shift Twin	2-23
NORA	N, T	60-64	Normalize A	2-24
NORD	N, T	60-44	Normalize Double	2-24
CONTROL				
HLT		00	Halt	2-24
NOP	M, T	10	No Operation	2-24
EXU	M, T	21	Execute	2-24
INT	M, T	07	Interpret	2-25
REP	M, T	23	Repeat	2-25
FLAG REGISTER				
FLAG	M	22	Flag	2-27
FIRS	M	22-0	Flag Indicator Reset/Set	2-27
FSTR	M	22-1	Flag Indicator Set Test/Reset	2-27
FRTS	M	22-2	Flag Indicator Reset Test/Set	2-28
FRST	M	22-3	Flag Indicator Reset/Set Test	2-28
SWT	M	22-4	SENSE Switch Test	2-28
INTERRUPTS				
EIR		0 02 20002	Enable Interrupts	3-4
DIR		0 02 20004	Disable Interrupts	3-4
IET		0 20 20004	Interrupt Enabled Test	3-4
IDT		0 20 20002	Interrupt Disabled Test	3-4
AIR		0 02 20020	Arm Interrupts	3-5

Legend: M = address field; *M = indirect address; T = tag field; N = number of shifts.

Price: $5.25

SDS 9300 COMPUTER REFERENCE MANUAL

90 00 50G

July 1969

SCIENTIFIC DATA SYSTEMS A XEROX COMPANY/701 South Aviation Boulevard/El Segundo, California 90245

Printed in U.S.A.

REVISIONS

This publication, 90 00 50G, dated July, 1969, is a minor revision of the SDS 9300 Computer Reference Manual, 90 00 50F.

The corrections to the previous edition are indicated by a vertical line in the margin of each affected page.

RELATED PUBLICATIONS

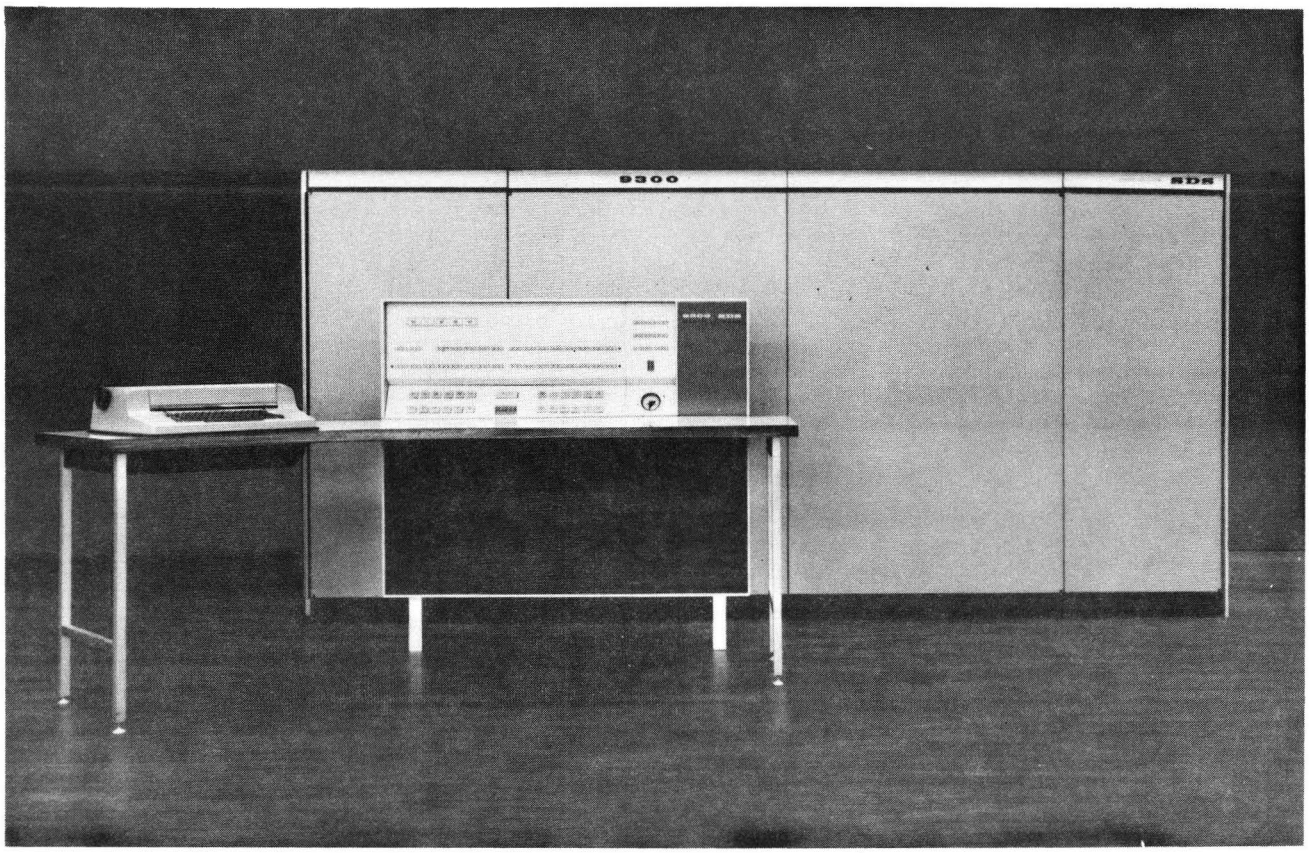
Title of Manual	Publication No.
SDS ALGOL 60 Reference	90 06 99
SDS FORTRAN IV Reference	90 08 49
SDS FORTRAN IV Operations	90 08 82
SDS SYMBOL and META-SYMBOL Reference	90 05 06
SDS 9300 MONITOR Reference	90 05 13
SDS Real-Time MONITOR Reference	90 11 08
SDS MANAGE Reference	90 10 46
SDS SORT/MERGE Reference	90 09 97
SDS Business Language Reference	90 10 22
SDS 9300 Computer Examiner Diagnostic System Technical	90 06 24

CONTENTS

APPENDIXES

FIGURES

TABLES



SDS 9300 Computer

1. GENERAL DESCRIPTION

INTRODUCTION

The SDS 9300, Figure 1-1, is a high-speed, general-purpose, digital computer designed for rapid, scientific computation and sophisticated, real-time control. SDS 9300 characteristics are:

- 24-bit word plus parity bit
- 48-bit word for floating-point arithmetic
- 3 index registers and indirect addressing with unlimited address cascading
- Basic 4096-word core memory, expandable to 32,768 words, 8192 and 16,384-word memory modules available
- Memory is nonvolatile in event of power failure (optional power failure feature permits saving contents of programmable registers); each memory module is functionally independent and directly addressable, with:

  0.7 microsecond time
  1.75 microseconds cycle time
- FORTRAN IV Compiler, META-SYMBOL Symbolic Assembler, and a monitor system as part of a complete software package
- Extensive repertoire of more than 110 instructions
- Execution times, including all accesses and indexing (using overlapped memories), in microseconds

  Fixed-Point

Add	1.75
Double Precision Add	3.5
Multiply	7.0
Shift (24 positions)	5.25

  Floating-Point (39-bit fraction, 9-bit exponent)

Add	14.0
Multiply	12.25

- Byte operations permit manipulation of 3-, 6-, 9-, 12-, 15-, 18-, or 21-bit bytes. Byte multiply instruction multiplies two 12-bit bytes in 7.0 microseconds.
- 6-bit flag register with set/reset/test instructions provides fast, easy-to-use program switches for logical decision making. Each position may be set or reset under program control and the status may be used to control program flow.
- 15 high-speed search operations operate at 1.75 microseconds per item.
- Multilevel indexing and indirect addressing, index registers contain a base and increment or decrement
- Repeat instruction operating with variable size increments or decrements
- Extensive shift and inter-register instructions for data manipulation
- Flexible and easily programmed subroutine execution
- One to four I/O communication channels (with interlacing capability), time-multiplexed with computer operation, providing input/output rates of over 288,000 words per second. Time-multiplexed input/output channels operate upon either words or characters. A 6-bit character is the standard character size; 6- and 12-bit characters, or 6-, 12-, and 24-bit characters can be specified as desired.
- A direct memory access system that allows input/output transfer to occur simultaneously with computer memory access, providing input/output rates of over 570,000 words per second
- Up to four direct access communication channels that incorporate the direct memory access system. Direct access channels operate upon words and characters. These channels accept 6-, 8-, 12-, and 24-bit characters. The number of characters per word is specified by the external device.
- Data multiplex channel that uses direct memory access connection and accepts/transmits information from external devices, or subchannels, which may operate simultaneously; thus, externally controlled and sequenced equipment may perform input/output buffering and control operations rather than the computer.
- Parity checking of memory input/output operations
- Input/output with scatter-write and gather-read
- Searching of magnetic tapes and discs can be accomplished independently of other computer operations, thereby requiring no computer time.
- A parallel word input/output system in addition to the channels to facilitate operating asynchronously on certain types of information under program control.
- Parallel word input/output in conjunction with repeat instruction yields high-speed input/output with a transfer rate of over 570,000 words per second.
- Up to 32,000 output control and input test signals
- Priority Interrupt System

  SDS optional hardware interrupts, 4 levels standard, 28 optional

  Special system interrupts, up to 992 optional levels
- Automatic program loading from cards, paper tape, or magnetic tape

- Standard input/output

  Time-multiplexed communication channel A
  Control console
  Automatic typewriter

- Complete display of all programmable registers with extensive manual controls
- Six SENSE switches, two manual interrupt switches and a selective halt provide complete capability for console control during execution
- No air conditioning required for central processor - Operating temperature range 10°C to 40°C
- All silicon semiconductors
- Power requirement of 8 KVA

- Optional input/output devices

  Photoelectric paper-tape reader and paper-tape punch, and spooler mounted on cart

  MAGPAK magnetic tape system

  Magnetic-tape units (IBM-compatible; binary and BCD), punched-card equipment, line printers, graph plotters

  Typewriter with electromechanical paper-tape reader and punch, auxiliary disc files

  Communications equipment, teletype consoles, display oscilloscopes

  A-D converters, digital multiplexer equipment, and other special system equipment

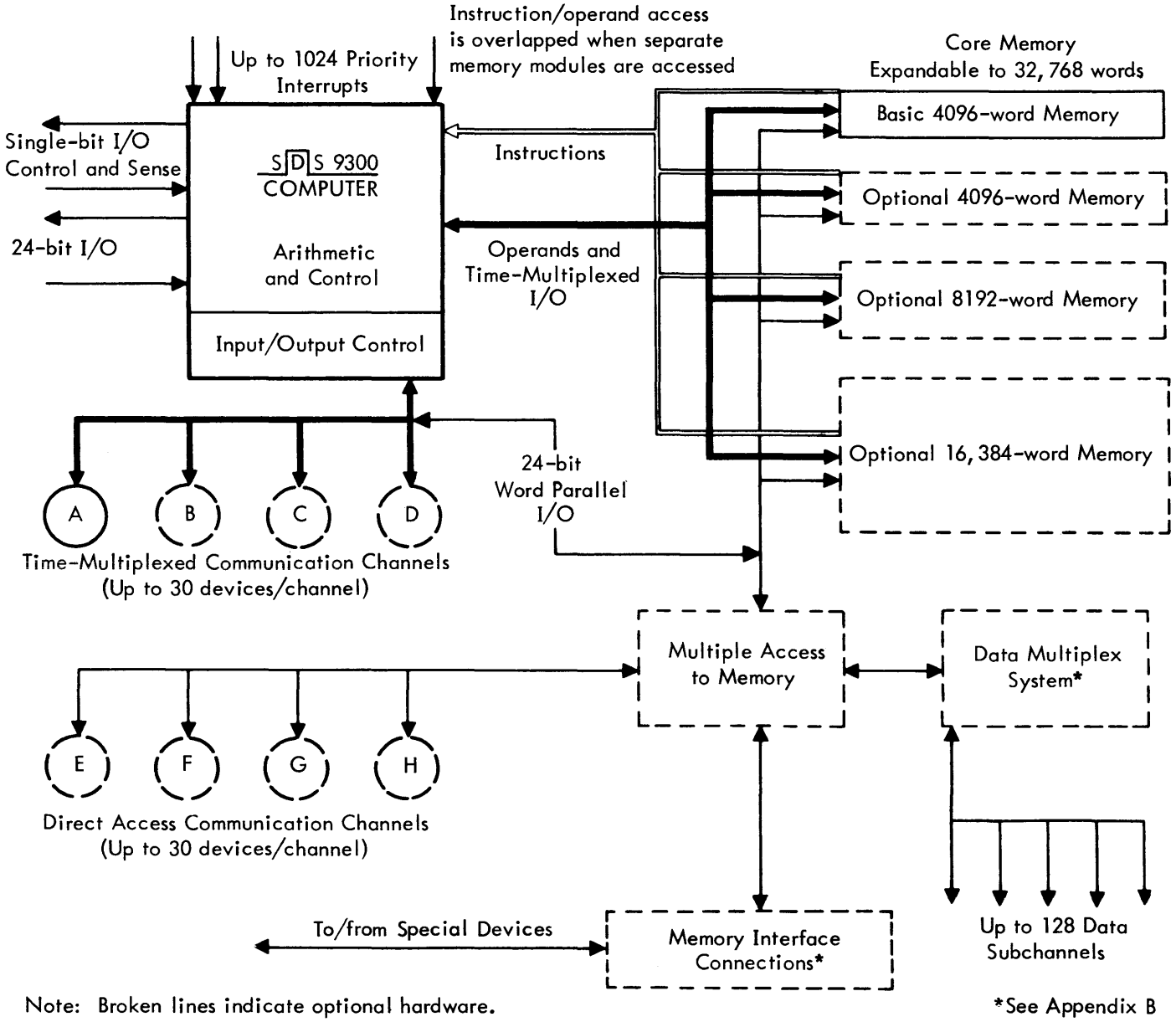


Figure 1-1. SDS 9300 Computer Configuration

CENTRAL PROCESSOR REGISTERS

The SDS 9300 contains 11 arithmetic and control registers and two optional arithmetic registers.

REGISTERS AVAILABLE TO THE PROGRAMMER (See Figure 1-2, dark lines).

The 24-bit accumulator (A register) is the main accumulator of the 9300.

The extended accumulator (B register) is a 24-bit extension of the A register. The B register contains the least significant portion of the double-length numbers.

There are three 24-bit index (X) registers, named X1, X2, and X3, to be used in address modification. Each index register is composed of a base address of 15 bits and a signed increment of 9 bits.

The program counter (P) register is a 15-bit register that contains the memory address of the current instruction. Unless modified by the program, the program counter is increased by one at the completion of each instruction.

The flag (F) register is a 6-bit register that may be set and/or sensed by the program. The first bit position of this register is the OVERFLOW indicator.

REGISTERS NOT AVAILABLE TO THE PROGRAMMER (See Figure 1-2, light lines).

The C register holds the 24-bit operand word as it is transmitted to, or received from, memory.

The D register holds the 24-bit instruction word as it is received from memory.

The 15-bit S register contains the address of the memory location to be accessed for either instruction or operand.

The 6-bit O register contains the instruction code of the instruction being executed.

The A' register is an optional, 15-bit register which temporarily extends the A register during the execution of floating-point instructions.

The B'register is an optional, 15-bit register which temporarily extends the B register during the execution of floating-point instructions.

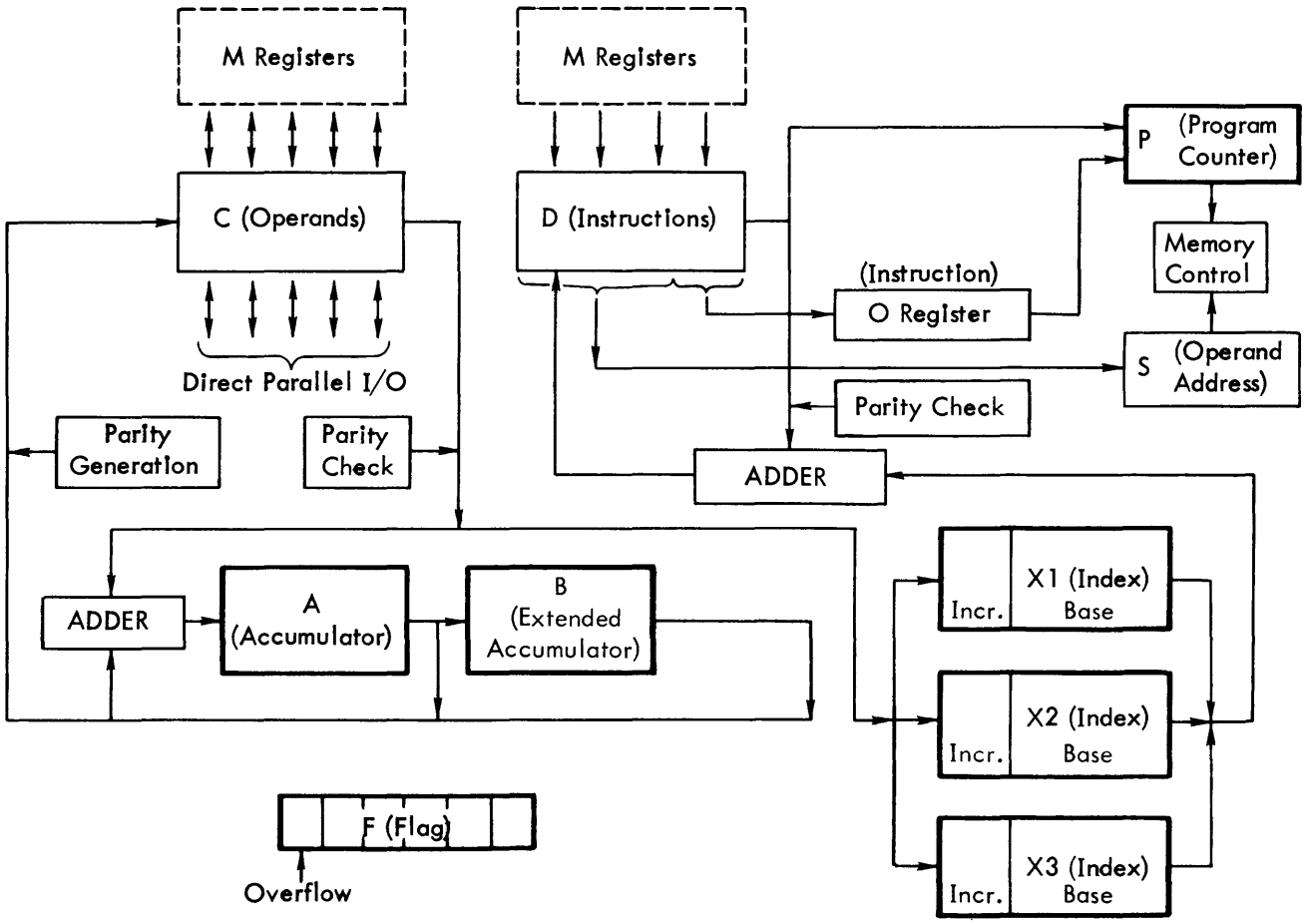


Figure 1-2. Basic Register Flow Diagram

SDS 9300 MEMORY

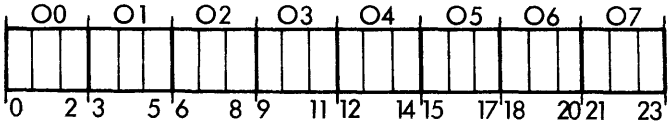
The basic 9300 memory consists of one random access, magnetic core, 4096-word module, with a word size of 24 bits plus parity. Additional 4096-, 8192-, 16,384-word modules are also available. All memory is directly addressable by the 9300 Central Processor and the communication channels. Memory words are addressed in octal notation as locations 00000 through 77777. When a system has 32,768 (32K) words of memory, the memory is a "wrap-around" or circular one where the next location after 77777 is 00000. If a system has less than 32K words, an access from an address larger than available in memory causes zeros to be accessed. An attempt to store into such a location essentially results in a "no-op" operation with the next instruction in sequence being executed.

Each module contains its own memory address register and read/write electronics. Each module operates independently of the other modules. When instructions and operands are held in different modules, memory accesses are automatically overlapped. In other words, if an instruction required two memory cycles for execution when both instruction and operand are in the same module, only one cycle is required if the instruction and data operand are in different modules.

Even parity is automatically generated for words stored in memory. All words are checked for even parity when they are brought from memory.

9300 MEMORY WORD FORMATS

A computer word is 24 binary digits (bits) long.

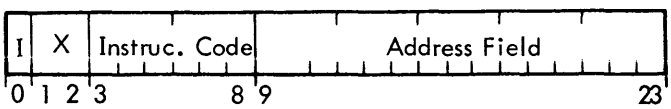


These bits are numbered (as shown above) from the left, or most significant end of the word, to the right, or least significant end of the word. All references to bit positions or bit numbers use this numbering format (e.g., bit 9 refers to bit position 9).

For simplicity of description, computer words are written in octal notation. Since one octal digit represents the absolute value of three binary digits, the 24-bit number, 000 001 010 011 100 101 110 111, is equivalent to the 8-digit octal number 01234567. The octal digits are also numbered (as shown above) and all references to octal positions for octal numbers use this numbering format (e.g., octal O3 refers to bits 9, 10, and 11).

INSTRUCTION WORD FORMAT

The instruction word format in the Central Processor is:



Bit position 0 contains the indirect address bit.

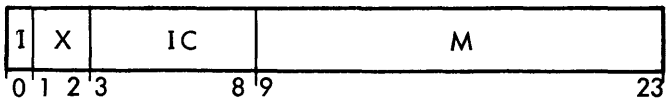
Bit positions 1 and 2 contain the index register bits.

Bit positions 3 through 8 contain the instruction code; the contents of this field determine the operation to be performed.

Bit positions 9 through 23 contain the address; the contents of this field usually represent the memory location of the operand called for by the instruction code.

It is convenient to use octal notation when referring to the "tag", instruction, and address fields. The tag field is a 1-digit octal field consisting of bit position 0, the indirect address bit, and bit positions 1 and 2, the index register bits. The instruction field is a 2-digit octal field consisting of bit positions 3 through 8; and the address field is a 5-digit octal field consisting of bit positions 9 through 23. Therefore, the form of the instruction is 0 00 00000.

In the instruction descriptions, instruction words are written as:



where "I" and "X" are the tag field, as defined previously; "IC" represents the 2-digit octal instruction code;"M" represents a generalized memory address.

For expressing instructions in examples to follow, standard META-SYMBOL assembler format is used. This format is:

```
LOC   LDA   M, T
```

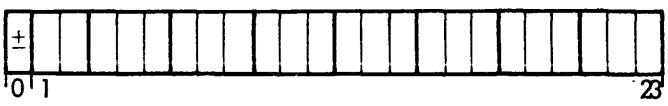
where LOC is the memory location of the instruction written in symbolic or numerical form, LDA is a representative mnemonic instruction code, M is a representative address written in symbolic or numerical notation, and T is a 1-digit octal integer (or a symbolic expression) that represents the index tag field. To express indirect addressing (a 1-bit in the indirect address bit position), an asterisk is prefixed to the address field:

```
LDA   *01000, T
LDA   *M, T
```

See Appendix B for a listing of META-SYMBOL mnemonics and expressions.

FIXED-POINT FORMAT

Fixed-point data words have the format:



These words are written as 8-digit octal numbers, with the sign being incorporated as the "leading bit" in the most significant octal digit.

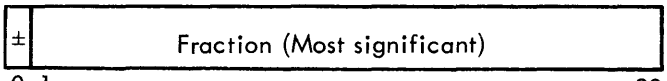
Fixed-point numbers are held in memory as 23-bit fractions with an assumed binary point to the left of bit position one. Numbers held in one word have the equivalent precision of over 6 decimal digits. In double-precision, (where one number is held in two words), numbers of over 13 decimal digits may be represented. The range of values of the fixed-point format is from greater than or equal to minus one to strictly less than plus one ($-1 \leq X < 1$). This inherently implies the use of scaling.

Fixed-point negative numbers are held in memory in two's complement form; they are operated on arithmetically in a two's complement number system. See Appendix A for a discussion of two's complement arithmetic.

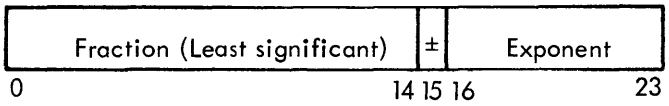
FLOATING-POINT FORMAT

Floating-point data words have the format:

Most Significant Word



Least Significant Word



The fractional portion of a floating-point number is a 39-bit proper fraction, with the leading bit being the sign bit, and with the assumed binary point just to the left of the most significant magnitude bit. The floating-point exponent is a 9-bit integer, with the leading bit being the sign bit. Both fraction and exponent are held in memory and operated on in two's complement form. If F represents the contents of the fractional field and E represents the contents of the exponent field, the number has the form $F \times 2^E$.

Floating-point numbers have over 11 decimal digits of precision and a decimally equivalent exponent range of 10^{-77} to 10^{+77}.

Floating-point numbers are operated on by optional floating-point instructions or by standard subroutines that are automatically called by the use of floating-point operation codes.

SPECIAL CHARACTERISTICS

Certain features in the SDS 9300 simplify programming and provide significant economies in memory and in program running time.

ADDRESS MODIFICATION

In the 9300, address modification is accomplished through indexing and indirect addressing, used singly or in combination. In both, the address is modified after the instruction has been called from memory. The instruction is preserved in memory in its original form.

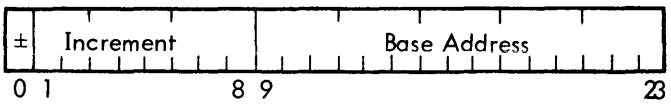
INDEXING

The computer contains three index registers for address modification. The use of an index register to modify the location of an operand does not increase instruction execution time. The relationship between the index registers and the index bits (tag field of the instruction) is:

Index Register	Designation	Index Bits 1 and 2 (Tag field)
1	X1	01
2	X2	10
3	X3	11

If the contents of the index bits in an instruction are nonzero, the contents of bits 9 through 23 of the designated index register are added to the contents of the address field of the instruction prior to execution.

Each index register contains an unsigned base address of 15 magnitude bits and a signed increment of 9 bits. The increment contains 8 magnitude bits and a sign bit and is held in two's complement form.



Index registers are modified by adding the signed increment value to the base address using two's complement arithmetic. Since the increment and base address fields are of unequal lengths, the sign bit (bit 0) of the increment field is extended six positions to the left prior to the addition. The resultant 15-bit sum is then stored in the base address field of the index register. The index register may be incremented by any value from -256_{10} to 255_{10} using a single instruction. Incrementing and testing for a "terminal condition" is done by the instruction, INCREASE INDEX AND BRANCH (BRX), as follows:

> If the index register has been negatively incremented, a terminal condition exists when the base address has been reduced below the zero value.
>
> If the index register has been positively incremented, a terminal condition exists when the resultant base address has been increased beyond the maximum address value (077777).

If the terminal condition exists, the next instruction is taken in sequence. If the terminal condition does not exist, program control is transferred to the location specified.

Index instructions are available to set an index to any value, to transfer the contents of an index to memory, and to set the index base independent of its increment value. Register change instructions are provided for transferring index register values from one index register to another, clearing selected portions of the index register, transferring index register values to and from the A and B registers, and loading the index register with the address of a program-selected location.

INDIRECT ADDRESSING

The indirect address bit is located in bit position zero of the instruction. This bit determines whether indirect addressing will be used with the instruction being executed.

If a zero is placed in the indirect address bit, the contents of the address field, bit positions 9 through 23, of the instruction are used as the 5-digit octal address of the operand called for by the instruction. If an index register is indicated, its contents are added to this address before the operand is accessed.

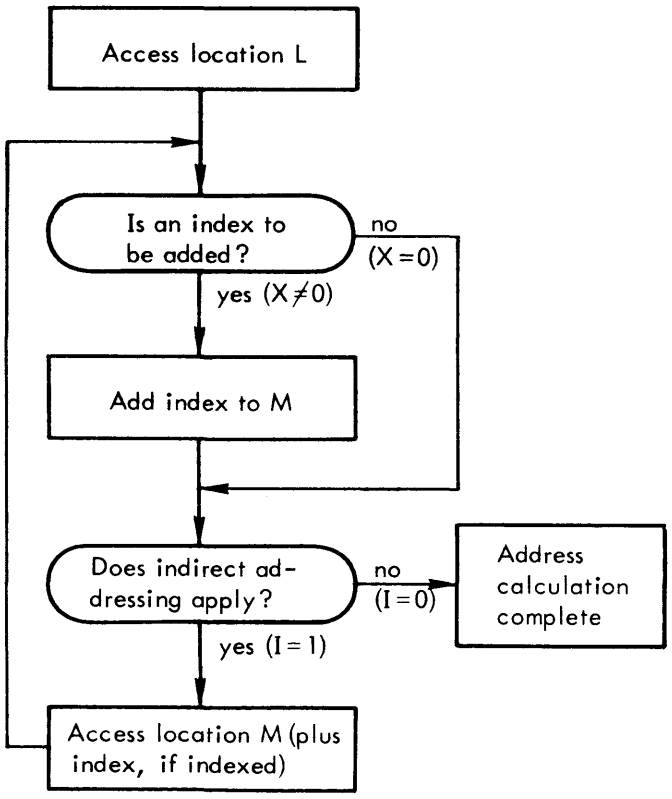


Figure 1-3. Address Calculation

If a one is placed in the indirect address bit, the operand, accessed as described above, is decoded as if it were an instruction without an instruction code.

That is, address decoding is reinitiated on the contents of location M. The process is iterative and multilevel indirect addressing is possible. Instruction execution time is increased one cycle time for each level of addressing. The process is outlined in Figure 1-3. The instruction to be executed is in location L.

The following examples illustrate the use of indexing and indirect addressing. Octal notation is used. The octal instruction code for LOAD A (LDA), used in the example, is 016.

Location	Contents	Effect
X1	00000001	
01000	00001001	
01001	40001002	
01002	00001003	
01003	00000002	
02000	0 16 01000	(01000) = 00001001 → A
02001	1 16 01000	(01000 + 1) = (01001) = 40001002 → A
02002	4 16 01000	((01000)) = (01001) = 40001002 → A
02003	5 16 01000	((01000 + 1)) = ((01001)) = (40001002) = ((01002)) = (01003) = 00000002 → A

SUBROUTINE EXECUTION

Three distinct methods of subroutine execution are provided. These are:

The normal closed subroutine where the input parameters are specified in appropriate registers such as the A register.

The addressed argument, closed subroutine where additional input parameters are found in a table whose location is specified by a memory address, and

The interrupt subroutine.

Normal closed subroutines are entered using a MARK PLACE AND BRANCH (BRM) which automatically stores the contents of the program counter (P) register and the flag register. The stored P register value is the address of the BRM instruction. Return to the main program is accomplished using a RETURN BRANCH (BRR) instruction, which adds one to the stored P register value and transfers control to that memory location. Automatic updating of the flag register, which contains various program indicators as well as the OVERFLOW indicator,

is provided with the execution of the BRR instruction. See Section 2, Branch Group, for a description of the branch instructions.

Interrupt subroutines are closed subroutines that are initiated by the detection of program-controlling interrupts which automatically cause the specified special interrupted subroutine to be entered. (See Interrupt, Section 3.) Interrupt subroutines are entered using a BRM instruction which automatically stores the contents of the P register and the Flag register. The value stored from the P register contains the address of the instruction to which program control should return after the interrupt has been serviced by the interrupt subroutine. A BRANCH AND CLEAR INTERRUPT (BRC) instruction is used to return control to the main program at the completion of the subroutine and simultaneously to inform the interrupt system that the interrupt has been completely processed.

Addressed argument subroutines provide a memory address to identify input parameters to the subroutine without affecting the machine registers. This process is accomplished by entering the subroutine using a BRANCH AND MARK PLACE OF ARGUMENT ADDRESS (BMA) instruction which stores the contents plus one of the P register with an indirect address bit of 1 in the effective memory location. The contents of the flag register are also stored in bits 3 through 8 of that location; the index bits are cleared. The flag register is then cleared. The subroutine, using indirect addressing, can acquire the input parameter address and/or the contents determined by that address. Since the flag register is cleared, no housekeeping of the OVERFLOW indicator or other flag register indicators is necessary. Return to the main program is accomplished using a BRR as in a normal closed subroutine.

Addressed argument subroutine facilities provide a convenient method for adding to the SDS 9300 instruction set using two-word macro instructions.

OVERFLOW

An overflow detector in the computer permits detection of erroneous arithmetic operations which occur during execution of a program. The OVERFLOW indicator (Of) is in bit position zero of the flag register. Of is turned on if any of the following occur:

A sum or difference resulting from an addition, subtraction, or negation that cannot be contained within the A register, or within the A and B registers in a double-precision operation.

Multiplication of N by N where N is 40000000, the largest negative number that can be represented in an SDS 9300 word. This product cannot be contained within the A and B registers.

A division operation where the absolute value of the numerator is equal to, or larger than, the absolute value of the denominator. The quotient cannot be contained within the A register.

An arithmetic left-shift operation which changes the sign bit (bit position 0) of the A register.

The status of Of can be tested and/or reset by the use of flag register control instructions included in the 9300 instruction set.

Overflow indication is cumulative. If Of is turned on, it remains on until turned off by the appropriate instruction.

The status of the flag register, including Of, is automatically preserved when closed or interrupt subroutines are executed. When a MARK PLACE AND BRANCH (BRM) instruction is executed, the contents of the flag register are preserved in bits 3 through 8 of the effective memory location. When a BRANCH AND MARK PLACE OF ARGUMENT ADDRESS (BMA) instruction is executed, the identical storage takes place and the flag register is cleared.

A RETURN BRANCH (BRR) instruction automatically merges the contents of the flag register with the contents of bit positions 3 through 8 of the effective memory and places the result back into the flag register.

These branch instructions are described in detail in Section 2, Branch Group.

REPEAT OPERATION

The instruction, REPEAT INSTRUCTION IN M (REP), provides automatic search capability. Fifteen search operations plus a number of high-speed repetitive operations are automatically performed using this instruction. REP uses index registers X1 and X2 for address and count control. Since the signed increment of index register X1 is used for address incrementing, repetitive operations may be performed on every Nth word in a list where N is greater than, or equal to, -256 and less than, or equal to, +255. Automatic searching is performed by repeating skip class instructions; an exit to the memory location following REP indicates that the condition was not found. Any 9300 instruction that accesses memory may be repeated. Each repeated operation is performed, typically, in one-third the time required for an equivalent program loop.

BYTE AND REGISTER MANIPULATION

A "byte" is defined as a group of bits smaller in length than a machine word. The SDS 9300 provides the programmer with the facility to manipulate bytes of 3, 6,

9, 12, 15, 18, or 21 bits in length. A set of register change instructions enables the programmer to perform clear, exchange, or one's complement operations on any single byte or combination of bytes in the A and B registers. Bytes in combinations may be contiguous or not, as desired by the user.

For example, a 6-bit byte located in A_{0-5} can be transferred into the B register and the contents of A_{0-5} cleared to zero in a single instruction without affecting the remainder of the A and B registers.

In addition to register change byte operations, the computer provides an instruction, TWIN MULTIPLY (TMU), that simultaneously multiplies two 12-bit bytes in the A register by two 12-bit bytes in memory, and produces two 24-bit products in the A and B registers.

The register change set includes instructions for the rapid and simultaneous transfer of information among the A, B, and index registers, either as full words or by selected fields. These instructions also provide the capability to set the base address of any index register with the address portion of the instruction.

FLOATING-POINT OPERATIONS

The use of floating-point numbers relieves the programmer of the burden of maintaining the binary point while arithmetic operations are being performed. The 9300 offers the user two methods for operating on floating-point numbers. Both methods use the previously defined floating-point format.

Optional floating-point instructions can be used to add, subtract, multiply, and divide floating-point numbers. These instructions automatically maintain the correct fraction and exponent relationship from execution to execution, as well as maintaining the numbers being operated on in proper floating-point form.

The same floating-point instruction codes can be used when the floating-point hardware option is not present in the computer. However, an automatic trap to memory location 030 results whenever such an instruction code is executed. (See Trap Locations, Section 3.) Control is transferred to memory location 030 without affecting the program counter. A standard SDS Floating-Point Package is provided which decodes and executes the proper floating-point instruction in a manner identical to the optional hardware, but at a slower speed. Thus, identical programs can be executed on computers with or without built-in floating-point hardware.

See Section 2 for a complete discussion of floating-point instructions as they are used in SDS Floating-Point Subroutines or in the built-in hardware form.

MEMORY OVERLAP TIMING

The execution time of an SDS 9300 program is affected by the optimum use of memory, input/output interface, and interrupt processing. These factors are discussed in this section.

INSTRUCTION OVERLAP

Instruction timing given in Sections 2, 3, and 4 assumes one memory cycle to obtain and interpret the instruction itself and, if necessary, one more cycle to obtain the operand and execute the instruction. If the central processor contains more than one memory module, the functions of obtaining the operand for one instruction and obtaining the next instruction might be performed in the same cycle. When this occurs, the instructions are said to overlap. Whenever overlap occurs, one cycle can be subtracted from the total number of cycles normally required to complete a given sequence of instructions.

Figure 1-4 shows two instructions in overlapped and non-overlapped execution.

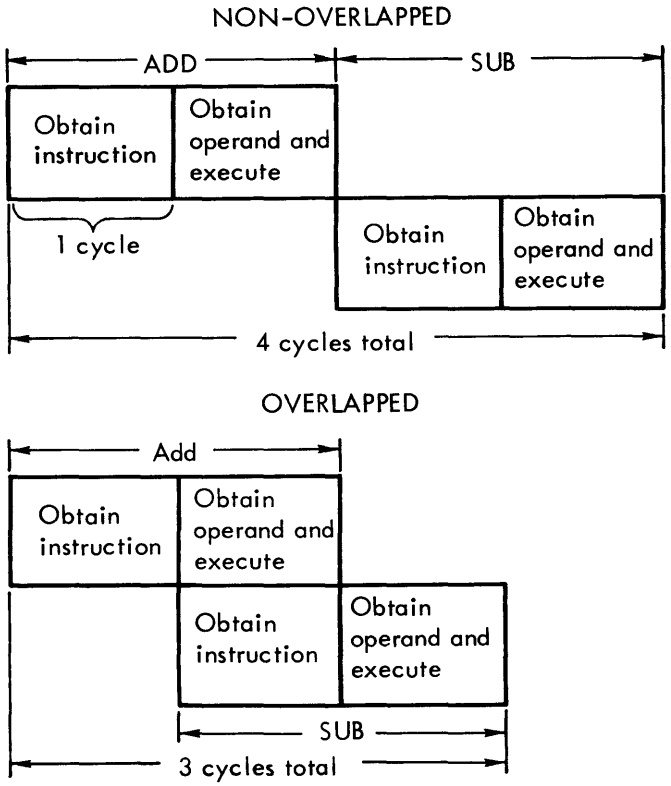


Figure 1-4. Non-Overlapped and Overlapped Execution of Instructions

Two contiguous instructions, J and K, overlap if the operand for J is in a memory module other than one which contains J and K. If the operand and instructions are contained in the same module, no overlap occurs. Also, no overlap occurs if:

Instruction J is:	or	Instruction K is:
EOM		EOM
LDX		SKS
XMX		NOP
EAX		Any Register Change
NOP		Any Shift
INT		Any Flag Test or Set
REP		
FLA		
FLS		
BRU		
BRR		
BRX		
BRC		
Any Register Change		
Any Shift		
Any Flag Test or Set		
Any Skip		

INTERRUPT TIMING

The occurrence of an interrupt may affect the timing of a sequence of normally overlapping instructions. If overlapping can occur in the sequence, then one or more of the overlapped cycles may be performed sequentially without overlap, depending upon the point at which the interrupt occurs.

An overlapped cycle is lost if an interrupt occurs during the execution of the first of two instructions which normally overlap. While executing the first instruction, the central processor obtains the next instruction word. Due to the presence of an interrupt, the instruction just obtained is not executed and, at the conclusion of the instruction currently being executed, the central processor obeys the interrupt. When the interrupt has been processed, a normal return to the main program causes the execution of the same instruction previously obtained but not used.

INPUT/OUTPUT TIMING

When a communication channel requires access to a memory module, it assumes control for the next memory cycle regardless of central processor requests. The channel has priority over the central processor.

If a channel needs access to the same memory module at the same time as the central processor, the channel takes its access first; the central processor waits one or two cycles. This priority procedure occurs even if the central processor is in the middle of executing an instruction. Total timing of the sequence being executed is thus increased by one (or two) memory cycle(s).

If the central processor is not using the same memory module as a direct access channel, then no time is lost. Both operations proceed in parallel without interference.

The priority for memory access is (1) direct access channels, (2) time-multiplexed channels, (3) operands to the central processor, and (4) instructions to the central processor.

Some instructions require several memory cycles for their execution, but do not access memory during every cycle. During the cycles when the central processor is executing these instructions, but not accessing memory, direct access channel access to memory may occur without affecting computation. The following instructions have such "free" cycles:

DPN	FLS
TMU	FLM
DIV	FLD
MUL	All Shifts
FLA	All Skips if the Skip Occurs

Figure 1-5 illustrates the free cycles in a multiply instruction with an overlapped add instruction following. If the next instruction could not be overlapped, there would be one additional free cycle.

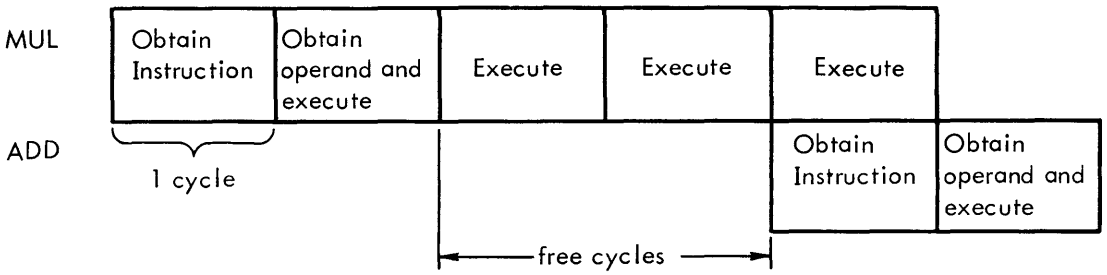


Figure 1-5. Example of Free Cycles

OPTIONAL MEMORY PROTECTION FEATURE

A single memory module (4K, 8K, or 16K) for the 9300 computer can be equipped with either a manual or a programmed memory protection feature (but not with both). If a computer has more than one memory module, individual modules can use either method; i.e., the manual or programmed feature can be intermixed among separate modules.

The memory protection feature equips a 9300 computer memory module with a write lock-out that protects the contents of selected areas of memory from being inadvertently destroyed or changed. (The memory protection feature is logically and electrically a part of the memory module, and, therefore, must be ordered for each module for which protection is desired.)

In the manual option, memory protection is controlled manually by switches. The manual method is important in real-time and on-line systems where an executive or master control program, or a system start-up/shut-down routine must be given permanent protection.

The programmed memory protection feature allows the computer to be time-shared in a multiprogrammed manner by several independent programs, supervised by a monitor that controls the memory lock-out system and guarantees that none of these programs or their data will be inadvertently destroyed by another.

Once a memory block has been protected, any reference to any of the protected addresses that would require a "write into memory" results in the following:

1. The offending instruction is completed normally, except that the "alter memory" portion of the storage cycle is inhibited; i.e., memory remains unchanged. "Write-requests" from the input/output channels will be similarly ignored. Memory will never be changed, thus preventing inadvertent destruction of data and instructions. All memory locations can always be read.

2. An internal interrupt (trap) will occur to a fixed location associated with the memory protection system. Location 04 is used in the SDS 9300 Computer. The interrupt routine may then determine the cause of the error and take appropriate remedial action.

PROGRAMMING CONSIDERATIONS

When the programmed memory protection feature is employed, memory modules are locked-out by an EOM/POT sequence. The program executes an EOM 020040 (ALMP: ALERT TO LOAD MEMORY PROTECT REGISTER) followed by a PARALLEL OUTPUT instruction. Various memory segments are controlled by bits in the output word: a 0 bit allows writing; a 1 bit sets the lock-out. Bit assignments depend on the address of the first word of the memory module to which the protection feature is attached, as shown in the following table:

Bit Number	Addresses Locked-Out	Bit Number	Addresses Locked-Out
23	00000 - 00777	13	34000 - 37777
22	01000 - 01777	12	40000 - 43777
21	02000 - 02777	11	44000 - 47777
20	03000 - 03777	10	50000 - 53777
19	04000 - 07777	9	54000 - 57777
18	10000 - 13777	8	60000 - 63777
17	14000 - 17777	7	64000 - 67777
16	20000 - 23777	6	70000 - 73777
15	24000 - 27777	5	74000 - 77777
14	30000 - 33777		

2. MACHINE INSTRUCTIONS

INTRODUCTION

This section contains a description of SDS 9300 instructions. They are grouped by functional category. Lists of instructions in functional, numerical, and alphabetical order are given in Appendix B, pages B-13, B-23, and B-29, respectively.

The following statements apply to the instruction descriptions:

All instruction times are in memory cycles, each cycle being 1.75 microseconds, include accessing the instruction, and are shown without overlap. Memory overlapping decreases by one machine cycle the execution of most instructions of two or more memory cycles where a memory access of an operand is required. (See Memory Overlap Timing, Section 1.)

Parentheses are used to denote "contents of." For example, "(A)" denotes "contents of the A register." The contents of registers and memory locations are expressed, in this manual, as octal numbers preceded by a signed (or unsigned) zero.

Subscripted numbers identify inclusive bit positions. For example, "$(A)_{0-11}$" indicates the contents of "bit positions 0 through 11 of the A register." (All numbers not preceded by a zero are decimal base, except for instructions, memory locations, the contents of memory locations, and the contents of registers.)

Indexing and indirect addressing apply to all instructions except as noted. Indexing does not change the instruction execution time. One additional memory cycle is required for each level of indirect addressing used.

The interrupt system can interrupt the program sequence at the end of any instruction except as noted.

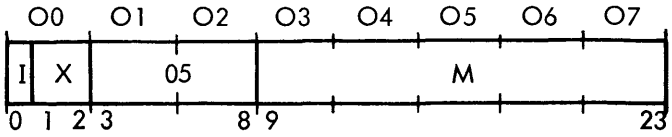
Registers or memory locations affected by each instruction are specified.

Any instruction can be repeated by REPEAT INSTRUCTION IN M (REP), except as noted. (See Control Group, REP.)

A diagram representing the instruction format is given for each instruction. The name of the instruction and its mnemonic code precede this diagram.

Example:

ADD ADD MEMORY TO A



The letter "M" represents the address part of the instruction. Some instructions will have octal numbers in this field, as denoted by the letter "C." These instructions do not refer to memory. If the letter "I" appears in bit position 0, the instruction operand can be indirectly addressed. The letter "X" in the index field (bit positions 1 and 2) denotes that the instruction may refer to an index register.

When discussing properties of the various instructions, including the indirect addressing facility, several terms have evolved to describe specific locations or addresses.

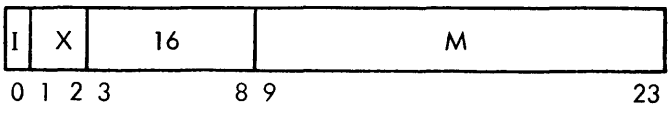
The term, effective memory location (EM), is used to identify the location in memory from which the final operand is to be taken at the conclusion of all indirect addressing and indexing. This term is sometimes shortened to "effective location."

The term, effective address, means the 5-digit octal address of the effective memory location.

The term, effective operand, means the contents of the effective memory location, or (EM).

LOAD/STORE GROUP

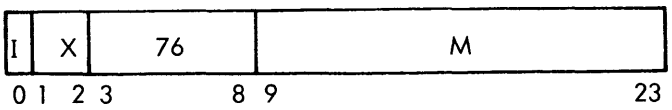
LDA LOAD A



The contents of the effective memory location are loaded into the A register.

Affected: (A) Timing: 2

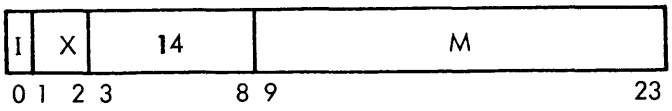
STA STORE A



The contents of the A register are stored in the effective memory location.

Affected: (EM) Timing: 3

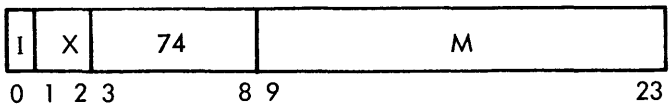
LDB LOAD B



The contents of the effective memory location are loaded into the B register.

Affected: (B) Timing: 2

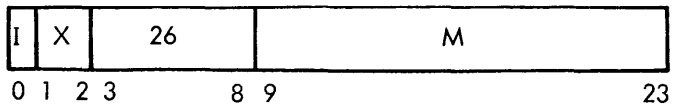
STB STORE B



The contents of the B register are stored in the effective memory location.

Affected: (EM) Timing: 3

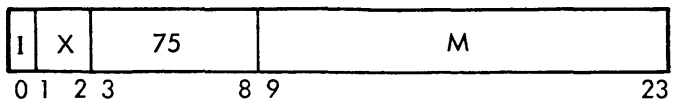
LDP LOAD DOUBLE PRECISION



The collective contents of the effective memory location and the effective memory location plus one are loaded into the A and B registers. For example, if the effective memory location is EM, then (EM) are loaded into the A register and (EM + 1) are loaded into the B register.

Affected: (A, B) Timing: 3

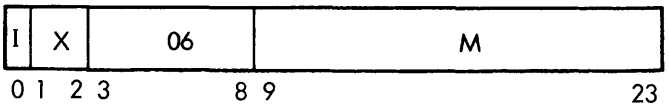
STD STORE DOUBLE PRECISION



The contents of the A and B registers are stored, respectively, into the effective memory location and the effective memory location plus one. For example, if the effective memory location is EM, the contents of the A register are stored in EM and the contents of the B register are stored in EM + 1.

Affected: (EM), (EM + 1) Timing: 4

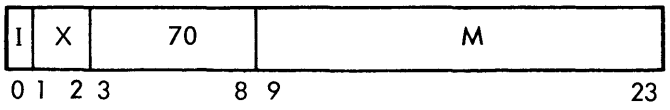
LDS LOAD SELECTIVE



The contents of the effective memory location are selectively loaded through a mask into the A register. The B register contains the selection mask. For each 1-bit in the B register the corresponding bit in the effective location is loaded into the A register. For each 0-bit in the B register the corresponding bit of the A register is unaffected.

Affected: (A) Timing: 2

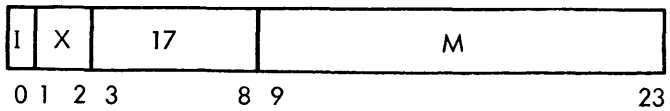
STS STORE SELECTIVE



The contents of the A register are selectively stored through a mask into the effective memory location. The B register contains the selection mask. For each 1-bit in the B register, the corresponding bit in the A register is stored in the effective location. For each 0-bit, the contents of the corresponding bit in the location is unaffected.

