XDS

Xerox Data Systems

Reference Manual

XDS 910 Computer

XDS 910 BASIC INSTRUCTIONS
(Central Processors)

Mnemonic		Code	Name	Page
LOAD/STORE				
LDA	A, T	76	Load A from Memory	8
STA	A, T	35	Store A in Memory	8
LDB	A, T	75	Load B from Memory	8
STB	A, T	36	Store B in Memory	8
LDX	A, T	71	Load Index from Memory	9
STX	A, T	37	Store Index in Memory	9
EAX	A, T	77	Copy Effective Address into Index	9
ARITHMETIC				
ADD	A, T	55	Add Memory to A	9
MIN	A, T	61	Memory Increment	9
MDE	A, T	60	Memory Decrement	9
SUB	A, T	54	Subtract Memory from A	9
MUS	A, T	64	Multiply Step	9
DIS	A, T	65	Divide Step	10
LOGICAL				
ETR	A, T	14	Extract	10
MRG	A, T	16	Merge	10
EOR	A, T	17	Exclusive OR	11
REGISTER CHANGE				
RCH		46	Register Change	11
XAB		0 46 00000	Exchange A and B	11
BAC		0 46 10000	Copy B into A, Clear B	11
ABC		0 46 20000	Copy A into B, Clear A	11
CLR		0 46 30000	Clear AB	11
BRANCH				
BRU	A, T	01	Branch Unconditionally	12
BRX	A, T	41	Increment Index and Branch	12
BRM	A, T	43	Mark Place and Branch	12
BRR	A, T	51	Return Branch	12

Mnemonic		Code	Name	Page
TEST/SKIP				
SKG	A, T	73	Skip if A Greater Than Memory	13
SKM	A, T	70	Skip if A = Memory on B Mask	13
SKA	A, T	72	Skip if A and M Do Not Compare Ones	13
SKN	A, T	53	Skip if Memory Negative	14
SHIFT				
RSH	N, T	0 66 000XX	Right Shift AB	14
RCY	N, T	0 66 200XX	Right Cycle AB	14
LSH	N, T	0 67 000XX	Left Shift AB	15
LCY	N, T	0 67 200XX	Left Cycle AB	15
NOD	N, T	0 67 100XX	Normalize, Decrement X	15
CONTROL				
HLT		00	Halt	15
NOP		20	No Operation	16
EXU	A, T	23	Execute Instruction in Memory	16
BREAKPOINT TESTS				
BPT 1		0 40 20400	Breakpoint No. 1 Test	16
BPT 2		0 40 20200	Breakpoint No. 2 Test	16
BPT 3		0 40 20100	Breakpoint No. 3 Test	16
BPT 4		0 40 20040	Breakpoint No. 4 Test	16
OVERFLOW				
OVT		0 40 20001	Overflow Test; Reset	16
ROV		0 02 20001	Reset Overflow	16
INTERRUPT				
EIR		0 02 20002	Enable Interrupts	20
DIR		0 02 20004	Disable Interrupts	20
IET		0 40 20004	Interrupt Enabled Test	20
IDT		0 40 20002	Interrupt Disabled Test	20
AIR		0 02 20020	Arm Interrupts	21

Legend: A = address, *A = indirect address; T = tag field; N = number of shifts

Price: $3.25

XDS 910 COMPUTER REFERENCE MANUAL

90 00 08D

February 1970

XDS

Xerox Data Systems/701 South Aviation Boulevard/El Segundo, California 90245

Printed in U.S.A.

REVISION

This publication, 90 00 08D, is a minor revision of the XDS 910 Computer Reference Manual, 90 00 08C, dated April 1966. Changes to the previous edition are indicated by a vertical line in the margin of the affected page.

RELATED PUBLICATIONS

Title	Publication No.
XDS SYMBOL and META-SYMBOL Reference Manual	90 05 06
XDS MONARCH Reference Manual	90 05 66
XDS 910/925 Programmed Operators Technical Manual	90 00 18
XDS 910/920 Computer EXAMINER Diagnostic System Technical Manual	90 00 19
XDS FORTRAN II Reference Manual	90 00 03
XDS 900 Series FORTRAN II Operations Manual	90 05 87
XDS ALGOL 60 Reference Manual	90 06 99
XDS Project Management System Reference Manual	90 08 18
XDS Business Language Reference Manual	90 10 22
XDS Sort/Merge Reference Manual	90 09 97
XDS 900 Series Utility and Debug Package (AID)	01 20 13

CONTENTS

APPENDIXES

ILLUSTRATIONS

TABLES

XDS 910 Computer

1. GENERAL DESCRIPTION

INTRODUCTION

The XDS 910, Figure 1, is a high-speed, low-cost, general-purpose, digital computer with the following characteristics:

- 24-bit word, plus parity bit
- Binary arithmetic
- Single-address instructions with

  Index Register
  Indirect Addressing
  Programmed Operators

- Basic core memory of 2048 or 4096 words, expandable to 16,384 words. All words are directly addressable with 8-microsecond cycle time
- 2048- and 4096-word memory modules available
- Typical execution times (including memory access and indexing) in microseconds:

  Fixed-Point Operations

Add	16
Multiply	248

  Floating-Point Operations

  24-bit Fraction (plus 9-bit exponent)

Add	432
Multiply	464

  39-bit Fraction (plus 9-bit exponent)

Add	896
Multiply	1696

- Program interchangeability with other XDS 900 Series Computers
- Parity checking of memory and I/O operations
- Priority Interrupt System

  Two standard XDS hardware interrupts; up to 38 more, optional

  Up to 896 optional special system interrupts

- Memory nonvolatile in event of power failure; optional power fail-safe feature permits saving contents of programmable registers
- Buffered input/output at rates in excess of 60,000 characters/second simultaneous with computation
- Standard input/output

  Display and manual control of internal registers

  Full-word input/output buffer

  The minimum 910 system includes either a photoelectric paper tape reader or a keyboard printer with paper tape reader and punch.

- Optional input/output devices

  Input/output typewriters

  Keyboard printer with paper tape reader and punch

  300-character/second paper tape readers, 60-character/second paper tape punches, paper tape spoolers

  MAGPAK Magnetic Tape Systems

  Magnetic tape units (IBM-compatible; binary and BCD), disc files

  Card readers, card punches, combination card reader/punch, line printers

  Off-line facility for printing directly from punched cards or magnetic tape

  Communications equipment, teletype consoles, display oscilloscopes, graph plotters

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

- MONARCH Monitor Routine, FORTRAN II Compiler, and META-SYMBOL Assembler, as part of complete software package
- All silicon semiconductors
- Operating temperature range: 10° to 55°C
- Dimensions (inches):

Double rack mounting:	65-1/2 x 48-1/4 x 25-1/2
Single rack mounting:	75-3/4 x 25-1/4 x 25-1/4

- Power: 110 v, 60 cps, 17 amp