Affected: (EM) Timing: 3

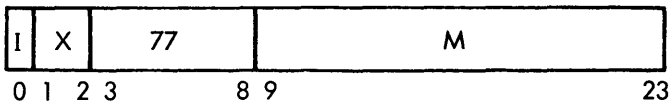
LDX LOAD INDEX



The entire 24-bit contents of the effective memory location are loaded into the index register specified by the contents of the index field. This procedure simultaneously sets both the base address and the increment value into the index register. This instruction cannot use an index register to modify the effective memory location. If indirect addressing is used, an index register cannot be used to modify the effective memory location at any level.

Affected: (X) Timing: 2

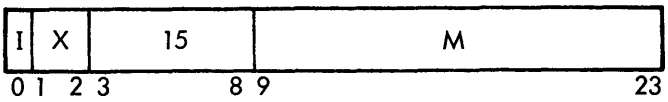
STX STORE INDEX



The contents of the index register specified in the index field of the instruction are stored in the effective memory location. All 24 bits of the index register are stored. This instruction cannot use an index register to modify the effective memory location. If indirect addressing is used, an index register cannot be used to modify the effective memory location at any level.

Affected: (EM) Timing: 3

EAX COPY EFFECTIVE ADDRESS INTO INDEX REGISTER 1



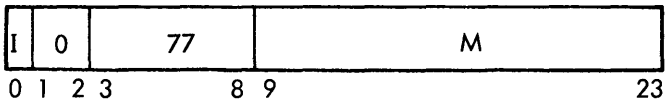
The address of the effective memory location is copied into the address portion of index register 1.

The addressing process for this instruction operates as in a LOAD A (LDA) instruction, except that instead of obtaining the contents of the effective memory location, the effective memory address itself is used as the operand. For example, if this instruction is executed with a zero indirect address bit and zeros in the index bit positions, the actual bit configuration in the address field of this instruction is copied into index register 1. Bit positions 0 through 8 of X1 are unchanged.

This instruction cannot be repeated with REP.

Affected: (X1) Timing: 2

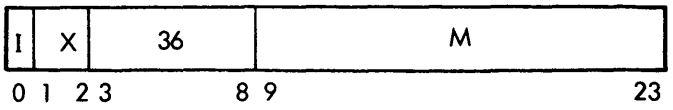
STZ STORE ZERO



A zero word is stored in the effective memory location. This instruction cannot use an index register to modify the effective memory location. This instruction cannot be repeated with REP.

Affected: (EM) Timing: 3

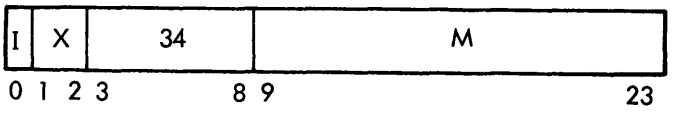
XMA EXCHANGE MEMORY AND A



The contents of the effective memory location are loaded into the A register and the contents of the A register are stored in the effective memory location.

Affected: (A), (EM) Timing: 3

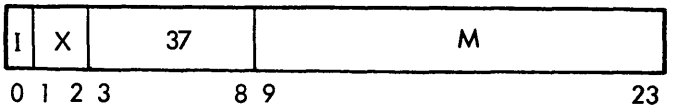
XMB EXCHANGE MEMORY AND B



The contents of the effective memory location are loaded into the B register and the contents of the B register are stored in the effective memory location.

Affected: (B), (EM) Timing: 3

XMX EXCHANGE MEMORY AND INDEX

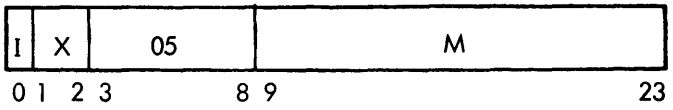


The contents of the effective memory location are loaded into the index register specified in the index field of the instruction, and the index register contents are stored in the same memory location. This instruction cannot use an index register to modify the effective memory location. If indirect addressing is used, an index register cannot be used to modify the effective memory location at any level.

Affected: (X), (EM) Timing: 3

BINARY ARITHMETIC GROUP

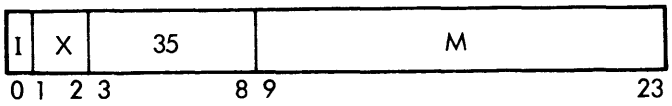
ADD ADD MEMORY TO A



The contents of the effective memory location are added to the A register and the result is placed in A. If both numbers have the same sign but the sign of the result is opposite, an overflow occurs and the OVERFLOW indicator is set. In this case, the sum is incorrect.

Affected: (A), Of Timing: 2

ADM ADD A TO MEMORY

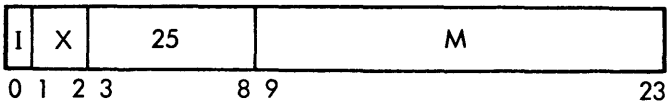


The contents of the A register are added to the contents of the effective memory location. The result is stored in the same location.

If both numbers have the same sign but the sign of the result is opposite, an overflow occurs and the OVERFLOW indicator is set. In this case, the sum is incorrect.

Affected: (EM), Of Timing: 3

DPA DOUBLE PRECISION ADD



The collective contents of the effective memory location and the effective location plus one are added to the collective contents of the A and B registers. The result is placed in the A and B registers.

If both numbers have the same sign but the sign of the result is opposite, an overflow occurs and the OVERFLOW indicator is set. In this case, the sum is incorrect.

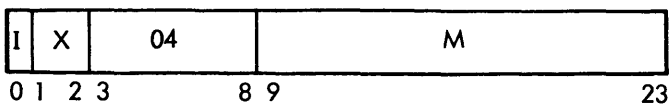
Affected: (A, B), Of Timing: 3

Example:

Assume locations EM and EM + 1 contain 01234567 and 07654321, respectively. Assume registers A and B contain 01233210 and 43211234, respectively. The DPA instruction performs the addition:

A	B	
01233210	43211234	
+ 01234567	07654321	from locations EM and EM + 1 results in A and B, respectively
02467777	53065555	

SUB SUBTRACT MEMORY FROM A

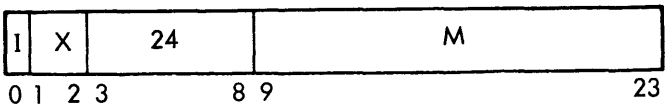


The contents of the effective memory location are subtracted from the A register and the result is placed in the A register.

If both numbers have the same sign after the subtrahend has been complemented for addition, but the sign of the result is opposite, an overflow occurs and the OVERFLOW indicator is set. In this case, the difference is incorrect.

Affected: (A), Of Timing: 2

DPS DOUBLE PRECISION SUBTRACT

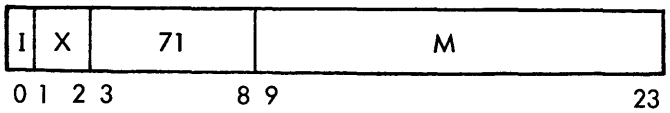


The collective contents of the effective memory location and the effective location plus one are subtracted from the collective contents of the A and B registers. The result is placed in the A and B registers.

If both numbers have the same sign after the subtrahend has been complemented for addition, but the sign of the result is opposite, an overflow occurs and the OVERFLOW indicator is set. In this case, the difference is incorrect.

Affected: (A, B), Of Timing: 3

MPO MEMORY PLUS ONE

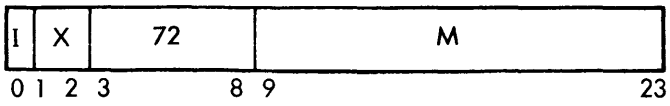


The contents of the effective memory location are increased by one. The result is placed in the same location.

Overflow does not cause the OVERFLOW indicator to be set.

Affected: (EM) Timing: 3

MPT MEMORY PLUS TWO

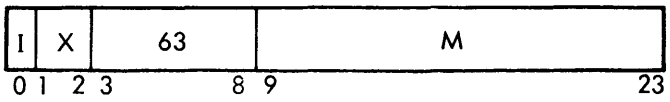


The contents of the effective memory location are increased by two. The result is placed in the same location.

Overflow does not cause the OVERFLOW indicator to be set.

Affected: (EM) Timing: 3

MUL MULTIPLY



The A register is multiplied by the contents of the effective memory location. The product is left in the A and B registers, with the more significant portion in A. The sign of the product is in A_0; the bit in B_0 is part of the product and is not treated as a sign bit. Since the product contains at most 46 significant bits, the content of B_{23} is not significant and is zero. The original contents of B do not affect the operation and are destroyed. If the product is greater than 2^{46}, overflow occurs and the OVERFLOW indicator is turned on.

This instruction cannot be repeated with REP.

Affected: (A, B), Of Timing: 5

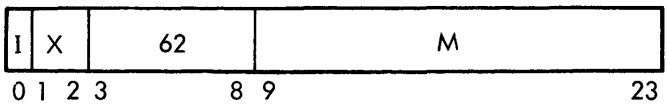
Example: MUL M

The multiplication of 3 by 3 is represented:

	(A)	(B)	(M)
Before execution:	00000003	Irrelevant	00000003
After execution:	00000000	00000022	00000003

Note that 022 scaled at 46 is equal to 011 scaled at 47.

DIV DIVIDE



The contents of the A and B registers treated as a double precision number are divided by the contents of the effective memory location. The quotient is placed in the A register, the remainder in the B register.

No overflow occurs if $-1 \leq [(A,B)/(M)] < 1$ (that is, if the contents of A and B divided by the contents of the effective location are greater than or equal to minus one but strictly less than plus one). If the quotient exceeds these boundaries, overflow occurs and the OVERFLOW indicator is set. In this latter case, the results are not arithmetically correct.

When dividing N, with scaling at Q1, by D, with scaling at Q2, the quotient Q is scaled at Q1-Q2. This instruction cannot be repeated with REP. See Appendix B-9 for a detailed description of this instruction.

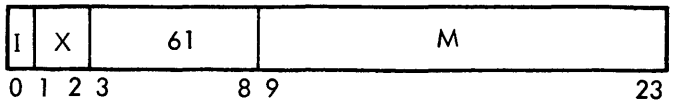
Affected: (A, B), Of Timing: 10

Example: DIV M

Dividing 7 scaled at binary 46 by 3 at binary 23 is represented:

	(A)	(B)	(M)
Before execution:	00000000	00000016	00000003
After execution:	00000002	00000001	00000003

TMU TWIN MULTIPLY



The contents of the A register and the contents of the effective memory location are each treated as a pair of 12-bit arithmetic quantities; bit positions 0 through 11 represent one unsigned quantity and bit positions 12 through 23 represent a second unsigned quantity. Bit positions 0 through 11 of the effective memory location are multiplied by bit positions 0 through 11 of the A register and the result is placed in the A register; simultaneously, bit positions 12 through 23 of the effective memory location are multiplied by bit positions 12 through 23 of the A register and the result is placed in the B register. The original contents of the B register are lost. No overflow can occur.

This instruction cannot be repeated with REP.

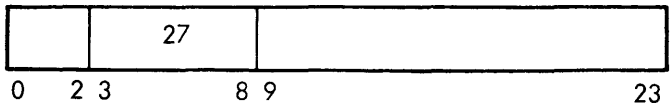
Twin Multiply (TMU) is useful in performing arithmetic operation on 12-bit input quantities where these quantities are packed two to a word. Also, logical integer multiplication can be performed on either single or double quantities.

Affected: (A, B) Timing: 5

Example: TMU M

	(A)	(B)	(M)
Before execution:	40070003	Immaterial	40070003
After execution:	20070061	00000011	40070003

DPN DOUBLE-PRECISION NEGATE



The double-precision fixed-point number contained in the A and B registers is replaced by the negative of that number.

Overflow occurs if the initial collective contents of A and B are the largest representable negative number [(A) = 40000000; (B) = 00000000]. In this case, the number remains unchanged.

This instruction cannot be repeated with REP.

Affected: (A, B), Of Timing: 3

FLOATING-POINT GROUP (Optional)

The following four instructions are optional floating-point instructions. They operate on normalized numbers in the floating-point format, and automatically leave all quantities normalized after execution.

All floating-point arithmetic operations assume that non-zero quantities are normalized prior to instruction execution. An attempt to perform a floating-point multiplication or division on an unnormalized number produces the identical result as using zero. An attempt to perform a floating-point addition or subtraction on an unnormalized number may produce erroneous results.

If a negative number is right-shifted in the process of equalizing exponents, the sign bit is propagated even if the shift exceeds 38 bit positions.

Example (Addition):

$$2020315205765 \times 2^{150} \quad = \quad 2020315205765 \times 2^{150}$$
$$5432100000000 \times 2^{0} \quad \quad \underline{7777777777777 \times 2^{150}}$$
$$2020315205764 \times 2^{150}$$

Example (Subtraction):

$$2020315205765 \times 2^{150} \quad = \quad 2020315205765 \times 2^{150}$$
$$2234567012345 \times 2^{0} \quad \quad \underline{7777777777777 \times 2^{150}}$$
$$2020315205764 \times 2^{150}$$

Floating-point operations with a zero result produce a fraction and an exponent of all zeros.

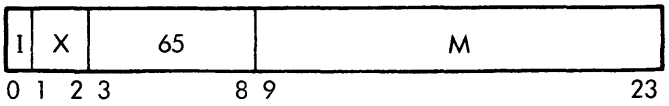
Overflow occurs in floating-point operations when the resultant positive exponent exceeds the 9-bit exponent field. Underflow occurs when the resultant negative exponent exceeds the 9-bit exponent field. Both overflow and underflow can occur in all floating-point operations and are automatically detected. These conditions cause the computer to execute the instruction in location 031 (see Trap Locations, Section 3). This location normally contains the entry to a corrective subroutine. Flag indicator 1 (the OVERFLOW indicator) is unaffected by a floating underflow or overflow.

In multiplication there is no overflow or underflow indication, regardless of exponents, if zero (or an unnormalized fraction) is used as a multiplier or multiplicand.

In division there is no overflow or underflow indication, if the numerator is zero (or an unnormalized fraction) and the denominator is a normalized non-zero number.

In division there is overflow indication regardless of exponents. if the denominator is zero (or an unnormalized fraction.) A quotient of zero fraction and zero exponent is produced as a result of these conditions.

FLA FLOATING ADD



The floating-point number contained in the effective memory location and the effective location plus one is added to the floating-point number in the A and B registers. The normalized floating-point sum is left in the A and B registers.

An attempt to use an unnormalized non-zero produces an erroneous result with the following exception. If zero is added to an unnormalized non-zero number in A, B, the number in A, B will become normalized and its exponent will be adjusted.

Exponent overflow or underflow can occur with this instruction. If either occurs, the computer executes the contents of location 031 as an instruction.

This instruction cannot be repeated with REP.

By using a special case of FLA, the floating-point number contained in the A and B registers can be normalized, with the contents of the exponent field of the number being decremented for each position of shift. To perform the normalize operation, the contents of the effective memory location and the effective memory location plus one must be all zeros. If the fractional portion of the number in the A and B registers is zero, the contents of the exponent field are set to zero.

Affected: (A, B) Timing: 6-11

Example: FLA M

Assume the floating-point binary number 0.101×2^{10} is stored in location M and M + 1 in floating-point format. This is written in octal form as 0.5×2^{2} and stored in memory as "24000000" since the high-order bit is the sign bit.

(M) = 24000000 (M + 1) = 00000002

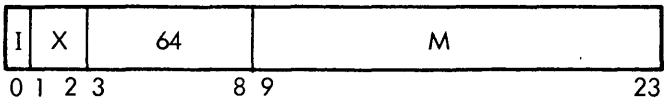
Add to it the same number in the A and B registers:

(A) = 24000000 (B) = 00000002

FLA M yields:

(A) = 24000000 (B) = 00000003

FLS FLOATING SUBTRACT



The floating-point number contained in the effective memory location and the effective location plus one is subtracted from the floating-point number contained in the A and B registers. The normalized floating-point difference is left in the A and B registers.

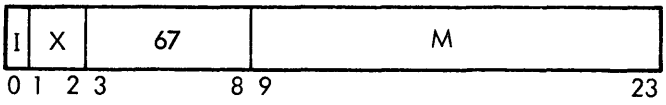
An attempt to perform a subtraction with an unnormalized number will produce an erroneous result.

Exponent overflow or underflow can occur with this instruction. If either occurs, the computer executes the contents of location 031 as an instruction.

This instruction cannot be repeated with REP.

Affected: (A, B) Timing: 6-11

FLM FLOATING MULTIPLY



The floating-point number contained in the effective memory location and the effective location plus one is multiplied by the floating-point number contained in the A and B registers. The normalized floating-point product is left in the A and B registers. The floating-point product is truncated at the right end to 39 bits.

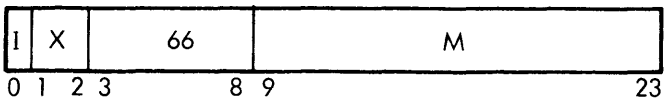
Exponent overflow or underflow can occur with this instruction. If either occurs, the computer executes the contents of location 031 as an instruction.

If either A and B registers or the two effective memory locations, or both, contain a floating-point zero, the contents of the A and B registers are made identically zero by clearing them.

This instruction cannot be repeated with REP.

Affected: (A, B) Timing: 8 cycles
if (A, B) = 0, 3 cycles
if (M, M + 1) = 0, 4 cycles

FLD FLOATING DIVIDE



The floating-point number contained in the A and B registers is divided by the floating-point number contained in the effective memory location and the effective location plus one. The normalized floating-point quotient is left in the A and B registers. The floating-point quotient is truncated to 39 bits at the right end.

Exponent overflow or underflow can occur with this instruction. If either occurs, the computer executes the contents of location 031 as an instruction.

If the contents of the A, B registers are a floating-point zero but the contents of the two effective memory locations are not a floating-point zero, the contents of the A, B registers are made identically zero by clearing them.

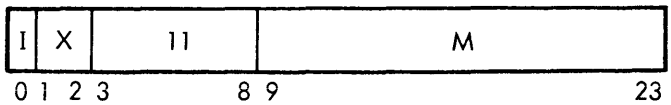
If the contents of the two effective memory locations are a floating-point zero, and the contents of the A, B registers any number, the contents of the A, B registers are made identically zero by clearing them. The overflow trap is taken in this case. This overflow is distinguishable since it is the only case where the A, B registers are identically zero during a floating-point operation when the overflow exit is taken.

This instruction cannot be repeated with REP.

Affected: (A, B) Timing: 17 cycles
if (A, B) = 0, (M, M+ 1) ≠ 0, 3 cycles
if (M, M + 1) = 0, 4 cycles

LOGICAL GROUP

ETR EXTRACT



A logical AND is performed between corresponding bits of the A register and the effective memory location. The result is placed in A. The operation is performed bit by corresponding bit as follows:

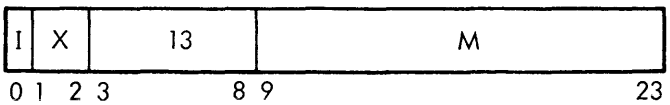
(A)	(EM)	Result in A
0	0	0
0	1	0
1	0	0
1	1	1

Affected: (A) Timing: 2

Example: ETR M

	(A)	(M)
Before execution:	64231567	00777700
After execution:	00231500	00777700

MRG MERGE



A logical inclusive OR is performed between corresponding bits of the A register and the effective memory

location. The result is left in A. The operation is performed bit by corresponding bit as follows:

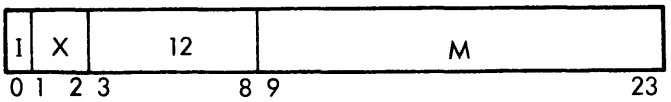
(A)	(EM)	Result in A
0	0	0
0	1	1
1	0	1
1	1	1

Affected: (A) Timing: 2

Example: MRG M

	(A)	(M)
Before execution:	06446254	02340712
After execution:	06746756	02340712

EOR EXCLUSIVE OR



A logical exclusive OR is performed between corresponding bits of the A register and the effective memory location. The result is left in A. The operation is performed bit by corresponding bit according to the following:

(A)	(EM)	Result in A
0	0	0
0	1	1
1	0	1
1	1	0

Affected: (A) Timing: 2

Example: EOR M

	(A)	(M)
Before execution:	34165031	70077021
After execution:	44112010	70077021

Selected bit positions of the A register may be logically inverted by proper configuration of the memory word. If every bit in the memory word is set to 1, a "one's complement" of A results.

Example: EOR M

	(A)	(M)
Before execution:	10357211	77777777
After execution:	67420566	77777777

REGISTER CHANGE GROUP

The facility to operate on and/or exchange data between the A, B, and the three index registers is provided within the set of micro-instructions in the Register Change group. The micro-instructions also provide the capability to perform all but two of these operations using only selected bytes of the registers involved in the operation. (The exceptions are COPY NEGATIVE OF A INTO A and ADDRESS TO INDEX BASE.) Generally, only those bytes designated by a 1-bit in the appropriate bit position of the instruction can be disturbed in any way.

All register change instructions use the same instruction code (040), and the address field of the instruction specifies the function to be performed. Three distinct modes are specified by the contents of the tag field (bits 0-2 of the instruction). The functions of the three modes are as follows:

Mode	Function
1	Interchange and/or modify information between selected bytes of registers A and B.
2	Interchange and/or modify information among selected bytes of registers A and B and the three index registers.
3	Load the address portion of a selected index register from the address field of the instruction.

In modes 1 and 2, the address portion of the instruction serves to extend the operation code (i.e., each of the address bits has a particular significance during instruction decoding and execution). In mode 3, the 15-bit address field is interpreted as an immediate operand, to be loaded into the address field of the designated index register.

When programmed in mode 1 or 2, the instruction may be given one of two mnemonics which are RCH or COPY. The assembler processes the address field and tag field of RCH instructions in the conventional manner, inserting the evaluated address field into bit positions 9-23 of the instruction and inserting the tag field into bit positions 0-2 of the instruction. In general, the address field is an octal number, representing the bit pattern that specifies the function to be performed.

The address field of COPY is interpreted differently. The field consists of a byte selection "mask" followed by one or more grouped expression lists that describe the desired operation(s). In the instruction examples to follow, a COPY instruction is also given, based on the information presented in Appendix B.

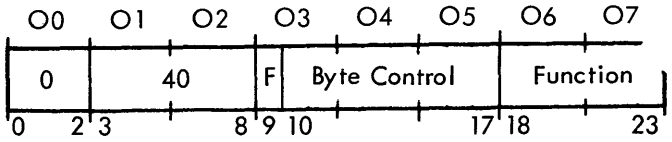
These instructions cannot be repeated with REP.

MODE 1: A AND B REGISTER CHANGE

The micro-instructions in this mode interchange and modify information between the A and B registers. From 3 to 24 bits may be operated upon in 3-bit bytes. The tag field must always be zero in this mode.

RCH REGISTER CHANGE

The instruction format for this operation is:



The function control bits are interpreted as follows:

Bit Position	Address Field Value	Function
9	040000	Copy negative of (A) into A
18	040	Merge inverse of (A) into B
19	020	Merge (A) into B
20	010	Clear B
21	04	Merge inverse of (B) into A
22	02	Merge (B) into A
23	01	Clear A

Note: If bit position 9 is 1, bit positions 21 and 22 must both be 0, but bit position 23 has no effect; however bit positions 18, 19, and 20 do operate, and with byte control.

Unless either bit 20 or 23 in the instruction is set to 1 (to clear the appropriate destination register), any of the operations other than COPY NEGATIVE OF A INTO A causes a merge into the selected bytes of the destination register.

The byte-control bits are interpreted as follows:

Instruction Word (RCH)		Affected Registers	
Bit Position	Address Field Value	Bit Positions Selected	Mask for COPY
10	020000	0-2	070000000
11	010000	3-5	07000000
12	04000	6-8	0700000
13	02000	9-11	070000
14	01000	12-14	07000
15	0400	15-17	0700
16	0200	18-20	070
17	0100	21-23	07

At least one byte-control bit must be present in all mode 1 RCH instructions (except COPY NEGATIVE OF A INTO A). If the mask (M) is omitted in a COPY instruction, the assembler automatically generates a mask of 077777777.

In the instruction examples to follow, the RCH instruction given is coded for all 1-bits in bit positions 10-17 of all instructions (except for COPY NEGATIVE OF A INTO A). A COPY example is also given, where M represents a byte selection mask and A, B, and X represent the A, B, and index registers (see Appendix B).

Affected: See examples Timing: 1

COPY NEGATIVE OF A INTO A

RCH 040000 COPY (-A, A) (0 40 40000)

The two's complement of the contents of A replaces the contents of A. This instruction does not operate under byte control; the contents of bit positions 10 - 17 of this instruction are ignored.

If the A register contains the number 40000000, the same number is left as the result of the execution of this instruction and the OVERFLOW indicator is set.

Affected: (A), Of

CLEAR A

RCH 037701 COPY (0, A) 0 40 37701

The contents of the selected bytes of A are set to zero; the rest of the contents of A is undisturbed.

Affected: (A)

Example: RCH 023101 COPY 070077007, (0, 5)

Bits 0 through 2, bits 9 through 14, and bits 21 through 23 of the A register are cleared by setting the appropriate "byte select" bits:

0 40 23101

	(A)
Before execution:	77777777
After execution:	07700770

CLEAR B

RCH 037710 COPY (0, B) 0 40 37710

The contents of the selected bytes of B are set to zero; the rest of the contents of B is undisturbed.

Affected: (B)

CLEAR A AND B

RCH 037711 COPY (0, (A, B)) 0 40 37711

The contents of the selected bytes of A and of B are set to zero; the remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

COPY A INTO B

RCH 037730 COPY (A, B) 0 40 37730

The contents of the selected bytes of A replace the contents of the same bytes of B. The contents of A and the rest of the contents of B are undisturbed.

Affected: (B)

COPY INVERSE OF A INTO B

RCH 037750 COPY (1-A, B) 0 40 37750

The logical inverse (one's complement) of the contents of the selected bytes of A replaces the contents of the same bytes of B. The contents of A and the rest of the contents of B are undisturbed.

Affected: (B)

COPY INVERSE OF A INTO B AND CLEAR A

RCH 037751 COPY (1-A, B), (0, A) 0 40 37751

The logical inverse (one's complement) of the contents of the selected bytes of A replaces the contents of the same bytes of B. The same bytes of A are then cleared. The rest of the contents of B and A is undisturbed.

Affected: (A), (B)

COPY B INTO A

RCH 037703 COPY (B, A) 0 40 37703

The contents of the selected bytes of B replace the contents of the same bytes of A. The contents of B and the rest of the contents of A are undisturbed.

Affected: (A)

EXCHANGE A AND B

RCH 037733 COPY (A, B), (B, A) 0 40 37733

The contents of the selected bytes of A are exchanged with the contents of the same bytes of B. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

EXCHANGE INVERSE OF A WITH B

RCH 037753 COPY (1-A, B), (B, A) 0 40 37753

The logical inverse (one's complement) of the contents of the selected bytes of A are exchanged with the contents of the selected bytes of B. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

EXCHANGE INVERSE OF B WITH A

RCH 037735 COPY (1-B, A), (A, B) 0 40 37735

The logical inverse of the contents of the selected bytes of B are exchanged with the contents of the selected bytes of A. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

EXCHANGE INVERSE OF A WITH INVERSE OF B

RCH 037755 COPY (1-A, B), (1-B, A) 0 40 37755

The logical inverse of the contents of the selected bytes of A are exchanged with the inverse of the contents of the selected bytes of B. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

COPY INVERSE OF B INTO A

RCH 037705 COPY (1-B, A) 0 40 37705

The logical inverse of the contents of the selected bytes of B replaces the contents of the same bytes of A. The contents of B and the rest of the contents of A are undisturbed.

Affected: (A), (B)

COPY INVERSE OF B INTO A AND CLEAR B

RCH 037715 COPY(1-B,A), (0,B) 0 40 37715

The logical inverse of the contents of the selected bytes of B replaces the contents of the same bytes of A. The contents of the same bytes of B are set to zero. The rest of the contents of A and B is undisturbed.

Affected: (A), (B)

COPY A INTO B AND CLEAR A

RCH 037731 COPY (A,B), (0,A) 0 40 37731

The contents of the selected bytes of A replace the contents of the same bytes of B and the contents of the same bytes of A are set to zero. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

COPY B INTO A AND CLEAR B

RCH 037713 COPY (B,A), (0,B) 0 40 37713

The contents of the selected bytes of B replace the contents of the same bytes of A and the contents of the same bytes of B are set to zero. The remaining bits of the contents of A and B are undisturbed.

Affected: (A), (B)

MERGE A INTO B

RCH 037720 COPY (A,B,B) 0 40 37720

A logical inclusive OR is performed between the contents of the selected bytes of A and B, with the result left in the same bytes of B. The contents of A and the rest of the contents of B are undisturbed.

Affected: (B)

MERGE B INTO A

RCH 037702 COPY (B,A,A) 0 40 37702

A logical inclusive OR is performed between the contents of the selected bytes of B and A with the result left in the same bytes of A. The contents of B and the rest of the contents of A are undisturbed.

Affected: (A)

MERGE A INTO B AND CLEAR A

RCH 037721 COPY (A,B,B), (0,A) 0 40 37721

A logical inclusive OR is performed between the contents of the selected bytes of A and B with the result left in the same bytes of B. The same bytes of the A register are then cleared. The rest of the contents of B and A is undisturbed.

Affected: (A), (B)

MERGE B INTO A AND CLEAR B

RCH 037712 COPY (B,A,A), (0,B) 0 40 37712

A logical inclusive OR is performed between the contents of the selected bytes of B and A, with the result left in the same bytes of A. The same bytes of the B register are then cleared. The rest of the contents of A and B is undisturbed.

Affected: (A), (B)

MERGE INVERSE OF A INTO B

RCH 037740 COPY (1-A,B,B) 0 40 37740

A logical inclusive OR is performed between the inverse of the contents of the selected bytes of A and the contents of the same bytes of B. The result is left in B. The contents of A and the rest of the contents of B are undisturbed.

Affected: (B)

MERGE INVERSE OF B INTO A

RCH 037704 COPY (1-B,A,A) 0 40 37704

A logical inclusive OR is performed between the inverse of the contents of the selected bytes of B and the contents of the same bytes of A. The result is left in A. The contents of B and the rest of the contents of A are undisturbed.

Affected: (A)

MERGE INVERSE OF A INTO B AND CLEAR A

RCH 037741 COPY (1-A,B,B), (0,A) 0 40 37741

A logical inclusive OR is performed between the inverse of the contents of the selected bytes of A and the contents

of the same bytes of B. The result is left in B. The same bytes of the A register are then cleared. The rest of the contents of B and A is undisturbed.

Affected: (A, B) Timing: 1

MERGE INVERSE OF B INTO A AND CLEAR B

RCH 037714 COPY (1-B, A, A), (0, B) 0 40 37714

A logical inclusive OR is performed between the inverse of the contents of the selected bytes of B and the contents of the same bytes of A. The result is left in A. The same bytes of the B register are then cleared. The rest of the contents of A and B is undisturbed.

Affected: (A), (B) Timing: 1

FORM MASK IN B

RCH 037770 COPY M, (-1, B) 0 40 37770

The contents of the selected bytes of B are replaced by ones. The contents of A and the rest of the contents of B are unchanged.

Affected: (B) Timing: 1

FORM MASK IN A

RCH 037707 COPY M, (-1, A) 0 40 37707

The contents of the selected bytes of A are replaced by ones. The contents of B and the rest of the contents of A are unchanged.

Affected: (A) Timing: 1

FORM MASK IN B AND CLEAR A

RCH 037771 COPY M, (-1, B), (0, A) 0 40 37771

The contents of the selected bytes of B are replaced by ones. The same bytes of the A register are cleared. The rest of the contents of B and A is unchanged.

Affected: (A), (B) Timing: 1

FORM MASK IN A AND CLEAR B

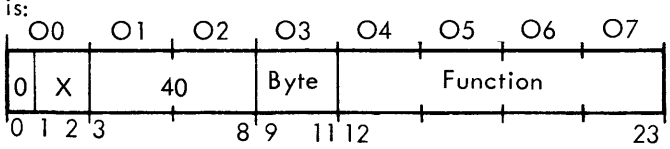
RCH 037717 COPY M, (-1, A), (0, B) 0 40 37717

The contents of the selected bytes of A are replaced by ones. The same bytes of the B register are cleared. The rest of the contents of A and B is unchanged.

Affected: (A), (B) Timing: 1

MODE 2: A/B/X REGISTER CHANGE

This mode allows interchange or modification of information in the A, B, and/or index registers. Byte groups 1, 2, and 3 which may be operated upon in this mode are, respectively, bits 0 through 8, bits 9 through 14, and bits 15 through 23. The tag field normally contains the number of the index register to be operated upon, depending on the setting of the function bits. Bit position 0 must contain a zero and bit positions 1 and 2 must not both be zero. The instruction format for this operation is:



The function control bits are interpreted as follows:

Bit Position	Address Field Value	Function
12	04000	Merge (X1) into index register designated by bits 1 and 2
13	02000	Merge (X2) into index register designated by bits 1 and 2
14	01000	Merge (X3) into index register designated by bits 1 and 2
15		Ignored
16	0200	Merge (B) into index register designated by bits 1 and 2
17	0100	Merge (A) into index register designated by bits 1 and 2
18	040	Merge contents of index register designated by bits 1 and 2 into B
19	020	Merge (A) into B
20	010	Clear B
21	04	Merge contents of index register designated by bits 1 and 2 into A
22	02	Merge (B) into A
23	01	Clear A

The byte-control bits are interpreted as follows:

Instruction Word (RCH)		Affected Registers	
Bit Position	Address Field Value	Bit Positions Selected	Mask for COPY
9	040000	0-8	077700000
10	020000	9-14	077000
11	010000	15-23	0777

At least one byte-control bit must be present in all mode 2 RCH instructions. If the mask (M) is omitted in a COPY instruction, the assembler automatically generates a mask of 077777777.

In the instruction examples to follow, the RCH instructions given are coded for 1-bits in bit positions 9-11 of all instructions.

The index register designated by bits 1 and 2 of a mode 2 RCH instruction is always cleared in the bytes specified. However, an index register designated by bits 12 through 14 is not cleared. Thus, if the index register designated by bits 1 and 2 is the same register as that designated by bits 12 through 14, the designated index register is merged back into itself. It they are not the same, the index register designated by bits 12 through 14 is not affected, but its contents are copied into the index register designated by bits 1 and 2.

Affected: See examples Timing: 1

COPY INDEX INTO A

RCH 07R005, X COPY (X, A) X 40 7R005

The contents of the selected bytes in index X replace the contents of the same bytes in A; the rest of the contents of A are undisturbed. The contents of the selected bytes in index R replace the contents of the same bytes in index X; the contents of index R are undisturbed. The COPY instruction above automatically produces a 1 in bit 12 if X is 1, a 1 in bit 13 if X is 2, and a 1 in bit 14 if X is 3.

Affected: (A), (X) if R does not correspond to X Timing: 1

COPY INDEX INTO B

RCH 07R050, X COPY (X, B) X 40 7R050

The contents of the selected bytes in index X replace the contents of the same bytes in B; the rest of the contents of B are undisturbed. The contents of the selected bytes in index R replace the contents of the same bytes in index X; the contents of index R are undisturbed. The COPY instruction above automatically produces a 1 in bit 12 if X is 1, a 1 in bit 13 if X is 2, and a 1 in bit 14 if X is 3.

Affected: (B), (X) if R does not correspond to X Timing: 1

COPY A INTO INDEX

RCH 070100 COPY (A, X) X 40 70100

The contents of the selected bytes of A replace the contents of the same bytes in index X. The contents of A and the rest of the contents of X are undisturbed.

Affected: (X) Timing: 1

COPY B INTO INDEX

RCH 070200, X COPY (B, X) X 40 70200

The contents of the selected bytes of B replace the contents of the same bytes in index X. The contents of B and the rest of the contents of X are undisturbed.

Affected: (X) Timing: 1

COPY INDEX 1 INTO SPECIFIED INDEX

RCH 074000, X COPY (X1, X) X 40 74000

The contents of the selected bytes of index 1 replace the contents of the same bytes in index X. The contents of index 1 and the rest of the contents of index X are undisturbed.

Affected: (X) Timing: 1

COPY INDEX 2 INTO SPECIFIED INDEX

RCH 072000, X COPY (X2, X) X 40 72000

The contents of the selected bytes of index 2 replace the contents of the same bytes in index X. The contents of index 2 and the rest of the contents of index X are undisturbed.

Affected: (X) Timing: 1

COPY INDEX 3 INTO SPECIFIED INDEX

RCH 071000, X COPY(X3, X) X 40 71000

The contents of the selected bytes of index 3 replace the contents of the same bytes in index X. The contents of index 3 and the rest of the contents of index X are undisturbed.

Affected: (X) Timing: 1

EXCHANGE INDEX AND A

RCH 070105, X COPY (X, A), (A, X) X 40 70105

The contents of the selected bytes of index X and of A are interchanged. The remaining bits in the contents of each register are undisturbed.

Affected: (X), (A) Timing: 1

EXCHANGE INDEX AND B

RCH 070250, X COPY (X, B), (B, X) X 40 70250

The contents of the selected bytes of index X and of B are interchanged. The remaining bits in the contents of each register are undisturbed.

Affected: (X), (B) Timing: 1

CLEAR INDEX

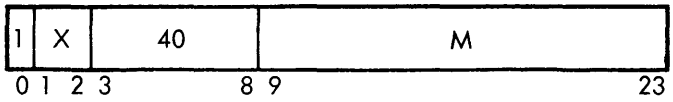
RCH 070000, X COPY (0, X) X 40 70000

The contents of the selected bytes of index X are set to zero. The rest of index X is undisturbed.

Affected: (X) Timing: 1

MODE 3: ADDRESS TO INDEX BASE

AXB ADDRESS TO INDEX BASE



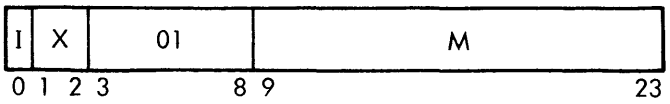
The execution of this instruction in mode 3 places bits 9 through 23 of the instruction field into bits 9 through 23 of the designated index register, X. The rest of index X is undisturbed. The contents of bit position 0 of the instruction must be equal to one.

Affected: (X) Timing: 1

BRANCH GROUP

Branch instructions alter the course of the program by changing the program counter, either conditionally or unconditionally. Branch instructions cannot be repeated with REP.

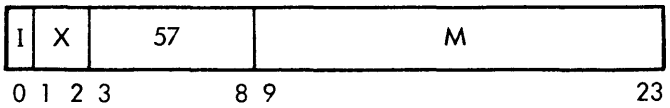
BRU BRANCH UNCONDITIONALLY



The next instruction is taken from the location determined by the effective address.

Affected: (P) Timing: 1

BRX INCREASE INDEX AND BRANCH



The increment value of the index register specified by the index field of instruction BRX is added to the base value of that index register and the result is placed in the index register. The next instruction is taken from the location determined by the effective address unless:

the increment is negative and the resultant base value is less than zero, that is, <00000

or

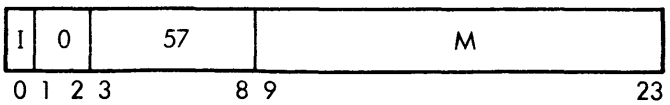
the increment is positive and the resultant base value is beyond the maximum address, that is, >077777.

Then, the next instruction is taken in sequence. Therefore, when the base value crosses the point between the highest address (077777) and lowest address (00000), or vice-versa, no branch occurs.

Indirect addressing may be used to determine the effective address, but indexing cannot be used on any level.

Affected: (P) Timing: 2 if branch
3 if no branch

BRC BRANCH AND CLEAR INTERRUPT



The next instruction is taken from the effective location and, at the same time, the highest priority interrupt channel that is set and has been recognized is cleared. Indirect addressing can be used to determine the effective address, but indexing cannot be used on any level. If the BRC instruction is indirectly addressed, bits 3 through 8 of the word containing the effective address are logically merged with the current contents of the

flag register and the result restored in the flag register; otherwise, the flag register is unchanged.

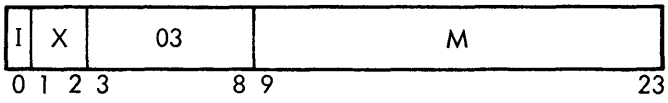
Affected: (P), (F) Timing: 2

Example: BRC *03000

Location	Contents
02000	4 57 03000
03000	0 53 04120

If the BRC instruction in location 02000 is executed and the flag register contains 021 before execution, the next instruction will be taken from location 04120, the highest priority interrupt channel in the active state will be returned to the inactive state, and the flag register will contain 073.

BRM MARK PLACE AND BRANCH



The contents of the program counter (P register) are stored in bit positions 9 through 23 of the effective memory location and control is transferred to the effective location plus one. The status of the flag register indicators are stored in positions 3 through 8 of the effective location and bit positions 0 through 2 are set to zero. An interrupt can occur following this instruction.

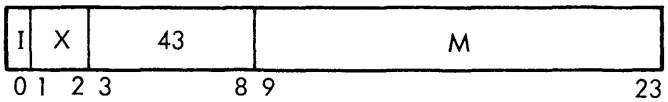
Affected: (P), (EM) Timing: 3

Example: BRM 02500, (P register) = 01517

	(F)	(Location 02500)	(P)
Before execution:	46	0 32 07621	01517
After execution:	46	0 46 01517	02501

This instruction is normally used to enter subroutines where a return to the main program will occur after the subroutine has been completed. BRR is used to return to the main program. BRC, indirectly addressed, is used to return to the main program from an interrupt-servicing routine.

BMA BRANCH AND MARK PLACE OF ARGUMENT ADDRESS



The contents of the program counter register plus one are stored in bit positions 9 through 23 of the effective memory location and control is transferred to the effective memory location plus one. Also occurring in the effective memory location are the following:

1. A one is stored in the indirect address bit
2. The contents of the flag register are stored in bit positions 3 through 8
3. Bit positions 1 and 2 are cleared to zero
4. The flag register is cleared.

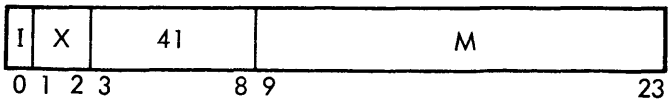
Affected: (F), (P), (EM) Timing: 3

Example: BMA 04030, (P register) = 02000

	(F)	(Location 04030)	(P)
Before execution:	35	4 27 05301	02000
After execution:	00	4 35 02001	04031

The BMA instruction is normally used to enter a subroutine which requires one or more arguments for its operation. In normal use, these arguments can be placed in an "argument list" in memory, with the address of the initial entry of this list placed in the location immediately following the BMA instruction. The subroutine can obtain this entry by indirectly addressing the subroutine entrance location. This procedure will access the contents of the location following the BMA since the BMA placed an indirect address bit in the subroutine entrance along with the address. RETURN BRANCH (BRR) is normally used to return to the main program at one location past the location of the argument address.

BRR RETURN BRANCH



The contents of the effective memory location are incremented by one and the least significant 15 bits are then stored in the program counter (P register). The contents of bit positions 3 through 8 of the effective location and the contents of the flag register are merged, with the result left in the flag register. The next instruction is taken from the address stored in the P register.

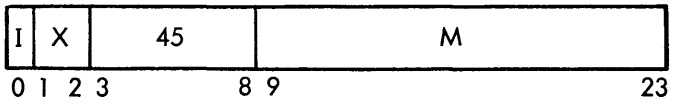
Affected: (F), (P) Timing: 3

Example: BRR 014200, (P register) = 03000

	(F)	(Location 014200)	(P)
Before execution:	23	0 65 0 0400	03000
After execution:	67	0 65 0 0400	00401

TEST AND SKIP GROUP

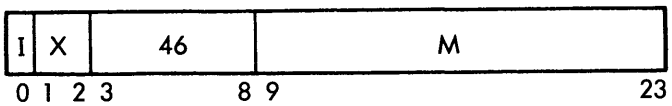
SKE SKIP IF A EQUALS MEMORY



The contents of the A register are compared with the contents of the effective memory location. If the contents of A are equal to the contents of the effective location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of A are not equal to the contents of the effective location, the next instruction in sequence is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

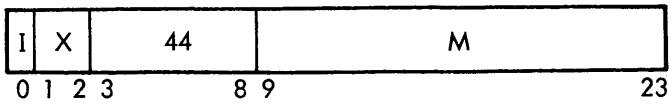
SKG SKIP IF A GREATER THAN MEMORY



The contents of the A register are algebraically compared with the contents of the effective memory location. If the contents of A are greater than the contents of the effective location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of A are less than or equal to the contents of the effective location, the next instruction in sequence is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

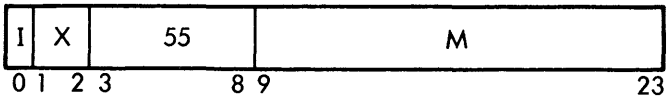
SKL SKIP IF A LESS THAN OR EQUAL TO MEMORY



The contents of the A register are algebraically compared with the contents of the effective memory location. If the contents of A are less than or equal to the contents of the effective location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of A are greater than the contents of the effective location, the next instruction in sequence is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

SKM SKIP IF A EQUALS MEMORY ON B MASK



Selected bits of the contents of the A register are compared with the corresponding bits in the contents of the effective memory location. If the selected bits in the contents of A are identical to the selected bits in the contents of the effective memory location, the next instruction in sequence is skipped and the following instruction is executed. If the selected bits in the contents of the A register are not identical to the selected bits of the contents of the effective location, the next instruction in sequence is executed.

The bits in A to be compared are selected by placing one-bits in the corresponding bit positions of the B register and zero-bits in the remaining bits of B.

The contents of A, B, and the effective location are considered to be unsigned, 24-bit, non-numeric quantities.

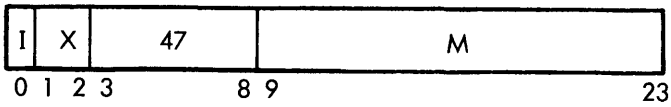
Affected: (P) Timing: 2 if no skip
3 if skip

Example: SKM M

(A)	(B)	(M)
00043007	00177000	57643200

Since a comparison is made in bit positions 8 through 14 only (as determined by B), and (A) = (M), a skip will occur.

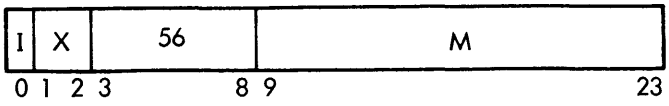
SKU SKIP IF A UNEQUAL TO MEMORY



The contents of the A register are compared with the contents of the effective memory location. If the contents of A are unequal to the contents of the effective location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of A are equal to the contents of the effective location, the next instruction in sequence is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

SKQ SKIP IF MASKED QUANTITY IN A IS GREATER THAN MEMORY



Selected bits of the contents of the A register are compared with the corresponding bits in the contents of the

effective memory location. If the selected bits in the contents of A are greater than the selected bits in the contents of the effective location, the next instruction in sequence is skipped and the following instruction in sequence is executed. If the selected bits in the contents of A are less than or equal to the selected bits in the contents of the effective location, the next instruction in sequence is executed.

The bits in A to be compared are selected by placing one-bits in the corresponding bit positions of the B register and zero bits in the remaining bits of B.

The contents, or any selected subset of the contents, of A and the effective memory location are considered to be 24-bit, or fewer, positive integers. The contents of the B register are considered to be an unsigned, 24-bit, non-numeric quantity.

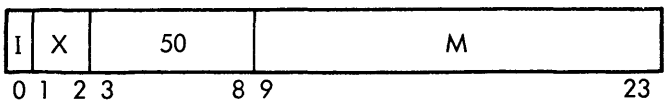
Affected: (P) Timing: 2 if no skip
3 if skip

Example: SKQ M

(A)	(B)	(M)
04703621	00077777	00010413

The mask in B will compare bit positions 9 through 23. Since those positions of memory are greater than those positions of A, no skip will occur.

SKF SKIP IF FLOATING EXPONENT IN B GREATER THAN OR EQUAL TO EXPONENT IN MEMORY



Bits 15 through 23 in the contents of the B register are compared with bits 15 through 23 in the contents of the effective memory location. If the contents of the 9-bit field of the B register are algebraically greater than or equal to the contents of the 9-bit field of the effective location, the next instruction in sequence is skipped and the following instruction is executed. If the contents of the 9-bit field of the B register are algebraically less than the contents of the 9-bit field of the effective location, the next instruction in sequence is executed.

The contents of the 9-bit field, bits 15 through 23, of the B register and the effective location are considered to be 8-bit numeric quantities (bits 16-23) with a sign in bit 9. All negative numbers are considered to be in two's complement form.

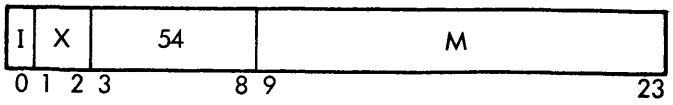
Affected: (P) Timing: 2 if no skip
3 if skip

Example: SKF M

(B)	(M)
34761204	05637763

Since the exponent in B is positive and therefore algebraically greater than the negative exponent in memory, a skip will occur.

SKA SKIP IF A AND MEMORY DO NOT COMPARE ONES ANYPLACE



The contents of the A register are compared, bit by bit, with the contents of the effective memory location. If the contents of the A register and the contents of the effective location do have at least one pair of 1-bits in corresponding bit positions, the next instruction in sequence is executed.

The instruction logically ANDs corresponding bits in A and memory, based on the following table:

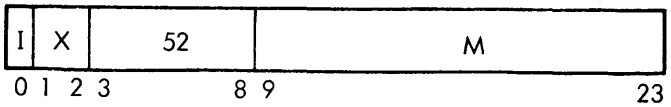
(A)	(Memory)	Result
0	0	0
0	1	0
1	0	0
1	1	1

If the result produces a 1 in any bit position, a skip will not occur. Different configurations of the memory word result in a wide variety of conditional instructions for use by the programmer. Some examples are:

Memory Configuration	Effect
40000000	Skip if A is positive
77777777	Skip if A = 0
00000001	Skip if A is even

Affected: (P) Timing: 2 if no skip
3 if skip

SKB SKIP IF MEMORY AND B COMPARE ONES ANYPLACE



The contents of the B register are compared, bit by bit, with the contents of the effective memory location. If the contents of the B register and the contents of the effective location do have at least one pair of 1-bits in corresponding bit positions, the next instruction in sequence is skipped and the following instruction is

executed. If the contents of the B register and the contents of the effective location do not have at least one pair of 1-bits in corresponding bit positions, the next instruction in sequence is executed.

The instruction logically ANDs corresponding bits in B and memory, based on the following table.

(B)	(Memory)	Result
0	0	0
0	1	0
1	0	0
1	1	1

If the result produces a 1 in any bit position, a skip will occur. Different configurations of the memory word result in a wide variety of conditional instructions for use by the programmer.

Some representative ones are:

Memory Configuration	Effect
40000000	Skip if B is negative
77777777	Skip if B $\neq$ 0
00000001	Skip if B is odd

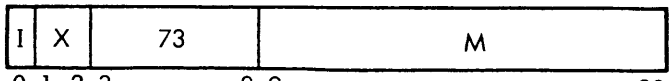
Affected: (P) Timing: 2 if no skip
3 if skip

SKN SKIP IF MEMORY NEGATIVE

If the contents of the effective memory location are negative, the next instruction in sequence is skipped and the following instruction is executed. If the contents of the effective location are positive or zero, the next instruction in sequence is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

SKR REDUCE MEMORY, SKIP IF NEGATIVE



The contents of the effective memory location are reduced by one and the result replaced into the same location and then tested. If the contents of the effective memory location are then negative, the next instruction in sequence is skipped and the following instruction is executed. If the contents of the effective location are positive or zero, the next instruction in sequence is executed.

Overflow will not set the OVERFLOW indicator.

Affected: (EM), (P) Timing: 3

SKP SKIP IF BIT SUM EVEN, PARITY TEST

Selected bits in the contents of the effective memory location are summed and the resulting sum is tested for being odd or even. If the selected-bit sum is even, the next instruction in sequence is skipped and the following instruction is executed. If the selected-bit sum is odd, the next instruction in sequence is executed.

The bits to be summed are selected by placing 1-bits in the corresponding bit positions of the B register and 0-bits in the remaining bits of B.

The contents of B and the effective location are considered to be unsigned, 24-bit, non-numeric quantities.

Due to the bit selection process, the roles of the B register and the effective location can be reversed, forming a test on the contents of the B register.

Affected: (P) Timing: 2 if no skip
3 if skip

Example: SKP M

(B)	(M)
04003777	61473556

The bits to be summed are 3 and 13-23. Since memory has an even number of one-bits in these positions, a skip will occur.

Logically this instruction sums the ones in the logical product (AND) of memory and B. Therefore, the "mask" could be in either memory or B.

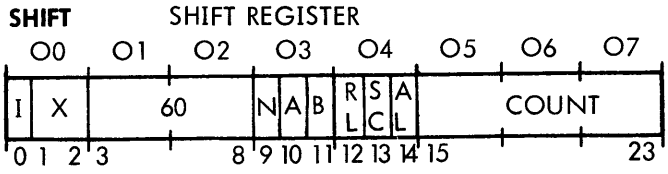
This instruction can be used for internal parity checking of data received from some input source.

SHIFTING OPERATIONS

The shift instructions operate on the contents of the A and B registers and offer complete facility for right shifting, left shifting, cycling, and normalizing the contents of these two registers. Each register may be shifted independently and both may be independently shifted simultaneously. The A and B registers may be

combined to form a double-length register whose double-length contents can also be shifted, cycled or normalized. This double-length register is referred to by the D in shift instruction mnemonics. When the contents of the A and B registers are shifted right together, bits from bit position 23 of the A register are shifted into bit position 0 of the B register. When the double register is shifted left, bits from bit position 0 of the B register are shifted into bit position 23 of the A register.

The instruction format for shifting, cycling, and normalizing is:



A 1 in bit position 9 is used to indicate a normalize instruction (see Normalize Instructions). All other shifts have a 0 in this position.

Bit positions 10-11 determine the combination of registers for a shift in the following manner:

(10)	(11)	Registers Shifted
0	0	A and B together as a double register
0	1	B only
1	0	A only
1	1	A and B independently

The type of shift is specified in positions 12-14 as shown below:

Bit Position	Type
12	0 = right 1 = left
13	0 = shift 1 = cycle
14	0 = arithmetic 1 = logical

SHIFT/CYCLE INSTRUCTIONS

The contents of the A, B, or double register may be shifted both logically and arithmetically. In an arithmetic right shift, with either the A, B, or the double register, the sign bit of the contents of the register does not shift right but the sign bit value 0 or 1 is copied into the next bit position as it is vacated for each bit shifted. Hence, in a right arithmetic shift, the sign bit is said to be extended to the right. In an arithmetic left shift, with either the A or the double register, the sign bits of the contents of the A register shift, but if during the shift a bit value different from the sign is shifted into the sign bit position of A, the OVERFLOW indicator is set. The B register is shifted left arithmetically in the same manner except that no test is made for 0 sign change and the OVERFLOW indicator is not set.

The contents of the A, B, or double register can be shifted logically, that is, the entire 24 or 48 bits of the contents of the register are considered as a unit. The vacated bit positions of a register during a left or right shift are set to zero.

The two registers, A and B, can be shifted at the same time, but independently in both the arithmetic and logical modes.

The contents of the A, B, or double registers may be cycled using the shift instructions. When the contents of a register are cycled, the bits which are shifted from one end of the register are copied into the other end of the register. The contents of registers A and B can also be independently cycled, but at the same time.

Examples:

	ARSA 8	LRSA 8	CRSA 8
(A) before execution	40372461	40372461	40372461
(A) after execution	77700765	00100765	14300765

Shift instructions cannot be repeated with REP.

When a shift instruction is executed, the amount of the shift is determined by bit positions 15 through 23 of the effective address of the instruction. These nine bits are treated as an unsigned count. If the initial count is equal to zero, no shifting occurs. If the initial count is greater than 48, it is set equal to 48 before shifting begins. Once the shift begins, the count is reduced by one for each position shifted until it reaches zero. The times in memory cycles for both left and right shifts with a count of C are:

Left Shift	Time	Right Shift
$C \leq 6$	2	$C \leq 3$
$6 < C \leq 28$	3	$3 < C \leq 14$
$28 < C \leq 48$	4	$14 < C \leq 25$
	5	$25 < C \leq 36$
	6	$36 < C \leq 47$
	7	$C = 48$

Since the type of shift and number of shifts are determined by bits 9 through 23 of the effective address, indirect addressing and/or indexing drastically alter the action specified in a shift instruction. When procuring the effective location for a shift instruction,

15-bit indexing is performed with all indirectly addressed operands, and

9-bit indexing is performed with all directly addressed operands.

That is, indexing with a direct address can affect only the 9-bit shift count.

The following examples of coding an ARITHMETIC RIGHT SHIFT A (ARSA) instruction give an idea of the variable facility offered in the shift instruction through the use of indexing and indirect addressing.

Examples:

The following sample instructions indicate three different ways that the shift instruction can be written to shift a number contained in register A four places to the right.

The shift instruction:

ARSA 04 (0 60 20004)

is a right shift instruction that causes the contents of register A to be shifted right four places. The index and indirect address fields are zero, so the instruction type and shift count are found in the instruction word itself.

The shift instruction:

ARSA 0, 1 (1 60 20000)

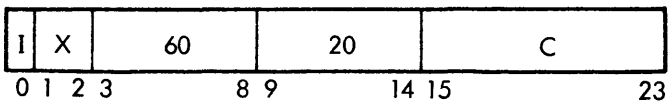
has an effective operand of 020000 + (index register 1). Assuming the contents of (index register 1) to be 00004, the effective operand is 020004. Therefore, the above instruction is again a right shift instruction with a count of four.

The shift instruction:

SHIFT *01000 (4 60 01000)

has an effective operand which is the contents of memory location 01000. Assuming the contents of location 01000 to be 00020004, the effective operand is 020004. Therefore, the above instruction is an arithmetic right shift instruction with a count of four.

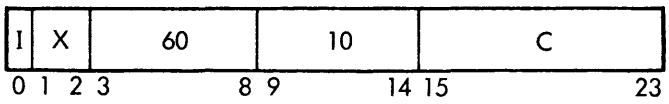
ARSA ARITHMETIC RIGHT SHIFT A



The contents of the A register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A does not shift, but its value is copied into the vacated bit positions of the shifted number. Bits shifted past bit position 23 in A are lost.

Affected: (A) Timing: 2-7

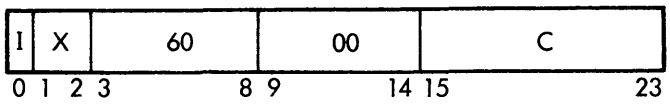
ARSB ARITHMETIC RIGHT SHIFT B



The contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of B does not shift, but its value is copied into the vacated bit positions of the shifted number. Bits shifted past bit position 23 in B are lost.

Affected: (B) Timing: 2-7

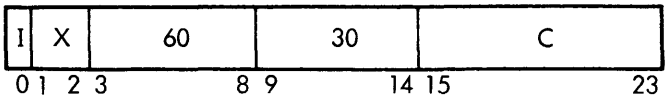
ARSD ARITHMETIC RIGHT SHIFT DOUBLE



The contents of the double register (that is, A and B registers) are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A does not shift, but its value is copied into the vacated bit positions of the shifted number. The bit in the sign position of B is shifted as any other bit in B. Bits shifted out of A_{23} are shifted into B_0. Bits shifted past position 23 in B are lost.

Affected: (A, B) Timing: 2-7

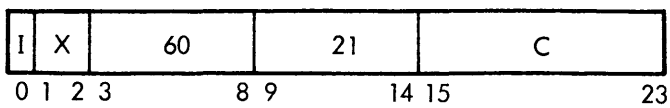
ARST ARITHMETIC RIGHT SHIFT TWIN (A AND B)



The contents of the A register and the contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A does not shift, but its value is copied into the vacated bit positions of the shifted number in A. The bit in the sign position of B does not shift, but its value is copied into the vacated bit positions of the shifted number in B. Bits shifted past position 23 in both A and B are lost. No bits shifted out of A_{23} are shifted into B_0.

Affected: (A), (B) Timing: 2-7

LRSA LOGICAL RIGHT SHIFT A

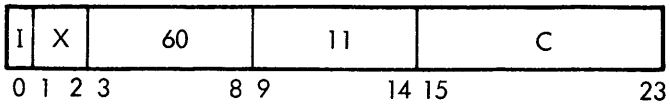


The contents of the A register are shifted right the number of places specified in bits 15 through 23 of the

effective operand. The bit in the sign position of A shifts along with the rest of the number and the vacated bit positions of the shifted number are filled with zeros. Bits shifted past position 23 in A are lost.

Affected: (A) Timing: 2-7

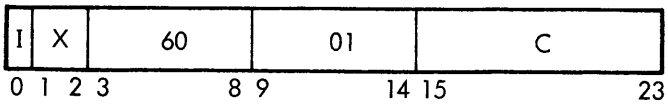
LRSB LOGICAL RIGHT SHIFT B



The contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of B shifts along with the rest of the number and the vacated bit positions of the shifted number are filled with zeros. Bits shifted past position 23 in B are lost.

Affected: (B) Timing: 2-7

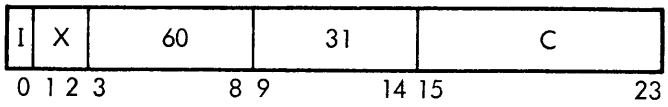
LRSD LOGICAL RIGHT SHIFT DOUBLE



The contents of the double register (that is, A and B together) are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A and the bit in the sign position of B are shifted as any other bits in A and B. The vacated bit positions of the shifted number in A are filled with zeros. Bits shifted out of A_{23} are shifted into B_0. Bits shifted past position 23 in B are lost.

Affected: (A, B) Timing: 2-7

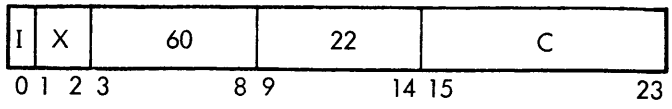
LRST LOGICAL RIGHT SHIFT TWIN (A AND B)



The contents of the A register and the contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A and the bit in the sign position of B are shifted as any other bits in A and B. The vacated bit positions of the shifted number in A and the vacated bit positions of the shifted number in B are filled with zeros. Bits shifted past position 23 in both A and B are lost. Bits shifted out of A_{23} are not shifted into B_0, but are lost.

Affected: (A), (B) Timing: 2-7

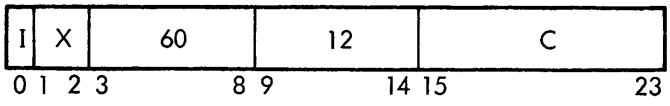
CRSA CIRCULAR RIGHT SHIFT A



The contents of the A register are shifted right the number of places specified in bits 15 through 23 of the effective operand. Bits which are shifted out of bit position 23 are copied into bit position 0. The register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (A) Timing: 2-7

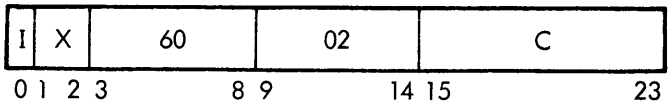
CRSB CIRCULAR RIGHT SHIFT B



The contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. Bits which are shifted out of bit position 23 are copied into bit position 0. The register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (B) Timing: 2-7

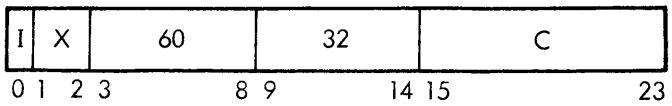
CRSD CIRCULAR RIGHT SHIFT DOUBLE



The contents of the double register are shifted right the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of B is shifted as any other bit in B. Bits shifted out of A_{23} are shifted into B_0. Bits which are shifted from bit position 23 of B are copied into bit position 0 of A. The double-length register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (A, B) Timing: 2-7

CRST CIRCULAR RIGHT SHIFT TWIN (A AND B)

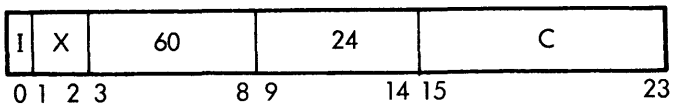


The contents of the A register and the contents of the B register are shifted right the number of places specified in bits 15 through 23 of the effective operand. Bits shifted from bit position 23 of A are copied into position 0 of A. Bits shifted from bit position 23 of B are copied

into position 0 of B. The bit in the sign position of A and the bit in the sign position of B are shifted as any other bits in A and B. No bits are transferred from one register to the other. Both registers are treated as if they were circular and are cycled onto themselves. No bits are lost.

Affected: (A), (B) Timing: 2-7

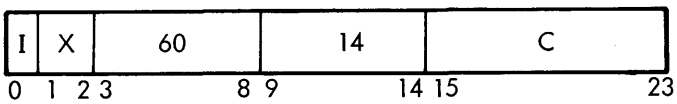
ALSA ARITHMETIC LEFT SHIFT A



The contents of the A register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits are shifted left through the sign position of A but when a bit which is different in value from the original sign is shifted into the sign position, the OVERFLOW indicator is set. Bits shifted past position 0 in A are lost. The vacated bit positions on the right end of the register are filled with zeros.

Affected: (A), Of Timing: 2-5

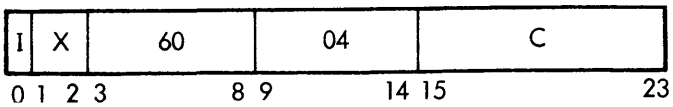
ALSB ARITHMETIC LEFT SHIFT B



The contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits are shifted left through the sign position of B but when a bit which is different in value from the original sign is shifted into the sign position, the OVERFLOW indicator is set. Bits shifted past position 0 in B are lost. The vacated bit positions on the right end of the register are filled with zeros.

Affected: (B), Of Timing: 2-5

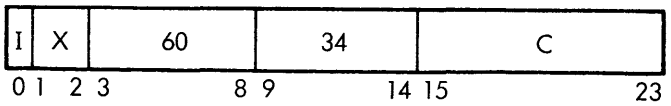
ALSD ARITHMETIC LEFT SHIFT DOUBLE



The contents of the double register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits are shifted left through the sign position of A, but when a bit is shifted into the sign position which is different in value from the original sign, the OVERFLOW indicator is set. Bits shifted out of B_0 are shifted into A_{23}. Bits shifted past position 0 in A are lost. The vacated bit positions on the right end of the B register are filled with zeros.

Affected: (A, B), Of Timing: 2-5

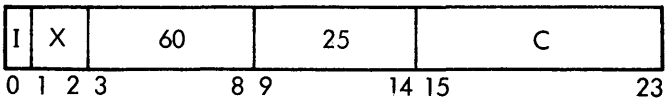
ALST ARITHMETIC LEFT SHIFT TWIN (A AND B)



The contents of the A register and the contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits are shifted left through the sign position of A, but when a bit which is different in value from the original sign is shifted into the sign position, the OVERFLOW indicator is set. Bits are shifted left through the sign position of B but changes in value of this bit do not cause the OVERFLOW indicator to be set. Bits shifted past position 0 in both A and B are lost. No bits shifted out of B_0 are shifted into A_{23}. The vacated bit positions on the right end of the A and B registers are filled with zeros.

Affected: (A), (B), Of Timing: 2-5

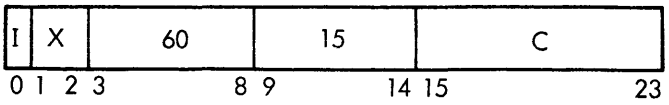
LLSA LOGICAL LEFT SHIFT A



The contents of the A register are shifted left the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A shifts along with the rest of the number. The vacated bit positions in the right end of the register are filled with zeros. Bits shifted past position 0 in A are lost.

Affected: (A) Timing: 2-5

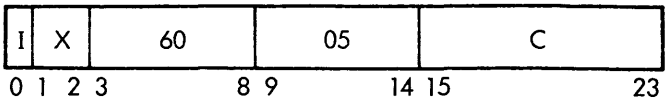
LLSB LOGICAL LEFT SHIFT B



The contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of B shifts along with the rest of the number. The vacated bit positions in the right end of the register are filled with zeros. Bits shifted past position 0 in B are lost.

Affected: (B) Timing: 2-5

LLSD LOGICAL LEFT SHIFT DOUBLE

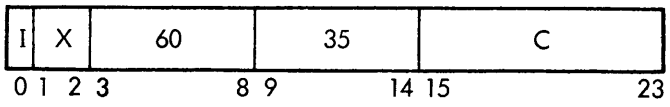


The contents of the double register are shifted left the number of places specified in bits 15 through 23 of the effective operand. The bits in the sign positions of the

A and B registers shift along with the rest of the number. The vacated bit positions in B of the shifted number are filled with zeros. Bits shifted out of B_0 are shifted into A_{23}. Bits shifted past position 0 in A are lost.

Affected: (A, B) Timing: 2-5

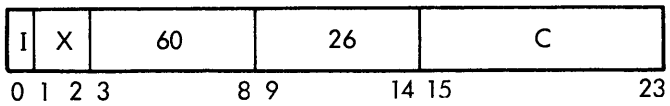
LLST LOGICAL LEFT SHIFT TWIN (A AND B)



The contents of the A register and the contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. The bit in the sign position of A and the bit in the sign position of B are shifted as any other bits in A and B. The vacated bit positions of the shifted number in A and the vacated bit positions of the shifted number in B are filled with zeros. Bits shifted past position 0 in A and B are lost. Bits shifted out of B_0 are not shifted into A_{23}.

Affected: (A), (B) Timing: 2-5

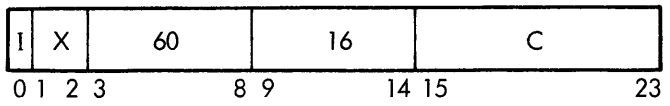
CLSA CIRCULAR LEFT SHIFT A



The contents of the A register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits which are shifted out of bit position 0 are copied into bit position 23. The sign bit shifts like any other bit. The register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (A) Timing: 2-5

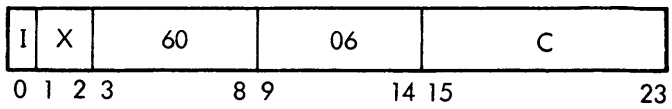
CLSB CIRCULAR LEFT SHIFT B



The contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits which are shifted out of bit position 0 are copied into bit position 23. The sign bit shifts like any other bit. The register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (B) Timing: 2-5

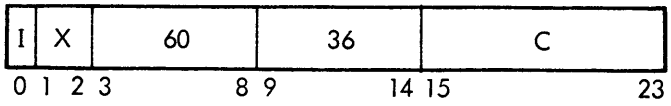
CLSD CIRCULAR LEFT SHIFT DOUBLE



The contents of the double register are shifted left the number of places specified in bits 15 through 23 of the effective operand. The bits in the sign positions of A and B are shifted as in any other bit in the number. Bits shifted out of B_0 are shifted into A_{23}. Bits which are shifted from bit position 0 of A are copied into bit position 23 of B. The double-length register is treated as if it were circular and is cycled onto itself. No bits are lost.

Affected: (A, B) Timing: 2-5

CLST CIRCULAR LEFT SHIFT TWIN (A AND B)



The contents of the A register and the contents of the B register are shifted left the number of places specified in bits 15 through 23 of the effective operand. Bits shifted from bit position 0 of A are copied into position 23 of A. Bits shifted from bit position 0 of B are copied into position 23 of B. The bit in the sign position of A and the bit in the sign position of B are shifted as any other bits in A and B. No bits are transferred from one register to the other. Both registers are treated as if they were circular and cycled onto themselves. No bits are lost.

Affected: (A), (B) Timing: 2-5

NORMALIZE INSTRUCTIONS:

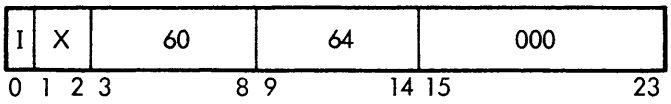
In the normalize instructions, the count (bits 15 through 23 of the instruction) is not inspected. Instead, the contents of the A (or double register) are shifted left until the contents of the sign bit of A and the contents of bit position one of A are not equal, and vacated bit positions are filled with zeros.

A count is kept of the number of shifts made, and when the normalize operation is complete, this count is subtracted from the contents of index register 1. If the count should exceed 48 (060), the indication is that the contents of the A (or double register) were initially equal to zero. In this case, 48 (060) is subtracted from index register 1. If index register 1 overflows negatively into bit position 15, the OVERFLOW indicator is set.

The timing for a normalize instruction is the same as for a left shift. The maximum time is five cycles.

It is important to note the overflow condition. The normalize count is subtracted from the five low-order octals, address portion, of X1 but overflow is determined on the basis of the low three octals, i.e., as though these octals contained a floating-point exponent. Therefore, a shift count of 20 subtracted from an X1 of 1010410 results in an X1 of 1010370 with overflow set, i.e., 01370 would be a valid address but 370 would not be a valid exponent.

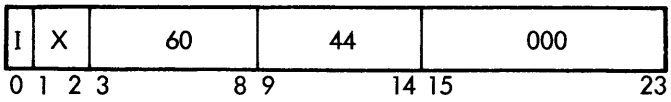
NORA NORMALIZE A



The contents of the A register are shifted left until a bit which is not equal to the bit in the sign position of A, position 0, appears in position 1 of A.

Affected: (A), (X1), Of Timing: 2-5

NORD NORMALIZE DOUBLE



The contents of the double register (that is, A and B together) are shifted left until a bit which is not equal to the bit in the sign position of A, position 0, appears in position 1 of A.

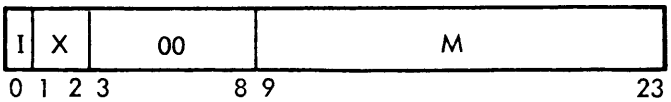
Affected: (A), (B), (X1), Of Timing: 2-5

Example:

NORD	(A)	(B)	(X1)	Of
Before execution:	00004632	76124035	00000000	0
After execution:	23153705	20164000	00077765	1

CONTROL GROUP

HLT HALT



When this instruction is executed, the computer halts computation, sets the HALT flip-flop, and turns on the HALT light on the console. To resume computation, the operator must first press the IDLE switch. The instruction whose address is contained in the P register will be executed when the operator presses either the RUN or STEP switch.

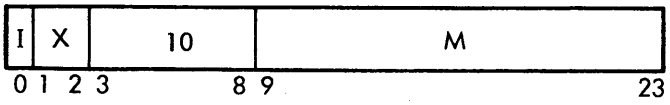
This instruction cannot be repeated with REP, and indirect addressing and indexing do not apply.

When HALT (HLT) is executed, all internal computation within the central processor will cease at the end of the present instruction being executed. If a data channel operation is in progress, it will continue until completed.

If an interrupt occurs while the computer is halted due to the execution of HALT (HLT), the interrupt is acknowledged as usual. The execution of BRANCH AND CLEAR INTERRUPT (BRC) to clear the interrupt and return to the main program causes the next instruction following HLT to be executed and the program to continue.

Affected: HALT (flip-flop and indicator) Timing: 1

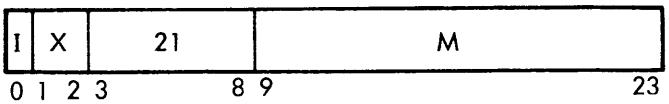
NOP NO OPERATION



The execution of this instruction has no effect on any register, indicator, or memory. This instruction can be repeated with REP, indirect addressing and indexing do apply, and address calculation proceeds in a manner identical to a load instruction. However, when the effective operand is accessed from memory, no further operation is performed.

Affected: None Timing: 2

EXU EXECUTE



The execution of this instruction causes the contents of the effective memory location to be executed as an instruction without altering the contents of the program counter. If the effective location is not a branch, skip, or another EXU instruction, the next instruction in sequence after the EXU is executed following the execution of the contents of the effective location.

If the contents of the effective memory location are a branch instruction, program control will go to the effective location of the branch and not to the next instruction in sequence following EXU.

If the contents of the effective memory location are a skip instruction, then, depending on the skip decision, program control will go to the next instruction or the next instruction plus one following the EXU.

If the contents of the effective memory location are another EXU, the above process is continued identically, with the normal return being the initial EXU instruction location plus one. This process can be cascaded indefinitely.

This instruction cannot be repeated with REP.

Affected: None Timing: 1

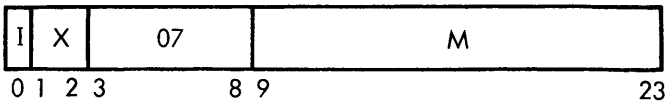
Example:

The SUBTRACT (SUB) instruction is executed in sequence using EXU:

Location	Instruction
0200	LDA 01000
0201	EXU 02000
0202	STA 01500
⋮	⋮
01000	00000005
⋮	⋮
01500	00000000
⋮	⋮
02000	SUB 03000
⋮	⋮
03000	00000004

After execution of STORE A (STA), the contents of location 01500 are 00000001.

INT INTERPRET

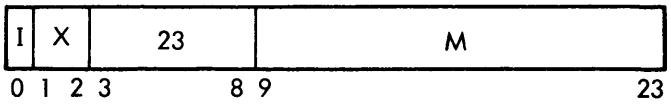


INT loads the contents of bit positions 3 through 8 of the effective memory location into bit positions 18 through 23 of index register 2 and sets bit positions 9 through 17 of index register 2 to zero; INT leaves bit positions 0 through 8 of index register 2 unaffected. This instruction can use an index register to modify the effective memory location at any level.

If bit position 1 of the effective location contains a 1, the computer skips the next instruction after INT and executes the following instruction. If bit position 1 contains a 0, the computer executes the next instruction after INT.

Affected: (X2) Timing: 2 if no skip
3 if skip

REP REPEAT INSTRUCTION IN M



This instruction provides the facility to execute an instruction repeatedly using these additional features:

1. The number of instruction executions desired can be given.

2. Skip class instructions can be used to make a repeated test on a sequence of data.

3. Between each data word operated upon, the addresses of the data words may be incremented or decremented by a 3-digit octal number. This facility allows repeated instructions to jump through a table and operate on selected data.

The instruction contained in the effective location is executed repeatedly until the specified number of data operands have been operated upon. Then the next instruction in sequence after the REP is executed. If the instruction in the effective location is a skip class test instruction, the instruction is executed repeatedly until the number of data operands have been operated upon or until a skip occurs. If no skip occurs, the next instruction in sequence after REP is executed. If a skip occurs, the next instruction in sequence after REP is skipped and the following instruction is executed.

Index registers X1 and X2 are used with REP.

Prior to execution,

1. Index register X2 must be loaded with the "negative repeat count," which is the two's complement of the maximum number of successive instruction executions desired,

2. Bits 9 through 23 of index register X1 must be loaded with the effective address to be used by the repeated instruction,

3. Bits 0 through 8 of index register X1 must be loaded with the desired increment value.

When REP is executed, the instruction code only of the effective memory location of the instruction to be repeated is used since the data addresses are taken from index register X1. Each time the repeated instruction is executed, index register X2 is counted up by one and the effective address in index register X1 is modified by adding the signed increment to the base address. When index register X2 equals zero, the instruction being repeated is terminated and the instruction sequence after REP is executed. If index register X2 is not equal to zero, the instruction is again repeated, using

the modified contents of index register X1 as the effective address. No other indirect addressing or indexing can occur in the repeated instruction except as described above. When the repeat sequence is terminated, index register X1 contains the last address referred to plus the increment. If X2 is initially zero, the instruction in M is repeated 32,768 times unless otherwise terminated. The repeat count counts upward and overflows into the increment field.

Double-precision instructions may be repeated by means of the following procedure. The starting address must be the location of the least significant half of the first operand. In other words, if the first double-precision operand of a group that is to be operated on in the repeat mode is in location M, then the address M + 1 is loaded into index register X1 before starting the repeat operation. For double-precision, the increment in index register 1 should be equal to or greater than two.

The user can use the complete indexing and indirect addressing facility to find the repeated instruction (noting, of course, that X1 and X2 must additionally correspond to the count increment and data address).

This instruction can effectively provide search capabilities when used with the indicated Skip Class Instruction:

1. Search for equality (SKE)
2. Search for inequality (SKU)
3. Search for selective equality (SKM)
4. Search for number larger than or equal to A (SKL)
5. Search for number strictly less than A (SKG)
6. Search for negative number (SKN)
7. Search for zero number (SKE, (A) = 0)
8. Search for non-zero number (SKU, (A) = 0)
9. Search for odd number (SKB, (B) = 00000001)
10. Search for even number (SKA, (A) = 00000001)
11. Search for number having ones in bit positions specified by B (SKM, (A) = 77777777, (B) = MASK)
12. Search for number having zeros in bit positions specified by B (SKM, (A) = 00000000, (B) = MASK)

In addition to search capability, the instruction provides these high-speed operations:

1. High-speed parallel input/output when the instruction to be repeated is either a PARALLEL INPUT (PIN) or PARALLEL OUTPUT (POT).
2. Selective memory clear when the instruction to be repeated is STORE A (STA) and the A register contains zero.
3. High-speed addition of successive numbers when the instruction to be repeated is ADD M TO A (ADD).
4. High-speed addition of a single constant to a group of successive memory locations if the instruction to be repeated is ADD A TO M (ADM).
5. Moving all words in a table to the following memory location if the instruction to be repeated is EXCHANGE M AND A (XMA).

Affected: (X1), (X2), and other registers as determined by the instruction to be repeated

Timing: 3+ N(T-1)
N = Repeat count
T = Execution time of repeated instruction

Example: Repeat Searches

Assume these initial conditions for the following examples of search operations using REPEAT INSTRUCTION IN M (REP).

Register/Location	Contents
Index X1	00101000
Index X2	00077774
Location AAA	32000000
01000	10000000
01001	20000001
01002	32000000
01003	04000000

1. SKIP IF A EQUALS M (SKE)

Assume bits 3 through 8 of location M contain 45, the instruction code for SKIP IF A EQUAL M (SKE).

When this sequence of coding:

```
        LDA     AAA
        REP     M           (M) = X 45 XXXXX
NOSKIP  BRU     NOTEQ
SKIP    BRU     EQUL
```

is executed, the instruction in location SKIP is executed next after the execution of REP. At this time, X1 contains 00101003 (the address plus one of the equal operand).

2. SKIP IF A LESS THAN OR EQUAL TO M (SKL)

Assume bits 3 through 8 of location M contain 44, the instruction code for SKIP IF A LESS THAN OR EQUAL TO M (SKL). Let the contents of location CCC be 02400000.

When this sequence of coding:

```
          LDA   CCC
          REP   M          (M) = X 44 XXXXX
NOSKIP    BRU   NLTE
SKIP      BRU   LESEQ
```

is executed, a search is made for a number to which 024000000 is less than or equal. In this case, it is not found. Therefore, instruction BRU NLTE is executed after REP. At this time, X1 contains 00101004.

3. SEARCH FOR ODD NUMBER IN TABLE

Assume bits 3 through 8 of location CCC contain 52 (octal for SKIP IF M AND B DO COMPARE ONES).

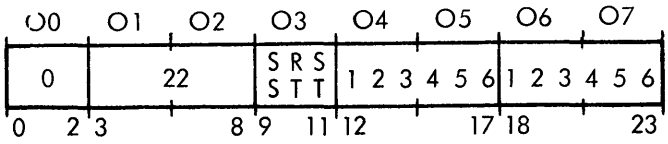
Let location BBB contain 00000001
and location CCC contain 05200000

When this coding:

```
          LDB   BBB
          REP   CCC
NOSKIP    BRU   EVEN
SKIP      BRU   ODD
```

is executed, a search occurs for a number with a 1 in its least significant bit.

FLAG FLAG REGISTER CONTROL AND SENSE SWITCH TESTS



This instruction sets, resets, and/or tests selected bit positions in the flag register. One mode of its operation also selectively tests the condition of the SENSE switches on the 9300 console. This instruction sets flag bits to one; it resets them to zero.

The flag register is a 6-bit register which is displayed on the control console (see Figure 5-1) as six indicators, one of which is the OVERFLOW indicator (bit one). The other five indicators are used by the programmer as he desires.

When instruction bit number 9 is made zero, the instruction refers only to the flag register. Bits 12 through 17 refer to reset operations; bits 18 through 23 refer to set operations. In this mode, bits 12 and 18 refer to flag indicator 1, bits 13 and 19 refer to flag indicator 2, and so on, with bits 17 and 23 referring to flag indicator 6. To reset or reset test an indicator, a one is placed in the corresponding bit position in bits 12 through 17. To set or set test an indicator, a one is placed in the corresponding bit position in bits 18 through 23.

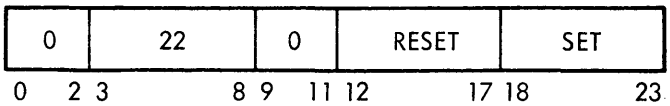
This instruction requires one cycle if no skip occurs, two cycles if a skip occurs.

When instruction control bit 9 is set to one, the instruction refers only to the SENSE switches. The six SENSE switches are referred to by the bits in 12 through 17 and 18 through 23 exactly as the flag indicators above.

Bits 10 and/or 11 can also be used when bit 9 is set to one. This produces a test which is the logical merge of the SWT with the test portion of FRTS and/or FSTR, i.e., no skip occurs if any test is met. No setting or resetting of the flags occurs in any case. Therefore, with bits 10 and 11 both set, a test is produced which does not skip if any SENSE switch or flag specified by bits 12 through 17 is reset or if any SENSE switch or flag specified by bits 18 through 23 is set.

This instruction cannot be repeated with REP.

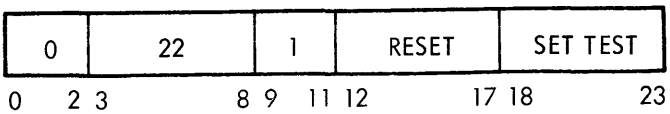
FIRS FLAG INDICATORS RESET AND SET



For each 1-bit in bit positions 12 through 17 the corresponding flag indicator is reset. For each 1-bit in bit positions 18 through 23 the corresponding flag indicator is set. If a flag is designated in both the reset and set fields, its state is unconditionally changed.

Affected: (F) Timing: 1

FSTR FLAG INDICATOR SET TEST/RESET

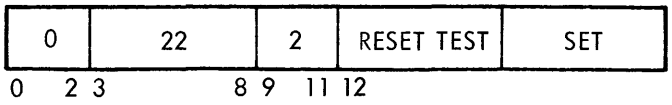


If any of the flag indicators specified in bit positions 18 through 23 is set, the next instruction in sequence is executed. If none of the specified flags is set, the next instruction in sequence is skipped and the following instruction is executed.

After testing, any flag specified by bit positions 12 through 17 is unconditionally reset to 0.

Affected: (F), (P) Timing: 1 if no skip
2 if skip

FRTS FLAG INDICATOR RESET TEST/SET

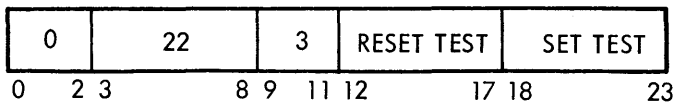


If any of the flag indicators specified in bit positions 12 through 17 is reset (equal to 0), the next instruction in sequence is executed. If none of the specified flags is reset, the next instruction in sequence is skipped and the following instruction is executed.

After testing, any flag specified in bit positions 18 through 23 is unconditionally set to 1.

Affected: (F), (P) Timing: 1 if no skip
2 if skip

FRST FLAG INDICATOR RESET/SET TEST



If any of the flag indicators specified in bit positions 12 through 17 is reset or if any of the flag indicators specified in bit positions 18 through 23 is set, the next instruction in sequence is executed.

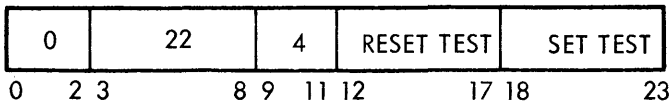
If none of the flag indicators specified in bit positions 12 through 17 is reset and if none of the flag indicators specified in bit positions 18 through 23 is set, the next instruction in sequence is skipped and the following instruction is executed.

If a flag is specified in both the reset and set fields, its status is ignored, and the instruction does not skip.

This instruction does not alter the condition of the flag register.

Affected: (P) Timing: 1 if no skip
2 if skip

SWT SENSE SWITCH TEST



If any of the SENSE switches specified in bit positions 12 through 17 is reset, or any of the SENSE switches specified in bit positions 18 through 23 is set, the next instruction in sequence is executed; otherwise, the next instruction in sequence is skipped and the following instruction is executed.

If a SENSE switch is specified in both the set and reset fields, its status is ignored and the instruction does not skip.

A SENSE switch is said to be set when it is on, or lit.

Affected: (P) Timing: 1 if no skip
2 if skip

3. INTERRUPT SYSTEM

The SDS 9300 Computer contains a Priority Interrupt System that provides added program control of input/output and compute operations, and allows immediate recognition of special external conditions on the basis of predetermined priority. The Priority Interrupt System is essentially a combination of hardware provisions and programming techniques. Various devices (such as the communication channels, interrupt buttons on the console, power fail-safe, real-time clock) can cause interruption of the program being executed by the computer by transmitting interrupt pulses to interrupt levels in the computer.

PRIORITY ASSIGNMENT

All interrupt devices used with a special computer installation are assigned unique, numbered priority levels (see Table 3-1) identified by octal numbers 00 through 01777, with the higher priority levels having a smaller number. Interrupt levels 00-037 are SDS optional hardware interrupt levels, are normally reserved for up to 32 special-purpose interrupt devices, and are normally added in pairs. Interrupt levels 040-01777 are special systems interrupt levels which can be added in any number up to 992 for general-purpose interrupts.

Table 3-1. Interrupt Locations

Location and Level	Name	Description
00	Power On	These two optional interrupts are always armed, always enabled, and are used with the power fail-safe system. When the power drops below safe limits, an interrupt occurs on 01. When power returns to a normal level, an interrupt occurs on 00.
01	Power Off	
02	Central Processor Parity	These two optional interrupts are armed or disarmed at the maintenance panel. When armed and a memory parity occurs, an interrupt is generated to the appropriate level depending on whether the parity was detected in the central processor or within a channel.
03	Input/Output Channel Parity	
04	Memory Protection	This optional interrupt level is activated if an instruction attempts to alter the contents of "protected" memory.
05	Unassigned	
06	Clock Sync	The program uses these optional interrupts for implementing a continuous or elapsed time clock (see Real-Time Clock Interrupts).
07	Clock Pulse	
010	Channel A Zero Word Count (End-of-Word)	These standard interrupts are used by channel A to signal the program when various events have occurred.
011	Channel A End-of-Record (End-of-Transmission)	
012	Channel B Zero Word Count (End-of-Word)	Supplied with the optional channel
013	Channel B End-of-Record (End-of-Transmission)	
014	Channel C Zero Word Count (End-of-Word)	Supplied with the optional channel
015	Channel C End-of-Record (End-of-Transmission)	
016	Channel D Zero Word Count (End-of-Word)	Supplied with the optional channel
017	Channel D End-of-Record (End-of-Transmission)	

Table 3-1. Interrupt Locations (cont.)

Location and Level	Name	Description
020 021	Channel E Zero Word Count Channel E End-of-Record	Supplied with the optional channel
022 023	Channel F Zero Word Count Channel F End-of-Record	Supplied with the optional channel
024 025	Channel G Zero Word Count Channel G End-of-Record	Supplied with the optional channel
026 027	Channel H Zero Word Count Channel H End-of-Record	Supplied with the optional channel
030 031	Floating-Point Instruction Trap Floating-Point Overflow-Underflow Trap	See Trap Locations
032 033	Console Button INTER 32 Console Button INTER 33	The two standard interrupt buttons on the console will cause interrupts to these locations. These are single-instruction interrupts, are always armed, always enabled, and need not be cleared by program.
034 035 036 037	Unassigned	
040 . . . 057	Group 1 Optional Special Systems Interrupts (address code 00)	All special systems interrupts are general-purpose, can be added in any number, and can be single-instruction or subroutine interrupts in any desired combination. If the optional Arm Interrupts Control Unit is not present, these interrupts are always armed. However, if the control unit is present, these interrupts are armed by EOM, POT. These interrupts are enabled if the interrupt system is enabled and are disabled if the interrupt system is disabled.
060 . . . 077	Group 2 Optional Special Systems Interrupts (address code 01)	
. . . 01760 . . . 01777	Group 62 Optional Special Systems Interrupts (address code 076)	

INTERRUPT LEVEL OPERATION

As shown in Figure 3-1, each interrupt level has three distinct operating states. In the inactive state, the level has not received a pulse from its assigned interrupt device. When the pulse is received, the level is unconditionally set to the waiting state. If no higher-priority level is in the waiting or active states, the interrupt level causes the computer to execute the instruction stored in the memory location corresponding to the priority number of the interrupt level as the next instruction. This recognition of an interrupt signal by the computer is accomplished at the end of the execution cycle of the instruction currently being executed, and advances the interrupt level to the active state. The instruction in the interrupt location is executed without incrementing the program counter (P register), and all lower-priority interrupt levels are inhibited until the interrupt level is cleared. The instruction placed in the interrupt location can be either a single instruction or a subroutine entry, depending on the type of interrupt level.

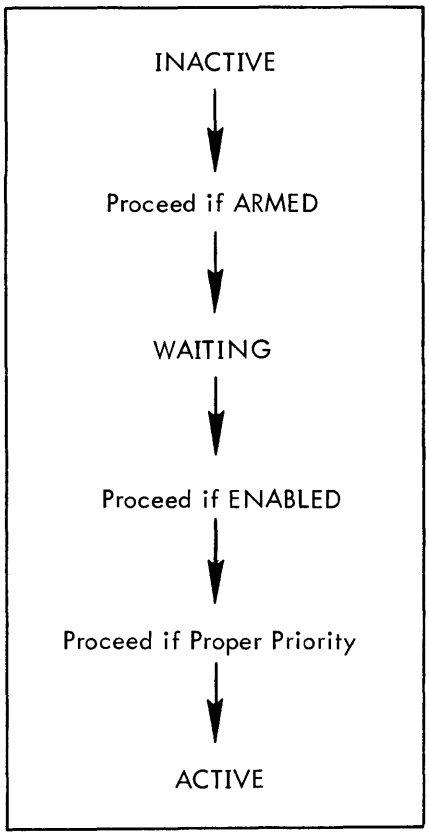


Figure 3-1. Interrupt Arm/Enable Response

SINGLE-INSTRUCTION INTERRUPT

If the interrupt level is a "single-instruction" interrupt level, the computer executes the instruction in the interrupt location, clears the interrupt level back to the inactive state, and executes the next instruction in sequence after the instruction at which the interrupt signal was acknowledged. For example, if a clock is connected to the computer so that it pulses an interrupt line at specified intervals, the program can maintain a real-time clock. If the clock is connected to interrupt level 07 (and location 07 contains the instruction MPO 02050), the computer adds 1 to location 02050 each time the clock pulse causes an interrupt. The main program can examine location 02050, whenever necessary, to determine how many time increments have elapsed since the clock was started.

Some of the optional interrupt levels 00-037 and any of the interrupt levels 040-01777 may be single-instruction interrupts, as required. Single-instruction interrupt levels require that the instruction have a timing of 2 or more cycles; otherwise, the interrupt level is not cleared after the single instruction is executed. Also, if the instruction in the interrupt location is a branch (and the branch should occur), the interrupt is cleared but there is no automatic return to the interrupted program.

SUBROUTINE INTERRUPT

If the interrupt level is a "subroutine" interrupt, the instruction in the interrupt location is normally a MARK PLACE AND BRANCH (BRM) instruction to a servicing subroutine which ends in a BRANCH AND CLEAR INTERRUPT (BRC) instruction, addressed to the first location of the subroutine. The BRM instruction places the current contents of the program counter (address of the next instruction in sequence after the interrupted instruction) in the first location of the servicing subroutine. During execution of the instructions within the servicing subroutine, a higher-priority interrupt can be acknowledged by the computer and the servicing subroutine is interrupted until the higher-priority interrupt has been processed. This allows interrupt levels to be arranged in the order of their importance and/or need for servicing. The BRC instruction at the end of the servicing subroutine clears the interrupt level back to the inactive state and returns program control to the next instruction in sequence in the interrupted program.

NON-INTERRUPTABLE INSTRUCTIONS

If an INCREASE INDEX AND BRANCH (BRX) instruction is being executed and the branch should occur, the computer does not acknowledge an interrupt signal until the instruction to which BRX branches is executed. Also, if an ENERGIZE OUTPUT M (EOM) or ENERGIZE OUTPUT TO DIRECT ACCESS CHANNEL (EOD) instruction is being executed, the computer does not acknowledge the interrupt signal until the instruction following the EOM or EOD is executed.

INTERRUPT ARM/ENABLE RESPONSE

Two control features are available to the programmer concerning the interrupt system. These features are arm/disarm and enable/disable. As shown in Figure 3-1, an interrupt level can proceed from the inactive to the waiting state only if it is "armed." If the level is "disarmed," the pulse is not "remembered" by the level. Once in the waiting state, the interrupt level remains in the waiting state until it has the highest priority of waiting interrupts. However, an interrupt level can proceed from the waiting to the active state only if the interrupt system is "enabled." If the interrupt system is disabled, the interrupt level remains in the waiting state until the interrupt system is enabled. Pressing the RESET button on the control console disarms, disables and clears all interrupt levels.

Some computer applications require that certain conditions always be immediately recognized and acted upon by the computer. For this reason, certain interrupt levels are subject only to priority considerations, and always cause an interrupt if an interrupt device pulses its interrupt line. This type of an interrupt is considered always armed and always enabled, and cannot be disarmed or disabled, except by rewiring the computer. For example, the standard interrupt buttons (INTER 32 and INTER 33 on the control panel) are of this type, as well as some optional interrupts (see Table 3-2). All communication channel interrupt levels are armed, disarmed, enabled, and disabled by means of the program.

ENABLE/DISABLE

The enable/disable feature is standard with the 9300 computer, and operates on the entire interrupt system (with the exception of power fail-safe and the console interrupt buttons). The interrupt system is enabled by execution of ENABLE INTERRUPTS (EIR) and is disabled by execution of DISABLE INTERRUPTS (DIR).

EIR ENABLE INTERRUPTS
EOM 020002 0 02 20002

This instruction is an EOM in the internal control mode (see Section 4, Primary Input/Output Instructions) which turns on the INTERRUPT ENABLE indicator on the computer control panel and unconditionally enables the entire interrupt system. If any interrupt levels are in the waiting state when EIR is executed, the one with the highest priority proceeds to the active state.

Affected: INTERRUPT ENABLE Timing: 1

DIR DISABLE INTERRUPTS
EOM 020004 0 02 20004

This instruction turns off the INTERRUPT ENABLE indicator and unconditionally disables the entire interrupt system (with the exception of power fail-safe and the console interrupt buttons). If any interrupt levels are in the active state when DIR is executed, they are all processed in the order of their priority. If any interrupts are in the waiting state when DIR is executed, they will remain in the waiting state until EIR is executed.

Affected: INTERRUPT ENABLE Timing: 1

IET INTERRUPT ENABLED TEST
(Skip if Interrupt System Enabled)
SKS 020004 0 20 20004

If the interrupt system is enabled when IET is executed, the computer skips the next instruction in sequence and executes the following instruction. If the interrupt system is disabled, the computer executes the next instruction in sequence (does not skip).

Affected: (P) Timing: 1 if no skip
2 if skip

IDT INTERRUPT DISABLED TEST
(Skip if Interrupt System Disabled)
SKS 020002 0 20 20002

If the interrupt system is disabled when IDT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the interrupt system is enabled, the computer executes the next instruction in sequence (does not skip).

Affected: (P) Timing: 1 if no skip
2 if skip

Table 3-2 summarizes how the various interrupt levels are affected by EIR/DIR instructions. Interrupt levels identified as "always enabled" are not affected by EIR/DIR.

Table 3-2. Interrupt Enable/Disable Conditions

Location	Function	Enable/Disable
00, 01	Power Fail-Safe	Always enabled
02, 03	Parity Error	
04	Memory Protection	
06, 07	Real-Time Clock	
010 . . . 027	Communication Channels	Enabled by EIR Disabled by DIR or RESET
032, 033	Console Interrupt Buttons	Always enabled
040 . . . 01777	Special Systems Interrupt levels	Enabled by EIR Disabled by DIR or RESET

ARM/DISARM

The arm/disarm feature is optional with the 9300 computer, although some interrupt levels are always armed and some are individually armed by EOM instructions. If the optional Arm Interrupt Control Unit is present as a part of the computer, all interrupt levels from 040 to 01777 are armed and/or disarmed in groups (i.e., 040-057, 060-077, etc.), and only by a specific combination of the instructions ARM INTERRUPTS (AIR) and PARALLEL OUTPUT (POT) and a control word. If the Arm Interrupt Control Unit is not present, these interrupt levels are considered to be always armed, but are still subject to control with enable/disable. Table 3-3 summarizes the arm/disarm feature for the various interrupts levels:

Table 3-3. Interrupt Arming Criteria

Location	Function	Arming Criteria
00, 01	Power Fail-Safe	Always armed
02, 03	Parity Error	Armed at maintenance panel
04	Memory Protection	Always armed
06	Clock Sync	Always armed
07	Clock Pulse	Arm with EOM 020100
010-017	TMCC	Arm with EIR (compatible mode) or Arm with EOM (extended mode)
020-027	DACC	Arm with EOM (extended mode only)
032, 033	Console Interrupts	Always armed
040-01777	Special System Interrupt levels	Always armed (or arm with AIR)

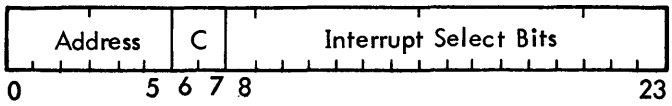
AIR ARM INTERRUPTS 0 02 20020
EOM 020020

AIR is an internal control EOM that prepares the Arm Interrupt Control Unit to receive a control word. The control word is transmitted to the control unit by a POT instruction (see Section 4, Primary Input/Output Instructions). The instruction sequence AIR, POT must be used for each group of 16 interrupt levels; otherwise, an unpredictable operation occurs. These instructions have no effect on the INTERRUPT ENABLE indicator, and the Control Unit is not affected by the indicator.

Affected: Arm Interrupt Control Unit Timing: 1

CONTROL WORD

The control word which the instruction POT addresses has the following format:



The address code (bits 0 through 5) identifies which group of interrupts is being addressed (e.g., an address code of octal 00 identifies interrupt levels 040-057. The C field (bits 6 and 7) specifies whether the interrupt levels selected by bits 8 through 23 of the control word are to be armed and/or disarmed. Bit position 8 of the control word represents the lowest-numbered (highest priority) interrupt within the group identified by the address code (e.g., 040, 060, etc.). Bit position 23 represents the highest-numbered (lowest-priority) level within the group.

The C field control functions are:

Bit Positions

6	7	Function
0	0	Not used
0	1	Arm only those interrupt levels that are selected by a 1 in bit positions 8-23. (Interrupt levels represented by a zero in bit positions 8-23 are not affected.)
1	0	Disarm only those interrupt levels that are selected by a zero in bit positions 8-23. (Interrupt levels represented by a 1 in bit positions 8-23 are not affected.)
1	1	Arm all interrupt levels selected by a 1 and disarm those levels selected by a zero.

Example:

The following partial program arms interrupt levels 050-067 and disarms levels 070-077.

Location	Instruction	Address	Comments
	EIR		Enable entire interrupt system (turns INTERRUPT ENABLE indicator on)
	AIR		Prepare the Arm Interrupt Control Unit to receive a control word
	POT	CW1	Transmit the control word in location CW1 to the Arm Interrupt Control Unit

Location	Instruction	Address	Comments
	AIR		An AIR must precede each POT
	POT	CW2	Transmit the control word in location CW2 to the Arm Interrupt Control Unit
	.		
	.		Other instructions in program
	.		
CW1	00200377		This control word only arms levels 050-057. If any of levels 040-047 are already armed or disarmed, they remain so.
CW2	01777400		This control word arms levels 060-067 and disarms levels 070-077, regardless of their previous state

CHANNEL INTERRUPT DESIGNATIONS

As shown in Table 3-1, each I/O channel has two interrupt levels. These reflect in two distinct uses of interrupts during channel input and output. Also, each A, B, C, and D channel level has two names that reflect their use in the extended or compatible I/O modes (see Section 4, Compatible/Extended Input/Output Modes).

END-OF-WORD/END-OF-TRANSMISSION INTERRUPT OPERATIONS; COMPATIBLE MODE

A program can use channel A as a single-word, direct, program-controlled, input/output buffer. Special I/O instructions applicable to channel A control this type of operation (see Section 4). In this mode, the program can specify that interrupts occur as each word is transferred from the channel buffer to the peripheral device on output, or as soon as the channel buffer is filled from the peripheral device on input. This is the End-of-Word (EOW) interrupt. The program can specify that an End-of-Transmission (EOT) interrupt occurs when the buffer detects a signal such as end-of-record from magnetic tape. During both input and output operations, this interrupt occurs when the peripheral device used in the transmission disconnects and the channel buffer becomes ready for another input/output operation.

These two interrupts also can control input/output termination for any time-multiplexed channel when the program is operating the channel buffers in the block transmission or "interlaced" compatible mode. In this mode, the End-of-Transmission interrupt also occurs when the channel buffer has sent a specified number of words from memory to a peripheral device as well as when the channel detects an end-of-record signal. However, the End-of-Word interrupt can occur on input only after a specified number of words have been read from a peripheral device, and then only if the channel buffer assembles another word. Thus, if the last specified word is read into memory before an end-of-record condition exists, the channel buffer reverts to the single-word mode of operation, in which case the End-of-Transmission interrupt occurs when the channel buffer receives the end-of-record signal. If channel A is being used, the remainder of the record can be stored in memory without the use of the interlace. However, channels B, C, and D operate with interlace only, and a character rate error may occur if the channel is neglected in this mode of transmission. If an end-of-record condition occurs first, only the End-of-Transmission interrupt occurs. No End-of-Word interrupt occurs during output.

The enable/disable instructions "enable and arm" or "disable and disarm" the End-of-Word and End-of-Transmission interrupts when the channel is not operating in the extended interlace mode. When the EIR instruction is executed, the interrupt system is enabled and these interrupts are also armed; when DIR is executed, the system is disabled and these interrupts are also disarmed.

ZERO WORD COUNT/END-OF-RECORD; EXTENDED MODE

When the SDS 9300 Input/Output System uses channels within its full capabilities, SDS 9300 input/output functions control interlaced block transmission operations (see Input/Output Functions, Section 4). The interrupts used with the extended input/output function control are Zero Word Count and End-of-Record. The Zero Word Count interrupt occurs when the last of the number of words specified is placed into or brought from memory. The End-of-Record interrupt occurs when the channel receives an end-of-record signal (gap). Input/output terminal functions can alter this latter occurrence for use with magnetic tapes.

EFFECTS OF THE ENABLE/DISABLE FEATURE ON ARMABLE INTERRUPTS

When operating an input/output channel in the extended mode, the interrupt enable feature controls the armable interrupts (Zero Word Count and End-of-Record). If a channel generates an extended mode I/O interrupt

while the system is disabled, the designated interrupt level goes to the waiting state. When the program again enables the interrupt system, the interrupt goes to the active state when its priority allows.

This feature greatly simplifies the programmers handling of multiple channel operations. The interrupt processing subroutine for one channel can disable the interrupt system while it processes the interrupt. During this time, the system receives all other interrupts in their respective levels and goes to the waiting state until the system is again enabled.

REAL-TIME CLOCK INTERRUPTS

When present in the system, interrupt levels 06 and 07 can function as a real-time clock. Interrupt 07 is a single-instruction interrupt. As a clock, interrupt 07 is designed to contain a skip-class instruction such that if a skip occurs (when 07 is activated), interrupt 06 is initiated. If the 07-level instruction is not a skip or if no skip occurs, interrupt 07 performs as any other single-instruction interrupt.

If SKR C is introduced as the single instruction associated with interrupt 07, the interrupt pair 06 and 07 constitute a special event counter. This is described in the following example:

If 10 is stored in location C and SKR C is in location 07, then interrupt 06 will occur after 11 occurrences of 07. When the inputs to interrupt 07 come from a real-time clock, the event counter becomes an interval timer with the remaining counts in the interval available internally at any time.

Two instructions arm and disarm these interrupts:

EOM 020100	Arms interrupt 07
EOM 020200	Disarms interrupt 07

TRAP LOCATIONS

Memory locations 030 and 031 are reserved as transfer locations in much the same way as the interrupt locations. The locations normally contain MARK PLACE AND BRANCH (BRM) instructions. The instruction in location 030 is executed when any of the four floating-point instructions is executed in a central processor that does not have optional floating-point hardware. This condition is called entering a "trap" location. The trap location is entered without altering the program counter so that the MARK PLACE AND BRANCH (BRM) marks the location in the main program that contains the floating-point instruction. This facility provides an automatic entry to floating-point subroutines via the use of floating-point instruction codes.

The instruction in location 031 is executed as a trap whenever an overflow occurs during the execution of a hardware floating-point instruction. The MARK PLACE AND BRANCH (BRM) in location 031 is normally used as an entry to a corrective subroutine. Such an overflow does not set the OVERFLOW indicator in the flag register.

Traps require no clearing as interrupts do.

When an interrupt and a trap occur simultaneously, the interrupt is delayed until the trap branch instruction is executed.

4. INPUT/OUTPUT SYSTEM

INTRODUCTION

The SDS 9300 has a flexible input/output system to complement its high internal processing speed and versatile instruction repertoire. The system can transmit data in word, character, or single-bit form to and from the computer at the speed of internal computation. The input/output system assumes control of conditions imposed by individual characteristics of a wide variety of devices, yet it leaves a high degree of input/output control to the programmer.

The I/O system provides for the following kinds of input and output:

1. Input/output of data words, each one of which is under direct control of the program
2. Communication channel input/output of characters or words, time-shared with normal accesses to memory and multiplexed with computation
3. Communication channel input/output of characters or words, fully buffered and simultaneous with computation
4. Direct parallel input/output of up to 24 bits of information to and from external equipment, completely controlled and sequenced externally from the central processor
5. Direct parallel input/output of up to 24 bits of information to and from external registers under program control
6. Single-bit input/output, such as equipment on/off status, sense switches, and pulsing and sensing of special devices

TIME-MULTIPLEXED COMMUNICATION CHANNELS

The SDS 9300 includes as standard equipment one time-multiplexed communication channel (TMCC), with provision for addition of three additional channels. These channels are capable of automatically controlling the flow of data to and from memory at rates up to one word every 3.5 microseconds. These channels run independently of the central processor and only interfere with it to transfer data to or from memory.

The time-multiplexed channels use the memory logic of the central processor to facilitate input and output of data words. The transfer of each word between a time-multiplexed channel buffer and memory requires two memory cycles. During this time, computation is delayed in the central processor. Priority for the use of the word input/output logic is in the order: channel D, C, B, A, with channel D having the highest priority. Any time-multiplexed channel operating with automatic interlace has priority over the central processor for memory access.

DIRECT MEMORY ACCESS SYSTEM

In addition to the time-multiplexed channels, a direct memory access system is included in the 9300. This system uses a separate path to memory from those used by the central processor. The separate path to memory allows data transfer through the direct access system without interfering with the central processor if the memory access is to a module which is not being addressed by the central processor. Up to four direct access communication channels (DACC), can be attached to the direct access system. These channels operate like time-multiplexed channels, except that they are faster and provide for a true overlap of input/output with processing.

Each direct access channel has its own independent memory logic. When a memory access is required to obtain or store a data word, computation is delayed one cycle if the access is in the same memory module being addressed by the central processor; if the module is not being addressed by the central processor, no time is lost and computation is unaffected. When two or more direct access channels require memory access simultaneously, priority is determined as described in Channel Memory Access Priority at the end of this section.

When a time-multiplexed channel and a direct access channel are accessing different modules, memory access overlap occurs as it does in central processor accesses.

Transmissions between direct access channels and core memory are under the control of the channel. At the onset of each memory cycle, the control unit interrogates all direct access channels to determine if one of them requires a transfer to or from computer memory. If such is the case, the computer connects the specified memory module to the selected direct access channel. If, simultaneously, the computer requires access to the same memory module, the channel requirement takes precedence and computation is delayed one memory cycle. If the computer and a direct access channel are not accessing the same memory module, the transfer takes place without affecting computation speed. Thus, internal computation and direct access channel transmissions occur simultaneously and independently when the computer and the channel are accessing different memory modules. Channel control logic permits the transfer of only one word per memory cycle to and from the computer memory independent of the number of

operating channels connected to the computer. Thus the maximum transfer rate for direct access is one word every memory cycle (approximately equal to 571,000 words per second, or in excess of two million characters per second).

Communication channels E, F, G, and H, if present in a system, are attached to the direct memory access system, and are therefore designated as direct access communication channels. Operation of these channels is discussed in Communication Channel Input/Output.

A data multiplex system, which uses the direct access memory connection, is also available as an option. This system consists of a data multiplex channel that accepts/transmits data words and memory addresses from many external devices or subchannels, all of which may be in operation at the same time. The system is capable of transmitting up to 571,000 words per second simultaneous with computation (see Appendix B).

In summary, considerable input/output flexibility and convenience is afforded the 9300 user. For example, the external equipment may include an interlace register which allows entire blocks of data to be entered into or read from memory. Telemetry data may be automatically multiplexed or decommutated, obviating sorting and sequencing within the computer.

COMMUNICATION CHANNEL DESCRIPTION

Up to 30 peripheral devices may be attached to one channel. Each of these devices has a unique two-digit, octal address by which it is selected for an input/output operation. To select the peripheral device, the program loads the proper unit address into the 6-bit unit address register (UAR) in the channel buffer. This address selects both the device and, if appropriate, the function to be performed. When any non-zero unit address is placed in the UAR, the peripheral unit addressed is said to be "connected" to the channel and it is said to be in the "active" state. When the UAR is loaded with a zero address, or any time that a terminal or initial condition causes the contents of UAR to be zero, the channel is "inactive". The zero in UAR also means that no peripheral unit is connected.

When the channel and the peripheral unit to be used have been connected, the channel must have information pertaining to the location in memory of the data to be transmitted or received and pertaining to the number or data words in the transfer.

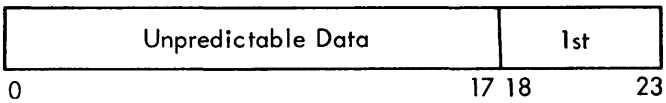
The number of data words to be in a data transfer is loaded into the word count register (WCR). This 15-bit register contains the data word count during the transmission. This count is decremented and replaced in the WCR for each word transmitted. When the word count is equal to zero, the transmission is complete.

The starting destination or source address in memory of the transmitted data is contained in the memory address register (MAR). The memory location to or from which data words are to be transmitted is loaded into the MAR at the same time the word count is entered. During transmission of the data, the contents of the MAR are incremented after each word just as the contents of WCR are decremented.

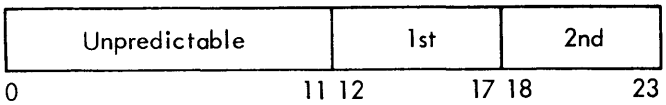
TIME-MULTIPLEXED CHANNEL REGISTERS

In the time-multiplexed channels A through D (see Figure 4-1), there are two other registers besides UAR, WCR, and MAR just discussed that are important to the programmer; these are the word assembly register (WAR) and the single-character register (SCR). The WAR, a 24-bit, word-sized buffer, contains the word of data actively being received or transmitted during an input or output operation. During input, 6-bit characters are received into the SCR and assembled one at a time into the WAR. Then the completed word is placed in memory. Depending on the number of characters per word specified, the word placed in memory during input has the form:

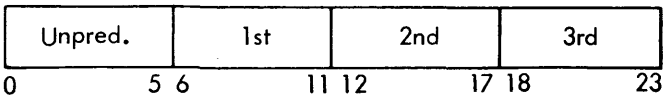
One character per word



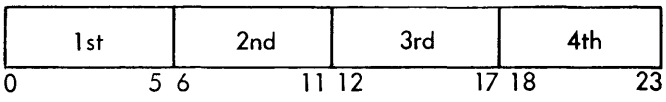
Two characters per word



Three characters per word



Four characters per word



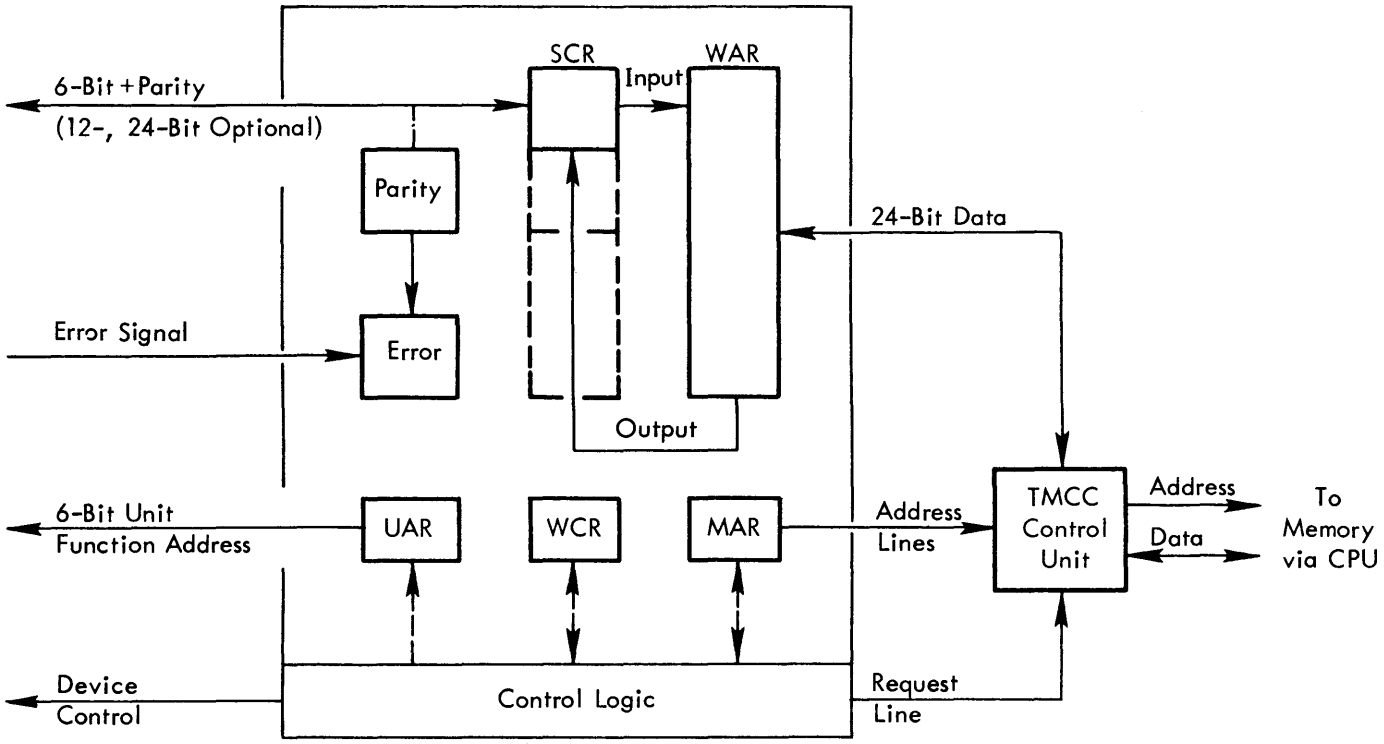


Figure 4-1. Typical SDS 9300 Time-Multiplexed Communication Channel, Block Diagram

When the end of an information record is detected by a channel buffer, the buffer automatically disengages from the device and is then "ready" for another operation. The buffer logic is reset, except that the state of the error indicator is maintained and the last word of the input is still in the word register. If the number of characters in the input record was not a multiple of the number of characters assembled into each computer word, then zeros are automatically forced into the least significant positions of the last word. This last word can then be stored in memory by a CHANNEL A INTO M WHEN READY (AIM) instruction after the buffer has disengaged. If the number of characters in the input record was a multiple of the number of characters assembled into each computer word, then the word remaining in the channel buffer is either the last group of characters from the input device, if they were not previously transferred to memory, or zeros if the last group of characters had been transferred to memory. In either case, it is safe to issue one such AIM instruction after the channel has disengaged without "hanging up" the computer.

During output, words are brought from memory into the WAR and disassembled into the SCR, one 6-bit character at a time. Depending on the characters per word format specified, the 6-bit characters within the word are output as follows:

Function	Format
Output one character from bits 0 through 5	One character per word
Output two characters from bits 0 through 5, 6 through 11	Two characters per word
Output three characters from bits 0-5, 6-11, 12-17	Three characters per word
Output four characters from bits 0-5, 6-11, 12-17, 18-23	Four characters per word

As required, the characters are transferred into the single-character register and output. After each character transfer, the word in the WAR is shifted left 6 bits to be ready for the next transfer. Only those characters needed from each word are used; when required, a new word is brought to the WAR for the next character. For special applications, a time-multiplexed channel may be equipped with a 12- or 24-bit single-character register. The external device which has a character size greater than 6 bits specifies to the channel what its size is, 12 or 24 bits. Standard 6-bit devices are unaffected by the installation of a wider SCR.

DIRECT ACCESS CHANNEL REGISTERS

In the direct access channels, E through H, the two other registers of importance are the word assembly

register (WAR) and input/output register (IOR). The WAR, as above, is a 24-bit word-sized buffer which, during a transmission, contains the information actively being transmitted to, or received from, the external device. Information is assembled into, or disassembled from, the WAR in either character or word format. The format is programmer-selectable. In word format, a data word of up to 24 bits received from a peripheral unit is placed directly into the WAR and then delivered directly to the IOR. When transmitting in the word format mode, words are delivered directly into the WAR from the IOR and from the WAR to the peripheral unit. When transmitting or receiving words, any size from one bit to 24 bits is acceptable. (See Figure 4-2.)

When operating in the character mode, one to four characters are packed into a word. These will normally be the standard 6-bit input/output character size. Characters of less than 6 bits can be handled in character format as defined by a particular installation's need. For character formats that use characters of less than 6 bits, the data transmission is actually in 6-bit character form with zeros filling out the remainder of the 6 bits. When operating in character format mode, the number of characters to be packed into, or unpacked from, each data word can be specified by program control. Under this format, one, two, three, or four characters may be packed into, or unpacked from, all words in a particular data transmission. This is true for all channels.

When receiving character data from a peripheral device, The first character of a word is received into bit positions 18 through 23 of the WAR. When the next character is received, the first 6 bits in bit positions 18 through 23 are shifted into bit positions 12 through 17 and the incoming character is placed into bit positions 18 through 23. The next incoming character causes the two 6-bit characters in bit positions 12 through 23 to be shifted to bit positions 6 through 17 and the incoming character is placed into bit positions 18 through 23. The next character causes another 6-bit left shift and then the character is placed in the vacated bit positions 18 through 23. At this point the WAR is filled completely with 24 bits. This information is now copied into the IOR to be placed into the proper memory location.

The above procedure would be followed when four characters per data word were specified for the data transmission. If three has been specified, the data word containing three 6-bit characters in bit positions 6 through 23 and zeros in bit positions 0 through 5 would be delivered to the IOR. The next incoming character would be accepted as the first of another set of three characters. If two characters had been specified, the data word containing two 6-bit characters in bit positions 12 through 23 and zeros in positions 0 through 11 would be delivered to the IOR. If one had been specified, the data word delivered to IOR would contain zero in bit positions 0 through 17 and one character in positions 18 through 23. When transmitting data using the character format mode, characters are taken from the most significant end of the WAR. If one character per word is specified, the 6-bit character in bit positions 0 through 5 is transmitted to the external device and then another full word of information is received from the IOR. If two characters per word are specified, the 6-bit character in positions 0 through 5

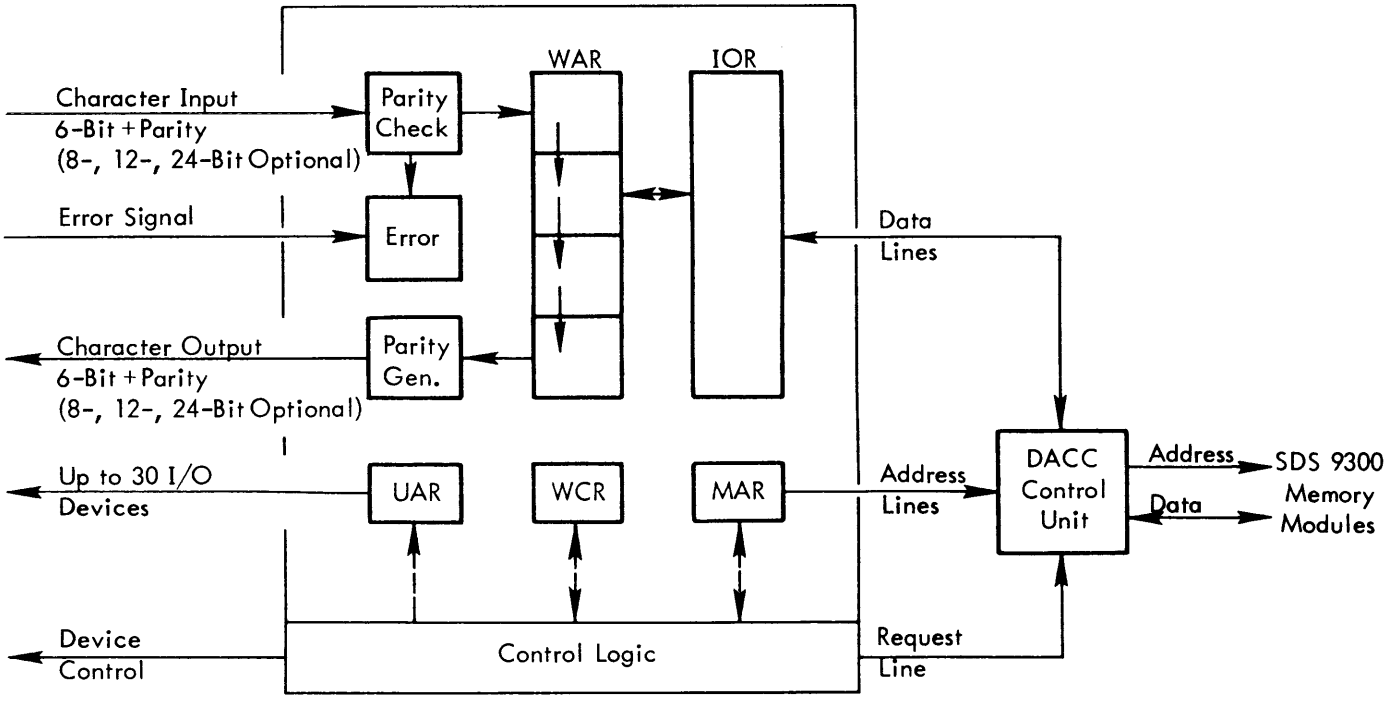


Figure 4-2. Typical SDS 9300 Direct Access Communication Channel, Block Diagram

is transmitted. Then the contents of bit positions 6 through 23 are shifted left into positions 0 through 17, the new 6-bit character in positions 0 through 5 is transmitted and another word is accepted from the IOR to be processed. If three characters are specified, the 6-bit character in bit positions 0 through 5 is transmitted. The contents are left-shifted 6 bits and the new contents of bit positions 0 through 5 are transmitted. The contents are left-shifted 6 bits again and the third character from positions 0 through 5 is transmitted. Then another word is received from the IOR to be processed. If four characters are specified, the above process is extended to one more 6-bit left shift and the final 6 bits of the word are transmitted before the next word is accepted from the IOR.

The IOR is a 24-bit register which is a full word buffer between the WAR and memory. The direct access channel control unit places words into the IOR, awaiting their transfer to WAR to be output. During input, the IOR receives full words from the WAR and places them into memory under control of the word count and memory address being used in the transmission. During multiple data word transfers, the WAR and the IOR simultaneously contain data information.

COMMUNICATION CHANNEL INPUT/OUTPUT

SDS 9300 communication channels provide fully buffered input/output control and transmission, multiplexed or simultaneous with computation. Up to eight data channels can be connected to the SDS 9300, all operating independently of each other.

Each channel can control as many as 30 input/output devices and automatically handles character/word assembly and disassembly, input/output parity detection and generation, data transmission to and from memory, and end-of-transmission detection.

All channels are bidirectional and can communicate with 6-bit character devices or word devices of up to 24 bits. In the case of character-oriented devices, the number of characters to be contained in each word during the transmission is specified by program when the channel operation is initiated.

Each channel consists of a channel buffer and a channel interlace. A channel buffer assembles and disassembles data words as they are transmitted between core memory and the peripheral equipment. Control of operations such as characters per word transmitted and direction of peripheral operation (as in magnetic tape forward/reverse) is maintained within the buffer.

Channel buffers are set up by a channel EOM or EOD. The execution of the channel EOM sets the operation controls as mentioned above, places the unit address in the buffer, and initiates data assembly/disassembly. The presence of the unit address activates the buffer, causing it to look for data coming from the peripheral device or from memory, as determined by the unit address.

A channel interlace controls the transfer of the data words going through the associated channel buffer. It supplies the memory address of data coming from or going to memory and maintains the word count determining the number of words transferred. The terminal interrupts, End-of-Record (EOR) and Zero Word Count (ZWC) come from the interlace and are under its control. Input/output termination functions are controlled by the interlace.

Transmission of data through a channel is initiated by two EOM instructions and a POT instruction. The first EOM alerts the interlace; that is, the interlace is activated and is instructed to expect a word count and starting address to be sent to it by the POT. The second EOM is an input/output mode EOM that specifies the interrupt and the terminal function to be used. It also can specify a 15th address bit and five more high-order word count bits; these expand the starting address to a maximum of 32,767 and the word count from 10 bits to 15 bits. This sequence is written: EOM (buffer control), EOM (input/output control) and POT. When the channel buffer is being set up at the same time, the interlace can be alerted with the buffer set-up EOM. When the buffer is already set up, the channel EOM, ALERT CHANNEL (ALC) may be used to alert the channel interlace.

When using channel A through D where the input/output terminal functions, interrupts, and additional count or address are not desired, only the EOM (Alert) and the POT are needed to set up the channel; the End-of-Record interrupt for the selected channel occurs if the Interrupt System is enabled. When using terminal function control, the eight channels are programmed in the same way. However, there is a distinction between channels A through D and channels E through H. Channels A through D are time-multiplexed channels; channels E through H are direct access channels.

PRIMARY INPUT/OUTPUT INSTRUCTIONS

EOM ENERGIZE OUTPUT M

The major instruction for preparing channel A, B, C, or D and an attached peripheral device to perform a data transmission or other peripheral activity is ENERGIZE OUTPUT M (EOM). This instruction operates in four distinct modes. These are buffer control (mode 0), input/output control (mode 1), internal control (mode 2), and system control (mode 3). In modes 2 and 3, EOM is used in non-communication channel operations such as special systems transmission. The different modes of operation are program selectable by the setting of two bits (10, 11) within the EOM instruction format as follows:

10	11	Mode	Area
0	0	0	Buffer Control
0	1	1	Input/Output Control
1	0	2	Internal Control
1	1	3	System Control

EOM in the buffer control mode (0) operates essentially as a set-up or preparation facility for data transmissions or other peripheral activities using the channel. The channel to be used, the peripheral unit on that channel, the operation to be performed, and the type of character format to be used are all detailed within the EOM in this mode. The use of BCD or binary transmission, the allowance or not of a leader (as in paper tape), and the direction of operation (as in forward direction for magnetic tape) are all detailed to the channel and its "connected" peripheral unit. Execution of such an EOM connects the specified peripheral unit to the channel buffer, starts the device (if it is in a ready condition), and alerts the channel interlace, if desired.

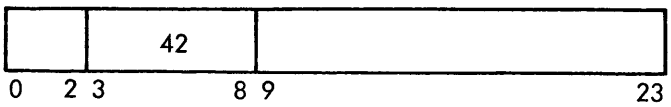
The EOM in the input/output control mode (1) is used to direct peripheral devices to perform non-transmitting operations such as rewind magnetic tape and upspace the printer. Selection of certain channel operations such as interrupt response and input/output terminal function desired is made with this EOM. It is also used to alert peripheral devices that a PARALLEL INPUT (PIN) or PARALLEL OUTPUT (POT) instruction is to follow. An extension of the word count to 15 bits for the number of words to be transmitted and an extension of the address specification to 15 bits can be given in this EOM.

The EOM in the internal control mode (2) is used to enable and disable the interrupt system. EOM in this mode is also used to prepare the system for the selective arming and disarming of the system interrupt levels.

EOM operating in the system control mode (3) is used to transmit information which is specifically coded for a given installation and system. Address capability is expanded for special system designations.

If an interrupt occurs during the execution of an EOM in any mode, it will not be acknowledged until the execution of the instruction following the EOM is completed.

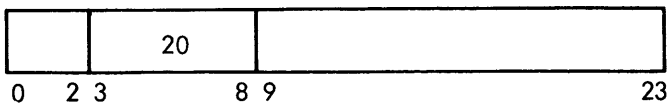
EOD ENERGIZE OUTPUT TO DIRECT ACCESS CHANNEL



The EOD instruction operates in buffer control (0) and input/output control (1) modes. It refers to channels E, F, G, and H and performs the same functions and operations as an EOM on channels A through D. The internal control (2) and system control (3) modes are available, as special systems require expanded capabilities.

Affected: DACC Timing: 1

SKS SKIP IF SIGNAL NOT SET



The principal instruction for testing the states and responses of a channel and its attached peripheral devices is the SKIP IF SIGNAL NOT SET (SKS) instruction. SKS is a skip class instruction yielding a decision and transfer capability to the input/output system. This instruction also operates in four distinct modes. These are special internal test (mode 0), channel and device test (mode 1), internal test (mode 2), and special system test (mode 3). In modes 2 and 3, the SKS is used to test non-channel-oriented functions. These different modes of operation are program selectable by the setting of two bits (10, 11) within the SKS instruction format:

10	11	Mode	Area
0	0	0	Special Internal Test
0	1	1	Channel and Device Test
1	0	2	Internal Test
1	1	3	Special System Test

In the channel and device test mode (1), SKS can be programmed to test a channel for channel active, word count equals zero, channel inter-record condition, and channel-error-indicator-shows-no-error. This mode also tests peripheral devices directly. These include testing indicators in a magnetic tape unit such as beginning of tape, end of tape, file protect ring present, and end-of-file. For example, an indicator within the printer might be addressed by an SKS to determine whether the paper is at the end-of-form.

In the internal test mode (2) SKS tests whether or not the interrupt system is enabled.

In the special internal test (0) and special system test (3) modes, SKS tests signals of special configurations as the specific systems require.

		Modes 0 & 2	Modes 1 & 3
Affected: (P)	Timing:	1 if no skip 2 if skip	2 if no skip 3 if skip

DIRECT PARALLEL INSTRUCTIONS

Two instructions, PARALLEL OUTPUT (POT) and PARALLEL INPUT (PIN), permit any word in core memory to be presented in parallel at a peripheral connector; or, inversely, permit signals sent to a connector to be stored in any core memory location. The execution of a POT or PIN instruction causes a signal to be sent to the external device involved in the input/output operation. This signal alerts the device to send its data word as soon as it is operational. When it becomes operational

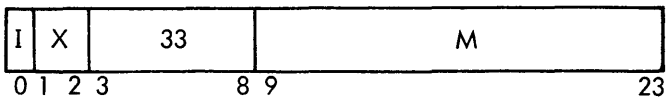
during a read or PIN operation, it transmits a ready signal to the central processor while at the same time presenting its data word. The computer places the received data word into a specified memory location. The computer "hangs up" during the execution of PIN until it receives the ready signal from the external device.

During the execution of a POT instruction, the central processor transmits a signal to the external device alerting it to receive a data word. When the device becomes operational, it transmits a ready signal to the central processor which releases the data word to the external device. The computer "hangs up" during the execution of POT until it receives the ready signal from the external device.

POT and PIN can be used effectively with other 9300 instructions to produce high-speed, synchronized, data transfers without the use of a communication channel. Selective input or output from and to a number of external holding devices is accomplished by preceding POT and PIN with an EOM to select the desired device. By preceding the POT or PIN with an SKS, the ready signal can be tested prior to execution of the parallel transfer instruction and the "hang-up" of the computer can be avoided. If the ready signal from the external device is used to initiate one of the priority interrupts, parallel input/output operation can occur as soon as the external device is able to transmit or receive. Since the ready signal initiating the interrupt is still present when the POT or PIN is executed, no "hang-up" will occur. These features allow the computer to execute parallel input/output operations at rates up to 70,000 words per second.

If POT and PIN are used in conjunction with the REPEAT INSTRUCTION IN M (REP), higher transmission rates of up to 190,000 words per second are possible. During transmissions of this type, no other computation can be performed. Direct access channel transfers can occur simultaneously and at their usual rates, provided that memory modules associated with the channels are different from memory modules associated with the parallel input/output transmissions.

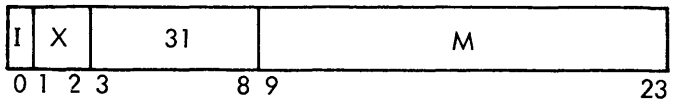
PIN PARALLEL INPUT



PIN stores the contents of 24 input lines in parallel in the effective memory location.

Affected: (EM) Timing: 4 + wait

POT PARALLEL OUTPUT

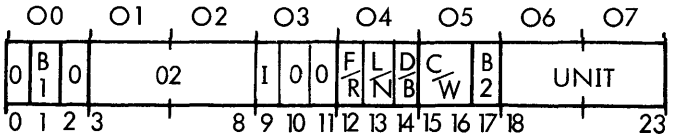


POT transmits the contents of the effective memory location in parallel to 24 output lines of an external device.

Affected: Output Device Timing: 3 + wait

COMMUNICATION CHANNEL EOM/EOD (BUFFER CONTROL MODE)

The ENERGIZE OUTPUT M (EOM) used in the buffer control mode is described in detail below. EOM in this mode alerts and connects the specified channel A, B, C, or D and selects the desired unit address. The detailed instruction format is:



Bit Position	Designation	Function
0, 2	0	Bit positions 0 and 2 are not used with EOM.
1, 17	B1 B2	Bit positions 1 and 17 specify the channel to be activated. Channel A is numbered 00, Channel B is 01, Channel C is 10, and Channel D is 11.
3-8	02	Bit positions 3 through 8 contain 02, the instruction code for EOM.
9	I	Bit position 9 alerts the buffer interlace. A 0-bit specifies non-interlace operation. A 1-bit specifies interlace operation. If a previous EOM has alerted the interlace, this bit is ignored.
10, 11	00	Bit positions 10 and 11 contain the EOM mode indicator for the buffer control mode (mode 0).
12	F/R	Bit position 12 specifies the direction in which the peripheral device will operate. A 0 specifies the forward direction. A 1 specifies the reverse direction.

Bit Position	Designation	Function
13	L/N	Bit position 13 specifies whether the device should be started with a leader as in paper tape. A 0 specifies a start with leader. A 1 specifies a start without leader.
14	D/B	Bit position 14 specifies the mode of character format. A 0 specifies BCD format. A 1 specifies Binary format.
15, 16	C/W	Bit positions 15 and 16 specify the number of characters to be assembled into, or disassembled from, each transmitted word. One character per word is specified by 00, two by 01, three by 10, and four by 11.
18-23	UNIT	Bit positions 18 through 23 specify the unit and the function to be performed with that unit. (See Table 4-1.)

EOD ENERGIZE OUTPUT ON DIRECT ACCESS CHANNEL

This instruction is used in the buffer control mode as described below. EOD in this mode alerts and connects the specified direct access channel (E, F, G, H) and the desired unit address.

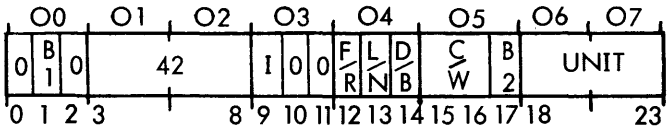


Table 4-1. Unit Address Codes

00	Disconnect	40	Not Used
01	Type Input No. 1	41	Type Output No. 1
02	Type Input No. 2	42	Type Output No. 2
03	Type Input No. 3	43	Type Output No. 3
04	Paper Tape Input No. 1	44	Paper Tape Punch Output No. 1
05	Paper Tape Input No. 2	45	Paper Tape Punch Output No. 2
06	Card Reader Input No. 1	46	Card Punch Output No. 1
07	Card Reader Input No. 2	47	Card Punch Output No. 2
10	Magnetic Tape Input No. 0	50	Magnetic Tape Output No. 0
11	Magnetic Tape Input No. 1	51	Magnetic Tape Output No. 1
12	Magnetic Tape Input No. 2	52	Magnetic Tape Output No. 2
13	Magnetic Tape Input No. 3	53	Magnetic Tape Output No. 3
14	Magnetic Tape Input No. 4	54	Magnetic Tape Output No. 4
15	Magnetic Tape Input No. 5	55	Magnetic Tape Output No. 5
16	Magnetic Tape Input No. 6	56	Magnetic Tape Output No. 6
17	Magnetic Tape Input No. 7	57	Magnetic Tape Output No. 7
20	-	60	High-Speed Printer Output No. 1
21	-	61	High-Speed Printer Output No. 2
22	-	62	-
23	-	63	-
24	-	64	Incremental Plotter Output No. 1
25	-	65	Incremental Plotter Output No. 2
26	Disc File or Auxiliary Drum Input No. 1	66	Disc File or Auxiliary Drum Output No. 1
27	Disc File or Auxiliary Drum Input No. 2	67	Disc File or Auxiliary Drum Output No. 2
30	Scan Magnetic Tape No. 0	70	Magnetic Tape Erase No. 0
31	Scan Magnetic Tape No. 1	71	Magnetic Tape Erase No. 1
32	Scan Magnetic Tape No. 2	72	Magnetic Tape Erase No. 2
33	Scan Magnetic Tape No. 3	73	Magnetic Tape Erase No. 3
34	Scan Magnetic Tape No. 4	74	Magnetic Tape Erase No. 4
35	Scan Magnetic Tape No. 5	75	Magnetic Tape Erase No. 5
36	Scan Magnetic Tape No. 6	76	Magnetic Tape Erase No. 6
37	Scan Magnetic Tape No. 7	77	Magnetic Tape Erase No. 7

Bit Position	Designation	Function
1, 17	B1 B2	Bit positions 1 and 17 specify the channel to be activated. Channel E is numbered 00, channel F is 01, channel G is 10, and channel H is 11.

All other indicators in the EOD are identical to EOM and function in the same way.

STANDARD EOM AND EOD CHANNEL INSTRUCTIONS

Several EOM and EOD function configurations have standard uses. These are given standard META-SYMBOL assembler-type mnemonics and are set aside as separate instructions.

ALC ALERT CHANNEL

The channel is alerted; however, the channel buffer is not affected in any way.

Channel	Mnemonic	EOM/EOD	Octal Configuration
A	ALC 0	EOM 050000	0 02 50000
B	ALC 1	EOM 050100	0 02 50100
C	ALC 2	EOM 050000,2	2 02 50000
D	ALC 3	EOM 050100,2	2 02 50100
E	ALC 4	EOD 050000	0 42 50000
F	ALC 5	EOD 050100	0 42 50100
G	ALC 6	EOD 050000,2	2 42 50000
H	ALC 7	EOD 050100,2	2 42 50100

DSC DISCONNECT CHANNEL

The channel is disconnected. Its unit address register is unconditionally set to 00 regardless of whether a device is currently being addressed by the channel. Any device which is connected to the channel is disconnected from the channel.

Channel	Mnemonic	EOM/EOD	Octal Configuration
A	DSC 0	EOM 0	0 02 00000
B	DSC 1	EOM 0100	0 02 00100
C	DSC 2	EOM 0,2	2 02 00000
D	DSC 3	EOM 0100,2	2 02 00100
E	DSC 4	EOD 0	0 42 00000
F	DSC 5	EOD 0100	0 42 00100
G	DSC 6	EOD 0,2	2 42 00000
H	DSC 7	EOD 0100,2	2 42 00100

ASC ALERT TO STORE ADDRESS FROM CHANNEL

The channel is alerted for a PIN instruction to follow. This instruction does not affect the operation of the channel. See Direct Parallel Instruction for a detailed discussion of PIN.

ASC is always used in conjunction with PIN to determine the current completion status of an I/O operation being performed by the selected channel. The two instructions should be written in the order ASC (EOM/EOD), PIN. When these two instructions have been executed, the contents of the effective memory location from the PIN instruction are as follows:

Bit positions 0 through 8 contain zero.

Bit positions 9 through 23 contain the contents of the memory address register of the channel.

Channel	Mnemonic	EOM/EOD	Octal Configuration
A	ASC 0	EOM 012000	0 02 12000
B	ASC 1	EOM 012100	0 02 12100
C	ASC 2	EOM 012000,2	2 02 12000
D	ASC 3	EOM 012100,2	2 02 12100
E	ASC 4	EOD 012000	0 42 12000
F	ASC 5	EOD 012100	0 42 12100
G	ASC 6	EOD 012000,2	2 42 12000
H	ASC 7	EOD 012100,2	2 42 12100

TOP TERMINATE OUTPUT ON CHANNEL

When the last word of a block has been delivered to the channel, this instruction is used to terminate channel output. After the execution of this instruction, the following occurs. After the last character in the channel buffer is delivered to the peripheral device, the channel is disconnected.

This instruction is always used to terminate a channel A, non-interlaced, output operation. It may be used with all communication channels if the particular function selected is terminal function 11 (IOSP) but no further data output is required (see Terminal Functions).

Channel	Mnemonic	EOM/EOD	Octal Configuration
A	TOP 0	EOM 014000	0 02 14000
B	TOP 1	EOM 014100	0 02 14100
C	TOP 2	EOM 014000,2	2 02 14000
D	TOP 3	EOM 014100,2	2 02 14100
E	TOP 4	EOD 014000	0 42 14000
F	TOP 5	EOD 014100	0 42 14100
G	TOP 6	EOD 014000,2	2 42 14000
H	TOP 7	EOD 014100,2	2 42 14100

COMPATIBLE/EXTENDED INPUT/OUTPUT MODES

The termination of an I/O operation and the interrupts that may be associated with that termination fall into two classes: compatible and extended. The choice of one of these two modes of input/output operation determines how the system behaves when the termination of an I/O operation occurs.

As mentioned in Section 3, Interrupt System, interrupts occurring at a single level (e.g., location 010, 012, etc.) can have different names (e.g., Zero Word Count and End-of-Word). These names reflect the different I/O mode in operation when the interrupt occurs. The differences include the timing of interrupt occurrence relative to the I/O operation and type of interrupt requested.

The compatible mode of operation for channels A, B, C, and D is directly compatible with the SDS 900 Series mode of I/O (interlaced) operation. The types of interrupts that can be requested are the End-of-Word and End-of-Transmission interrupts.

The extended mode for all channels expands the I/O capabilities to include the terminal and arming functions discussed below. The types of interrupts that can be requested are the Zero Word Count and End-of-Record interrupts.

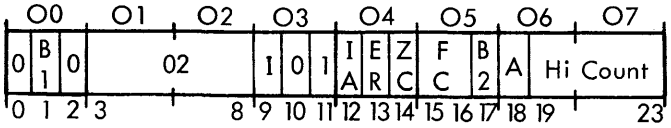
The I/O mode is selected in the input/output EOM (EOD) via bit 12, the interrupt arm bit. A 0-bit makes the system operate in the compatible mode; a 1-bit sets the system in the extended mode.

In particular, the interrupt arm (IA) bit determines whether any of the extended functions operate; that is, a 0 in IA means that the other extended mode controls, bits 13, 14, 15, and 16,have no effect.

INPUT/OUTPUT CONTROL MODE EOM/EOD

The input/output EOM(EOD) selects the I/O operation mode. When the extended mode is selected, this EOM also selects (arms) interrupts that are to be made operational and selects the desired terminal function. EOM applies to channels A, B, C, and D; EOD applies to channels E, F, G, and H.

EOM ENERGIZE OUTPUT M

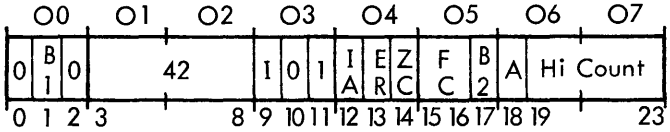


Bit Position	Designation	Function
0, 2	0	Bit positions 0 and 2 are ignored.
1, 17	B1, B2	Bit positions 1 and 17 specify the channel.
3-8	02	Bit positions 3 through 8 contain 02, the instruction code for EOM
9	I	Bit position 9 alerts the interlace. If the interlace has been alerted by a previous EOM, this bit is ignored.
10, 11	01	Bit positions 10 and 11 contain the EOM indicator for the input/output control mode (mode 1).
12	IA	Bit position 12 selects the mode of I/O operation. A 0 specifies the compatible mode. The operation of bits 13, 14, 15, and 16 are disallowed. Channels A, B, C, and D only can operate in this mode, which is completely 900 Series-compatible; if interrupts are required, the user enables the interrupt system (EIR), thus enabling and arming the End-of-Word and End-of-Transmission interrupts. A 1 specifies the extended mode. All channels can operate in this mode. This allows the use of bits 13, 14, 15, and 16. If interrupts are required, the user arms the associated ones by placing 1-bits in bit 13 and/or 14. The terminal function to be used is selected via bits 15 and 16. Note that a 1-bit in 13 and/or 14 does the following: 1. Arms that interrupt during this complete I/O operation; disconnecting this channel disarms the interrupt. 2. Once armed by bits 13 and/or 14, the interrupt can be enabled by the enable/disable feature of the interrupt system. If a channel generates an extended mode I/O interrupt while the system is disabled, the designated interrupt level goes to the waiting state. When the program again enables the interrupt system, the interrupt goes to the active state when its priority allows.
13	ER	Bit position 13 controls the arming of the End-of-Record (EOR) interrupt. A 1-bit arms the interrupt. A 0-bit disarms the interrupt.

Bit Position	Designation	Function
14	ZC	Bit position 14 controls the arming of the Zero Word Count (ZWC) interrupt. A 1-bit arms the interrupt. A 0-bit disarms the interrupt.
15, 16	FC	Bit positions 15 and 16 specify the terminal condition function to be performed with the transmission. These are defined in Extended Mode Terminal Functions.
18	A	Bit position 18 is the high-order address bit.
19-23	Hi Count	Bit positions 19 through 23 contain the most significant five bits of the 15-bit word count. These positions specify a word count greater than 1023.

EOD ENERGIZE OUTPUT ON DIRECT ACCESS CHANNEL

This instruction, used in the input/output control mode (mode 1), is described below.



The functions of EOD are identical to EOM in this mode and operate on channels E, F, G, and H. Direct access communication channels operate only in the extended interlace mode; therefore, a DACC does not examine bit 12, but assumes it to be a 1. Note that with the complete omission of this second EOD, a DACC operates in the terminal function 00 (IORD) mode.

COMPATIBLE MODE TERMINAL FUNCTIONS

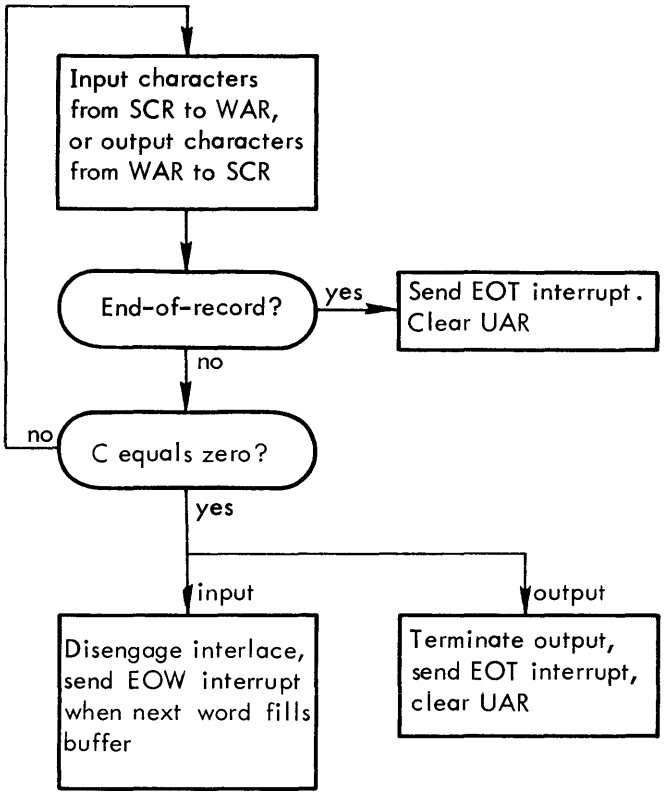
The following description and diagram illustrates the automatic terminal functions of a time-multiplexed communication channel when operating in the compatible interlace mode of data transmission.

INPUT

Read C words. If C equals zero before the end-of-record is detected, the interlace is disengaged and the End-of-Word (EOW) interrupt signal is generated when the next word fills the word assembly register. Since the channel buffer continues to input characters, a character rate error occurs if the record is longer than C+1 words, unless the peripheral device is disconnected from the buffer or the remainder of the record is otherwise disposed of. At the end-of-record, the peripheral device is disconnected, the channel becomes inactive, and the channel generates an EOT interrupt.

OUTPUT

Write C words. When C equals zero, output is terminated, the channel is disconnected and the EOT interrupt is generated.



EXTENDED MODE TERMINAL FUNCTIONS

A 2-bit function code in bit positions 15 and 16 of the input/output EOM/EOD controls the termination of input/output operations in the extended mode as follows:

Bit Position 15	16	Terminal Function
0	0	Input/Output of a Record and Disconnect (IORD)
0	1	Input/Output until Signal then Disconnect (IOSD)
1	0	Input/Output of a Record and Proceed (IORP)
1	1	Input/Output until Signal then Proceed (IOSP)

These functions are described below with the letter C representing the specified word count of the transmission. Following each of the discussions is a diagram representing the automatic terminal actions of the channel while under control of the specified terminal function in the extended interlace mode of data transmission.

The following table summarizes the terminal functions that should be used with various devices. The IOSP can always be used with any device; however, it causes a disconnect only when an end-of-record signal is received by the channel from a peripheral device on input.

Device	Input	Output
Typewriter	IOSD	IOSD
Paper Tape	IORD, IOSD (IORP can be used but there is no advantage to doing so)	IOSD
Cards	IORD, IOSD (this should not normally be used to disconnect in the middle of a half-read card)	IOSD, IORD
Printer		IOSD, IORD
Magnetic Tape	IORD, IORP	IORD, IORP

INPUT/OUTPUT OF A RECORD AND DISCONNECT (IORD)

A program should not use IORD with devices that do not have end-of-record conditions on output (e.g., devices such as the paper tape punch and typewriter). These devices do terminate output but give the program no indication when they receive the last characters.

INPUT

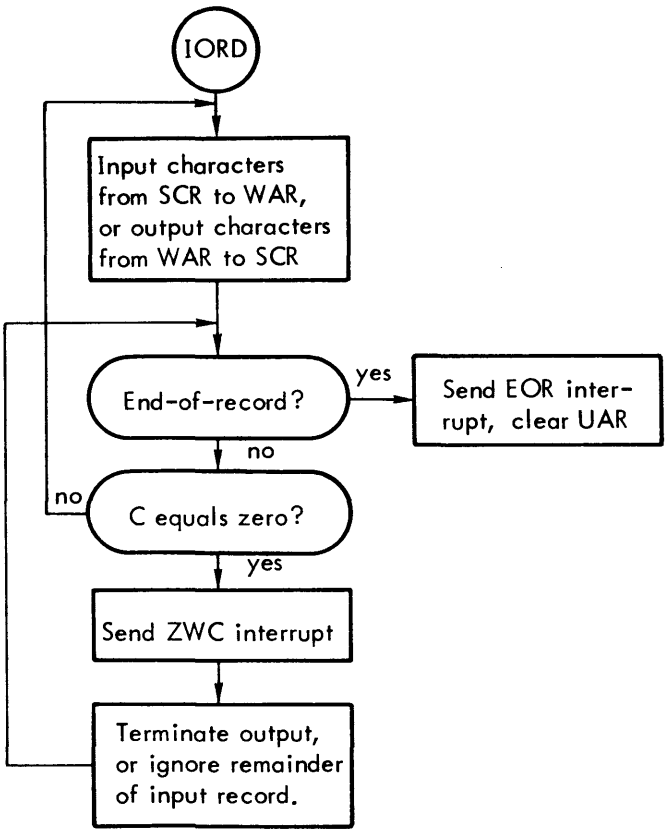
Read C words. If C equals zero before the end-of-record is detected, the rest of the record is ignored. At the end-of-record, the peripheral device is disconnected and the channel becomes inactive.

OUTPUT

Write C words. When C equals zero, output is terminated (i.e., the device is signaled that the last characters have been transmitted). When the peripheral device has generated the end-of-record and, if necessary, checked the validity of the record, it sends an end-of-record response to the channel buffer. When received by the buffer, the end-of-record signal generates an End-of-Record (EOR) interrupt (if armed) and disconnects the channel.

The line printer generates the end-of-record response when it completes the printing of a line. If the printer encounters any print error or faults, it sends a signal to the channel that sets the channel error indicator; this can occur since the printer has not disconnected from the channel. The IORD is useful when the program is to print several lines and the program is not otherwise to use the channel between lines. When the printer completes each line, it causes an EOR interrupt (assumed to be armed), notifying the program that it can immediately transmit the next paper control instruction and the next line image.

The unbuffered card punch operates similarly. It generates the end-of-record response after punching each row If any faults occur during the punching of the entire card, the card punch sends a signal to the channel that sets the channel error indicator; this occurs after punching the last row (row 9).



INPUT/OUTPUT UNTIL SIGNAL THEN DISCONNECT (IOSD)

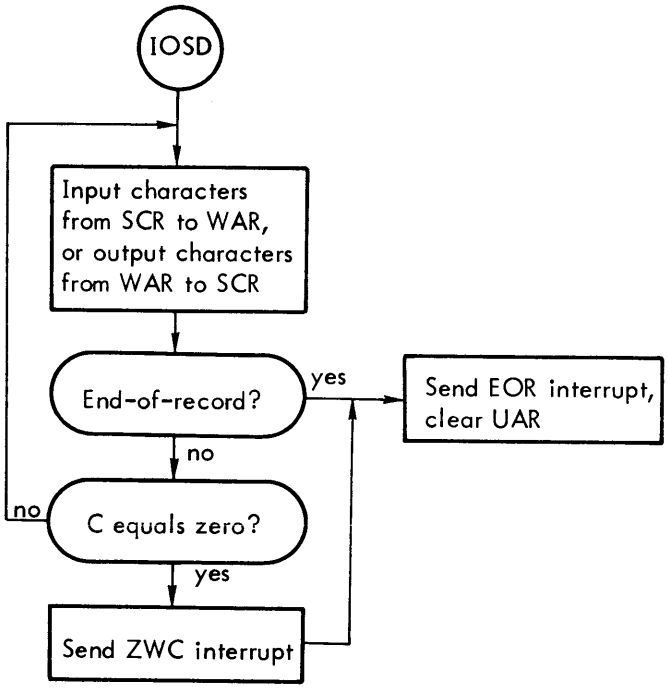
The IOSD is designed for use on devices which are normally operated on the basis of the word count only. Typewriters and paper-tape devices are of this type, as are the printer and card punch when the user does not wish to stay connected until the operation is complete.

INPUT

Read C words. When C equals zero or when the end-of-record is encountered, the device is disconnected and the channel become inactive. If the channel disconnects because of zero count, an EOR interrupt (if armed) will be generated in addition to the Zero Word Count (ZWC) interrupt; if both are armed, C = 0 will occur first.

OUTPUT

Write C words. When C equals zero and when the last character has been transmitted, the channel disconnects the device and becomes inactive. If an end-of-record signal is received before the count reaches zero, the channel will disconnect immediately.



INPUT/OUTPUT OF A RECORD AND PROCEED (IORP)

A program should not use IORP with devices that do not generate end-of-record responses upon output termination; such devices are paper tape and typewriter. These devices do terminate output but give the program no indication when they receive the last character.

The IORP should also not be used with the printer and card punch since these devices expect the channel to disconnect after they send EOR.

INPUT

Read C words. If the channel counts C down to zero before the peripheral device encounters the end-of-record, the channel ignores the rest of the record (to the end-of-record). When the peripheral device sends the end-of-record signal to the channel, the channel sets its end-of-record indicator; this signal sets the End-of-Record (EOR) interrupt (if armed). The channel does not disconnect. The channel is now in an "inter-record" condition.

When the peripheral is magnetic tape, the tape continues to move when the tape handler encounters the end-of-record. The end-of-record occurs when the tape read-heads encounter tape gap; this also causes a tape signal to "come high". If the program executes a new read-tape or scan-tape EOM during the intra-gap time (approximately one millisecond while the tape gap signal is high), the tape remains in motion and proceeds to read or scan the next record. If the program executes no such EOM before the tape gap signal drops, the channel disconnects and the tape comes to a stop. No additional interrupt occurs. This is the only condition that causes a channel to disconnect automatically.

All other input devices remain connected until the program takes further action. The paper tape reader remains in motion; the program should issue a DISCONNECT CHANNEL instruction if the program is not reading any more tape. To proceed after the end-of-record occurs, the program first executes a buffer control mode EOM to re-initialize the channel and then reloads the interlace portion of the channel (the program can alert the Interlace via the buffer control EOM). Otherwise, the channel immediately terminates any attempt to use its interlace portion since the channel is aware that it is still active and in the end-of-record condition. When the program continues from an inter-record condition, the program should use an extended-mode terminal function. An IORP should not be used to read devices which do not have EOR signals (e.g., the typewriter and paper tape punch).

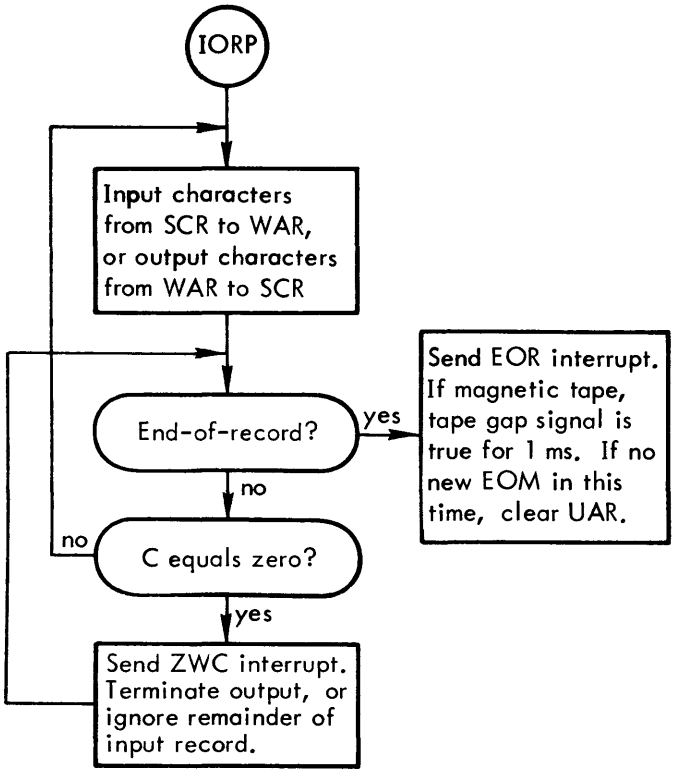
OUTPUT

Write C words. When the channel interlace counts C down to zero, the interlace notifies the channel buffer that it has received the last word that is to be output; when the buffer outputs this last word, it sends a signal to the connected peripheral device indicating that the device has the last word now. When the peripheral device receives, outputs, and checks the validity of this last word, it sends an end-of-record response to the channel buffer. When received by the buffer, the end-of-record signal generates an EOR interrupt (if armed) and sets the inter-record indicator; the channel does not disconnect.

When the peripheral device is magnetic tape, the tape continues to move after it signals end-of-record. As in

reading tape, the signal causes the tape gap signal to come high. If the program executes a new write-tape or erase-tape EOM during the intra-gap time (approximately one millisecond), the tape remains in motion and proceeds to write or erase a new record. If the program executes no such EOM before the tape gap signal drops, the channel disconnects and the tape comes to a stop. No interrupt occurs at this time. This is the only condition which causes a channel to disconnect automatically.

To proceed after the end-of-record occurs, the program first executes a buffer control mode EOM to re-initialize the channel UAR and then reloads the interlace portion of the channel (the program can alert the interlace via the buffer control EOM). Otherwise, the channel immediately terminates any attempt to use its interlace portion, since the channel is aware that it is still active and in the end-of-record condition. When the program continues from an inter-record condition, the program should use an extended-mode terminal function.



INPUT/OUTPUT UNTIL SIGNAL THEN PROCEED (IOSP)

INPUT

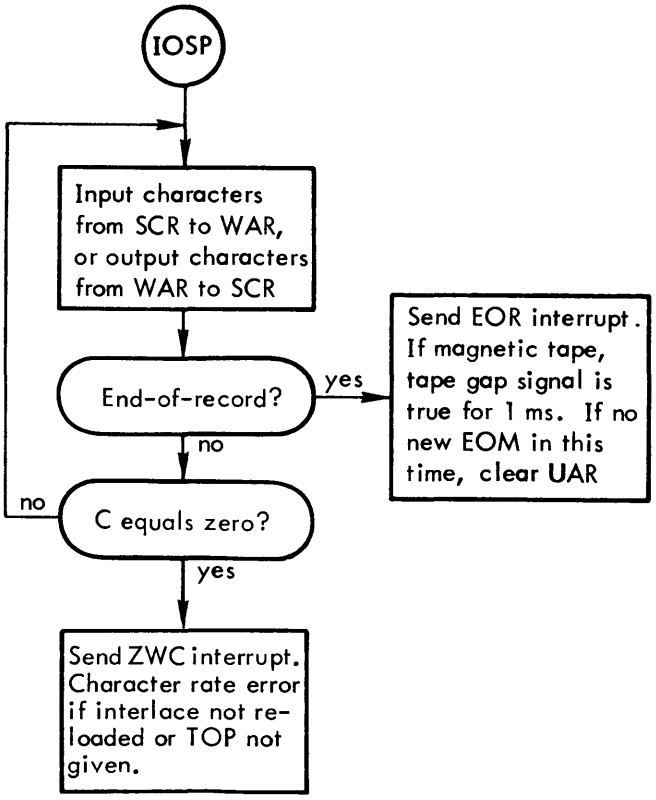
Read C words. If the channel counts C down to zero before the peripheral device encounters the end-of-record the channel generates a Zero Word Count (ZWC) interrupt (if armed). The program should reload the interlace portion of the channel to continue reading the record. As far as the peripheral device knows, nothing happens at this time. Failure to reload the interlace before the peripheral device sends enough characters to overfill the channel buffer causes a rate error; this sets the channel error indicator.

When the peripheral device encounters the end-of-record, IOSP operates identically like the IORP function.

OUTPUT

Write C words. When the channel counts C down to zero, the channel generates a ZWC interrupt (if armed); the channel does not terminate output. The program should reload the interlace portion of the channel to continue writing in the same record. Failure to reload the interlace before the buffer transmits all of the characters in its registers and before the peripheral device requests the next character from the buffer is a rate error; this sets the channel error indicator.

If the program executes a TERMINATE OUTPUT (TOP) instruction after the channel has counted C down to zero, the channel terminates the output and operates as an IORP from this point on.

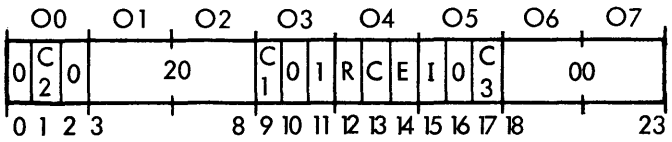


CHANNEL AND DEVICE SKS

The SKIP IF SIGNAL NOT SET (SKS) is used in the channel and device test mode (mode 1) as described

below. The SKS tests the indicators in a channel as well as devices attached to it. To test the channel, unit address 00 is used. Instruction format is:

CHANNEL TESTS



Bit Position	Designation	Function
3-8	20	Bit positions 3 through 8 contain 20, the instruction code for SKS.
10, 11	01	Bit positions 10 and 11 contain the mode selection for mode 1.
9, 1, 17	C1 C2 C3	Bit positions 9, 1, and 17 are used as an octal digit to specify the channel to be tested. Channel A is 0, channel B is 1, and so on, channel H being 7.
12	R	Test for ready. A 1-bit selects the test.
13	C	Test if indicator for word count equal to zero is set. A 1-bit selects the test.
14	E	Test for error indicator reset. A 1-bit selects the test.
15	I	Test for inter-record condition. A 1-bit selects the test.
18-23	00	Bit positions 18 through 23 are zero to specify a data channel test.

Affected: (P) Timing: 2 if no skip
3 if skip

STANDARD SKS INSTRUCTIONS

Several SKS function configurations have standard uses. These are given standard assembler-type mnemonics and are always used as shown.

CAT CHANNEL ACTIVE TEST
(Skip if Channel Not Active)

If the channel is ready to accept a new input/output instruction, the next instruction in sequence is skipped and the following instruction is executed. If the channel is active, busy servicing a command, or in the process of disconnecting a peripheral unit, the next instruction in sequence is executed.

Channel	Mnemonic	SKS	Octal Configuration
A	CAT 0	SKS 014000	0 20 14000
B	CAT 1	SKS 014100	0 20 14100
C	CAT 2	SKS 014000,2	2 20 14000
D	CAT 3	SKS 014100,2	2 20 14100
E	CAT 4	SKS 054000	0 20 54000
F	CAT 5	SKS 054100	0 20 54100
G	CAT 6	SKS 054000,2	2 20 54000
H	CAT 7	SKS 054100,2	2 20 54100

CET CHANNEL ERROR TEST
(Skip if no Error on Channel)

The error indicator in the channel is tested for being in the set condition. If the error indicator has not been set, the next instruction in sequence is skipped and the following instruction is executed. If the error indicator has been set, the next instruction in sequence is executed.

Channel	Mnemonic	SKS	Octal Configuration
A	CET 0	SKS 011000	0 20 11000
B	CET 1	SKS 011100	0 20 11100
C	CET 2	SKS 011000,2	2 20 11000
D	CET 3	SKS 011100,2	2 20 11100
E	CET 4	SKS 051000	0 20 51000
F	CET 5	SKS 051100	0 20 51100
G	CET 6	SKS 051000,2	2 20 51000
H	CET 7	SKS 051100,2	2 20 51100

CZT CHANNEL ZERO COUNT TEST
(Skip if Channel Word Count is Zero)

A test is made on whether the contents of the word count register in the channel have been reduced to zero. If the contents of WCR are zero, the next instruction in sequence is skipped and the following instruction is executed. If the contents of WCR are non-zero, the next instruction in sequence is executed.

Channel	Mnemonic	SKS	Octal Configuration
A	CZT 0	SKS 012000	0 20 12000
B	CZT 1	SKS 012100	0 20 12100
C	CZT 2	SKS 012000,2	2 20 12000
D	CZT 3	SKS 012100,2	2 20 12100
E	CZT 4	SKS 052000	0 20 52000
F	CZT 5	SKS 052100	0 20 52100
G	CZT 6	SKS 052000,2	2 20 52000
H	CZT 7	SKS 052100,2	2 20 52100

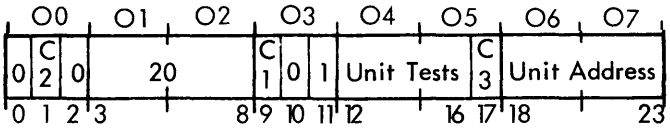
CIT CHANNEL INTER-RECORD TEST
(Skip if Inter-record Condition Present)

A test is made to determine if the channel is in the inter-record condition. This condition is present or true only after an IOSP or IORP instruction has encountered an end-of-record on input or output, and remains true as long as the channel is active (i.e., until the device is disconnected), or until a continuing EOM followed by the loading of a new instruction occurs. If the inter-record condition is true, the next instruction in sequence is skipped and the following instruction is executed. If the inter-record condition is not true, the next instruction in sequence is executed.

Channel	Mnemonic	SKS	Octal Configuration
A	CIT 0	SKS 010400	0 20 10400
B	CIT 1	SKS 010500	0 20 10500
C	CIT 2	SKS 010400,2	2 20 10400
D	CIT 3	SKS 010500,2	2 20 10500
E	CIT 4	SKS 050400	0 20 50400
F	CIT 5	SKS 050500	0 20 50500
G	CIT 6	SKS 050400,2	2 20 50400
H	CIT 7	SKS 050500,2	2 20 50500

DEVICE TESTS

The SKIP IF SIGNAL NOT SET (SKS) below is used in the channel and device test mode (mode 1) to test the condition of the peripheral devices in the system directly. The individual instructions are described in Section 6, Peripheral Equipment.



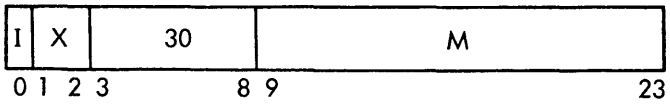
Bit Position	Designation	Function
0, 2	0	Bit positions 0 and 2 are set to zero; they are not used in this mode.
9, 1, 17	C1 C2 C3	Bit positions 9, 1 and 17 are used as an octal digit to specify the channel. Channel A is 0, channel B is 1, and so on.
3-8	20	Bit positions 3 through 8 contain the instruction code for SKS.
10, 11	01	Bit positions 10 and 11 contain the mode selection for mode 1.
12-16	Unit Tests	Bit positions 12 through 16 select the particular test and are system dependent.

SINGLE WORD TRANSFER VIA CHANNEL A

INSTRUCTIONS

Channel A can be programmed as a single-word input/output buffer. Data transfer is performed under direct program control or with the aid of the interrupt system. Interlace is not used with these instructions. The following two instructions perform the transfer.

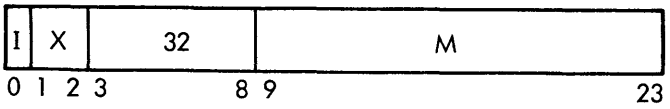
MIA MEMORY INTO CHANNEL A WHEN EMPTY



The contents of the effective memory location are transferred into the channel A word buffer. The central processor "hangs up" until the buffer is empty and ready to accept the data word.

Affected: Channel A buffer Timing: 2 + wait

AIM CHANNEL A INTO MEMORY WHEN FULL



The contents of the channel A word buffer are transferred into the effective memory location. The central processor "hangs up" until the buffer is full and ready to deliver the data word.

Affected: (EM) Timing: 3 + wait

SINGLE-WORD OPERATIONS

The single-word buffer operation of channel A is used in two ways. Data word transfers between channel A and memory can be performed under direct program control. The connect EOM and the AIM or MIA are given in sequence and the computer "hangs up" until the buffer is ready to perform the transfer. This delay is usually due to buffer tie-up while the buffer is actively transferring the previously requested data word.

The tie-up of the central processor is eliminated if the Priority Interrupt System is used with AIM and MIA. Using the Interrupt System allows the program to connect the device to be used in the transfer, to enable the interrupt, and to continue processing in the main program. When the buffer is ready to receive from, or transfer to, memory, the End-of-Word interrupt to location 010 notifies the program that the buffer is ready. A service routine is entered via a BRANCH AND MARK PLACE (BRM) instruction in location 010. This routine contains the MIA or AIM which can execute immediately without computer tie-up.

During single-word operations, the buffer error indicator is set if a parity error or incorrect timing error is detected. The incorrect timing error occurs when characters enter the buffer: during input, before the previous word is removed; during output, if characters are needed for output before a word is supplied to the buffer. The transmission is not terminated upon detection of any of these errors.

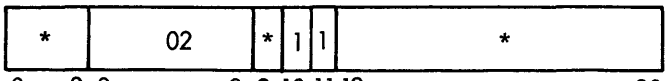
An end-of-record termination can be detected using the interrupt system. During output, use of the TERMINATE OUTPUT (TOP) after the final MIA or AIM causes an interrupt to location 011 when that final data word has been processed by the buffer. This interrupt inhibits the End-of-Word interrupt. During input, the End-of-Transmission interrupt is sent to location 011 when the end-of-record is detected. During input from devices which do not generate an end-of-record signal, the transmission is terminated using a DSC to disconnect the channel. No End-of-Transmission interrupt is received when terminating in this way.

SINGLE-BIT INPUT/OUTPUT

Operating in the system mode (mode 3), the two instructions, ENERGIZE OUTPUT M (EOM) and SKIP IF SIGNAL NOT SET (SKS), are used to provide single-bit input/output transmissions.

Execution of an EOM operating in the system mode causes a 2-microsecond signal to be transmitted to one of a possible 65,536 signal destinations. The system mode EOM format is:

EOM ENERGIZE OUTPUT M



Bit positions 0, 1, and 2 are additional special system address bits.

Bit positions 3 through 8 contain the EOM instruction code, 02.

Bit position 9 is reserved for use with the Data Multiplex option.

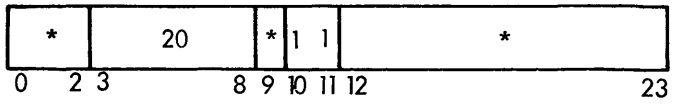
Bit positions 10 and 11 contain the system mode indicator (mode 3).

Bit positions 12 through 23 contain the 12-bit address field which specifies the special system destinations.

Affected: Special System Device Timing: 1

The SKS system test format is as follows with each corresponding bit-set being identical to the system EOM format.

SKS SKIP IF SIGNAL NOT SET



Execution of an SKS in the system test mode causes a 16-bit address to be presented to the collection of special system devices. If the external device whose address is presented is supplying a set signal to the central processor, the next instruction in sequence from the SKS is executed. If no signal is set, the next instruction in sequence is skipped and the following instruction is executed.

Affected: (P) Timing: 2 if no skip
3 if skip

PROGRAMMING THE INTERLACE REGISTER

Programming a block transmission of data using the full facility of the 9300 Input/Output System includes an "alert" EOM, an interlace set-up EOM, a POT instruction, and a buffer set-up EOM. The primary restriction on the use of these instructions is that the POT instruction be preceded by an interlace alert EOM. The alert EOM can be a separate instruction (ALC), or it can be included in a buffer set-up EOM. Thus, any of the following sequences can be used for block transmission programming:

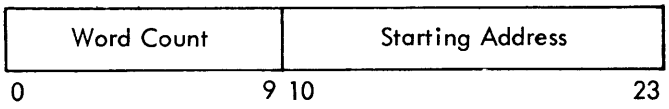
1. ALC Alert interlace (mode 1)
   EOM Set up buffer (mode 0)
   EOM Set up interlace (mode 1)
   POT Word count and starting address

2. EOM Alert interlace and set up buffer (mode 0)
   EOM Set up interlace (mode 1)
   POT Word count and starting address

3. ALC Alert interlace (mode 1)
   EOM Set up interlace (mode 1)
   POT Word count and starting address
   EOM Set up buffer (mode 0)

In sequences 1 and 2, the peripheral device is started before the interlace is engaged; in sequence 3, the peripheral device is started after the interlace is engaged. Thus, sequence 3 permits the computer operator to "step" through this sequence of instructions. It is important to note that all EOM/EOD instructions for a communication channel must have the appropriate setting of bit positions 1 and 17. In particular, this applies to the input/output control mode EOM/EOD which has no channel mnemonic.

The data word transmitted to the interlace register by the POT instruction has the following format:



The word count is right-justified in bit positions 0 through 9 of the word, providing for input/output of up to 1023 words. The starting address of the input/output operation is right-justified in bit positions 10 through 23 of the word. The input/output control EOM/EOD preceding the POT instruction provides 5 more word-count bits (extending the word count up to 32,767) and a high-order starting address bit (for starting addresses above 037777).

A sample sequence from a magnetic tape read operation is given below.

Location	Instruction	Comments
.		
.		
.		
01000	RTD *0, 1, 4 (0 02 42611)	This EOM specifies channel A, activates the interlace, is in the Buffer Control mode, specifies forward direction of tape motion with no leader and BCD character format, selects four characters per word assembly mode, and connects the unit to read tape number 1.
01001	EOM 015001 (0 02 15001)	This EOM is in the I/O control mode (1), selects the channel interrupt mode, disarms the End-of-Record interrupt, arms the Zero Word Count interrupt, selects terminal function 00 and specifies high order word count of 01.
01002	POT 01020 (0 31 01020)	This POT transmits to the channel the contents of location 01020. The location contains the word count and the starting location for data input.
.		
.		
.		
01020	00313500	This location contains the low order 10 bits of the word count and the low order 14 bits of the starting address.

The channel assembles the starting address from the EOM, bit 18, and from the word transmitted by the POT. In this sample, the starting address for the read operation is 013500. The word count is assembled from the same EOM, bits 19-23, and from the word transmitted by the POT. In this sample, the word count is 02006. This is assembled as follows. Bits 19 through 23 of the EOM are 000 01; bits 0 through 9 of the transmitted word are 0 000 000 110. Assembling these bits into one 15-bit count, 000 010 000 000 110, the word count becomes 02006.

These three instructions read one magnetic tape record of 02006 word length into memory starting at location 013500. When the word count equals zero during the transmission, an interrupt is sent to channel A interrupt level 011. Any further information is ignored and when the tape reaches the end-of-record it is stopped, disconnected, and the channel becomes inactive.

In another mode (compatible mode) of channel operation, the second EOM may be omitted. This mode allows a word count of up to 1023 (01777) words and starting addresses up to 16,383 (037777). The End-of-Word and End-of-Transmission interrupts are used when interrupts are desired. They can be armed and enabled or disarmed and disabled by the enable/disable instructions. Since the extended mode input/output functions that are specified in the second EOM cannot be used, the latter two interrupts are used along with SKS tests to determine the terminal conditions of input/output transmissions. (This I/O mode operates only for TMC channels A, B, C, and D.)

A sample line print sequence programmed in this mode follows:

Location	Instruction	Comments
.		
.		
.		
01000	PLP *0, 1, 4 (0 02 42660)	This EOM specifies channel A, activates the interlace, specifies four characters per word, and connects the unit address for Printer 1.
01001	POT 01030 (0 31 01030)	This POT transmits to the channel the contents of location 01030.
.		
.		
.		
01030	02042000	The location contains the word count (0401) and the starting address (02000) for output.

Since the input/output facility is less comprehensive in this mode, the user must be aware of the terminal conditions that will occur. For output, the mode is equivalent to function 00; that is when C words have been

transmitted, the output is terminated, and when the character has been sent, the device is disconnected.

If the interrupt system is enabled, an End-of-Record interrupt to location 011 occurs when the device disconnects. No interrupt occurs on level 010. See the Terminal Functions discussion for details.

For input, this mode is equivalent to functions 00 (IORD) and 01 (IOSD) if the end-of-record is encountered before the word count is reduced to zero. If the word count is reduced to zero before the end-of-record is encountered, the interlace portion of the channel disengages all its control of the channel buffer. The buffer continues to assemble characters until a word is completed. If the interrupt system is enabled, the buffer then generates an End-of-Word interrupt on level 010. The program has approximately 1.5 character times to reload the interlace if reading is to continue; otherwise, a character rate error occurs. On channel A, the contents of the buffer can be stored with the AIM instruction. This mode of channel operation should generally not be used for input, unless the length of the input record is fixed and known.

CHANNEL MEMORY ACCESS PRIORITY

During each memory cycle the control unit interrogates each channel to determine if it needs access to memory. If only one channel requires memory access, the channel is allowed to proceed immediately. If more than one channel requires memory access, the one that is allowed to proceed is determined on the basis of a fixed and a variable priority. The fixed priority is in the order (highest to lowest): direct access channel, time-multiplexed channel, and central processor. Time-multiplexed channels have fixed priority in the order (highest to lowest): D, C, B, A. Direct access channels have variable priority that is normally determined by a comparison of the word assembly register in each channel. The channel whose word assembly register has the fewest number of characters remaining to be filled is selected for memory access. For example, if the word assembly register in channel E has one character position unfilled and the word assembly register in channel F has three character positions unfilled, channel E is selected. Thus, each channel increases its priority level as each character is read into the word assembly register. If the contents of the register in two or more channels are equal in characters to be filled and no other channel in the set has fewer characters to be filled, priority is determined in sequence, with channel H having top priority.

Note that the number of characters to be placed in the word assembly register at any time is dependent on the characters per word count specified for the transmission. Assume, for example, that in channel E the character count is three characters per word and in channel F the character count is four characters per word. If both channel F and channel E need access to memory simultaneously, and if both have two characters filled in their respective word assembly registers, then channel E gets first memory access since it has only one character place to be filled.

5. CONTROL CONSOLE

The basic 9300 computer system includes a console for operator control of the computer. This console is connected directly to the central processor and contains a control panel (see Figure 5-1), with switches for operating the system and for displaying the contents of operational registers. The console also contains an input/output typewriter (connected to channel A) for control.

SWITCHES

POWER

The POWER switch turns the computer system power on or off. When power is on, the switch is lighted.

LOAD

The four LOAD switches, DRUM, MAG TAPE, CARDS, and PAPER TAPE are used for initial program loading. This fill procedure is:

1. With the computer in IDLE, press RESET. This clears the D register (INSTRUCTION REGISTER) and the program counter (PROGRAM LOCATION).
2. Press RUN. The computer now executes the instruction in the D register (which is a HALT). The program counter advances to 00001.
3. Press one of the four LOAD switches. Each of these switches causes an AIM 2 (0 32 00002) instruction to be inserted into the D register and the HALT to be cleared. Index 1 is loaded with 001 77771. Depending on which switch is pressed, one of the following four devices on channel A will be activated.

   Paper Tape Reader No. 1 (unit address 04)
   Card Reader No. 1 (unit address 06)
   Disc File No. 1 (unit address 26)
   Magnetic Tape No. 0 (unit address 10)

   The LOAD switch also prepares the channel to operate in the forward, binary, four character per word mode.

A "bootstrap" program contained on one of the above I/O devices must be in position to be read as the first input from the device. A typical bootstrap program is:

Location	Contents	Instruction
00002	1 32 00012	AIM 012, 1
00003	1 57 00002	BRX 02, 1
00004	1 17 00011	LDX 011, 1
00005	1 32 00000	AIM 0, 1
00006	0 20 14000	CAT 0
00007	1 57 00005	BRX 05, 1
00010	0 00 00000	HLT
00011	0 01 02000	BRU 02000

The AIM 00002 instruction that is forced into the D register stores the first word of the bootstrap program in location 2. The contents of location 2 are then executed.

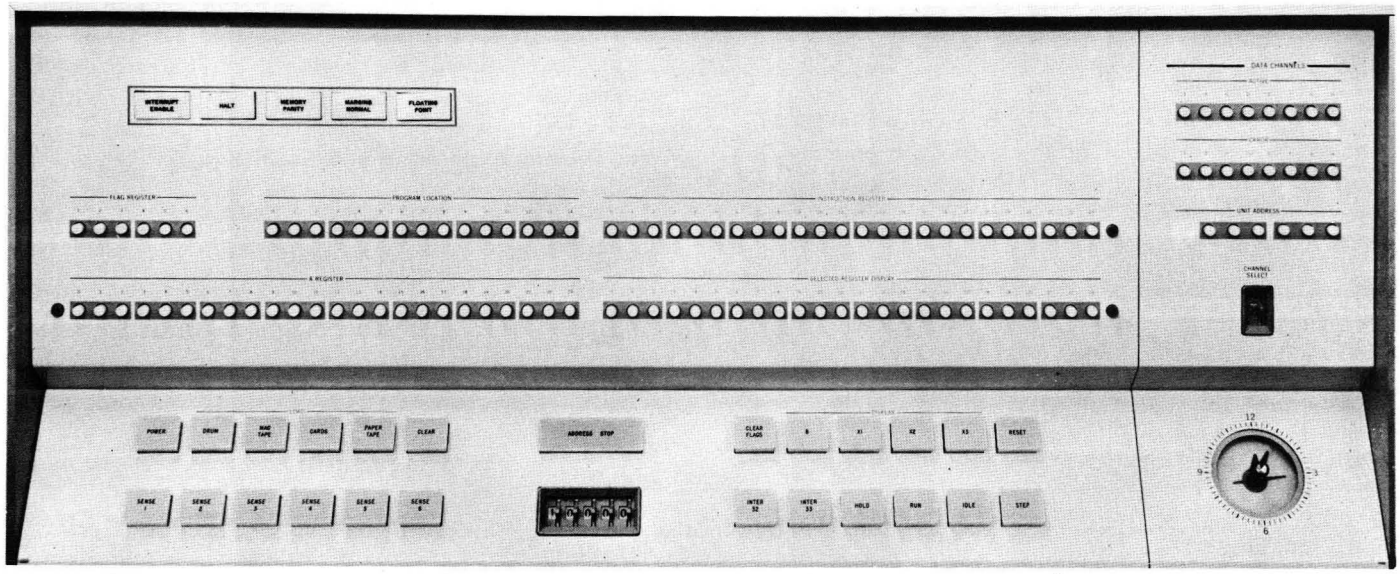


Figure 5-1. SDS 9300 Control Panel

The AIM in 2 is indexed with index X1 which contains -7. The effective address of the AIM is then 00003 so that the second word is stored in 3. This word is a BRX back to the AIM. These two instructions then load the remainder of the bootstrap program. The remaining six words can be those needed for the specific loading that is to be done. The program shown loads one record of any length into memory beginning with location 02000, and causes a halt when the record is read into memory.

The channel active test in location 6 skips when the end-of-record has been reached. In a bootstrap from paper tape or magnetic tape the record may be of any length. From cards the record is 40 words.

CLEAR, CLEAR FLAGS

The two switches, CLEAR and CLEAR FLAGS, are used together to clear all of memory. Pressing the CLEAR FLAGS switch alone clears the FLAG REGISTER to zero. If the computer is in IDLE and both clear switches are pressed concurrently, they light and memory is cleared. They must be held down for approximately 60 milliseconds to clear all of memory. If the computer is in RUN, or if only one clear switch is pressed, memory does not clear. The contents of the A, B, and index registers are not disturbed.

DISPLAY

The four DISPLAY switches, B, X1, X2, and X3, are used to select the register to be displayed in SELECTED REGISTER DISPLAY lights when the computer is in IDLE. The switches are interlocked and only one can be pressed at a time. The switch last pressed remains lit to indicate which of the registers, B, X1, X2, or X3, is displayed. During RUN, the B register is displayed regardless of which switch is pressed.

RESET

When the computer is in IDLE, this switch resets all interrupts and communication channels, clears the D register (INSTRUCTION REGISTER), the program counter (PROGRAM LOCATION), and all logical indicators in the computer.

ADDRESS STOP

The ADDRESS STOP switch, when on, will cause the computer to halt if the program counter should ever equal the address in the ADDRESS STOP dials. This switch is lit when on.

The ADDRESS STOP dials are five, 8-position, thumbnail wheel switches that can be set to octal addresses 00000 to 77777 and are used to select a program location at which to stop computation.

SENSE

The six SENSE switches give the operator manual control of the program while it is being executed. The program can test the status of these switches with the SWT instruction. The switches are of the "push-push" type and are lighted when on, or set, and not lighted when off, or reset.

INTER 32, INTER 33

The two interrupt switches, INTER 32 and INTER 33, are used to generate interrupts to either location 032 or 033. The program can place BRM instructions in these locations and allow the operator to directly execute desired programs.

HOLD

When the HOLD switch is lit, the current contents of the program counter (PROGRAM LOCATION) is inhibited from counting. Instructions inserted into the D register (INSTRUCTION REGISTER) and executed will not step the program counter.

RUN

When the computer is in IDLE and this switch is pressed, the computer begins to execute instructions starting with the one in the D register (INSTRUCTION REGISTER). If a HALT (HLT) instruction is executed, the computer stops and turns on the HALT light. To restart, the operator must press IDLE and then RUN. The RUN switch is lit when pressed.

IDLE

When the computer is in RUN, depressing this switch changes the computer to the IDLE state after it has completed execution of the instruction being processed. When the computer is in IDLE, SELECTED REGISTER DISPLAY lights display the selected register and information may be inserted in all registers.

STEP

The STEP switch is only operative when the computer is in IDLE. Pressing it will cause one instruction to be executed. Pressing repeatedly will cause additional instructions to be executed.

DISPLAYS

The registers displayed on the console directly reflect the contents of the hardware registers. If a display is cleared or changed by the operator, the contents of the actual register are identically changed.

INTERRUPT ENABLE

The INTERRUPT ENABLE indicator is on whenever the interrupt system is enabled. Pressing the RESET switch disables the interrupt system and turns the indicator off.

HALT

The HALT indicator is on whenever the computer is in RUN and is executing a HALT (HLT) instruction. It is cleared by pressing the IDLE or STEP switch.

MEMORY PARITY

If an operand access from memory encounters a parity error, this indicator is turned on. The computer goes to the Parity Error interrupt, if present. Pressing the RESET switch resets the parity detector and turns the indicator off.

MARGINS NORMAL

When the computer is in the normal operating mode, this indicator is lighted. If a test condition has been set on the maintenance console, the indicator is off.

FLOATING POINT (OPTIONAL)

When the floating point option is installed and the computer is operating in the floating point mode, this indicator is lighted.

FLAG REGISTER

This display consists of six indicators which display the state of the six program flags. Flag 1 is the OVERFLOW indicator. Pressing the CLEAR FLAGS switch clears FLAG REGISTER and the six program flags.

PROGRAM LOCATION

The program counter (P register) is a 15-bit register that contains the location of the next instruction to be executed. The program counter may be changed by inserting a BRU instruction into the D register (INSTRUCTION REGISTER) and executing it. When the computer is in IDLE, PROGRAM LOCATION displays the location of the instruction to be executed next.

INSTRUCTION REGISTER

This display consists of 24 indicator buttons, which are lighted, or not, depending on the contents of the D register. The 24-bit D register holds each instruction before, and possibly during, the time it is executed. When the computer is in IDLE, INSTRUCTION REGISTER displays the instruction to be executed next. A clear button is on the right of the display. When the computer is in IDLE, a new instruction may be inserted in the D register by pressing these indicator buttons, and may then be executed by pressing RUN or STEP. The RESET switch clears the register and the display.

A REGISTER

This display consists of 24 indicator buttons, which are lighted, or not, depending on the contents of the A register. The A register is a 24-bit register which is used for most arithmetic operations. A clear button is on the left of the A REGISTER display. When the computer is in IDLE, the contents of the A register may be set to any desired configuration by pressing the appropriate indicator buttons.

SELECTED REGISTER DISPLAY

This display consists of 24 binary indicator buttons with a clear button to the right of the display. When the computer is in RUN, SELECTED REGISTER DISPLAY always displays B. When the computer is in IDLE, the register selected by one of the DISPLAY switches on the switch panel is displayed. Any of the three index registers, X1, X2, X3, or the B register may be selected, and the contents of the selected register can be cleared and set to any desired configuration by pressing the appropriate indicator buttons.

COMMUNICATION CHANNEL ACTIVE

These eight indicators, one for each channel, reflect the operating conditions of all communication channels. The corresponding indicators are lighted as the channels become ACTIVE and are turned off as the channels leave the ACTIVE state.

COMMUNICATION CHANNEL ERROR

These indicators, one for each channel, indicate an I/O error on that channel. The indicators are turned off by a device select EOM to the channel. The RESET switch also clears these lights.

COMMUNICATION CHANNEL UNIT ADDRESS

The contents of the unit address register of the channel selected by the channel select switch are displayed in these six indicators.

6. PERIPHERAL EQUIPMENT

This section describes some of the input/output devices that can be attached to a channel, specifies the EOM and SKS instructions for each device, and provides standard programming approaches for hardware conditions peculiar to each device. In the programming examples, all octal integers are preceded by a zero unless otherwise specified, decimal integers are not preceded by a zero, and all instructions are coded for channel A without interlace.

INPUT/OUTPUT TYPEWRITER

The electric input/output typewriter is used for operator control, error or status messages, and similar functions. The typewriter has no ready test and is considered always ready.

TYPEWRITER INSTRUCTIONS

The typewriter instructions to follow are coded without interlace, using channel A at 4 characters/word, on unit number 1.

RKB 0,1,4 READ KEYBOARD
EOM 02601 0 02 02601

This instruction connects the typewriter to the channel, turns on the typewriter (lights the input light), and initializes the channel to assemble 4 characters/word.

When a typewriter input operation immediately follows typewriter output, the program must allow 40 milliseconds (22,840 computer cycles) after the channel disconnects before executing RKB. Otherwise, the last character transmitted to the typewriter may reappear as the first character read back into the channel.

TYP 0,1,4 TYPE TYPEWRITER
EOM 02641 0 02 02641

This instruction connects the typewriter to the channel, turns on the typewriter, and initializes the buffer to output 4 characters/word.

TERMINATING TYPEWRITER INPUT/OUTPUT

Since the typewriter is not a record-oriented device, it provides no terminating signals. Thus, the program must disconnect the typewriter at the end of an input with a DISCONNECT CHANNEL (DSC) instruction. If typewriter output is accomplished using interlace, the interlace control automatically terminates output (clears the unit address register in the channel). If single-word transmission is used for typewriter output, the program must terminate the output operation with a TERMINATE OUTPUT (TOP) instruction. If the channel unit address register is not cleared after a typewriter input or output, the CHANNEL ACTIVE TEST (CAT) will not cause the computer to skip an instruction.

ERROR CONDITIONS

The typewriter does not generate error signals, but if an input or output parity error or character rate error is detected by the channel, the error flip-flop in the channel is set and the DATA CHANNEL ERROR indicator on the control panel is turned on.

Example: Typewriter Output

The message:

ASSEMBLY DONE
ENTER NEW PROGRAM

is typed out under program control. The SDS 9300 internal codes for these characters are stored beginning in location 02000. The carriage return code, 052, and the space code, 012, are inserted where needed. The End-of-Record interrupt is requested. The routine is written as a closed subroutine which uses interrupts. Channel A and Typewriter Number 1 are used.

Location	Instruction	Address	Comments
01000	PZE		This instruction is an assembler instruction, used here as a convenient way to reserve the entry location for subroutine use.
	STZ	SWICH	This clears the location called SWICH. SWICH is later used to indicate to the main program that output is complete.

Location	Instruction	Address	Comments
	TYP	*0, 1, 4	This instruction connects Typewriter Number 1 to channel A for output, specifies four characters per word mode, and alerts channel A interlace. The instruction is an EOM with octal configuration, 0 02 42641.
	EXU	WRITE	This instruction causes the Input/Output Control EOM in location WRITE to be executed.
	POT	WRITE+1	This instruction sends the word count and starting address in WRITE+1 to the channel.
	BRR	01000	This instruction branches back to the main program.
WRITE	EOM 00403720	016200	This EOM specifies output function code 01 (IOSD) and the End-of-Record interrupt. The word in WRITE+1 specifies that eight words will be output from memory beginning in location 03720. According to output function 01 (IOSD) when the word count equals zero during the transmission, the device is disconnected when the last character is out; at this time, the interrupt occurs.

The main program is processed while the output operation is being performed by the channel. When finished with the output, an interrupt will be transmitted to interrupt level 011.

011	BRM	OKAY	This instruction, placed in location 011, branches and marks to location OKAY elsewhere in memory.
OKAY	PZE		This instruction saves the entry location
	MPO	SWICH	This instruction increments location SWICH as an indicator for the main program.
	BRC	*OKAY	This instruction branches to the main program and clears the active interrupt, level 011.

This is the internal code for the output message:

	A S S E	M B L Y	D O N	E C/R E N
2000	21 62 62 25	44 22 43 70	12 24 46 45	25 52 25 45
	T E R	N E W	P R O G	R A M
2004	63 25 51 12	45 25 66 12	47 51 46 27	51 21 44 12

Example: Typewriter Input

Control is requested by the operator to input four control characters. Control is obtained by pressing INTER 32 which will transmit an interrupt to interrupt level 032. The terminal interrupts are not requested in this example.

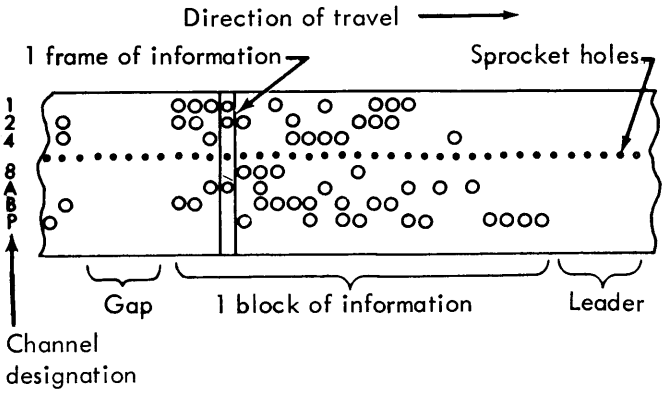
Location	Instruction	Address	Comments
032	BRM	INPUT	This instruction in interrupt location 032 branches to the input subroutine.
INPUT	PZE		This instruction saves the entry location.
	RKB	*0, 1, 4	This instruction connects channel A to Typewriter 1, specifies the four characters per word format, and alerts the interlace.
			The input request light on the typewriter is lit.

Location	Instruction	Address	Comments
			The octal configuration of the instruction is 0 02 42601. The asterisk prefixed to the address field of read and write controlling EOM instructions indicates the setting of the interlace alert bit (9).
	EXU	CHARS	This instruction executes the instruction at location CHARS.
	POT	CHARS+1	This instruction transmits to the channel the word count and starting address.
	CAT	0	This instruction tests for channel not active. If the channel is active when CAT is executed, the next instruction in sequence is executed. If the channel is inactive, the next instruction is skipped and the following one is executed.
	BRU	$-1	This instruction branches to the CAT instruction. The dollar sign and accompanying signed integer in the address field is an assembler declaration for the indicated number of locations prior to or following the current one. Plus indicates following.
	BRU	CHECK	This instruction branches to an assumed routine to determine what characters were typed in.
CHARS	EOM 00047640	014200	This EOM specifies input function 01 (IOSD) and no interrupt at the end of transmission. The word in CHARS+1 specifies that one word can be input into location 07640. Only one word will be accepted before the channel disconnects and goes inactive. The count equaling zero causes channel disconnect.

PAPER TAPE INPUT/OUTPUT

PAPER TAPE FORMAT

The paper tape uses six hole positions for information and one for odd parity check in each frame. The paper tape is one inch wide, with ten frames of information per inch in the direction of travel.



Information is organized on the tape in blocks. A block is any number of information frames, set off by a gap (in which only the sprocket hole is punched) at either end. Gap in front of a block is called "leader."

PAPER TAPE READER

The paper tape reader is primarily used for loading programs and/or data into memory. The reader is always ready for operation and no ready test is required. Before executing the EOM instruction to read a tape, the tape must be loaded into the reader. The loading procedure is:

1. Place the tape actuator into the LOAD position.
2. Insert the tape (from left to right) into the tape guide, with channel P toward the operator. (If a spool of tape is used, mount the spool on the spooler and thread the tape into the take-up spool.)
3. Place the tape actuator into the RUN position.

RPT 0,1,4 READ PAPER TAPE
EOM 02604 0 02 02604

This instruction connects paper tape reader number 1 to channel A, starts the tape moving, and transmits a block of information (1 character at a time) to the buffer. The reader ignores leader and, unless otherwise instructed by another EOM, stops within one frame of gap, generates on end-of-record signal, and disconnects from the channel buffer (clears the unit address register).

In some operations, a tape may consist of only one block, such as a source language tape prepared off-line. In this case, the program need not read the entire block at one time, but may stop the reader between frames with a DSC instruction, and then start again to read the remainder or another portion of the block. However, the paper tape reader must not be restarted until at least 30 milliseconds (approximately 17,130 computer cycles)

have elapsed following the previous read operation. Since the paper tape reader stops between frames, no frame is missed between subsequent read operations.

TERMINATING PAPER TAPE INPUT

Once a paper tape read operation is started, the paper tape reader should not be disconnected (by DSC) until at least 4 characters have been read, to prevent damaging the read mechanism. Also, if only a portion of a block is to be read in the compatible mode, DSC must be executed within 0.3 millisecond (approximately 171 computer cycles) after the last character is read. Otherwise, characters continue to enter the channel and a character rate error occurs. (The program may also store the unwanted remainder of the record into an unused portion of memory. When the reader disconnects, after reading the last character, an End-of-Record interrupt occurs if the interrupt is armed and enabled.)

ERROR CONDITIONS

If a parity or character rate error occurs during a paper tape read operation, the channel error flip-flop is set and the DATA CHANNEL ERROR indicator on the computer control panel is turned ON.

Example: Read Paper Tape

This program will read a block of 64 characters from paper tape. The four characters per word format makes the input 16 words. The tape station is turned on. Zero Word Count, interrrupt level 010, is requested for the operation on channel A. The routine is written as a closed subroutine which uses interrupts.

Location	Instruction	Address	Comments
01000	PZE		This assembler instruction saves a place for the entry location.
	STZ	SWICH	This instruction clears location SWICH which will be used as an input-finished indicator.
	RPT	*0, 1, 4	This instruction connects Paper Tape Reader 1 to channel A, specifies the four characters per word format, and alerts the interlace. The octal configuration of this EOM instruction is 0 02 42604.
	EXU	REED	This instruction executes the instruction at location REED.
	POT	REED+ 1	This instruction transmits to the channel the word count and starting address.
	BRR	01000	This instruction branches back to the main program for processing while the input operation is in progress.
REED	EOM 01003720	015200	This EOM specifies input function 01 (IOSD), and the Zero Word Count interrupt. The word in REED+ 1 specifies that input into memory begins in location 03720 and that 16 words will be read before the operation is completed. When the word count equals zero, the interrupt occurs. Then the channel disconnects. When the tape read operation is complete, the Zero Word Count interrupt occurs at level 010.
010	BRM	FNISH	This instruction, in location 010 for this example, branches and marks to location FNISH.
FNISH	PZE		This instruction saves the entry location.
	MPO	SWICH	This instruction sets an input-finished switch for use by the main program.
	BRC	*FNISH	This instruction branches back to the main program and clears interrupt level 010 from the active state.

A test to the channel, CET, for parity error during the read operation can be made before the BRC instruction.

PAPER TAPE PUNCH

The paper tape punch is primarily used for punching programs and/or data to be later loaded back into memory. The punch is always ready for operation and no ready test is required. Before executing the EOM to punch a tape, the operator should determine if there is enough tape on the supply reel for the punching operation and that the tape is properly threaded. For extensive punching operations, the tape should be threaded onto a take-up reel. After each roll of tape has been punched, the operator must empty the chad box and brush all loose chad from the tape guide. Otherwise, the punch may jam during a punching operation.

If the toggle switch on the punch panel is placed in the RUN position, the punch motor runs continuously. If the switch is in the AUTO position, the punch motor is turned on only when the punch is addressed by the buffer (with an automatic delay to allow the motor to reach punching speed) or when the FEED button on the punch panel is pressed. Tape leader may be manually punched by pressing the FEED button until the desired amount of leader is produced. The following punch tape instructions are coded for channel A, using unit number 1 at 4 characters/word, without interlace.

PPT 0,1,4 PUNCH PAPER TAPE WITHOUT LEADER
EOM 02644 0 02 02644

This instruction connects the paper tape punch to the channel, starts the punch motor (if not already on), and initializes the buffer to output 4 characters per word. Since bit position 13 contains a 1, no leader is generated before punching the first frame.

PTL 0,1,4 PUNCH PAPER TAPE WITH LEADER
EOM 0644 0 02 00644

This instruction is identical to PPT, except that bit position 13 contains a 0, to specify that the punch generate approximately 1 inch of leader preceding the first frame. PTL may be used to form separate blocks of information on a single tape, when successive punching operations are executed.

TERMINATING PAPER TAPE OUTPUT

The paper tape punch continues to punch as long as it receives characters from the channel, regardless of the infrequency of transmission. The punch operates at 60 characters per second, asynchronously. If the channel does not supply characters to the punch fast enough for operation at 60 cps, the punch waits for each character, losing no data and creating no blank frames, unless so instructed by a PTL instruction. Thus, the program must disconnect the tape punch at the end of the output operation. Otherwise, the channel unit address register is not cleared, and the computer will not skip the next instruction when CAT is subsequently executed. If the punch operation is accomplished under interlace control, a TERMINATE OUTPUT (TOP) instruction is automatically generated. If single-word transmission is used, the program must contain the TOP instruction.

The paper tape punch does not automatically produce gap after punching a block of information. If gap is desired, the operator may depress the FEED button to produce the desired gap. Also, the program may instruct the punch to produce a 1-inch gap by executing PTL followed immediately by a TOP instruction.

ERROR CONDITIONS

If a parity error occurs during a paper tape punch operation, the channel error flip-flop is set and the DATA CHANNEL ERROR indicator on the computer control panel is turned on.

Example: Punch Paper Tape

This program will punch one block ot 20 words beginning in location 02000. A 1-inch leader precedes the block. The routine is a closed subroutine which uses interrupts.

Location	Instruction	Address	Comments
01000	PZE		This assembler instruction saves a place for the entry location.
	STZ	WHERE	This instruction clears a switch location used as an indicator to the main program for completion of the punch operation.
	PTL	*0, 1, 4	This instruction connects channel A to Paper Tape Punch 1, specifies four characters per word mode, and alerts the interlace. The instruction specifies that leader is to be punched, and if not already on, the punch motor is turned on. The octal configuration of this EOM is 0 02 40644.
	EXU	PUN20	This instruction executes the I/O control EOM that sets the interrupt and selects output function 00 (IORD).

Location	Instruction	Address	Comments
	POT	PUN20+1	This instruction transmits to the channel the word count and starting address of the transmission.
	BRR	01000	This instruction branches back to the main program.
PUN20	EOM 01202000	015000	The EOM specifies output function 00 (IORD) and the Zero Word Count interrupt. The word in PUN20 specifies that 20 words will be output from memory to the punch beginning at location 02000. According to output function 00, when the word count equals zero during the transmission, the interrupt occurs. The word has not been fully transmitted at this time. When it is and the output is complete, the channel disconnects.

When the Zero Word Count interrupt occurs:

```
010     BRM     END
END     PZE
        MPO     WHERE
        BRC     *END
```

CARD INPUT/OUTPUT

CARD FORMAT

Two formats are available for reading and punching 80-column cards: Hollerith and binary. Hollerith format, as shown in Figure 6-1, consists of up to 80 Hollerith-coded characters per card, with each character represented by a single column. Thus, a card may represent up to 80 characters (20 words at 4 characters/word) in Hollerith format.

Binary format consists of two 6-bit characters per column. The top 6 rows (12-3) of column 1 form the first character (with the most significant bit in row 12), the bottom 6 rows (4-9) form the next character (with the most significant bit in row 4). Thus, a single card may represent up to 160 characters (40 words at 4 characters/word) in binary format.

CARD READER

Before initiation of a card read operation, the card reader should be loaded and tested as follows:

1. Loading procedure:
   a. Press POWER ON switch.
   b. Place cards into hopper (face down with row 12 towards the operator) and place plastic weight on the cards.
   c. Press START switch.
2. Testing procedure:
   a. Test channel (CAT)
   b. Test card reader (CRT)

CARD READER INSTRUCTIONS

If the card reader is in a ready condition when the read card EOM is executed, the reader reads 1 card (column by column, starting with column 1), transmits 80 Hollerith (or 160 BCI) characters to the channel, generates an end-of-record signal, and waits for the next EOM. The card reader instructions to follow are coded without interlace, using channel A at 4 characters/word, for unit number 1.

RCD 0,1,4 READ CARD DECIMAL (Hollerith)
EOM 02606 0 02 02606

This instruction alerts the card reader, causes a card to feed from the hopper, and specifies the Hollerith format. As each column is read, it is translated to SDS internal code.

RCB 0,1,4 READ CARD BINARY
EOM 03606 0 02 03606

This instruction alerts the card reader, causes a card to feed from the hopper, and specifies the binary format. As each column is read, it is transmitted as two 6-bit binary-coded characters.

The reading mode may be changed between card columns by executing EOM instructions with the appropriate format code. This provides a means of reading cards that have some fields punched in Hollerith and others in binary. At times, only the first portion of a card has information required by the program. In order to save the computer time required to process the unwanted information, the reader may be instructed to skip the remainder of the card.

SRC 0,1 SKIP REMAINDER OF CARD BEING READ
EOM 012006 0 02 12006

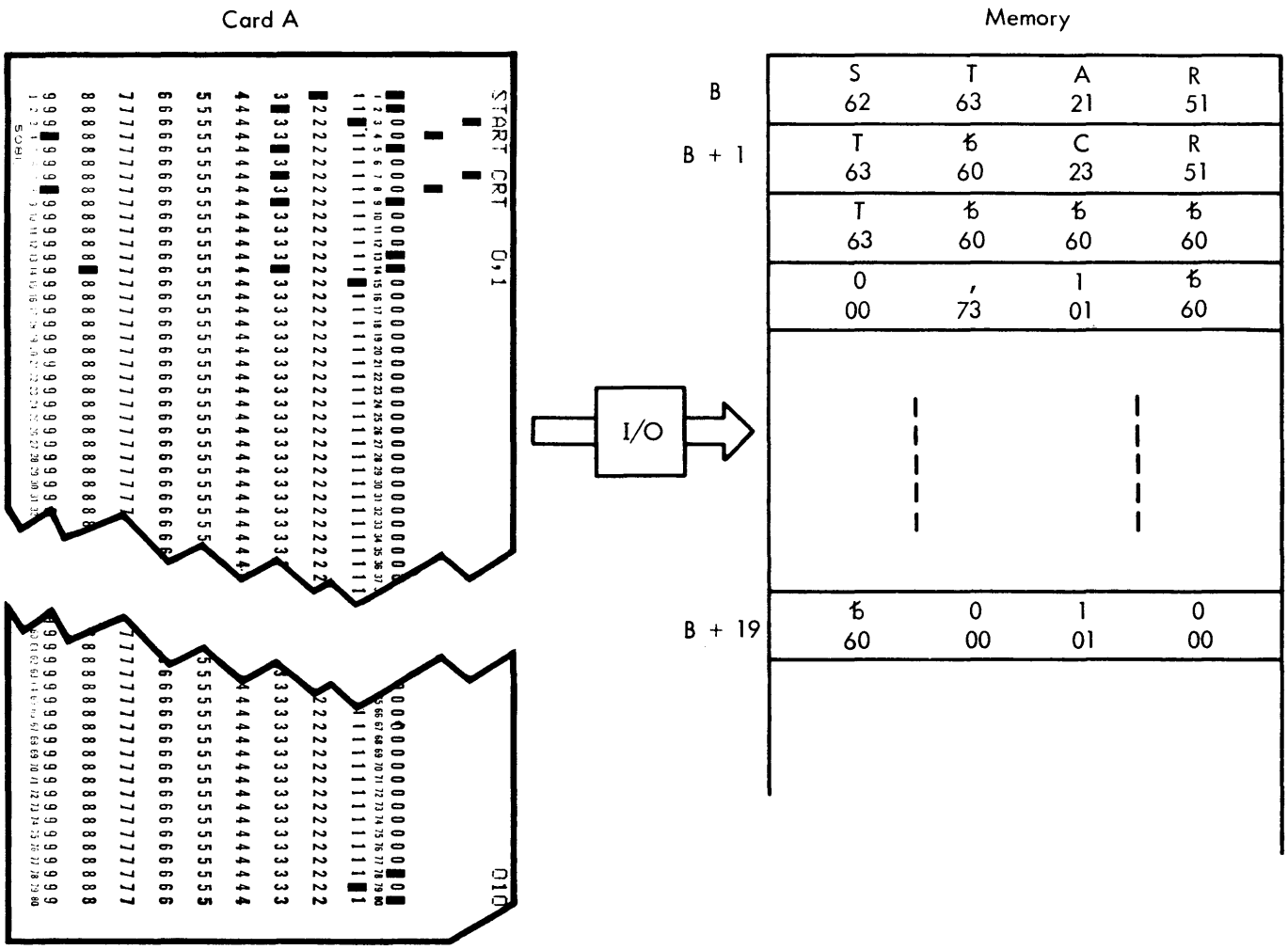


Figure 6-1. Card Read Into Memory in Hollerith

This instruction causes the reader to stop transmission of characters to the channel. The remaining characters are not checked for validity, but a read check, feed check, or end-of-record condition still cause an End-of-Record interrupt and disconnect the card reader from the channel.

CARD READER TESTS

The card reader tests to follow are coded for channel A, using unit number 1.

CRT 0, 1 CARD READER READY TEST
(Skip if Card Reader Ready)

SKS 012006 0 20 12006

The card reader is ready to feed and read when all of the following conditions exist:

1. POWER ON switch is on
2. Hopper is not empty
3. Stacker is not full
4. Feed mechanism is operating properly
5. Read mechanism is operating properly
6. START switch has been pressed
7. No feed or read cycle is in process

If the card reader is ready when CRT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the card reader is not ready, the computer executes the next instruction in sequence (does not skip). This ready test should be made before each EOM instruction that initiates a read cycle.

FCT 0, 1 FIRST COLUMN TEST
(Skip if Not First Column)

SKS 014006 0 20 14006

This test determines if the first column is about to be read by the card reader. Since the time elapsing

between the execution of a card reader EOM and the reading of the first column is approximately 85 milliseconds (48,450 computer cycles), this test allows the computer to perform other operations during this time. If FCT is executed less than 1.2 milliseconds (approximately 685 computer cycles) before the first column is due to be read, the computer skips the next instruction in sequence and executes the following instruction. If FCT is executed 1.2 milliseconds (or more) before the first column is due to be read, the computer executes the next instruction in sequence (does not skip).

CFT 0,1 CARD READER END-OF-FILE TEST (Skip if Not End of File)

SKS 011006 0 20 11006

This test determines if an end-of-file (EOF) condition exists for the card reader. The EOF condition exists when the hopper is empty and the EOF ON indicator switch is lighted. (The END OF FILE indicator is also lighted when an EOF condition exists.) If the EOF condition exists, the computer executes the next instruction in sequence (does not skip), and the EOF condition continues until the operator adds cards to the hopper or resets the EOF ON switch. If the EOF condition does not exist, the computer skips the next instruction in sequence and executes the following instruction.

ERROR AND DISCONNECT CONDITIONS

If the card reader has been instructed to read a card, the card reader response to error and disconnect conditions is as follows:

Condition	Card Reader Response
1. Feed malfunction	Disengage card reader motor
	Turn on FEED CHECK indicator
	Turn on NOT READY indicator
	Set error flip-flop in channel (test with CET)
	Disconnect card reader from buffer (clear unit address register)
	Generate End-of-Record interrupt signal
2. Read malfunction	Turn on READ CHECK indicator (other responses are identical to Feed check)
3. Validity error	Turn on VALIDITY CHECK indicator
	Set error flip-flop in buffer
4. End of card (end of read cycle)	Disconnect card reader from buffer
	Generate End-of-Record interrupt signal
	Disengage card reader motor
	Wait for new EOM

When reading cards in the single-word mode of transmission, a CHANNEL ACTIVE TEST (CAT) should be issued before each AIM to ensure that the card reader has not become disconnected (read or feed check). Otherwise, the computer will "hang up" on the AIM should the buffer become disconnected before the desired number of columns has been read.

CONTROLS AND INDICATORS

The card reader control panel provides the following controls and indicators:

POWER ON	Pressing this switch causes the POWER ON and NOT READY indicator to be lighted.
NOT READY	This indicator is lighted whenever the card reader is in a not ready condition (and POWER ON has been pressed).
START	Pressing this switch (after POWER ON has been pressed) puts the reader in a ready condition (turns off the NOT READY indicator).
EOF ON	If this switch is on (lighted) and the card hopper is empty, the end-of-file condition is satisfied. If the switch is off (not lighted), the end-of-file condition is inhibited - whether the hopper is empty or not.
END OF FILE	This indicator turns on (lights) whenever the end-of-file condition is satisfied.
FEED CHECK	This indicator turns on whenever an improper feed cycle occurs.
READ CHECK	This indicator turns on whenever a malfunction occurs in the read station during a read cycle.
VALIDITY CHECK	This indicator turns on whenever an invalid character is read during a Hollerith read (RCD) operation.
RESET	This switch is used to clear (turn off) the FEED CHECK, READ CHECK, and VALIDITY CHECK indicators.
STOP	Pressing this switch causes a not ready condition, turns on the NOT READY indicator, and stops the card reader after the card currently being read.
POWER OFF	Pressing this switch removes power from the card reader and turns off all indicators, except for EOF ON and END OF FILE.

Example: Read Card

This program reads one card in Hollerith mode. It is written as a closed subroutine that uses interrupts.

Location	Instruction	Address	Comments
01000	PZE		This assembler instruction saves a location for the subroutine entry.
	CRT	0, 1	This instruction is the card reader ready test for Card Reader 1 on channel A.
			If not ready, the next instruction is executed. If ready, the next one is skipped and the following instruction is executed. The octal configuration is 0 20 12006.
	BRU	$ - 1	This instruction branches back to the test on not ready. An exit to a not-ready corrective routine can be put here.
	RCD	*0, 1, 4	This instruction connects Card Reader 1 to channel A, alerts the interlace, and starts a card moving toward the read station. Hollerith mode is specified. The octal configuration for this instruction is 0 02 42606.
	EXU	READ	This instruction executes the I/O control EOM at location READ.
	POT	READ+ 1	This instruction transmits to the channel the word count and starting address.
	BRR	01000	This instruction branches back to the main program.
READ	EOM 01203720	015200	This EOM specifies input function 01 (IOSD) and the Zero Word Count interrupt. The word in READ+1 specifies that a record will be read into memory beginning at location 03720. A 20-word limit is specified.

The main program is processed while the card read operation is being performed by the channel. When finished with the input, an interrupt will be transmitted to interrupt level 010, the Zero Word Count interrupt location for channel A.

010	BRM	TEST	This instruction, placed in location 010 for this example, branches and marks to location TEST.
TEST	PZE		This instruction saves a location for the routine entry.
	CET	0	This instruction tests for an error on channel A. Its octal configuration is 0 20 11000.
	BRM	ERR	This instruction is executed if there is an error on channel A. It is assumed that ERR is the entry to a corrective subroutine.
	BRC	*TEST	This instruction returns control to the main program and clears interrupt level 010. This instruction is executed if no error is detected.

CARD PUNCH

Before initiation of a card punch operation, the card punch should be loaded and tested as follows:

1. Loading procedure:
   a. Turn the POWER switch ON
   b. Load the hopper with blank cards
   c. Press the START pushbutton on the control panel. (This procedure initializes the coupler and establishes the ready condition for feeding and punching the cards.)
2. Testing procedure:
   a. Test channel (CAT)
   b. Test card punch (CPT)

CARD PUNCH INSTRUCTIONS

If the card punch is ready when the punch card EOM is executed, the punch punches one 80-digit row in a card (starting with row 12) and then waits for a new EOM. Since the card punch operates by rows, the card punch program must present the entire card image to the

coupler 12 times for each card. The coupler examines the card image, and loads the punch buffer with the appropriate row image before each row is punched. After each row is punched, the punch buffer is cleared and the coupler waits for the next EOM. The card punch instructions to follow are coded without interlace, using channel A at 4 characters/word, for unit number 1.

PCD 0, 1, 4 PUNCH CARD DECIMAL (Hollerith)
EOM 02646 0 02 02646

This instruction alerts the punch, causes a card to feed past the punch station, and specifies the Hollerith format. A transmission of 80 characters (20 words at 4 characters per word) must follow this instruction. The EOM and transmission of characters must be executed 12 times for each card to be punched.

PCB 0, 1, 4 PUNCH CARD BINARY
EOM 03646 0 02 03646

This instruction is identical to PCD, except that the binary format is specified.

The EOM must be followed each time by a transmission of 160 characters (40 words at 4 characters/word). When the single-word mode of transmission is used for punching a card, each character transmission for a row must be followed by a TERMINATE OUTPUT (TOP) instruction. TOP is automatically generated with interlaced outputs.

CARD PUNCH TESTS

The card punch tests to follow are coded for channel A, using unit number 1.

PBT 0, 1 PUNCH BUFFER TEST
(Skip if Punch Buffer Ready)
SKS 012046 0 20 12046

This instruction is used to test the status of the punch buffer. If the punch buffer is clear (empty) and ready for loading when PBT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the punch buffer is not clear when PBT is executed, the computer executes the next instruction in sequence (does not skip). The punch buffer is always clear if the punch is ready to feed and punch.

CPT 0, 1 CARD PUNCH READY TEST
(Skip if Card Punch Ready)
SKS 014046 0 20 14046

The card punch is ready to feed and punch a card when all of the following conditions exist:

1. POWER switch is ON
2. Hopper is not empty
3. Stacker is not full
4. Chip box is not full
5. Feed mechanism is operating properly
6. START pushbutton has been pressed
7. No feed or punch cycle is in process

If the card punch is ready when CPT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the card punch is not ready, the computer executes the next instruction in sequence (does not skip). This ready test should be made before each EOM instruction that initiates a punch cycle.

ERROR CONDITIONS

If the card punch has been instructed to feed and punch a card and the card does not feed properly (or the punch buffer is not loaded at punch time), the error flip-flop in the channel is set.

Example: Punch Card

This program punches one card in Hollerith mode. It is written as a closed subroutine which uses interrupts. Index register X3 is used to count the 12 times the card image is presented to the punch.

Location	Instruction	Address	Comments
01000	PZE		This assembler instruction saves the location for the subroutine entry.
	STZ	SWICH	This instruction clears a switch to be used later.
	LDA STA	01000 ENTR2	This pair of LDA and STA place the main program mark address in location ENTR2.
	MPO	ENTR2	MPO adds one to the stored contents.
MCRDS	LDX	ROWS, 3	This LDX instruction initializes index register X3 with 077700013, which is a base of 11 decimal and an increment of -1.

Location	Instruction	Address	Comments
	CPT	0, 1	This instruction tests the card punch for a ready condition. The card punch is number 1 on channel A.
	BRU	$ - 1	This instruction is executed if the punch is not ready. It branches back to the test, CPT. An exit to a time loop with the facility to tell the operator that the card punch will not become ready can be placed here.
GETRW	PCD	*0, 1, 4	This instruction is executed if the punch is ready. It alerts channel A with interlace, connects Card Punch 1 to channel A, and starts a card moving toward the punch station. Four characters per word and Hollerith format are specified.
	EXU	PNCH	This instruction executes the EOM located in PNCH.
	POT	PNCH + 1	This instruction transmits to the channel the word count and starting address.
	BRU	*ENTR2	This instruction branches back to the main program.
PNCH	EOM 01202000	015000	This EOM specifies output function 00 (IORD) and the Zero Word Count interrupt. The word in PNCH + 1 specifies that 20 words will be output from memory beginning in location 02000. Note that the card image must be sent to the channel 12 times to punch a card
ROWS	77700013		

The main program is processed while the output is being performed by the channel. When finished with the output, an interrupt will be transmitted to interrupt level 011, the End-of-Record location for channel A.

011	BRM	ENTR2	This instruction branches and marks to location ENTR2.
ENTR2	PZE		This assembler instruction saves a location for routine entry.
	BRC	$ + 1	This instruction is used to clear the interrupt. It is cleared here to allow other types of return later in the program. It branches to the next location.
	BRX	GETRW, 3	This instruction adds the increment in index X3 to the base in index X3. If the base has not been reduced below zero, the next instruction executed at location GETRW. When the base is reduced to zero, the next instruction in sequence is executed. This index counts "row times" on each card.
	MPO	SWICH	This instruction sets a switch to indicate to the main program that the punch operation is complete.
	BRC	*ENTR2	This instruction returns control to the main program.

LINE PRINTER

SDS buffered line printers are capable of printing up to 1000 lines per minute at 132 characters per line, with a standard set of 56 characters. Printing is accomplished by means of a rotating character drum and a bank of 132 print hammers. The drum passes 56 different characters, in lines of 132 each, past the hammer bank. Upon command from the computer, the selected print hammers drive the paper against the ribbon and onto the appropriate character typeface as it passes the print position. The characters are transmitted sequentially for storage in the printer buffer before printing. A programmable format tape loop provides fixed (or preselected) space control. Upspacing of 1 to 7 lines, as well as page control, may be accomplished by program instructions.

An optional, off-line facility allows the program or the operator to initiate card-to-printer or magnetic tape-to-printer operations simultaneous with computation (see Off-Line Printing).

PRINTER CONTROLS

The printer controls, Figure 6-2, for SDS line printers consist of eight switches and indicators.

The POWER/ON switch is an alternate action switch. The computer must be turned on for this switch to be

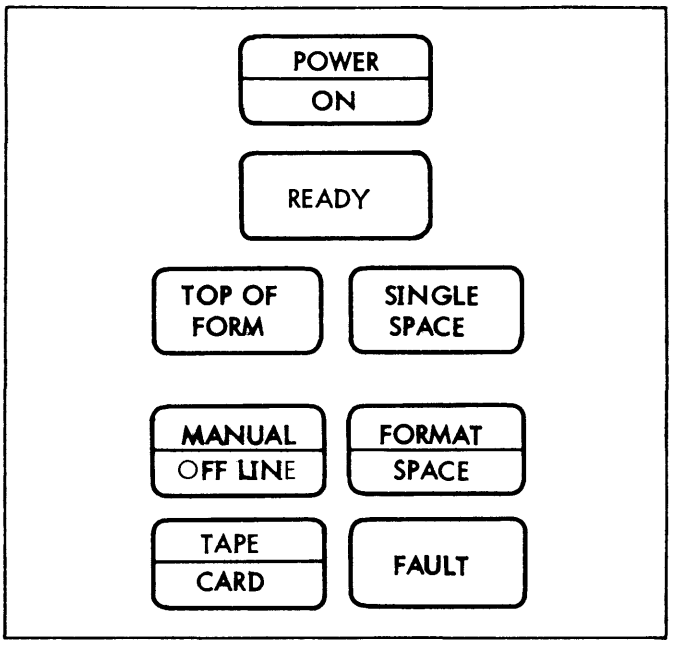


Figure 6-2. Printer Control Indicator Lights and Switches

activated. Pressing POWER/ON lights the top half of the indicator, turns on the motors and hammer driver power supply, and starts a timer that allows the motors to reach proper speed. After 20 seconds the bottom half lights, indicating that the printer is operable.

When the printer is initially turned on, the READY indicator is off. When pressed, it is turned on if:

1. paper is loaded in the line printer,
2. the lower half of the POWER/ON switch is lighted, and
3. the hammer power supply is on.

This indicator automatically goes off when the above conditions are not realized. The printer is ready for either on-line or off-line operation when READY is turned on. READY is reset to preclude computer intervention while changing paper or ribbon, or operating the TOP OF FORM or SINGLE SPACE switches.

Pressing TOP OF FORM causes the printer to position paper according to format tape channel 1. This indicator is lighted only when the format tape is positioned at channel 1, that is, top-of-form on a standard tape loop. This switch is operative when there is paper in the printer and the READY indicator is off.

Pressing SINGLE SPACE causes the printer to upspace paper one single space, independently of the vertical format tape. This switch is operative when there is paper in the machine and READY is off.

The FAULT indicator lights when the printer detects a parity error as information transfers from the buffer to the print hammers, or when it detects a parity error in incoming data from magnetic tape or cards during an off-line operation. It remains lighted until the next EOM addresses the printer. The condition of the light corresponds to the status of a program-testable fault indicator in the printer.

MANUAL/OFF LINE[†] is a combination of a switch and two independent indicators. The program or the operator may initiate off-line operation, which is indicated by the illumination of the bottom half of this switch (OFF LINE). If the operator presses this switch to initiate off line operation, the top half (MANUAL) is also lighted, and remains lighted until the operator presses the switch again. OFF LINE is normally reset when the end-of-file is detected from the input unit. Pressing READY (when READY is lighted) also resets OFF LINE, that is, by switching the printer from the "ready" to the "not ready" state.

The FORMAT/SPACE[†] switch is used in off-line operation. The operator may use either mode, spacing a single space after each line of print, or using the first character stored on tape or cards as a vertical format character.

The TAPE/CARD[†] switch selects the desired input device.

PAPER TAPE FORMAT LOOP

A paper tape format loop, placed in the printer, allows upspacing to proceed to prespecified vertical positions on the print page. The format loop is an eight-channel paper tape. Putting a punch in the specified channel at the desired vertical spacing selects the channel upspace. Channel 1 is the top of form channel, channel 7 is the bottom of form channel, and channel 0 is the single upspace channel. In the off-line mode with SPACE control, channel 0 controls single spacing. When printing with no format loop inserted in the printer, single upspacing occurs regardless of the channel specified.

LINE PRINTER INSTRUCTIONS

PLP 0,1,4 PRINT LINE PRINTER
EOM 02660 0 02 02660

This instruction connects the line printer to channel A and specifies a character transmission of 4 characters per word.

This instruction is followed by the transmission of up to 132 characters. If the character count is less than 132, the characters are printed left-justified on the page. If the character count is more than 132, the printer produces an undetectable error.

[†]If an off-line coupler is not attached to the printer, the MANUAL/OFF LINE, FORMAT/SPACE, and TAPE/CARD indicators neither light nor affect printer operation.

The following control instructions are coded for channel A using unit number 1:

POL 0,1 PRINTER OFF-LINE
EOM 012060 0 02 12060

This instruction places the printer off-line and initiates an off-line print operation. The selected input device (card reader 1 or magnetic tape unit 7) also goes off-line. (See Off-Line Printing.)

PSC 0,1,n PRINTER SKIP TO FORMAT CHANNEL n
EOM 01n460 0 02 1n460

This instruction causes the printer to eject paper until the paper tape format loop detects the first punched hole in the channel specified by the number n (0 to 7). (See PSP for timing.)

PSP 0,1,n PRINTER UPSPACE n LINES
EOM 01n660 0 02 1n660

This instruction causes the printer to upspace n (0 to 7) lines. Consecutive upspace instructions must be separated by a sufficient time delay. Otherwise, the two PSP instructions may be merged by the printer.

Approximate completion times for PSP (from initiation of instruction to paper stop) are:

Upspace 1 line: 25 milliseconds (14,275 cycles)

Upspace more than 1 line: add 10 milliseconds (5,690 cycles) for each additional line.

LINE PRINTER TESTS

The line printer tests to follow are coded for channel A, using unit number 1:

PFT 0,1 PRINTER FAULT TEST
(Skip if no Printer Fault)
SKS 011060 0 20 11060

This test determines if the printer has detected a parity error during a transfer of information from the printer buffer to the print hammers. If such an error occurs, a fault detector is set and the FAULT indicator is lighted. If the fault detector is set when PFT is executed, the computer executes the next instruction in sequence (does not skip). If the fault detector is not set, the computer skips the next instruction in sequence and executes the following instruction.

PRT 0,1 PRINTER READY TEST
(Skip if Printer Ready)
SKS 012060 0 20 12060

This instruction tests the printer for a "ready" condition. The criteria for a printer "ready" condition are:

1. Paper is loaded in the machine,
2. The lower half of the POWER ON switch is lighted, and
3. The hammer power supply is on.

If the printer is ready when PRT is executed, the computer skips the next instruction in sequence and executes the following instruction. If the printer is not ready, the computer executes the next instruction in sequence (does not skip). Since the printer tests ready while ejecting paper, the program should allow a definite time interval to pass (see PSP) after a PSC or PSP instruction before executing a new PSC or PSP. A dummy PLP instruction may be issued between two space instructions (PSC or PSP). This instruction will provide the timing required. A Ready test may be used to determine when the second paper space instruction may be sent.

EPT 0,1 END OF PAGE TEST
(Skip if Not End of Page
SKS 014060 0 20 14060

This instruction tests the printer for paper position. If the paper is positioned at the end of page (specified by format channel 7), the computer executes the next instruction in sequence (does not skip). If the paper is not positioned at the specified end of page, the computer skips the next instruction in sequence and executes the following instruction.

TERMINATING LINE PRINTER OUTPUT

When the single-word mode of transmission is used for printing on the line printer, each character transmission for a line must be followed by a TERMINATE OUTPUT (TOP) instruction. TOP is automatically generated with interlaced outputs.

ERROR CONDITIONS

1. Print fault - parity error during transfer of character information from print buffer to print hammers.
2. Buffer error - parity or character rate error during transfer of information through buffer.
3. Input fault - parity error in incoming data from cards or magnetic tape (during off-line operation only).

Example: Print Two Lines

This program will print two lines with a single upspace between them. The paper will be positioned at the top of the page. It is assumed that the printer is ready to print or is becoming ready after a print operation. This program, written as a closed subroutine, uses channel A, line printer 1, and the Zero Word Count interrupt.

Location	Instruction	Address	Comments
01200	PZE		This assembler instruction reserves a location for subroutine entry.
	STZ	SWICH	This instruction initializes a location, SWICH, which is used to indicate printing completed.
	PRT	0, 1	This instruction tests for printer ready. The octal configuration for this instruction is 0 20 12060.
	BRU	$ - 1	This instruction returns control to the ready test. It is executed if the printer is not ready.
	PSC	0, 1, 1	This instruction instructs the printer to move paper to the top of the page. The octal configuration for this instruction is 0 02 11460.
	PLP	*0, 1, 4	This instruction connects printer 1 to channel A and specifies four characters per word transfer mode. The interlace is also alerted. The octal configuration for this instruction is 0 02 42660.
	EXU	PRINT	This instruction executes the EOM at location PRINT.
	POT	PRINT+1	This instruction transmits the word count and starting address.
	BRR	01200	This instruction branches back to the main program while the line is being printed.
PRINT	EOM 02043720	015000	This EOM specifies output function 00 (IORD) and the Zero Word Count interrupt. The word in PRINT+1 specifies that 33 words will be output from memory beginning in location 03720.

The main program is processed while the data transfer is being completed. When completed, the Zero Word Count interrupt will be transmitted to interrupt level 010. This indicates that all the data from memory has been obtained, not that the printing of the line is complete. The latter will be determined by testing.

010	BRM	UPSPC	This branches and marks to location UPSPC elsewhere in memory.
UPSPC	PZE		This instruction reserves a location for an entry.
	RCH SKE BRC	037701 SWICH *UPSPC	These instructions determine if the two lines have been printed. If so, the BRC branches and clears the interrupt to the main program. This is the final exit.
	PRT	0, 1	This instruction tests for printer ready. As soon as the current line is printed, the printer will be ready.
	BRU	$ - 1	This instruction is executed if the printer is not ready.
	PSP	0, 1, 1	This instruction (octal 0 02 11660) upspaces the printer one line.
HEAR	PLP	*0, 1, 4	This instruction sets up the printer with interlace.
	EXU	PRNT	This instruction executes the EOM instruction in location PRNT.
	POT	PRNT+1	This instruction transmits the word count and starting address.
	MPO	SWICH	This instruction flags the printing of the second line.
	BRC	*UPSPC	This instruction branches and clears the interrupt to the main program to await completion of the data transfer.
PRNT	EOM 02042041	015000	This EOM specifies output function 00 (IORD) and the Zero Word Count interrupt. The word in PRNT+1 specifies that 33 words will be output from memory beginning in location 02041. The channel disconnects at the end of the output.

Note that the print and control instructions starting at location HEAR are executed before the printer has had time to completely upspace the paper as requested. These instructions cause an immediate transfer of data into the print buffer, and printing begins immediately after upspacing is completed.

OFF-LINE PRINTING

The optional, off-line facility allows the line printer to produce printed records from card or magnetic tape sources without computer attention. The character transmission proceeds directly from the source to the printer and the channel may still be used by the computer for other input/output operations (e.g., card reading on card reader 2, card punch, paper tape read/punch, disk read/write, etc.). Once initiated, the printing operation is controlled by the source and proceeds until the source generates an end-of-file signal (see card input and magnetic tape input for appropriate end-of-file conditions).

The FAULT indicator lights when a parity error is detected during the reading of a tape record; the off-line printer rereads the record in an attempt to read good data. If this reread record contains an error, FAULT lights, the off-line operation terminates, and the printer goes back on-line if physically connected to the computer and the MANUAL indicator is off. When a validity check occurs during a card read, FAULT lights, the operation terminates, and the printer goes back on-line if the MANUAL indicator is off. The next EOM addressing the printer resets FAULT if the printer is on-line. If the MANUAL indicator is on, the error condition may be cleared by pressing READY off and then on again. If a fault occurs in an off-line operation initiated by the computer, the usual method for clearing the error is:

1. Press MANUAL on.
2. Press READY off.
3. Press READY on.
4. Press MANUAL off.

In a manually-initiated off-line operation, steps 1 and 4 are not required.

Off-line printing can be formatted as desired through the use of a single upspace or the format control mode (see Table 6-1). Off-line printing terminates by an end-of-file indicator from either device. Upon termination of an off-line operation, a physically connected off-line printer system returns on-line, provided the MANUAL indicator is off.

Table 6-1. Format Control Characters

Code	Character	Function
00	0	Skip to format channel 0
01	1	Skip to format channel 1
02	2	Skip to format channel 2
03	3	Skip to format channel 3
04	4	Skip to format channel 4
05	5	Skip to format channel 5
06	6	Skip to format channel 6
07	7	Skip to format channel 7

Table 6-1. Format Control Characters (Cont.)

Code	Character	Function
40	-(hyphen)	Do not space
41	J	Upspace 1 line
42	K	Upspace 2 lines
43	L	Upspace 3 lines
44	M	Upspace 4 lines
45	N	Upspace 5 lines
46	O	Upspace 6 lines
47	P	Upspace 7 lines

PRINTING OFF-LINE UNDER OPERATOR CONTROL

The procedure for operator control of off-line printing is:

1. Switch on the desired input device. (Magnetic tape is selected by dialing it to logical tape number 7.)
2. Place paper at top of form, as desired, by means of the TOP OF FORM switch.
3. Select desired input device by means of the TAPE/CARD switch.
4. Select either the FORMAT or SPACE mode as required.
5. Press MANUAL/OFF LINE switch.
6. Press READY switch on, which initiates actual data transfer.

PRINTING OFF-LINE UNDER COMPUTER CONTROL

The procedure for computer control of off-line printing is:

1. Turn the equipment on.
2. Prepare the desired input device for operation.
3. Select desired input device by means of the TAPE/CARD switch.
4. Select either the FORMAT or SPACE mode as required.
5. Press the READY switch on.
6. Under program control, test the tape or card unit and the line printer for "ready" condition.
7. Then, to start transfer of data, give the POL instruction to print off-line.

OFF-LINE PRINT TERMINATION

Off-line printing terminates when an end-of-file indicator from the magnetic tape unit or card reader occurs. When printing from magnetic tape, the print operation terminates when the first character read from a record is the end-of-file code, octal 17.

When printing from cards, the print operation terminates when the end-of-file signal comes from the reader. This occurs when the card hopper becomes empty and the EOF ON switch on the reader is on (END OF FILE indicator lights). If the hopper becomes empty when EOF ON is not lighted, the printer waits for more cards to be placed in the hopper and the reader to become ready. When the reader is again ready, printing resumes.

MAGNETIC TAPE INPUT/OUTPUT

MAGNETIC TAPE FORMAT

All magnetic tape units used by the SDS 9300 System are IBM-compatible. The tape is one-half inch wide Mylar base material, 1.5 mils thick. Tape reels (10.5-inch, plastic) can contain up to 2400 feet of tape. A reflective marker is placed on the back of the tape, approximately ten feet from its beginning, to indicate the load point. The leading ten feet are used for threading tape through the guides on the unit. The load-point marker is on the Mylar side of the tape along the edge nearest the operator when the tape is mounted. A similar marker is placed along the other edge of the tape to mark the end-of-reel. About 14 feet of tape are reserved between the end-of-reel marker and the end of the tape. This space includes at least ten feet of leader and enough tape to hold a record of 9600 characters in 200 bpi density after the end-of-reel marker is sensed.

Characters are recorded on tape in seven parallel tracks. A change in the magnetic flux in a track is used to record a 1-bit for a given character position. No change in magnetic flux indicates a 0-bit. Six of the tracks are used for information; the seventh track is a parity check. Both even and odd parity are used. Tape can be recorded in binary mode using odd parity. In this mode the six-bit characters from the channel are recorded without alteration. Data also can be recorded in binary-coded decimal (BCD) mode using even parity. In this mode, characters from the channel are transformed to IBM standard BCD interchange code (see Appendix A).

Information on tape is arranged in blocks that may contain one or more records. A record may be any length within the capacity of available core storage in the computer. Records or blocks of records are separated on tape by a record gap (section of blank tape) about 3/4-inch long. In writing, the gap is automatically produced at the end of a record or block. Reading begins with the first character sensed after the gap and continues until the next gap is encountered.

An inter-record gap, followed by a special, single-character record, is used to mark the end of a file of information. The character is a tape mark (0001111) and is recorded by writing a one-word record in BCD with one-character-per-word format. One or more files may be written on a reel of tape. On reading an end-of-file record, the tape control unit stops the tape and sets its end-of-file indicator, which may be tested by the program.

The tape control unit will consider any record which contains only tape mark (0001111) characters an end-of-file. All such characters will be read into memory as requested.

As information is written, an odd-even count is made of the number of 1-bits in each channel. At the end of each record a bit is written for each channel so that the total number of 1-bits in each track will be even. This check is always even whether the character parity is even or odd. The character containing these check bits is called the longitudinal parity character and is written slightly past the end of recorded information in the block.

Since the longitudinal check character always reflects an even parity check for each channel, in the BCD mode, the check character itself will always have an even number of 1-bits. In the binary mode, however, the check character may have either an even or an odd number of 1-bits. This means that a reverse scan over a binary record may result in turning on the error indicator in the channel even though the record itself is correct. As a general rule, the error indicator should be ignored after a reverse read or reverse scan operation.

It is possible to write tape in a one-, two-, or three-character-per-word mode provided characters can be supplied at a sufficient rate. On reading, however, the tape unit uses the character count to ascertain when it has read two characters and can look for gap. If a one-character-per-word read were started, a single noise character would stop the tape. In reverse scan a one-character-per-word operation would cause the tape to stop after detecting the longitudinal check character at the end of the record with the tape positioned in the area of recorded information.

All scan operations must be in three- or four-character-per-word mode or the tape will not stop when it reaches gap.

As a general rule, tape units should be programmed for three or four characters per word if possible. The write-tape-mark operation is an exception to this rule.

The TAPE READY TEST (TRT) should be used between tape operations of opposite direction to ensure that the tape unit stops and reverses. It is advisable to terminate tape writing by erasing several inches of tape whenever subsequent resumption of recording is anticipated. This will eliminate the effects of a possible extraneous character which might arise through subsequent tape repositioning.

MAGNETIC TAPE UNIT TESTS

The magnetic tape unit tests to follow are coded for channel A, with n being the number (0-7) of the magnetic tape unit.

TRT 0, n TAPE READY TEST
(Skip if Tape Not Ready)
SKS 01041n 0 20 1041n

Tape unit n is tested for not ready. If the tape is not ready, the next instruction in sequence is skipped and the following instruction is executed. If the tape is ready, the next instruction in sequence is executed.

A tape is not ready:

1. If there is no physical unit set to the logical unit number being tested.
2. If the selected unit is not in the automatic mode, or
3. If the tape is in motion for any operation.

FPT 0, n FILE PROTECT TEST
(Skip if Tape Not File Protected)
SKS 01401n 0 20 1401n

Tape unit n is tested for file protect ring. If the file protect ring is inserted, the next instruction in sequence is skipped and the following instruction is executed. If not inserted, the next instruction in sequence is executed. The skip will not occur if there is no logical unit n on the channel. This instruction should be used before any write operation to determine whether it is possible to perform the write operation.

BTT 0, n BEGINNING OF TAPE TEST
(Skip if Not Beginning of Tape)
SKS 01201n 0 20 1201n

Tape unit n is tested at the beginning of the tape. If it is not positioned on the load-point marker, the next instruction in sequence is skipped and the following instruction is executed. If positioned at the load-point marker, the next instruction in sequence is executed. The skip will not occur if there is no logical unit n on the channel.

ETT 0, n END OF TAPE TEST
(Skip if Not End of Tape)
SKS 01101n 0 20 1101n

Tape unit n is tested to see if it has sensed the end of the tape. If the tape unit has not sensed the end-of-reel marker, the next instruction in sequence is skipped and the following instruction is executed. If the end-of-reel marker has been sensed, the next instruction in sequence is executed. The end-of-reel condition is reset when the tape is moved backwards over the end-of-reel marker. The skip will not occur if there is no logical unit n on the channel.

DT2 0, n DENSITY TEST, 200 BPI†
(Skip if Not 200 BPI)
SKS 01621n 0 20 1621n

Tape unit n is tested for being set at 200 bpi density. If not, the next instruction in sequence is skipped and the following instruction is executed. If so, the next instruction is executed.

DT5 0, n DENSITY TEST, 556 BPI†
(Skip if Not 556 BPI)
SKS 01661n 0 20 1661n

Tape unit n is tested for being set at 556 bpi density. If not, the next instruction in sequence is skipped and the following instruction is executed. If so, the next instruction in sequence is executed.

DT8 0, n DENSITY TEST, 800 BPI†
(Skip if Not 800 BPI)
SKS 01721n 0 20 1721n

Tape unit n is tested for being set at 800 bpi density. If not, the next instruction in sequence is skipped and the following instruction is executed. If so, the next instruction in sequence is executed.

TFT 0 TAPE END-OF-FILE TEST†
(Skip if Not End of File)
SKS 013610 0 20 13610

The tape control unit is tested to determine if a tape under its control encountered an end-of-file during the last read or scan operation. If not, the next instruction in sequence is skipped and the following instruction is executed. If end-of-file was encountered, the next instruction in sequence is executed.

The end-of-file indicator remains set until another tape operation is called for.

TGT 0 TAPE GAP TEST†
(Skip if Tape Not in Gap)
SKS 012610 0 20 12610

The tape control unit is tested to see if a tape under its control is in motion in the gap following a record; if not, the computer skips the next instruction in sequence and executes the following instruction; if so, the computer executes the next instruction in sequence.

†Note: These instructions apply only to 41.7-kc and 96-kc magnetic tape systems.

When the tape unit detects the gap at the end of a record and has checked the longitudinal parity character, it generates the gap signal. This signal remains true for approximately one millisecond (approximately 571 computer cycles). During this time, the test instruction does not skip, and the tape may be given a command to continue in the direction it is going. If so programmed, the tape continues without stopping. If the record encountered should be an end-of-file, the gap signal does not become true, the tape always stops, and the test instruction skips.

MAGPAK TEST
(Skip if Not MAGPAK)

SKS 01021n 0 20 1021n

Tape unit n is tested for being a MAGPAK. If the tape unit is not a MAGPAK, the computer skips the next instruction in sequence and executes the following instruction. If the tape unit is a MAGPAK, the computer executes the next instruction in sequence.

READING MAGNETIC TAPE

Once a tape is started with a read binary or read BCD EOM/EOD, it continues until an end-of-record gap is detected. If the computer does not instruct it to continue, it will then stop in the middle of that gap. When the tape stops, the tape unit disconnects from the channel. If an end-of-file is encountered, the tape control unit sets its EOF indicator. This indicator can be tested by the central processor and will remain set until a new EOM/EOD is given to a tape unit on that channel. The tape always stops after the tape mark.

The EOF character (0001111) is read into memory along with its check character. In a four-character-per-word read, this will appear in the first word of the input area as a 17170000 word.

Once a record has been written on tape, it cannot be certain that any subsequent record previously written can be read. This means that a record in the middle of a file cannot be updated or rewritten if following records need to be read.

Any errors detected, either by the channel (in the character parity) or by the control unit (with longitudinal parity), sets the error indicator in the channel. When an error is detected in reading, the tape should be backspaced over the erroneous record and a reread attempted.

If the end-of-reel marker is encountered while reading, the end-of-reel indicator in the tape unit is set and may be interrogated by the program at any time. An end-of-file is normally used to indicate the end-of-record information on tape. It is possible, however, to use the end-of-reel indicator to mark the last record on the reel.

BACKSPACE

A backspace record is implemented using the scan feature. A scan reverse EOM/EOD is used to start the tape in reverse. The channel is then loaded with a 00 function (IORD) with a 0 count, and the EOR interrupt armed. When the channel signals that the operation is complete, the tape is situated with the read-write head in front of the last record scanned.

SCAN

A scan operation is similar to a read operation except that the channel shifts the characters read through its word assembly register, but does not consider a word complete until a tape gap is encountered. When the gap is reached, the channel uses the last four characters in the word assembly register as the only word read from the record. When scanning in reverse at 4 characters/word, the word consists of the last four characters scanned, which are the first four logical characters of the record. These characters will be assembled in reverse. For example, if the first four characters of the record were 1234 and the record was scanned in reverse, these would appear as 4321 in the word stored for that record.

The same operation occurs in the forward scan with the last four characters of the record forming the word stored. The scan is useful for reverse searching on the first word of the records in the file being searched. In this case, the tape is started in a reverse scan at 4 characters/word and the channel interlace loaded with a terminal function with a word count of 1, and the Zero Word Count interrupt is armed. When the beginning of the record is reached, the channel interrupts the program. The program checks the first word in the record against a search key. If they agree, then the program need only wait for the channel to become inactive and the record may be read forward. If the record is not the desired one, the program gives another scan reverse without waiting for the channel to become inactive, and reloads the channel interlace to scan the next record.

NOTE: The reverse scan must be performed at 4 characters/word only; otherwise, an undefined operation occurs when the scan is attempted.

RTB 0, n, 4 READ TAPE IN BINARY
EOM 0361n 0 02 0361n

Tape unit n is started in a Binary read mode.

RTD 0, n, 4 READ TAPE IN DECIMAL (BCD)
EOM 0261n 0 02 0261n

Tape unit n is started in a BCD read mode.

SFB 0, n, 4 SCAN FORWARD IN BINARY
EOM 0363n 0 02 0363n

Tape unit n is started forward in a Binary scan mode.

SFD 0, n, 4 SCAN FORWARD IN DECIMAL (BCD)
EOM 0263n 0 02 0263n

Tape unit n is started forward in a BCD scan mode.

SRB 0, n, 4 SCAN REVERSE IN BINARY
EOM 0763n 0 02 0763n

Tape unit n is started in reverse in a Binary scan mode.

SRD 0, n, 4 SCAN REVERSE IN DECIMAL (BCD)
EOM 0663n 0 02 0663n

Tape unit n is started in reverse in a BCD scan mode.

Example: Read Magnetic Tape

This program reads one record from Magnetic Tape Unit 1 on channel A. The program is written as a subroutine that uses the End-of-Record interrupt. It is assumed that the tape is not at the beginning or the end of tape.

Location	Instruction	Address	Comments
01000	PZE		This assembler instruction saves a location for the subroutine entry.
	TRT	0, 1	This instruction tests for ready on Magnetic Tape 1, channel A. If Magnetic Tape 1 is ready to perform an input/output operation, the next instruction in sequence is executed; if not, the next instruction is skipped and the following one is executed. The octal configuration for the command is 0 20 10411.
	BRU	$+2	This instruction skips one instruction.
	BRU	$-2	This instruction branches back to TRT. An exit to a routine that determines reasons for the non-ready condition can be placed here.
	RTD	*0, 1, 4	This instruction alerts channel A, activates interlace, connects it to Magnetic Tape 1, and starts tape motion. The four characters per word and BCD format are specified.
	EXU	REDTP	This instruction executes the EOM located in location REDTP.
	POT	REDTP+1	This instruction transmits to the channel the word count and starting address.
	BRR	01000	This instruction branches back to the main program.
REDTP	EOM 06202000	016000	This EOM specifies input function 00 (IORD) and the End-of-Record interrupt. The word in REDTP+1 specifies that one record or 100 words, whichever is smaller, will be read into memory beginning in location 02000. Any remaining words in the record after the first 100 will be ignored.

The main program is processed while the input operation is being performed by the channel. When finished, the End-of-Record interrupt will be transmitted to location 011.

011	BRM	COMPL	This instruction in interrupt location 011 branches and marks to COMPL to finish the read operation.
COMPL	PZE		This instruction saves a location for the routine entry.
	CET	0	This instruction tests for error in channel A. If an error is detected, the next instruction in sequence is executed; if not, the next one is skipped and the following instruction is executed. The octal configuration is 0 20 22000.
	BRM	ERTST	This instruction branches to an assumed routine to reread the block a few times and, if the error continues, informs the operator.
	BRC	*COMPL	This instruction returns control to the main program and clears interrupt level 011.

MAGNETIC TAPE UNIT CONTROLS

The following instructions are used for control of magnetic tape units. These instructions are EOM's in the input/output control mode (1).

REW 0, n REWIND
EOM 01401n 0 02 1401n

Tape unit n is started in a rewind. Once started, the tape continues in rewind until the beginning of tape is

sensed; it then stops, and after 1 second (to allow the drive capstans to return to normal speed) generates a ready signal.

RTS 0 CONVERT READ TO SCAN
EOM 014000 0 02 14000

The tape unit currently in a read mode on the channel is instructed to convert from the read mode of operation to the scan mode of operation.

SRR 0 SKIP REMAINDER OF RECORD[†]
EOM 013610 0 02 13610

The tape unit currently on the channel is instructed to skip the remainder of the record being read.

WRITING MAGNETIC TAPE

Once a tape unit is ready and the file protect ring is on the tape reel, that is, the file protect test is false, a write operation can be initiated. The tape will start and remain in motion until the termination signal from the buffer is received. The tape control unit will then write the remaining characters of the record and the longitudinal check character. When the check character is read by the read-after-write head, the tape will signal the channel that gap has been reached. If no further write instruction is received within one millisecond, the tape is stopped and disconnected.

An end-of-file character should be written (or a segment of tape erased) after a series of records have been written, if the user wishes to backspace or rewind and then expects to return at some later time to record additional information at the end of the previous series of records. This practice provides positive identification of the end of a record and facilitates return to a specific location on the tape. If this method is not used, there is a possibility that the tape will not subsequently stop in the same location at the end of the series of records as it did when the last record was written. This would leave a segment of tape in the gap which has not been written and may cause erroneous operation when the tape is read.

In addition to writing under program control, magnetic tape can also be erased under program control. Tape may be erased by addressing it with an erase unit address. When a tape is addressed with an erase unit address, it operates as though it were in a write mode, except that no information is recorded. The program or interlace supplies the count of the number of words to be erased.

This type of erase is useful for the correction of a write error. When a write error occurs, an ERASE TAPE REVERSE (ERT) is given to start the tape in reverse. Then the same count, used to write the record originally, is loaded to control the erase. This procedure ensures that the tape always returns to the beginning of the erroneous record, even if a bad spot on the tape might appear as a gap. The record may now be rewritten. If the write still produces an error, the record is erased backwards and then an erase forward, using the same count, bypasses the section of tape where the difficulty occurred. The record may now be rewritten on a new section of tape.

The erase procedure is used to produce 3.75 inches of blank tape between the load point and the first record. This is accomplished by erasing 150 words at 200 bpi density, 417 words at 556 bpi density, or 600 words at 800 bpi density.

Writing an end-of-file record is accomplished using a one-character-per-word, BCD, write instruction. Then the channel interlace is loaded with a count of 1 and the address of a word containing the tape mark character (17) in the left-most position. EOM or EOD instructions to the tape units specify start-without-leader since the tape unit generates gap on all write operations automatically. Thus, it is not necessary for the starting EOM to call for leader. A leader instruction should never be included in a magnetic tape program, because an attempt to generate leader may cause an erroneous operation.

WTB 0, n, 4 WRITE TAPE IN BINARY
EOM 0365n 0 02 0365n

Tape unit n is started in a Binary write mode.

WTD 0, n, 4 WRITE TAPE IN DECIMAL (BCD)
EOM 0265n 0 02 0265n

Tape unit n is started in a BCD write mode.

EFT 0, n, 4 ERASE TAPE FORWARD
EOM 0367n 0 02 0367n

Tape unit n is started in an erase mode.

ERT 0, n, 4 ERASE TAPE IN REVERSE
EOM 0767n 0 02 0767n

Tape unit n is started in reverse in an erase mode.

[†]Note: This instruction applies only to 41.7-kc and 96-kc magnetic tape systems.

Example: Gather-Write Magnetic Tape

This program writes one record on magnetic tape. The data written in the record are gathered from three non-contiguous areas of memory. The program is written as a closed subroutine that uses the Zero Word Count interrupt. Channel A and Magnetic Tape 1 on channel A with interlace are used.

This program is written to clarify programming for magnetic tapes. Extra programming is not included to save the contents of the A or index registers for the main program.

A scatter-read operation can be performed with an almost identical program. The difference is the exchange of the read instruction (RTD) with the write instruction (WTD) and the deletion of the file-protect testing instruction.

Location	Instruction	Address	Comments
1000	PZE		This assembler instruction saves a location for the subroutine entry.
	STZ	COUNT	This instruction clears location COUNT for use later as a switch.
	TRT	0, 1	This instruction tests Magnetic Tape 1 on channel A for being ready.
	BRU	$ + 2	This instruction branches two locations ahead. This instruction is executed if the magnetic tape is ready.
	BRU	$ - 2	This instruction branches back to the Tape Ready Test.
	FPT	0, 1	This instruction tests whether the file protect ring is present on the tape reel. If so, the next instruction is skipped and the following one is executed. The octal configuration of the instruction is 0 20 14011.
	BRM	OPER	This instruction branches and marks to an assumed routine to call the operator and instruct him to insert file protect ring on Magnetic Tape 1.
	LDA STA MPO	01000 FAST FAST	These three instructions place the marked subroutine entry location plus one into location FAST.
	WTD	*0, 1, 4	This instruction connects Magnetic Tape 1 to channel A, specifies BCD transfer mode, and starts the tape moving. Four characters per word mode is specified. The octal configuration of the instruction is 0 02 42651.
	BRU	FAST + 1	This instruction branches around location FAST.
FAST	PZE		This instruction saves a location for entry to the multiple write area of the subroutine.
	LDX	COUNT, 1	This instruction loads index X1 with the contents of COUNT. This is used in picking up the proper input/output control instructions.
	LDA SKU BRC	OKAY COUNT *FAST	These three instructions determine when the write operation is complete. When it is, location COUNT will contain the number 6 and the active interrupt, level 010, is cleared.
	ALC	0	This instruction alerts the interlace in channel A for subsequent loading.
	EXU	A, 1	This instruction executes the EOM located in address A modified by index X1.
	POT	A + 1, 1	This instruction transmits to the channel the word count and starting address.
	MPT	COUNT	This instruction adds 2 to the contents of COUNT.
	BRC	*FAST	This instruction branches back to the main program.

The main program is processed while the output is being performed by the channel. When output is completed, the channel transmits the Zero Word Count signal to interrupt location 010.

Location	Instruction	Address	Comments
010	BRM	FAST	This instruction branches and marks at location FAST.

This is repeated for the output words in A+2 and in A+4. Then the test in location FAST+3 causes a final BRANCH AND CLEAR interrupt back to the main program.

A	EOM 06202000	015600	This EOM specifies output function 11 (IOSP) and the Zero Word Count interrupt. The word in A+1 specifies that 100 words will be output from memory beginning in location 02000.
A+2	EOM 14402500	015600	This EOM specifies output function 11 (IOSP) and the Zero Word Count interrupt. The word in A+3 specifies 200 words from memory beginning in location 02500.
A+4	EOM 06203000	015000	This EOM specifies output function 00 (IORD) and the Zero Word Count interrupt. The word in A+5 specifies 100 words from memory beginning in location 03000. Upon completion of the output of this sub-record, the channel disconnects.
OKAY	00000006		This is the stored number 6 used in the completion tests.

APPENDIX A

SDS CHARACTER CODES

Characters		SDS Internal Code	Card Code	Magnetic Tape BCD Code on Tape
Typewriter	Printer			
ø	0	00	0	12
1	1	01	1	01
2	2	02	2	02
3	3	03	3	03
4	4	04	4	04
5	5	05	5	05
6	6	06	6	06
7	7	07	7	07
8	8	10	8	10
9	9	11	9	11
Space	Blank	12	8-2	12 (3)
# or =	=	13	8-3	13
@ or '	'	14	8-4	14
:	:	15	8-5	15
>	>	16	8-6	16
✓	✓	17	8-7	17
& or +	+	20	12	60
A	A	21	12-1	61
B	B	22	12-2	62
C	C	23	12-3	63
D	D	24	12-4	64
E	E	25	12-5	65
F	F	26	12-6	66
G	G	27	12-7	67
H	H	30	12-8	70
I	I	31	12-9	71
Backspace ? (1)	? (5)	32	12-0 (4)	72
	.	33	12-8-3	73
¤ or)	)	34	12-8-4	74
[	[	35	12-8-5	75
<	<	36	12-8-6	76
‡ Stop	‡ (5)	37 (2)	12-8-7	77

Characters		SDS Internal Code	Card Code	Magnetic Tape BCD Code on Tape
Typewriter	Printer			
-	-	40	11	40
J	J	41	11-1	41
K	K	42	11-2	42
L	L	43	11-3	43
M	M	44	11-4	44
N	N	45	11-5	45
O	O	46	11-6	46
P	P	47	11-7	47
Q	Q	50	11-8	50
R	R	51	11-9	51
Car. Ret. ! (1)	! (5)	52	11-0 (4)	52
$	$	53	11-8-3	53
*	*	54	11-8-4	54
]	]	55	11-8-5	55
;	;	56	11-8-6	56
Δ	Δ	57	11-8-7	57
ß	Blank	60	Blank	20
/	/	61	0-1	21
S	S	62	0-2	22
T	T	63	0-3	23
U	U	64	0-4	24
V	V	65	0-5	25
W	W	66	0-6	26
X	X	67	0-7	27
Y	Y	70	0-8	30
Z	Z	71	0-9	31
Tab ‡ (1)	‡ (5)	72	0-8-2	32
,	,	73	0-8-3	33
% or (	(	74	0-8-4	34
~	~ (5)	75	0-8-5	35
\	\	76	0-8-6	36
* Delete	* (5)	77 (2)	0-8-7	37

NOTES:

(1) The characters ? ! and ‡ are for input only. The functions Backspace, Carriage Return, or Tab always occur on output.

(2) On the off-line paper tape preparation unit, 37 serves as a stop code and 77 as a code delete.

(3) The internal code 12 is written on tape as a 12 in BCD. When read, this code is always converted to 00.

(4) The codes 12-0 and 11-0 are generated by the card punch; however, the card reader will also accept 12-8-2 for 32 and 11-8-2 for 52 to maintain compatibility with earlier systems.

(5) For the 64-character printers only.

TABLE OF POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625

OCTAL - DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000	1024
to	to
2777	1535
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000	1536
to	to
3777	2047
(Octal)	(Decimal)

Octal-Decimal Integer Conversion Table

4000 (Octal)	2048 (Decimal)
to	to
4777 (Octal)	2559 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 (Octal)	2560 (Decimal)
to	to
5777 (Octal)	3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

Octal-Decimal Integer Conversion Table

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000	3072
to	to
6777	3583
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	1022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000	3584
to	to
7777	4095
(Octal)	(Decimal)

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

Octal-Decimal Fraction Conversion Table

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

TWO'S COMPLEMENT ARITHMETIC

SDS computer systems hold negative numbers in memory in binary two's complement form. The two's complement of a binary number is formed by adding one to the one's complement (logical inverse) of the number. This convention allows the sign of a number to be used as an integral part of the number in all arithmetic operations and obviates the need for keeping track of a detached sign with computer logic.

In SDS systems, the sign bit is in the first bit position to the left of the most significant magnitude bit. Thus, if an SDS computer word was only 6 bits long instead of 24, some common decimal values would be represented in binary format as follows:

Decimal Number	Octal Equivalent	Complement Plus 1	Binary Equivalent
3	03	-	000 011
2	02	-	000 010
1	01	-	000 001
0	00	-	000 000
-1	(-)01	77	111 111
-2	(-)02	76	111 110
-3	(-)03	75	111 101
31	37	-	011 111
-31	(-)37	41	100 001

This table suggests the following algorithms:

1. To find the binary, two's complement of a negative decimal number:

   a. Find the octal equivalent of the absolute of the number

   b. Form the complement and add one

   c. Express as a binary number.

   The result is the binary, two's complement equivalent.

2. To find the decimal equivalent of a binary two's complement number:

   a. Express as an octal number

   b. Subtract one and form the complement

   c. Find the decimal equivalent.

   The negative of the result is the decimal equivalent.

The following examples show how two's complement numbers automatically yield the correct result when used arithmetically in the computer.

Decimal Number	Binary Equivalent
+ 20	010 100
- 03	+ 111 101
+17	1010 011 = $21_8 = 17_{10}$
	↳lost carry

Note that the carry out of the most significant (sign bit) position is lost. Nevertheless, the value remaining is the correct answer.

Decimal Number	Binary Equivalent
- 32	100 000
+ 24	011 000
-8	111 000 = $(-)10_8 = -8_{10}$

When performing additions or subtractions in the computer, carries out of the sign bit do not always signify a true overflow condition or cause the OVERFLOW indicator to be set. In an addition, it is impossible to produce an overflow if the signs of the operands are unlike. The computer sets the OVERFLOW indicator in an addition only when the signs of the two operands are the same, but the sign of the result is opposite. In a subtraction, which in the computer is accomplished by forming the two's complement of the subtrahend and then adding to the minuend, the test for overflow is similar to that for addition. That is, overflow occurs when both numbers have the same sign after the subtrahend has been complemented but the sign of the result is opposite.

APPENDIX B

OPTIONAL EQUIPMENT

DATA MULTIPLEXING SYSTEM

The standard I/O systems provided with the SDS 9300 computer provide for operation with all standard SDS peripheral equipments and for high-performance special devices. The Data Multiplexing System (DMS) provides an alternate I/O system that is of particular use in dealing with multiple sources of data and for systems which may have very high data rates (see Figure B-1).

The SDS 9300 computer has essentially two major paths along which I/O data can flow to and from memory. The first path is the same that is used by the main frame itself. The PIN/POT operations use the first path. In addition to this path, which is primarily under the control of the main frame, there is a second path that is completely under the control of the units attached to it. The second path has priority over the first for access to memory. This path is made available with the multiple access to memory feature.

MULTIPLE ACCESS TO MEMORY FEATURE

The multiple access to memory (MAM) feature provides the necessary modules on both main frame and memory to permit memories to be accessed via the second path. A word can be transferred over the path in either direction in one cycle. If the computer is equipped with two or more memories and the main frame is communicating with one memory while some other device is using the second path to another memory, then there is no interference with computation. If both the main frame and an I/O device using the second path simultaneously address the same memory, the second path has priority; the program loses one cycle while the second path transmits.

The multiple access feature is required for the attachment of four direct access communication channels (DACC), data multiplex channels (DMC), or memory interface connections (MIC). These devices all incorporate a priority scheme for determining the assignment of the second path.

A practically unlimited number of MICs in addition to the four DACCs and the Data Multiplexing System can be attached to a computer system. Each MIC has the necessary priority control to operate with other MICs and the DACCs and the DMS. Both MICs and DACCs can be arranged so as to produce any required configuration of priorities.

DATA MULTIPLEXING BASIC ELEMENTS

A Data Multiplexing System consists of two basic elements:

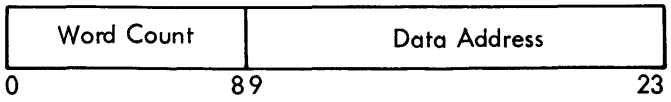
1. The data multiplex channel for communicating with several data sources/destinations and for synchronizing I/O operations with memory, MICs, DACCs, and other DMCs.
2. One or more data subchannels (DSC) for interfacing between peripheral and systems devices and the DMC.

DATA MULTIPLEX CHANNEL

The data multiplex channel (DMC) is the basic unit for the data multiplexing system. It connects to the second path to memory via the multiple access to memory (MAM) feature. A DMC consists of a 24-bit register and control logic to enable the DMS to perform a variety of functions. All addresses and data are transmitted between the DMC and subchannels via a bus system. The data and address are connected to memory via the MAM only when a transfer is to be made. All program control required for a given I/O operation operates directly on the individual subchannel, not the DMC.

The DMC is equipped with an internal interlace feature. This feature allows a subchannel to specify the address of a word in memory where the data address and count are to be found. When operating with internal interlace, the subchannel supplies the address of its interlace word instead of the actual data address. The DMC reads out the interlace word, increments the address portion, decrements the count, restores the word and then accepts the data from or transmits the data to the subchannel. The DMC also supplies a signal to the subchannel if the decremented count is zero.

The format of the internal interlace word or word pair is as shown:



The 9-bit word count allows for block lengths to 512 words. With the 9300, transmissions using internal interlace require 3 cycles per word.

The DMC also provides for automatic memory incrementing. The counting capability of the DMC register permits the entire 24-bit register or either the upper or lower 12 bits to be incremented. When such a memory increment operation is to be performed, the subchannel

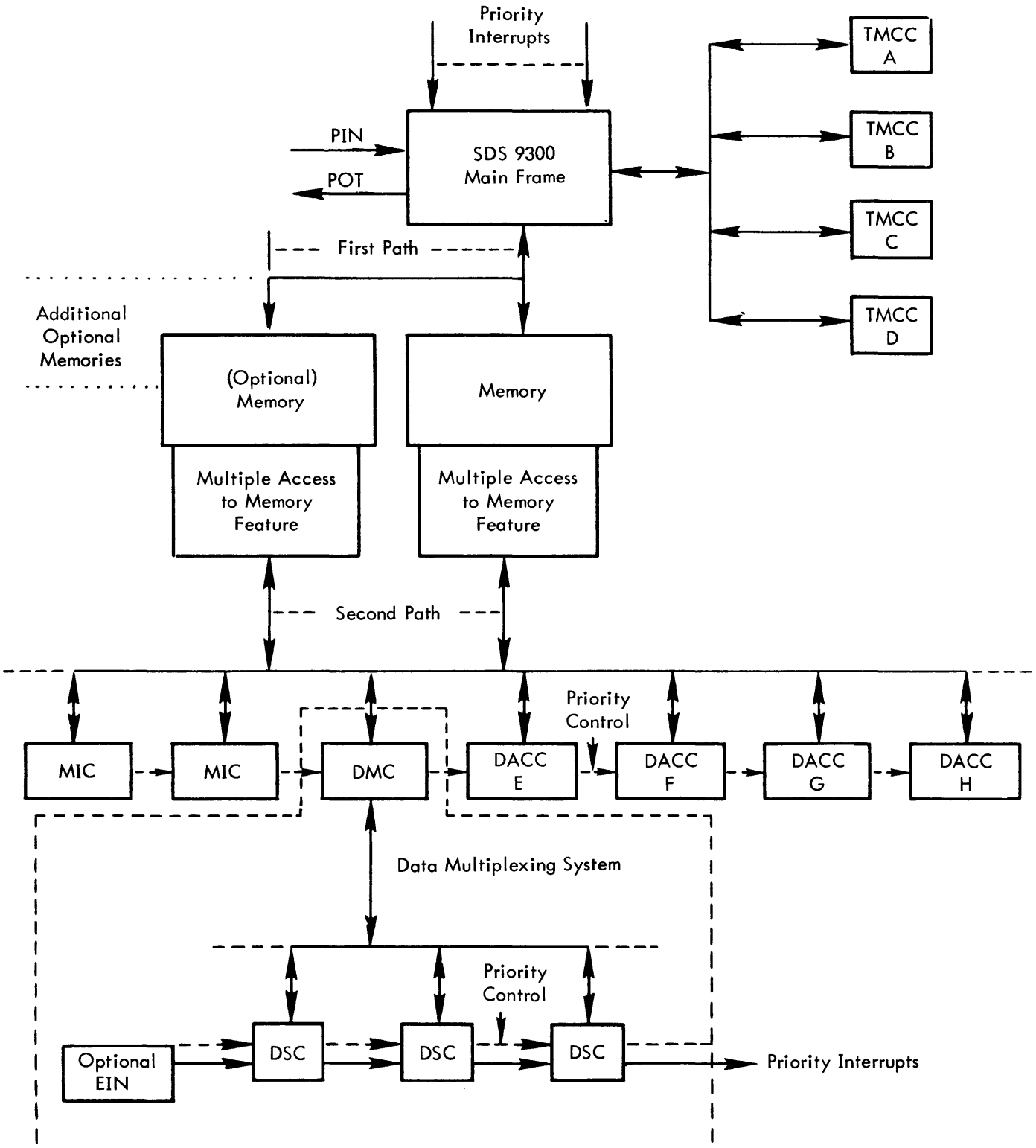


Figure B-1. SDS 9300 Overall Computer Configuration

signals the DMC with a special increment line and supplies the address. The DMC reads out the word, increments it and then restores it. If the word was zero after the incrementing, the DMC signals the subchannel which may then interrupt the program. The maximum incrementing rate is 1 count every 2 cycles. Parity is always checked or generated by the DMC.

DATA SUBCHANNELS (DSC-N)

There are a number of subchannels which can be attached to the DMC. The two described below are standard subchannels. They are distinguished primarily by the size of the data unit. Full word is 24 bits plus parity for 9300. However, 12-bit information (half words) is transmitted to the 9300 by the DSC-I. Subchannels can arm and generate program interrupts but do not include the interrupt levels themselves. The signals must be routed to optional interrupt levels if the interrupt features are to be used.

The subchannels use a priority scheme to determine which may transmit to the DMC at any given time. This is similar to the scheme used by the MICs, DMCs and in transmitting to memory. Up to 128 DSCs can be connected to a DMC. A DSC can use the internal interlace feature of the DMC to control its transmission or it can be equipped with an external interlace (EIN).

A DSC using internal interlace has two words assigned to it. These two words are adjacent even/odd locations and are fixed for a given subchannel. The program can select either the even or odd location. If the even location is selected, the subchannel will automatically switch to the odd location when the count field of the even word is zero. The program can also select whether or not the subchannel will switch back to the even word when the count field of the odd word is zero. The subchannel will generate an interrupt signal when the count field of either word reaches zero. Transmission termination occurs when the odd word's count equals zero if the subchannel does not switch back to the even word.

The two-word internal interlace allows a subchannel to handle continuous data by alternately working from one memory area or another. By allowing the subchannel to switch automatically from one interlace word to the other, the program is relieved of the necessity for making real-time responses to the zero count condition. Using first the even then the odd interlace word allows maximum word count of 1024 for a pair of interlace words.

CHARACTER SUBCHANNEL (DSC-I)

The DSC-I contains a 12-bit data register that can assemble and disassemble two 6-bit characters, and transmit one or two 6-bit characters or one 12-bit character. It checks and generates the parity of characters to enable it to couple with standard SDS peripherals. The DSC-I has a unit address register. For the 9300, it can be used for multiple typewriters or other character-oriented devices. However, it only uses 12 bits of the full 24-bit word.

The subchannel can operate with either internal or external interlace. It has one mode of output and two modes of input. During output, it transmits until the odd internal interlace word count is zero and then terminates if interlace cycling is not requested. The output can also be terminated if the device sends an END signal to the channel. This END signal may cause the DSC-I to generate an interrupt to the program.

Input, like output, can always be terminated due to an external END signal. The program can also specify if the DSC is to terminate and disconnect on zero count or disconnect only on the END signal. In either case, however, all transmission to memory is terminated after the odd interlace count reaches zero if interlace cycling is not requested.

FULL-WORD SUBCHANNEL (DSC-II)

The DSC-II is a general purpose subchannel designed to allow communication with word-oriented input/output units such as analog-digital and digital-analog converters. It contains no storage for data. The external device must be capable of holding the data during the transmission to/from the DMC. (An A-to-D converter would have such capability). Like the DSC-I, the DSC-II can operate with either internal or external interlace. Its operation in this respect is identical to that of the DSC-I. The DSC-II also contains control logic to facilitate memory increment operations in conjunction with the DMC.

EXTERNAL INTERLACE

The external interlace (EIN) can be attached to the DSC to control the transmission of its data to/from memory. The EIN consists of a 15-bit address register and a 9-bit count register. These registers are loaded automatically when the subchannel is activated, the information coming from the internal interlace memory locations. Once the EIN is set up, it will control the transmissions of the DSC at a maximum rate of 1 word per memory cycle. After each word is transmitted, the EIN increments its address register and decrements its count. When the count equals zero, the EIN signals the DSC, which can then generate a program interrupt and/or notify the external device. Transmission normally terminates on zero count. Sequencing of interlace words is identical to the sequence of operation performed for internal interlace, except that only two memory cycles are used for interlace word processing. The first is to access the interlace word initially; the second is to restore the interlace word when the count reaches zero.

PROGRAM CONTROL OF DATA SUBCHANNELS

Transmission of data between a DSC and computer memory is controlled by two 24-bit interlace control words unique to the DSC and wired into fixed adjacent locations in memory. During a transmission the DMC/DSC uses the two interlace control words for determination of transmission address and record length.

The DSCs are numbered from 0 to 0376 in even octal numbers; this permits a maximum of 128 subchannels. The memory locations of the interlace control word pairs associated with the DSCs are numbered X0000, X0001 for DSC-0, X0002, X0003 for DSC-2...X0376, X0377 for DSC-0376. DSC-I numbering need not be contiguous, but DSC-II's are configured one or two in a module and are numbered with adjacent numbers. If a system contains multiple DSC-II modules (each with 1 or 2 subchannels), the module numbering need not be contiguous; 4, 0 and 0224, 0220 and 0314 is a typical possibility for five DSC-II subchannels. Transmissions to and from the DSC and memory may be under internal interlace control or, when so equipped, under external interlace control.

INTERNAL INTERLACE

During an internal interlace transmission, the DMC controls the interlacing operation in the following order:

1. Access Interlace Word

   The DMC accesses the interlace word assigned to the requesting subchannel.

2. Process Interlace Word

   The DMC increments the 15-bit address portion of the word and decrements the 9-bit word count.

3. Test for Zero and Set Indicator

   Next, the DMC tests the word count for zero and if it is zero, set an indicator in the pertinent DSC.

4. Restore

   The DMC then places the new word count/address values back into memory using the assigned address of requesting subchannel.

5. Access/Store as Requested

   The DMC accesses or stores the transmitted word as requested using the incremental address (see above).

6. Stop or Continue

   The DSC checks its zero count indicator and

   a. if zero and working on the even interlace word, the DSC continues operation using the odd interlace word,

   b. if zero, working on the odd interlace word and the cycle bit is set, the DSC continues using the even interlace word,

   c. if zero, working on the odd interlace word and the cycle bit is reset, the DSC terminates the operation on a DSC-II or responds as required by the function control on a DSC-I.

   d. if not zero, the DSC returns operation to the DMC to continue at 1 (above).

Note that the first address used is the "address specified plus one" and the first word count is the "word count specified minus one". In particular, an initial word count of zero causes a 512-word block to be transmitted.

EXTERNAL INTERLACE

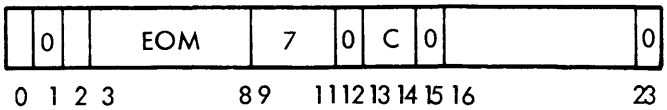
During transmissions utilizing external interlace control, the interlacing operation proceeds as described above except that when the DSC is activated, the DSC with external interlace (EIN) requests the DMC to access the desired interlace control word. The interlace control word is sent to the EIN. Thereafter, data transmissions to and from the DSC to memory utilize the interlace address and word count supplied by the EIN.

Data transmissions using the EIN require only one cycle while those data transmissions using internal interlace require three cycles. Should a transmission result in the EIN detecting a zero word count condition, the DSC-EIN will restore the external interlace word and will proceed according to 6 (above). Any termination of a DSC operation prior to zero word count due to any externally derived halt signal also causes a restoring of the EIN interlace control word.

DSC PROGRAMMING

An EOM, POT sequence selects, alerts, and controls the subchannel; an EOM, SKS sequence selects and tests the status and conditions of the subchannel.

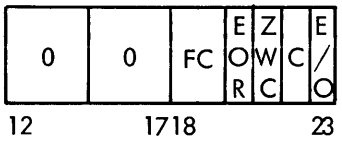
The EOM has the form:



and is referred to as the "select EOM".

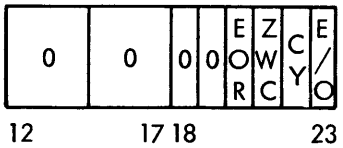
Bit positions 16-23 contain the DSC number being alerted; these numbers are the even numbers from 0 to 0376 for the DSC and the C field (bits 13, 14) specifies one of three modes to which the DSC is alerted. When followed by a POT instruction, the modes have the following meaning:

C	Effect
00	The subchannel decodes the lower 12 bits (12-23) of the "POTted" word as the lower 12 bits of a buffer control mode EOM. For DSC-I, this will select a device with the unit address field, set the character/word count, specify binary or BCD format, forward or reverse, and leader or no leader. For DSC-II, the 12 bits activate the subchannel and select the proper unit (if more than one is attached to the DSC).
01	The subchannel decodes the lower 12 bits of the "POTted" word as the lower 12 bits of an input/output control mode EOM. If bits 18 through 23 are zero, the "POTted" word addresses the selected DSC. For DSC-I, these bits perform such functions as rewind tape, space paper, etc. For DSC-II, these bits perform such functions as required by the selected device attached to the DSC.
10	The subchannel decodes the lower 12 bits of the "POTted" word for controlling the interlace and interrupts. The control type EOM should precede the buffer control EOM. For DSC-I the form is:



FC is a 2-bit function code similar to the TMCC/DACC terminal function codes. The remaining bits function as described below for DSC-II.

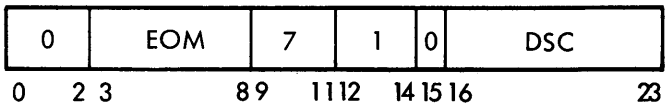
For DSC-II, the form is:



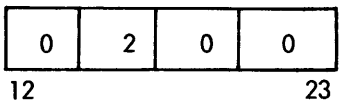
Bit Position	Function
20	A 1 in the EOR bit arms the End-of-Record interrupt for this channel.
21	A 1 in the ZWC bit arms the Zero Word Count interrupt.
22	A 1 in the CY bit (cycle) sets the cycle mode such that the interlace will switch from the odd word back to the even word at the zeroing of the odd word count. If ZWC and CY are set, a zero count interrupt is generated each time the interlace switches (to either word - even or odd). If CY is set to 0, the interlace will not proceed after the count of the odd word is zero; and a zero count interrupt occurs only when the count of the odd word is zero.
23	A 0 in the E/O bit selects the even interlace word as the first interlace word in a transmission; note that when starting on the even word, the interlace aways switches to the odd word for further control when the even word count goes to zero. A 1 in E/O sets the odd interlace word as the first interlace word in a transmission; the interlace ceases control when the odd word count reaches zero unless the C bit is set to cycle.

TERMINATING DSC INPUT/OUTPUT

Once the cycle bit has been set, the interlace continues to cycle back and forth between the even/odd interlace words. An EOM, POT sequence is used to terminate the cycle. The EOM is:



The lower 12 bits of the "POTted" word must be:

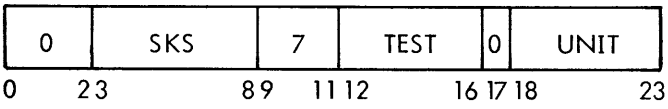


The interlace terminates the next time the count reaches zero in the odd interlace word.

For example, to terminate the cycle on DSC 4, use the following sequence:

```
        EOM 071004
        POT 010000
         .
         .
         .
010000  00000200
```

The SKS to test subchannels has the form:



A select EOM with C equal to zero (C=0) permits the SKS to be directed to the subchannel or to the device attached to it. The UNIT field specifies the device to be tested; the TEST field is defined for the particular device.

When testing the subchannel, the UNIT field is set to 00. The TEST field contains the same testing format as SKS for testing a TMCC. For example, to test DSC 4 for error, use the following sequence:

```
EOM 070004
SKS 071000
```

MEMORY INTERFACE CONNECTION

Once a computer is equipped with a multiple access to memory feature, one or more memory interface connections (MIC) can be attached. The MIC is a general interface between the computer and the outside world that allows special devices to be connected to the computer. The MIC converts between the 4-volt logic levels used in the computer and the 8 volts used outside. It preserves the integrity of the memory by generating the parity of incoming data words. It will also check the parity of words read from memory to indicate memory failures. If incoming data is supplied with parity, the MIC will check for odd parity as it generates the internal memory parity and respond with a signal that indicates if the transmission was correct. The device that is connected to the MIC must store both the data and the address until the transmission to/from memory is completed.

The memory interface connection is useful in many systems applications where the fully extended capabilities of the Data Multiplexing System are not required.

AUTOMATIC POWER FAIL-SAFE SYSTEM

The computer core memory holds its information with all power removed, but information in the computer registers is destroyed by loss of power. Upon failure of main power to the computer, this system provides that the contents of all registers and other volatile information are automatically stored in core memory; also, further writing into core storage is inhibited during the decay period of the computer dc power supply outputs. Erroneous memory control is prevented during power-off and power-on operations. Power-off/-on interrupt routines permit proper resumption of a program, automatically, after power is restored.

This solid-state system consists of ac power-sensing and memory-sequencing circuitry, two high-priority interrupt channels, and a "shut-down/start-up" programming sequence.

The SKIP IF SIGNAL NOT SET (SKS) instruction is an aid in programming this option. Its address is 024000. If the OFF interrupt (01) has just occurred, the computer executes the next instruction in sequence (does not skip).

MEMORY PARITY INTERRUPTS

SDS computers incorporate an extensive memory parity checking system. The inclusion of parity generation and checking circuitry assures the integrity of all data and instructions transferred among the memory, the central processing unit, and input/output channels.

In normal operation a switch on the computer console specifies the action to be performed by the computer when a memory parity error is detected. Two actions are available: the computer halts with the parity indicator lighted; or the computer ignores the parity error and proceeds with the program.

In many real-time applications it is desirable to keep the computer running when a parity error is detected. Also, the program must be notified of the error without stopping computation.

An optional feature provides this capability by means of two levels of armed interrupts. One interrupt level is associated with the central processor and the Time-Multiplexed Communication Channels; the other interrupt level with the Direct Access Communication Channels and the Data Multiplexing System. Memory parity errors detected from these two sources produce a priority interrupt associated with the cause. The processing routine associated with the interrupt can then take appropriate action, such as re-initiate the failed operation, notify the operator, or enter a diagnostic routine. Such action allows memory parity errors to be recognized and handled properly without hindering the computer's performance of real-time or on-line calculations.

REAL-TIME CLOCK

The Real-Time Clock (RTC) provides a flexible time-orientation system for the SDS 9300 Computer. It derives time pulses from the 60-cycle computer power supply. These pulses are then used to produce a timing mark every 16.67 milliseconds, or optionally every 8.33 milliseconds. The Real-Time Clock can also accept timing marks from a customer-supplied input, thereby allowing time measurement to any required resolution for special applications. These timing marks are supplied at standard SDS logic levels to the computer's RTC circuitry.

The timing marks are then used by the computer and its interrupt system to provide either an elapsed-time counter or a continuously incrementing time counter depending on the needs of the customer. The RTC operates in either mode depending only on the computer's stored program.

Location	Type	Computer	Description
06	Normal	9300	CLOCK SYNC
07	Single Instruction	9300	CLOCK PULSE

The Clock Pulse and Clock Sync interrupts function together to provide elapsed-time, event counter, or time-of-day clocks.

The Clock Pulse interrupt is a single-instruction interrupt. (Note: See Single Instruction Interrupts in Section 3.) An MPO instruction is usually placed in the Clock Pulse interrupt location. When MPO is used as a single-instruction interrupt subroutine, it causes the contents of the effective address to be incremented by one. Furthermore, if the new (incremented) contents of the effective address is 0000, a Clock Sync interrupt is generated. The Clock Sync interrupt can be generated in no other way.

ELAPSED-TIME CLOCK

The elapsed-time clock times the length of a program or subroutine, or initiates or discontinues processing at program-determined time intervals. An arbitrary memory location is reserved as a counter. When initialized, this cell contains the 2's complement of the number of time intervals to be counted. The Clock Pulse interrupt location contains an SKR instruction.

Each Clock Pulse interrupt results in decrementing the clock count by one. When the count is finished, an interrupt to the Clock Sync location occurs. A supervisory or other appropriate control program can then be entered to perform the customer-desired operation.

CONTINUOUSLY INCREMENTING CLOCK

The continuously incrementing clock maintains "time-of-day" for the computer. One memory location serves to count the timing marks. In this case, the Clock Pulse is used to increment this location. (The Clock Pulse interrupt location contains an MPO instruction.) A simple, straightforward subroutine can be entered to reconstruct the exact time-of-day from this twenty-four bit count.

ARM/DISARM

The Clock Pulse interrupt can be armed and disarmed with these instructions.

EOM Effective Address	Action
20200	Disarm Clock Pulse Interrupt
20100	Arm Clock Pulse Interrupt

The Clock Sync interrupt is always armed.

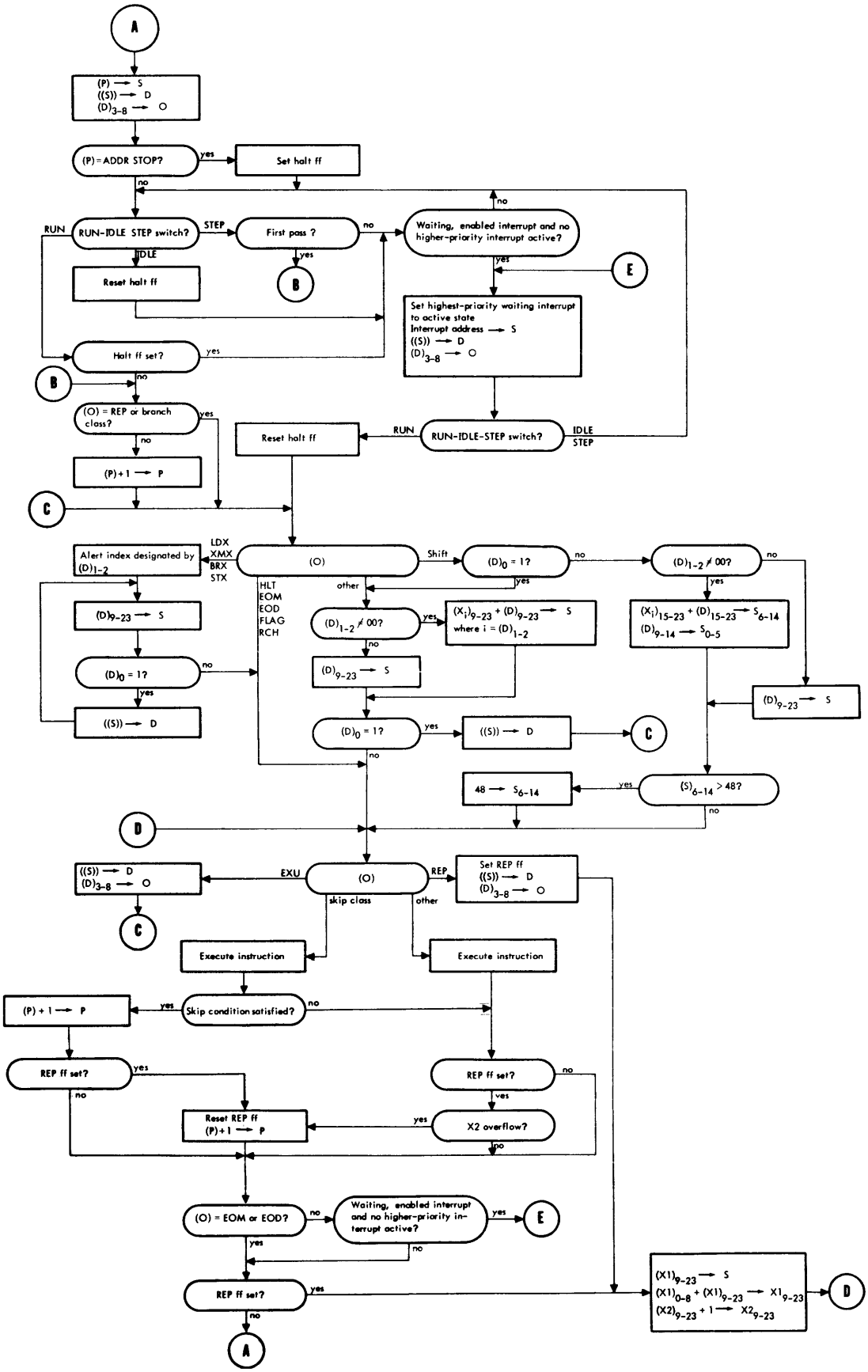


Figure B-2. 9300 Computer Instruction Execution Cycle

INSTRUCTION EXECUTION CYCLE

A symbolic diagram of the SDS 9300 instruction execution cycle is shown in Figure B-2. The diagram illustrates the major operations involved during execution of instructions by the 9300 Computer, including the effects of the RUN-IDLE-STEP switch, normal interrupt processing, effective address calculation, and the REPEAT instruction. The diagram does not in all cases precisely depict actual computer operations; however, insofar as the programmer is concerned, the diagram is a valid representation of the instruction execution process.

The symbolic notation used in the diagram is consistent with that used in other portions of this reference manual. The symbolic terms are defined as follows:

Term	Definition
D	The register that holds an instruction while it is being decoded
$(D)_0$	Indirect address bit
$(D)_{1-2}$	Index register designator
$(D)_{3-8}$	Operation code field
$(D)_{9-23}$	Address field
Halt ff	An internal flip-flop that causes the computer to cease executing instructions as long as the flip-flop is set
O	Operation code register
(O)	Contents of the operation code register
P	Program counter
(P)	Contents of the program counter
REP ff	An imaginary indicator for the purpose of diagramming the action of the REPEAT instruction
S	Memory address register
(S)	Contents of the memory address register
((S))	Contents of the location whose address is in the memory address register
(X_i)	Contents of the index register designated by $(D)_{1-2}$
X1	Index register 1
(X1)	Contents of index register 1
X2	Index register 2
(X2)	Contents of index register 2

At the top of the diagram, reference point "A", assume that the program counter (P) contains the address of the next instruction to be executed, the D register contains the next instruction, the O register contains the operation code of the next instruction, and the computer is in the IDLE mode.

If the RUN switch is pressed, the computer proceeds to decode the instruction in the D register. Since this is the first instruction to be executed in the RUN mode, no interrupt acknowledgement can occur at this time. The Halt flip-flop (ff) was reset in the IDLE mode.

If the instruction references memory (i.e., is not a HALT (HLT), ENERGIZE OUTPUT M (EOM), ENERGIZE OUTPUT D (EOD), FLAG, or REGISTER CHANGE (RCH) instruction), the computer performs indexing and/or indirect addressing, as specified by the indirect address bit, the index register designation, and the operation code of the instruction.

Note that for the instructions LOAD INDEX (LDX), EXCHANGE MEMORY AND INDEX (XMX), INCREASE INDEX AND BRANCH (BRX), and STORE INDEX (STX), the address field is not modified by an index register. Also, note that indirectly addressed shift instructions are indexed as normal instructions until the last level of indirect addressing is completed, at which time indexing applies only to the 9 low-order bits of the address field.

If the instruction is a REPEAT (REP) instruction or is the instruction operand of the REP instruction, the diagram illustrates the action taken by the computer with each iteration of the subject instruction, including the termination of the REP instruction itself.

After the instruction is executed, the computer determines whether an interrupt condition is present. If all of the conditions are satisfied for acknowledging an interrupt condition, the computer sets the highest priority waiting interrupt to the active state and then accesses the instruction in the location assigned to the acknowledged interrupt level.

If no interrupt condition is present, the computer accesses the next instruction from the location whose address is the new contents of the P register, loads the instruction into the D register, and loads the operation code of the new instruction into the O register.

DIVISION INSTRUCTION

During execution of the DIVIDE (DIV) instruction, the contents of the A,B registers taken as a double-precision number are divided by the single-precision contents of the effective address. If the numerator is a single-precision number, the program should clear the B Register prior to executing the DIV, or erroneous results may occur. Although a double-length numerator is used, DIV is a single-precision operation; it should not be confused with a double-precision divide operation that uses a double-length denominator and produces a double-length quotient.

After the execution of DIV, the single-precision quotient replaces the contents of the A Register, and the remaining portion of the numerator that has not been divided (undivided remainder) replaces the contents of the B register. The quotient is signed in accordance with algebraic convention, that is, positive if numerator and denominator signs are alike, but negative otherwise. However, DIV generates only 23 magnitude bits and, if the magnitude of the quotient is so small as to require more than 23 bits to resolve, DIV may produce a zero quotient regardless of the required sign; but the remainder reflects the undivided portion of the original numerator. The binary scaling of the quotient is equal to the numerator scale factor minus the denominator scale factor.

The undivided remainder replaces the contents of the B Register and has the same sign as the original numerator. It is scaled, in B, at numerator scaling minus 23. By definition, the undivided remainder is that quantity which must be added to the product of the denominator and the quotient to produce the original numerator. The results of the DIV instruction are consistent with definition. It should be noted that the program must shift the remainder right 23 places before adding it to the product of denominator and quotient to maintain proper scaling.

Overflow is possible and the computer sets the Overflow indicator if:

a. $(A, B)/(M) \geq 1$, or

b. $(A, B)/(M) < -1$

Example:

Let

$(A, B) = 58.75B30$

$(M) = 10B8$

Binary scaling of the quotient = 30 - 8 = 22

Binary scaling of the remainder = 30 - 23 = 7

Complete quotient should be 5.875

With the binary point at 22 and only 23 bits generated in the quotient, the contents of A will be 5.5. In other words, only 55 of the original 58.75 units in the numerator are divided, leaving 3.75 units undivided. The undivided remainder is thus 3.75B7 in B.

Example:

Let

$(A, B) = 44.625B28$

$(M) = 7B7$

Quotient scaling = 28 - 7 = 21

Remainder scaling = 28 - 23 = 5

Complete quotient should be 44.625/7 = 6.375

With the binary scale point at 21, only 6.25 will be generated in A. Thus, only 7 x 6.25 = 43.75 units are divided leaving 44.625 - 43.75 = .875 undivided. Therefore, the undivided remainder is .875B5 in B.

META-SYMBOL COPY INSTRUCTION

The operand field of COPY consists of a byte selection "mask" followed by one or more grouped expression lists that describe the desired operation(s). The programmer need be concerned only with operational legitimacy and not with its specification via bit patterns.

Examples:

Label	Operation	Operand	Effect
	COPY	(0, (A, B))	Clear A and B
	COPY	(A, B)	Copy A into B
	COPY	(A, B), (B, A)	Exchange A and B
	COPY	077, (A, B, B)	Merge the low-order 6 bits of A and B in B.

Unless a merge is specified, the assembler automatically sets the "clear" bit. Thus, the second line causes the generation of 0 40 37703.

Format:

Label	Operation	Operand
LABEL	COPY	E, (E11, . . . , E1N), (E21, E2N), . . . , (EM1, . . . , EMN)

Since parenthetical notation is used in the operand field, parentheses have not been used to denote "optional." As usual, the label is optional and may or may not be external. The first operand and all successive operand lists are also optional.

Rules:

1. The byte selection mask, if present, is the first expression to appear in the operand field. It is not enclosed within parentheses. In the absence of this expression, the assembler assumes the mask 077777777 to be implicitly specified. Actually, the assembler cannot insert the mask directly into the byte-selection position of the instruction, since the 24-bit value must be mapped into 3 or 8 bits. However, it is convenient to think of the mask in this manner. Since the mask may be an expression, it need not always be written as an octal number. Unless the programmer indicates that the specified index register be cleared (in a mode 2 register change), the assembler automatically sets one of the bits 12, 13, or 14 to prevent the register from being cleared.

   Examples:

Label	Operation	Operand	Effect
EXP	EQU	0777	
HI3	EQU	070000000	
	COPY	EXP, (B, 1), (0, B)	$(B)_{15-23} \longrightarrow X1_{15-23}$, $0 \longrightarrow (B)_{15-23}$
	COPY	HI3, (A, B)	$(A)_{0-2} \longrightarrow B_{0-2}$

2. Following the mask, one or more parenthetical expression lists appear, separated by commas. Within a list, two or more expressions (or expression groups) appear. The first of these specifies the source of information flow, and the last specifies the destination. In the case of three or more successive expressions, logical OR is implied. Thus, COPY operations are specified by ordered groupings of values.

The following definitions relate the value of an expression to the 24-bit source value/register or destination register. Where actual registers are not involved (0 and -1), it is convenient to imagine the existence of two fictitious registers always containing all zeros and all ones, respectively.

Value		Meaning
-5	-(A)	The negative (two's complement) of (A)
-4	$\overline{(A)}$	The inverse (one's complement) of (A)
-3	$\overline{(B)}$	The inverse (one's complement) of (B)
-1	-1	
0	0	
1	(X1)	
2	(X2)	
3	(X3)	
4	(B)	
5	(A)	

To refer to the registers mnemonically, the programmer must precede his program by equality directives such as:

```
A       EQU     5
B       EQU     4
X2      EQU     2
IA      EQU     -4
IB      EQU     -3
ONES    EQU     -1
```

Thus, the programmer can specify any legitimate register change without having to write the necessary bit pattern explicitly, and without being restricted to a preselected set of mnemonic codes. Also the assembler diagnoses the variable field for legitimacy.

Examples:

Mnemonic Notation	Absolute	Interpretation
COPY (A, B), (B, A)	COPY (5, 4), (4, 5)	Exchange A and B
COPY (IA, B), (0, A) COPY (1-A, B), (0, A)	COPY (-4, 4), (0, 5)	Copy inverse of A into B and clear A
COPY 070, (ONES, B) COPY 070, (-1, B)	COPY 070, (-1, 4)	Form mask in B_{18-20}

SDS 9300 INSTRUCTION LIST — FUNCTIONAL CATEGORIES

Designation	Instruction Code	Name	Function	Timing
LOAD-STORE				
LDA	16	Load A	(M) → A	2
STA	76	Store A	(A) → M	3
LDB	14	Load B	(M) → B	2
STB	74	Store B	(B) → M	3
EAX	15	Copy Effective Address into Index Register 1	M → X1	2
LDX	X - 17	Load Index	(M) → X	2
STX	X - 77	Store Index	(X) → M	3
STZ	0 - 77	Store Zero	0 → M	3
LDP	26	Load Double Precision	(M, M+1) → A, B	3
STD	75	Store Double Precision	(A, B) → M, M+1	4
XMA	36	Exchange M and A	(M) ↔ (A)	3
XMB	34	Exchange M and B	(M) ↔ (B)	3
XMX	X - 37	Exchange Memory and Index	(M) ↔ (X)	3
LDS	06	Load Selective	$(M)(B)+(A)(\overline{B}) \rightarrow A$	2
STS	70	Store Selective	$(A)(B)+(M)(\overline{B}) \rightarrow M$	3
ARITHMETIC				
ADD	05	Add M to A	(A)+(M) → A	2
DPA	25	Double Precision Add	(A, B)+(M, M+1) → A, B	3
SUB	04	Subtract	(A) - (M) → A	2
DPS	24	Double Precision Subtract	(A, B)-(M, M+1) → A, B	3
MPO	71	Memory Plus One	(M)+1 → M	3
MPT	72	Memory Plus Two	(M)+2 → M	3
MUL	63	Multiply	(A)×(M) → A, B	5
DIV	62	Divide	(A, B)÷(M) → A, Rem → B	10
ADM	35	Add A to M	(A)+(M) → M	3
TMU	61	Twin Multiply	$(A)_{0-11} \times (M)_{0-11} \rightarrow A$; $(A)_{12-23} \times (M)_{12-23} \rightarrow B$	5
DPN	27	Double Precision Negate	-(A, B) → A, B	3

Designation	Instruction Code	Name	Function	Timing
FLOATING POINT				
FLA	65	Floating Add	(A, B) + (M, M+1) → A, B Floating	6-11
FLS	64	Floating Subtract	(A, B) - (M, M+1) → A, B Floating	6-11
FLM	67	Floating Multiply	(A, B) × (M, M+1) → A, B Floating	8
FLD	66	Floating Divide	(A, B) ÷ (M, M+1) → A, B Floating	17
LOGICAL				
ETR	11	Extract	(A)(M) → A	2
MRG	13	Merge	(A) OR (M) → A	2
EOR	12	Exclusive OR	(A) $(\overline{M})$ OR $(\overline{A})$ (M) → Z	2
REGISTER CHANGE				
Mode 1				
RCH, COPY	40			
	0 40 40000	Copy Negative of A into A	-(A) → A	1
	0 40 XXX01	Clear A	0 → A	1
	0 40 XXX10	Clear B	0 → B	1
	0 40 XXX11	Clear AB	0 → A, B	1
	0 40 XXX30	Copy A into B	(A) → B	1
	0 40 XXX50	Copy Inverse of A into B	$(\overline{A})$ → B	1
	0 40 XXX51	Copy Inverse of A into B, Clear A	$(\overline{A})$ → B, 0 → A	1
	0 40 XXX03	Copy B into A	(B) → A	1
	0 40 XXX33	Exchange A and B	(A) ↔ (B)	1
	0 40 XXX53	Exchange Inverse of A with B	$(\overline{A})$ ↔ (B)	1
	0 40 XXX35	Exchange Inverse of B with A	$(\overline{B})$ ↔ (A)	1
	0 40 XXX55	Exhange Inverse of A with Inverse of B	$(\overline{A})$ ↔ $(\overline{B})$	1
	0 40 XXX05	Copy Inverse of B into A	$(\overline{B})$ → A	1
	0 40 XXX15	Copy Inverse of B into A, Clear B	$(\overline{B})$ → A, 0 → B	1
	0 40 XXX31	Copy A into B, Clear A	(A) → B, 0 → A	1
	0 40 XXX13	Copy B into A, Clear B	(B) → A, 0 → B	1

Designation	Instruction Code	Name	Function	Timing
REGISTER CHANGE (Cont.)				
Mode 1 (Cont.)				
	0 40 XXX20	Merge A into B	(A) OR (B) → B	1
	0 40 XXX02	Merge B into A	(B) OR(A) → A	1
	0 40 XXX21	Merge A into B, Clear A	(A) OR (B) → B, 0 → A	1
	0 40 XXX12	Merge B into A, Clear B	(B) OR(A) → A, 0 → B	1
	0 40 XXX40	Merge Inverse of A into B	$(\overline{A})$ OR (B) → B	1
	0 40 XXX04	Merge Inverse of B into A	$(\overline{B})$ OR(A) → A	1
	0 40 XXX41	Merge Inverse of A into B, Clear A	$(\overline{A})$ OR (B) → B, 0 → A	1
	0 40 XXX14	Merge Inverse of B into A, Clear B	$(\overline{B})$ OR(A) → A, 0 → B	1
	0 40 XXX70	Form Mask in B	(A) OR $(\overline{A})$ → B	1
	0 40 XXX07	Form Mask in A	(B) OR$(\overline{B})$ → A	1
	0 40 XXX71	Form Mask in B, Clear A	(A) OR $(\overline{A})$ → B, 0 → A	1
	0 40 XXX17	Form Mask in A, Clear B	(B) OR$(\overline{B})$ → A, 0 → B	1
Mode 2				
RCH, COPY	40			
	X 40 XX005	Copy Index into A	(X) → A	1
	X 40 XX050	Copy Index into B	(X) → B	1
	X 40 X0100	Copy A into Index	(A) → X	1
	X 40 X0200	Copy B into Index	(B) → X	1
	X 40 X4000	Copy Index 1 into Index	(X1) → X	1
	X 40 X2000	Copy Index 2 into Index	(X2) → X	1
	X 40 X1000	Copy Index 3 into Index	(X3) → X	1
	X 40 X0105	Exchange Index and A	(X) ↔ (A)	1
	X 40 X0250	Exchange Index and B	(X) ↔ (B)	1
	X 40 X0000	Clear Index	0 → X	1

Designation	Instruction Code	Name	Function	Timing
Mode 3				
AXB	4X 40 XXXXX	Address to Index Base	$(P)_{9-23} \longrightarrow X_{9-23}$	1
BRANCH				
BRU	01	Branch Unconditionally	$M \longrightarrow P$	1
BRX	X-57	Increase Index and Branch	$(X)_{0-8} + (X)_{9-23} \longrightarrow X_{9-23}$ If no overflow $M \longrightarrow P$	2,3
BRC	0-57	Branch and Clear Interrupt	$M \longrightarrow P$, Clear Interrupt If Indirect $(M)_{3-8} + (F) \longrightarrow F$	2
BRM	03	Mark Place and Branch	$(P) \longrightarrow M_{9-23}$, $(F) \longrightarrow M_{3-8}$ $0 \longrightarrow M_{0-2}$, $M+1 \longrightarrow P$	3
BMA	43	Branch and Mark Place of Argument Address	$(P)+1 \longrightarrow M_{9-23}$, $(F) \longrightarrow M_{3-8}$ $0 \longrightarrow M_{1-2}$, $1 \longrightarrow M_0$, $0 \longrightarrow F$ $M + 1 \longrightarrow P$	3
BRR	41	Return Branch	$(M)_{9-23} + 1 \longrightarrow P$ $(M)_{3-8} + (F) \longrightarrow F$	2
TEST/SKIP				
SKE	45	Skip if A Equals M	If (A) = (M), Skip	2,3
SKU	47	Skip if A Unequal to M	If $(A) \neq (M)$, Skip	2,3
SKG	46	Skip if A Greater than M	If (A) > (M), Skip	2,3
SKL	44	Skip if A Less than or Equal to M	If $(A) \leq (M)$, Skip	2,3
SKR	73	Reduce M, Skip if Negative	$(M) - 1 \longrightarrow M$; if (M) <0, Skip	3
SKM	55	Skip if A = M on B Mask	If (A)(B) = (M)(B), Skip	2,3
SKN	53	Skip if M Negative	If (M) <0, Skip	2,3
SKA	54	Skip if M and A Do Not Compare Ones	If (A)(M) = 0, Skip	2,3
SKB	52	Skip if M and B Do Compare Ones	If $(B)(M) \neq 0$, Skip	2,3
SKP	51	Skip if Bit Sum Even	If sum of (B)(M) Even, Skip	2,3
SKS	20	Skip if Signal Not Set		mode 0&2; 1,2 mode 1&3; 2,3

Designation	Instruction Code	Name	Function	Timing
TEST/SKIP (Cont.)				
SKF	50	Skip if Floating Exponent in B $\geq$ M	If $(B)_{15-23} \geq (M)_{15-23,}$ Skip	2,3
SKQ	56	Skip if Masked Quantity in A Greater than M	If (A)(B) > (M)(B), Skip	2,3
SHIFT				
SHIFT	60	Shift (used with indirect addressing)		2-7
ARSA	60-20	Arithmetic Right Shift A	$(A)_{1-23} \longrightarrow$	2-7
ARSB	60-10	Arithmetic Right Shift B	$(B)_{1-23} \longrightarrow$	2-7
ARSD	60-00	Arithmetic Right Shift Double	$(A)_{1-23} \longrightarrow (B) \longrightarrow$	2-7
ARST	60-30	Arithmetic Right Shift (Twin A and B)	$(A)_{1-23} \longrightarrow$ $(B)_{1-23} \longrightarrow$	2-7
LRSA	60-21	Logical Right Shift A	$(A) \longrightarrow$	2-7
LRSB	60-11	Logical Right Shift B	$(B) \longrightarrow$	2-7
LRSD	60-01	Logical Right Shift Double	$(A) \longrightarrow (B) \longrightarrow$	2-7
LRST	60-31	Logical Right Shift Twin (A and B)	$(A) \longrightarrow$ $(B) \longrightarrow$	2-7
CRSA	60-22	Circular Right Shift A	$(A) \longrightarrow (A)$	2-7
CRSB	60-12	Circular Right Shift B	$(B) \longrightarrow (B)$	2-7
CRSD	60-02	Circular Right Shift Double	$(A) \longrightarrow (B) \longrightarrow (A)$	2-7
CRST	60-32	Circular Right Shift Twin (A and B)	$(A) \longrightarrow (A)$ $(B) \longrightarrow (B)$	2-7
ALSA	60-24	Arithmetic Left Shift A	$\longleftarrow (\pm A)$	2-5
ALSB	60-14	Arithmetic Left Shift B	$\longleftarrow (\pm B)$	2-5
ALSD	60-04	Arithmetic Left Shift Double	$\longleftarrow (\pm A) \longleftarrow (\pm B)$	2-5
LSAB	60-34	Arithmetic Left Shift Twin (A and B)	$\longleftarrow (\pm A)$ $\longleftarrow (\pm B)$	2-5
LLSA	60-25	Logical Left Shift A	$\longleftarrow (A)$	2-5
LLSB	60-15	Logical Left Shift B	$\longleftarrow (B)$	2-5
LLSD	60-05	Logical Left Shift Double	$\longleftarrow (A) \longleftarrow (B)$	2-5

Designation	Instruction Code	Name	Function	Timing
SHIFT (Cont.)				
LLST	60-35	Logical Left Shift Twin (A and B)	← (A) ← (B)	2-5
CLSA	60-26	Circular Left Shift A	(A) ← (A)	2-5
CLSB	60-16	Circular Left Shift B	(B) ← (B)	2-5
CLSD	60-06	Circular Left Shift Double	(A) ← (B) ← (A)	2-5
CLST	60-36	Circular Left Shift Twin (A and B)	(A) ← (A) (B) ← (B)	2-5
NORA	60-64	Normalize A	← (±A), (X1) - N → X1	2-5
NORD	60-44	Normalize Double	← (±A) ← (B), (X1) - N → X1	2-5
FLAG REGISTER				
FLAG	22	Flag (Single operand expression)		1, 2
FIRS	22-0	Flag Indicator Reset/Set	R(F) M_{12-17} S(F) M_{18-23}	1
FSTR	22-1	Flag Indicator Set Test/Reset	If (F)M_{18-23} Not Set, Skip	1, 2
FRTS	22-2	Flag Indicator Reset Test/Set	If (F)M_{12-17} Not Reset, Skip S(F)M_{18-23}	1, 2
FRST	22-3	Flag Indicator Reset/ Set Test	If (F)M_{12-17} Not Reset and If (F)M_{18-23} Not Set, Skip	1, 2
SWT	22-4	SENSE Switch Test	If (S.S.)M_{12-17} Not Reset, and If (S.S.)M_{18-23} Not Set, Skip	1, 2
CONTROL				
HLT	00	Halt	Halts Computation	1
NOP	10	No Operation	--	2
EXU	21	Execute	Execute the instruction M	1
INT	07	Interpret	$(M)_{3-8}$ → $X2_{18-23}$ If $(M)_1 = 1$, Skip	2, 3
REP	23	Repeat Instruction in M	Repeat instruction in M until X2 = 0	3+N(T-1)

Designation	Instruction Code	Name	Function	Timing
INTERRUPTS				
EIR	0 02 20002	Enable Interrupt System		1
DIR	0 02 20004	Disable Interrupt System		1
AIR	0 02 20020	Arm Interrupts		1
IET	0 20 20004	Interrupt Enabled Test	Skip if Interrupt System Enabled	1, 2
IDT	0 20 20002	Interrupt Disabled Test	Skip if Interrupt System Disabled	1, 2
INPUT/OUTPUT				
EOM	02	Energize Output M	1.75 μsec pulse to device M	1
EOD	42	Energize Output to Direct Access Channel		1
PIN	33	Parallel Input	External Lines ⟶ M	4
POT	31	Parallel Output	(M) ⟶ External Lines	3
MIA	30	Memory into Channel A buffer	(M) ⟶ A	2
AIM	32	Channel A buffer into Memory	(A) ⟶ M	3
CHANNEL CONTROL				
DSC 0	0 02 00000	Disconnect Channel A		1
DSC 1	0 02 00100	Disconnect Channel B		1
DSC 2	2 02 00000	Disconnect Channel C		1
DSC 3	2 02 00100	Disconnect Channel D		1
DSC 4	0 42 00000	Disconnect Channel E		1
DSC 5	0 42 00100	Disconnect Channel F		1
DSC 6	2 42 00000	Disconnect Channel G		1
DSC 7	2 42 00100	Disconnect Channel H		1
ALC 0	0 02 50000	Alert Channel A		1
ALC 1	0 02 50100	Alert Channel B		1
ALC 2	2 02 50000	Alert Channel C		1
ALC 3	2 02 50100	Alert Channel D		1
ALC 4	0 42 50000	Alert Channel E		1
ALC 5	0 42 50100	Alert Channel F		1
ALC 6	2 42 50000	Alert Channel G		1
ALC 7	2 42 50100	Alert Channel H		1

Designation		Instruction Code	Name	Function	Timing
CHANNEL CONTROL (Cont.)					
ASC	0	0 02 12000	Alert to Store Address in Channel A		1
ASC	1	0 02 12100	Alert to Store Address in Channel B		1
ASC	2	2 02 12000	Alert to Store Address in Channel C		1
ASC	3	2 02 12100	Alert to Store Address in Channel D		1
ASC	4	0 42 12000	Alert to Store Address in Channel E		1
ASC	5	0 42 12100	Alert to Store Address in Channel F		1
ASC	6	2 42 12000	Alert to Store Address in Channel G		1
ASC	7	2 42 12100	Alert to Store Address in Channel H		1
TOP	0	0 02 14000	Terminate Output on Channel A		1
TOP	1	0 02 14100	Terminate Output on Channel B		1
TOP	2	2 02 14000	Terminate Output on Channel C		1
TOP	3	2 02 14100	Terminate Output on Channel D		1
TOP	4	0 42 14000	Terminate Output on Channel E		1
TOP	5	0 42 14100	Terminate Output on Channel F		1
TOP	6	2 42 14000	Terminate Output on Channel G		1
TOP	7	2 42 14100	Terminate Output on Channel H		1
CAT	0	0 20 14000	Channel A Active Test		2,3
CAT	1	0 20 14100	Channel B Active Test		2,3
CAT	2	2 20 14000	Channel C Active Test		2,3
CAT	3	2 20 14100	Channel D Active Test		2,3
CAT	4	0 20 54000	Channel E Active Test		2,3
CAT	5	0 20 54100	Channel F Active Test		2,3
CAT	6	2 20 54000	Channel G Active Test		2,3
CAT	7	2 20 54100	Channel H Active Test		2,3
CET	0	0 20 11000	Channel A Error Test		2,3
CET	1	0 20 11100	Channel B Error Test		2,3
CET	2	2 20 11000	Channel C Error Test		2,3
CET	3	2 20 11100	Channel D Error Test		2,3
CET	4	0 20 51000	Channel E Error Test		2,3
CET	5	0 20 51100	Channel F Error Test		2,3
CET	6	2 20 51000	Channel G Error Test		2,3
CET	7	2 20 51100	Channel H Error Test		2,3

Designation		Instruction Code	Name	Function	Timing
CHANNEL CONTROL (Cont.)					
CIT	0	0 20 10400	Channel A Inter-record Test		2,3
CIT	1	0 20 10500	Channel B Inter-record Test		2,3
CIT	2	2 20 10400	Channel C Inter-record Test		2,3
CIT	3	2 20 10500	Channel D Inter-record Test		2,3
CIT	4	0 20 50400	Channel E Inter-record Test		2,3
CIT	5	0 20 50500	Channel F Inter-record Test		2,3
CIT	6	2 20 50400	Channel G Inter-record Test		2,3
CIT	7	2 20 50500	Channel H Inter-record Test		2,3
CZT	0	0 20 12000	Channel A Zero Count Test		2,3
CZT	1	0 20 12100	Channel B Zero Count Test		2,3
CZT	2	2 20 12000	Channel C Zero Count Test		2,3
CZT	3	2 20 12100	Channel D Zero Count Test		2,3
CZT	4	0 20 52000	Channel E Zero Count Test		2,3
CZT	5	0 20 52100	Channel F Zero Count Test		2,3
CZT	6	2 20 52000	Channel G Zero Count Test		2,3
CZT	7	2 20 52100	Channel H Zero Count Test		2,3

SDS 9300 INSTRUCTION LIST — NUMERICAL ORDER

Instruction Code	Designation	Name	Page
00	HLT	Halt	2-24
01	BRU	Branch Unconditionally	2-14
02	EOM	Energize Output M	4-5
0 02 00000	DSC 0	Disconnect Channel A	4-9
0 02 00100	DSC 1	Disconnect Channel B	4-9
2 02 00000	DSC 2	Disconnect Channel C	4-9
2 02 00100	DSC 3	Disconnect Channel D	4-9
0 02 12000	ASC 0	Alert to Store Address in Channel A	4-9
0 02 12100	ASC 1	Alert to Store Address in Channel B	4-9
2 02 12000	ASC 2	Alert to Store Address in Channel C	4-9
2 02 12100	ASC 3	Alert to Store Address in Channel D	4-9
0 02 14000	TOP 0	Terminate Output on Channel A	4-9
0 02 14100	TOP 1	Terminate Output on Channel B	4-9
2 02 14000	TOP 2	Terminate Output on Channel C	4-9
2 02 14100	TOP 3	Terminate Output on Channel D	4-9
0 02 20002	EIR	Enable Interrupt System	3-4
0 02 20004	DIR	Disable Interrupt System	3-4
0 02 20020	AIR	Arm Interrupts	3-5
0 02 50000	ALC 0	Alert Channel A	4-9
0 02 50100	ALC 1	Alert Channel B	4-9
2 02 50000	ALC 2	Alert Channel C	4-9
2 02 50100	ALC 3	Alert Channel D	4-9
03	BRM	Mark Place and Branch	2-15
04	SUB	Subtract	2-4
05	ADD	Add M to A	2-3
06	LDS	Load Selective	2-2
07	INT	Interpret	2-25
10	NOP	No Operation	2-24
11	ETR	Extract	2-7
12	EOR	Exclusive OR	2-8
13	MRG	Merge	2-7
14	LDB	Load B	2-2

Instruction Code	Designation	Name	Page
15	EAX	Copy Effective Address into Index Register 1	2-3
16	LDA	Load A	2-1
X-17	LDX	Load Index	2-2
20	SKS	Skip if Signal Not Set	4-6
0 20 10400	CIT 0	Channel A Inter-record Test	4-16
0 20 10500	CIT 1	Channel B Inter-record Test	4-16
2 20 10400	CIT 2	Channel C Inter-record Test	4-16
2 20 10500	CIT 3	Channel D Inter-record Test	4-16
0 20 11000	CET 0	Channel A Error Test	4-15
0 20 11100	CET 1	Channel B Error Test	4-15
2 20 11000	CET 2	Channel C Error Test	4-15
2 20 11100	CET 3	Channel D Error Test	4-15
0 20 12000	CZT 0	Channel A Zero Count Test	4-15
0 20 12100	CZT 1	Channel B Zero Count Test	4-15
2 20 12000	CZT 2	Channel C Zero Count Test	4-15
2 20 12100	CZT 3	Channel D Zero Count Test	4-15
0 20 14000	CAT 0	Channel A Active Test	4-15
0 20 14100	CAT 1	Channel B Active Test	4-15
2 20 14000	CAT 2	Channel C Active Test	4-15
2 20 14100	CAT 3	Channel D Active Test	4-15
0 20 20002	IDT	Interrupt Disabled Test	3-4
0 20 20004	IET	Interrupt Enabled Test	3-4
0 20 50400	CIT 4	Channel E Inter-record Test	4-16
0 20 50500	CIT 5	Channel F Inter-record Test	4-16
2 20 50400	CIT 6	Channel G Inter-record Test	4-16
2 20 50500	CIT 7	Channel H Inter-record Test	4-16
0 20 51000	CET 4	Channel E Error Test	4-15
0 20 51100	CET 5	Channel F Error Test	4-15
2 20 51000	CET 6	Channel G Error Test	4-15
2 20 51100	CET 7	Channel H Error Test	4-15
0 20 52000	CZT 4	Channel E Zero Count Test	4-15
0 20 52100	CZT 5	Channel F Zero Count Test	4-15
2 20 52000	CZT 6	Channel G Zero Count Test	4-15
2 20 52100	CZT 7	Channel H Zero Count Test	4-15

Instruction Code	Designation	Name	Page
0 20 54000	CAT 4	Channel E Active Test	4-15
0 20 54100	CAT 5	Channel F Active Test	4-15
2 20 54000	CAT 6	Channel G Active Test	4-15
2 20 54100	CAT 7	Channel H Active Test	4-15
21	EXU	Execute	2-24
22	FLAG	Flag	2-24
22-0	FIRS	Flag Indicator Reset/Set	2-27
22-1	FSTR	Flag Indicator Set Test/Reset	2-27
22-2	FRTS	Flag Indicator Reset Test/Set	2-28
22-3	FRST	Flag Indicator Reset/Set Test	2-28
22-4	SWT	SENSE Switch Test	2-28
23	REP	Repeat Instruction in M	2-25
24	DPS	Double Precision Subtract	2-4
25	DPA	Double Precision Add	2-4
26	LDP	Load Double Precision	2-2
27	DPN	Double Precision Negate	2-5
30	MIA	Memory into A	4-16
31	POT	Parallel Output	4-7
32	AIM	A into Memory	4-16
33	PIN	Parallel Input	4-7
34	XMB	Exchange M and B	2-3
35	ADM	Add A to M	2-4
36	XMA	Exchange M and A	2-3
X-37	XMX	Exchange Memory and Index	2-3
40	RCH, COPY	Register Change (Modes 1 and 2)	2-9
0 40 XXX01		Clear A	2-9
0 40 XXX02		Merge B into A	2-11
0 40 XXX03		Copy B into A	2-10
0 40 XXX04		Merge Inverse of B into A	2-11
0 40 XXX05		Copy Inverse of B into A	2-10
0 40 XXX07		Form Mask in A	2-12
0 40 XXX10		Clear B	2-9
0 40 XXX11		Clear AB	2-10
0 40 XXX12		Merge B into A, Clear B	2-11
0 40 XXX13		Copy B into A, Clear B	2-11
0 40 XXX14		Merge Inverse of B into A, Clear B	2-12

Instruction Code	Designation	Name	Page
0 40 XXX15		Copy Inverse of B into A, Clear B	2-11
0 40 XXX17		Form Mask in A, Clear B	2-12
0 40 XXX20		Merge A into B	2-11
0 40 XXX21		Merge A into B, Clear A	2-11
0 40 XXX30		Copy A into B	2-10
0 40 XXX31		Copy A into B, Clear A	2-11
0 40 XXX33		Exchange A and B	2-10
0 40 XXX35		Exchange Inverse of B with A	2-10
0 40 XXX40		Merge Inverse of A into B	2-11
0 40 XXX41		Merge Inverse of A into B, Clear A	2-11
0 40 XXX50		Copy Inverse of A into B	2-10
0 40 XXX51		Copy Inverse of A into B, Clear A	2-10
0 40 XXX53		Exchange Inverse of A with B	2-10
0 40 XXX55		Exchange Inverse of A with Inverse of B	2-10
0 40 XXX70		Form Mask in B	2-12
0 40 XXX71		Form Mask in B, Clear A	2-12
0 40 40000		Copy Negative of A into A	2-9
X 40 XX005		Copy Index into A	2-13
X 40 XX050		Copy Index into B	2-13
X 40 X0000		Clear Index	2-14
X 40 X0100		Copy A into Index	2-13
X 40 X0105		Exchange Index and A	2-14
X 40 X0200		Copy B into Index	2-13
X 40 X0250		Exchange Index and B	2-14
X 40 X1000		Copy Index 3 into Index	2-13
X 40 X2000		Copy Index 2 into Index	2-13
X 40 X4000		Copy Index 1 into Index	2-13
4X 40 XXXXX	AXB	Address to Index Base	2-14
41	BRR	Return Branch	2-15
42	EOD	Energize Output Direct Access Channel	4-6
0 42 00000	DSC 4	Disconnect Channel E	4-9
0 42 00100	DSC 5	Disconnect Channel F	4-9
2 42 00000	DSC 6	Disconnect Channel G	4-9
2 42 00100	DSC 7	Disconnect Channel H	4-9
0 42 12000	ASC 4	Alert to Store Address in Channel E	4-9
0 42 12100	ASC 5	Alert to Store Address in Channel F	4-9
2 42 12000	ASC 6	Alert to Store Address in Channel G	4-9
2 42 12100	ASC 7	Alert to Store Address in Channel H	4-9

Instruction Code	Designation	Name	Page
0 42 14000	TOP E	Terminate Output on Channel E	4-9
0 42 14100	TOP F	Terminate Output on Channel F	4-9
2 42 14000	TOP G	Terminate Output on Channel G	4-9
2 42 14100	TOP H	Terminate Output on Channel H	4-9
0 42 50000	ALC 4	Alert Channel E	4-9
0 42 50100	ALC 5	Alert Channel F	4-9
2 42 50000	ALC 6	Alert Channel G	4-9
2 42 50100	ALC 7	Alert Channel H	4-9
43	BMA	Branch and Mark Place of Argument Address	2-15
44	SKL	Skip if A Less than M	2-16
45	SKE	Skip if A Equals M	2-16
46	SKG	Skip if A Greater than M	2-16
47	SKU	Skip if A Unequal to M	2-16
50	SKF	Skip if Floating Exponent in B $\geq$ M	2-17
51	SKP	Skip if Bit Sum Even	2-18
52	SKB	Skip if M and B Do Compare Ones	2-17
53	SKN	Skip if M Negative	2-18
54	SKA	Skip if M and A Do Not Compare Ones	2-17
55	SKM	Skip if A = M on B Mask	2-16
56	SKQ	Skip if Masked Quantity in A Greater than M	2-16
0-57	BRC	Branch and Clear Interrupt	2-14
X-57	BRX	Increase Index and Branch	2-14
60	SHIFT	Shift	2-19
60-00	ARSD	Arithmetic Right Shift Double	2-20
60-01	LRSD	Logical Right Shift Double	2-21
60-02	CRSD	Circular Right Shift Double	2-21
60-04	ALSD	Arithmetic Left Shift Double	2-22
60-05	LLSD	Logical Left Shift Double	2-22
60-06	CLSD	Circular Left Shift Double	2-23
60-10	ARSB	Arithmetic Right Shift B	2-20
60-11	LRSB	Logical Right Shift B	2-21
60-12	CRSB	Circular Right Shift B	2-21
60-14	ALSB	Arithmetic Left Shift B	2-22
60-15	LLSB	Logical Left Shift B	2-22
60-16	CLSB	Circular Left Shift B	2-23
60-20	ARSA	Arithmetic Right Shift A	2-20
60-21	LRSA	Logical Right Shift A	2-20

Instruction Code	Designation	Name	Page
60-22	CRSA	Circular Right Shift A	2-21
60-24	ALSA	Arithmetic Left Shift A	2-22
60-25	LLSA	Logical Left Shift A	2-22
60-26	CLSA	Circular Left Shift A	2-23
60-30	ARST	Arithmetic Right Shift Twin (A and B)	2-20
60-31	LRST	Logical Right Shift Twin (A and B)	2-21
60-32	CRST	Circular Right Shift Twin (A and B)	2-21
60-34	ALST	Arithmetic Left Shift Twin (A and B)	2-22
60-35	LLST	Logical Left Shift Twin (A and B)	2-23
60-36	CLST	Circular Left Shift Twin (A and B)	2-23
60-44	NORD	Normalize Double	2-24
60-64	NORA	Normalize A	2-24
61	TMU	Twin Multiply	2-5
62	DIV	Divide	2-5
63	MUL	Multiply	2-5
64	FLS	Floating Subtract	2-7
65	FLA	Floating Add	2-6
66	FLD	Floating Divide	2-7
67	FLM	Floating Multiply	2-7
70	STS	Store Selective	2-2
71	MPO	Memory Plus One	2-4
72	MPT	Memory Plus Two	2-4
73	SKR	Reduce M, Skip if Negative	2-18
74	STB	Store B	2-2
75	STD	Store Double Precision	2-2
76	STA	Store A	2-2
0-77	STZ	Store Zero	2-3
X-77	STX	Store Index	2-3

SDS 9300 INSTRUCTION LIST — ALPHABETICAL ORDER

Designation	Instruction Code	Name	Page
ADD	05	Add M to A	2-3
ADM	35	Add A to M	2-4
AIM	32	A into Memory	4-16
AIR	0 02 20020	Arm Interrupts	3-5
ALC 0	0 02 50000	Alert Channel A	4-9
ALC 1	0 02 50100	Alert Channel B	4-9
ALC 2	2 02 50000	Alert Channel C	4-9
ALC 3	2 02 50100	Alert Channel D	4-9
ALC 4	0 42 50000	Alert Channel E	4-9
ALC 5	0 42 50100	Alert Channel F	4-9
ALC 6	2 42 50000	Alert Channel G	4-9
ALC 7	2 42 50100	Alert Channel H	4-9
ALSA	60-24	Arithmetic Left Shift A	2-22
ALSB	60-14	Arithmetic Left Shift B	2-22
ALSD	60-04	Arithmetic Left Shift Double	2-22
ALST	60-34	Arithmetic Left Shift Twin (A and B)	2-22
ARSA	60-20	Arithmetic Right Shift A	2-20
ARSB	60-10	Arithmetic Right Shift B	2-20
ARSD	60-00	Arithmetic Right Shift Double	2-20
ARST	60-30	Arithmetic Right Shift Twin (A and B)	2-20
ASC 0	0 02 12000	Alert to Store Address in Channel A	4-9
ASC 1	0 02 12100	Alert to Store Address in Channel B	4-9
ASC 2	2 02 12000	Alert to Store Address in Channel C	4-9
ASC 3	2 02 12100	Alert to Store Address in Channel D	4-9
ASC 4	0 42 12000	Alert to Store Address in Channel E	4-9
ASC 5	0 42 12100	Alert to Store Address in Channel F	4-9
ASC 6	2 42 12000	Alert to Store Address in Channel G	4-9
ASC 7	2 42 12100	Alert to Store Address in Channel H	4-9
AXB	4X 40 XXXXX	Address to Index Base	2-14
BMA	43	Branch and Mark Place of Argument Address	2-15
BRC	0-57	Branch and Clear Interrupt	2-14
BRM	03	Mark Place and Branch	2-15
BRR	41	Return Branch	2-15

Designation	Instruction Code	Name	Page
BRU	01	Branch Unconditionally	2-14
BRX	X-57	Increase Index and Branch	2-14
CAT 0	0 20 14000	Channel A Active Test	4-15
CAT 1	0 20 14100	Channel B Active Test	4-15
CAT 2	2 20 14000	Channel C Active Test	4-15
CAT 3	2 20 14100	Channel D Active Test	4-15
CAT 4	0 20 54000	Channel E Active Test	4-15
CAT 5	0 20 54100	Channel F Active Test	4-15
CAT 6	2 20 54000	Channel G Active Test	4-15
CAT 7	2 20 54100	Channel H Active Test	4-15
CET 0	0 20 11000	Channel A Error Test	4-15
CET 1	0 20 11100	Channel B Error Test	4-15
CET 2	2 20 11000	Channel C Error Test	4-15
CET 3	2 20 11100	Channel D Error Test	4-15
CET 4	0 20 51000	Channel E Error Test	4-15
CET 5	0 20 51100	Channel F Error Test	4-15
CET 6	2 20 51000	Channel G Error Test	4-15
CET 7	2 20 51100	Channel H Error Test	4-15
CIT 0	0 20 10400	Channel A Inter-record Test	4-16
CIT 1	0 20 10500	Channel B Inter-record Test	4-16
CIT 2	2 20 10400	Channel C Inter-record Test	4-16
CIT 3	2 20 10500	Channel D Inter-record Test	4-16
CIT 4	0 20 50400	Channel E Inter-record Test	4-16
CIT 5	0 20 50500	Channel F Inter-record Test	4-16
CIT 6	2 20 50400	Channel G Inter-record Test	4-16
CIT 7	2 20 50500	Channel H Inter-record Test	4-16
CLSA	60-26	Circular Left Shift A	2-23
CLSB	60-16	Circular Left Shift B	2-23
CLSD	60-06	Circular Left Shift Double	2-23
CLST	60-36	Circular Left Shift Twin (A and B)	2-23
COPY	40	Copy	2-9
CRSA	60-22	Circular Right Shift A	2-21
CRSB	60-12	Circular Right Shift B	2-21
CRSD	60-02	Circular Right Shift Double	2-21
CRST	60-32	Circular Right Shift Twin (A and B)	2-21

Designation	Instruction Code	Name	Page
CZT 0	0 20 12000	Channel A Zero Count Test	4-15
CZT 1	0 20 12100	Channel B Zero Count Test	4-15
CZT 2	2 20 12000	Channel C Zero Count Test	4-15
CZT 3	2 20 12100	Channel D Zero Count Test	4-15
CZT 4	0 20 52000	Channel E Zero Count Test	4-15
CZT 5	0 20 52100	Channel F Zero Count Test	4-15
CZT 6	2 20 52000	Channel G Zero Count Test	4-15
CZT 7	2 20 52100	Channel H Zero Count Test	4-15
DIR	0 02 00004	Disable Interrupt System	3-4
DIV	62	Divide	2-5
DPA	25	Double Precision Add	2-4
DPN	27	Double Precision Negate	2-5
DPS	24	Double Precision Subtract	2-4
DSC 0	0 02 00000	Disconnect Channel A	4-9
DSC 1	0 02 00100	Disconnect Channel B	4-9
DSC 2	2 02 00000	Disconnect Channel C	4-9
DSC 3	2 02 00100	Disconnect Channel D	4-9
DSC 4	0 42 00000	Disconnect Channel E	4-9
DSC 5	0 42 00100	Disconnect Channel F	4-9
DSC 6	2 42 00000	Disconnect Channel G	4-9
DSC 7	2 42 00100	Disconnect Channel H	4-9
EAX	15	Copy Effective Address into Index Register 1	2-3
EIR	0 02 20002	Enable Interrupt System	3-4
EOD	42	Energize Output Direct Access Channel	4-6
EOM	02	Energize Output M	4-5
EOR	12	Exclusive OR	2-8
ETR	11	Extract	2-7
EXU	21	Execute	2-24
FIRS	22-0	Flag Indicator Reset/Set	2-27
FLA	65	Floating Add	2-6
FLAG	22	Flag	2-27
FLD	66	Floating Divide	2-7
FLM	67	Floating Multiply	2-7
FLS	64	Floating Subtract	2-7
FRST	22-3	Flag Indicator Reset/Set Test	2-28
FRTS	22-3	Flag Indicator Reset Test/Set	2-28

Designation	Instruction Code	Name	Page
FSTR	22-1	Flag Indicator Set Test/Reset	2-27
HLT	00	Halt	2-24
IDT	0 20 20002	Interrupt Disabled Test	3-4
IET	0 20 20004	Interrupt Enabled Test	3-4
INT	07	Interpret	2-25
LDA	16	Load A	2-1
LDB	14	Load B	2-2
LDP	26	Load Double Precision	2-2
LDS	06	Load Selective	2-2
LDX	X-17	Load Index	2-2
LLSA	60-25	Logical Left Shift A	2-22
LLSB	60-15	Logical Left Shift B	2-22
LLSD	60-05	Logical Left Shift Double	2-22
LLST	60-35	Logical Left Shift Twin (A and B)	2-23
LRSA	60-21	Logical Right Shift A	2-20
LRSB	60-11	Logical Right Shift B	2-21
LRSD	60-01	Logical Right Shift Double	2-21
LRST	60-31	Logical Right Shift Twin (A and B)	2-21
MIA	30	Memory into A	4-16
MPO	71	Memory Plus One	2-4
MPT	72	Memory Plus Two	2-4
MRG	13	Merge	2-7
MUL	63	Multiply	2-5
NOP	10	No Operation	2-24
NORA	60-64	Normalize A	2-24
NORD	60-44	Normalize Double	2-24
PIN	33	Parallel Input	4-7
POT	31	Parallel Output	4-7
RCH	40	Register Change	2-9
REP	23	Repeat Instruction in M	2-25
SHIFT	60	Shift	2-19
SKA	54	Skip if M and A Do Not Compare Ones	2-17
SKB	52	Skip if M and B Do Compare Ones	2-17
SKE	45	Skip if A Equals M	2-16
SKF	50	Skip if Floating Exponent in B $\geq$ M	2-17
SKG	46	Skip if A Greater than M	2-16
SKL	44	Skip if A Less than or Equal to M	2-16

Designation	Instruction Code	Name	Page
SKM	55	Skip if A = M on B Mask	2-16
SKN	53	Skip if M Negative	2-18
SKP	51	Skip if Bit Sum Even	2-18
SKQ	56	Skip if Masked Quantity in A Greater than M	2-16
SKR	73	Reduce M, Skip if Negative	2-18
SKS	20	Skip if Signal Not Set	4-6
SKU	47	Skip if A Unequal to M	2-16
STA	76	Store A	2-2
STB	74	Store B	2-2
STD	75	Store Double Precision	2-2
STS	70	Store Selective	2-2
STX	X-77	Store Index	2-3
STZ	0-77	Store Zero	2-3
SUB	04	Subtract	2-4
SWT	22-4	Sense Switch Test	
TMU	61	Twin Multiply	2-5
TOP 0	0 02 14000	Terminate Output on Channel A	4-9
TOP 1	0 02 14100	Terminate Output on Channel B	4-9
TOP 2	2 02 14000	Terminate Output on Channel C	4-9
TOP 3	2 02 14100	Terminate Output on Channel D	4-9
TOP 4	0 42 14000	Terminate Output on Channel E	4-9
TOP 5	0 42 14100	Terminate Output on Channel F	4-9
TOP 6	2 42 14000	Terminate Output on Channel G	4-9
TOP 7	2 42 14100	Terminate Output on Channel H	4-9
XMA	36	Exchange M and A	2-3
XMB	34	Exchange M and B	2-3
XMX	X-37	Exchange Memory and Index	2-3

INDEX

SDS 9300 INPUT/OUTPUT INSTRUCTIONS

Channel Instructions and Tests

Mnemonic		Code	Name	Page
PRIMARY I/O				
EOM	M, T	02	Energize Ouput M	4-5
EOD	M, T	42	Energize Output to DAC	4-6
SKS	M, T	20	Skip if Signal not Set	4-6
POT	M, T	31	Parallel Output	4-7
PIN	M, T	33	Parallel Input	4-7
MIA	M, T	30	Memory into Channel A	4-16
AIM	M, T	32	Channel A into Memory	4-16

Mnemonic		Code	Name	Page
CHANNEL†				
DSC	0	0 02 00000	Disconnect Channel	4-9
ALC	0	0 02 50000	Alert Channel	4-9
ASC	0	0 02 12000	Alert to Store Address	4-9
TOP	0	0 02 14000	Terminate Output	4-9
CAT	0	0 20 14000	Channel Active Test	4-15
CET	0	0 20 11000	Channel Error Test	4-15
CZT	0	0 20 12000	Channel Zero Count Test	4-15
CIT	0	0 20 10400	Channel Inter-Record Test	4-16

Peripheral Device Instructions and Tests

Mnemonic		Code	Name	Page
PAPER TAPE†				
RPT	0, 1, 4	0 02 02604	Read Paper Tape	6-3
PTL	0, 1, 4	0 02 00644	Punch Paper Tape, Leader	6-5
PPT	0, 1, 4	0 02 02644	Punch Paper Tape, No Leader	6-5
CARD†				
RCD	0, 1, 4	0 02 02606	Read Card Decimal (Hollerith)	6-6
RCB	0, 1, 4	0 02 03606	Read Card Binary	6-6
SRC	0, 1	0 02 12006	Skip Remainder of Card	6-6
CRT	0, 1	0 20 12006	Card Reader Ready Test	6-7
FCT	0, 1	0 20 14006	First Column Test	6-7
CFT	0, 1	0 20 11006	Card Reader EOF Test	6-8
PCD	0, 1, 4	0 02 02646	Punch Card Decimal (Hollerith)	6-10
PCB	0, 1, 4	0 02 03646	Punch Card Binary	6-10
CPT	0, 1	0 20 14046	Card Punch Ready Test	6-10
PBT	1, 1	0 20 12046	Punch Buffer Test	6-10
LINE PRINTER†				
PLP	0, 1, 4	0 02 02660	Print Line Printer	6-12
PRT	0, 1	0 20 12060	Printer Ready Test	6-13
EPT	0, 1	0 20 14060	End of Page Test	6-13
PFT	0, 1	0 20 11060	Printer Fault Test	6-13
PSC	0, 1, n	0 02 1n460	Printer Skip to Format n	6-13
PSP	0, 1, n	0 02 1n660	Printer Upspace n Lines	6-13
POL	0, 1	0 02 12060	Print Off-Line	6-13

Mnemonic		Code	Name	Page
TYPEWRITER†				
RKB	0, 1, 4	0 02 02601	Read Keyboard	6-1
TYP	0, 1, 4	0 02 02641	Writer Typewriter	6-1
MAGNETIC TAPE†				
TRT	0, 1	0 20 10411	Tape Ready Test	6-17
FPT	0, 1	0 20 14011	File Protect Test	6-17
BTT	0, 1	0 20 12011	Beginning of Tape Test	6-17
TGT	0	0 20 12610	Tape Gap Test	6-17
ETT	0, 1	0 20 11011	End of Tape Test	6-17
DT2	0, 1	0 20 16211	Density Test, 200 BPI	6-17
DT5	0, 1	0 20 16611	Density Test, 556 BPI	6-17
DT8	0, 1	0 20 17211	Density Test, 800 BPI	6-17
TFT	0	0 20 13610	Tape EOF Test	6-17
SKS	10211	0 20 10211	MAGPAK Test	6-18
RTD	0, 1, 4	0 02 02611	Read Tape Decimal (BCD)	6-18
RTB	0, 1, 4	0 02 03611	Read Tape Binary	6-18
SFD	0, 1, 4	0 02 02631	Scan Forward Decimal (BCD)	6-18
SFB	0, 1, 4	0 02 03631	Scan Forward Binary	6-18
SRD	0, 1, 4	0 02 06631	Scan Reverse Decimal (BCD)	6-19
SRB	0, 1, 4	0 02 07631	Scan Reverse Binary	6-19
REW	0, 1	0 02 14011	Rewind Tape	6-19
WTD	0, 1, 4	0 02 02651	Write Tape Decimal (BCD)	6-20
WTB	0, 1, 4	0 02 03651	Write Tape Binary	6-20
EFT	0, 1, 4	0 02 03671	Erase Forward Tape	6-20
ERT	0, 1, 4	0 02 07671	Erase Reverse Tape	6-20
RTS	0	0 02 14000	Convert Read to Scan	6-20
SRR	0	0 02 13610	Skip Remainder of Record	6-20

Legend: M = address field; *M = indirect address; T = tag field; n = number (0-7)

†Mnemonics and octal codes given are for channel A, device number 1, and 4 characters/word format

SDS

Scientific Data Systems A XEROX COMPANY

701 South Aviation Blvd. / El Segundo, California 90245 (213) 772-4511 / Cable SCIDATA / Telex 674839 / TWX 910-325-6908

EASTERN TECHNOLOGY CENTER
12150 Parklawn Drive
Rockville, Maryland 20852
(301) 933-5900

PRINTED CIRCUITS DEPT.
600 East Bonita Avenue
Pomona, Calif. 91767
(714) 624-8011

TECHNICAL TRAINING
5250 West Century Blvd.
Los Angeles, Calif. 90045
(213) 772-4511

INTERNATIONAL MANUFACTURING SUBSIDIARY

Scientific Data Systems Israel, Ltd.
P.O. Box 5101
Haifa, Israel
04-530253, 04-64589
Telex 922 4474

SALES OFFICES

Western Region

5045 N. 12th St.
Phoenix, Arizona 85014
(602) 264-9324

1360 So. Anaheim Blvd.
Anaheim, Calif. 92805
(714) 774-0461

*5250 West Century Blvd.
Los Angeles, Calif. 90045
(213) 772-4511

Vista Del Lago Office Center
122 Saratoga Avenue
Santa Clara, Calif. 95050
(408) 246-8330

3333 South Bannock
Suite 400
Englewood, Colo. 80110
(303) 761-2645

Fountain Professional Bldg
9004 Menaul Blvd., N.E.
Albuquerque, N.M. 87112
(505) 298-7683

El Paso Natural Gas Bldg.
Suite 201
315 E. 2nd South Street
Salt Lake City, Utah 84111
(801) 322-0501

400 Bldg., Suite 415
400 108th Avenue, N.E.
Bellevue, Wash. 98004
(206) 454-3991

Midwestern Region

*2720 Des Plaines Avenue
Des Plaines, Illinois 60018
(312) 824-8147

Clausen Bldg., Suite 310
16000 W. Nine Mile Road
Southfield, Michigan 48124
(313) 353-7360

4367 Woodson Road
St. Louis, Missouri 63134
(314) 423-6200

Seven Parkway Center
Suite 238
Pittsburgh, Pa. 15220
(412) 921-3640

Southern Region

State National Bank Bldg.
Suite 620
200 W. Court Square
Huntsville, Alabama 35801
(205) 539-5131

Orlando Executive Center
1080 Woodcock Road
Orlando, Florida 32803
(305) 841-6371

2964 Peachtree Road, N.W.
Suite 350
Atlanta, Georgia 30305
(404) 261-5323

Jefferson Bank Bldg.
Suite 720
3525 N. Causeway Blvd.
Metairie, Louisiana 70002
(504) 837-1515

4900 S. Lewis Avenue
Tulsa, Oklahoma 74105
(918) 743-7753

8383 Stemmons Freeway
Suite 233
Dallas, Texas 75247
(214) 637-4340

*2300 West Loop South
Suite 150
Houston, Texas 77027
(713) 623-0510

Eastern Region

20 Walnut Street
Wellesley Hills, Mass. 02181
(617) 237-2300

Brearley Office Building
190 Moore Street
Hackensack, N. J. 07601
(201) 489-0100

280 North Central Avenue
Hartsdale, New York 10530
(914) 948-2929

*1301 Avenue of the Americas
New York City, N.Y. 10019
(212) 765-1230

673 Panorama Trail West
Rochester, New York 14625
(716) 586-1500

P.O. Box 168
535 Pennsylvania Ave.
Ft. Washington Industrial Park
Ft. Washington, Pa. 19034
(215) 643-2130

Washington (D.C.) Operations

*2351 Research Blvd.
Rockville, Maryland 20850
(301) 948-8190

Canada

864 Lady Ellen Place
Ottawa 3, Ontario
(613) 722-8387

Oil Exchange Building
309 7th Avenue, S.W.
Calgary 2, Alberta
(403) 265-8134

280 Belfield Road
Rexdale 605, Ontario
(416) 677-8422

INTERNATIONAL OFFICES & REPRESENTATIVES

European/African Headquarters

Scientific Data Systems
I.L.I. House, Olympic Way
Wembley Park (London)
Middlesex, England
(01) 903-2511, Telex 27992

Sweden

Nordisk Elektronik AB
Stureplan 3
Stockholm 7
(08) 24 83 40

Denmark

A/S Nordisk Elektronik
Danasvej 2
Copenhagen V
EVA 8285/EVA 8238

Norway

Nordisk Elektronik(Norge) A/S
Middelthunsgt. 27
Oslo 3
(2) 60 25 90

France

Compagnie Internationale pour l'Informatique, C.I.I.
Executive and Sales Offices
66, Route de Versailles
78-Louveciennes
Yvelines
951 86 00 (Paris area)
Manufacturing and Engineering
Rue Jean Jaures
78-Les Clayes-sous-Bois
Yvelines

Israel

Elbit Computers Ltd.
Subsidiary of Elron Electronic Industries Ltd.
88 Hagiborim Street
Haifa
6 4613

*Regional Headquarters

90 00 50G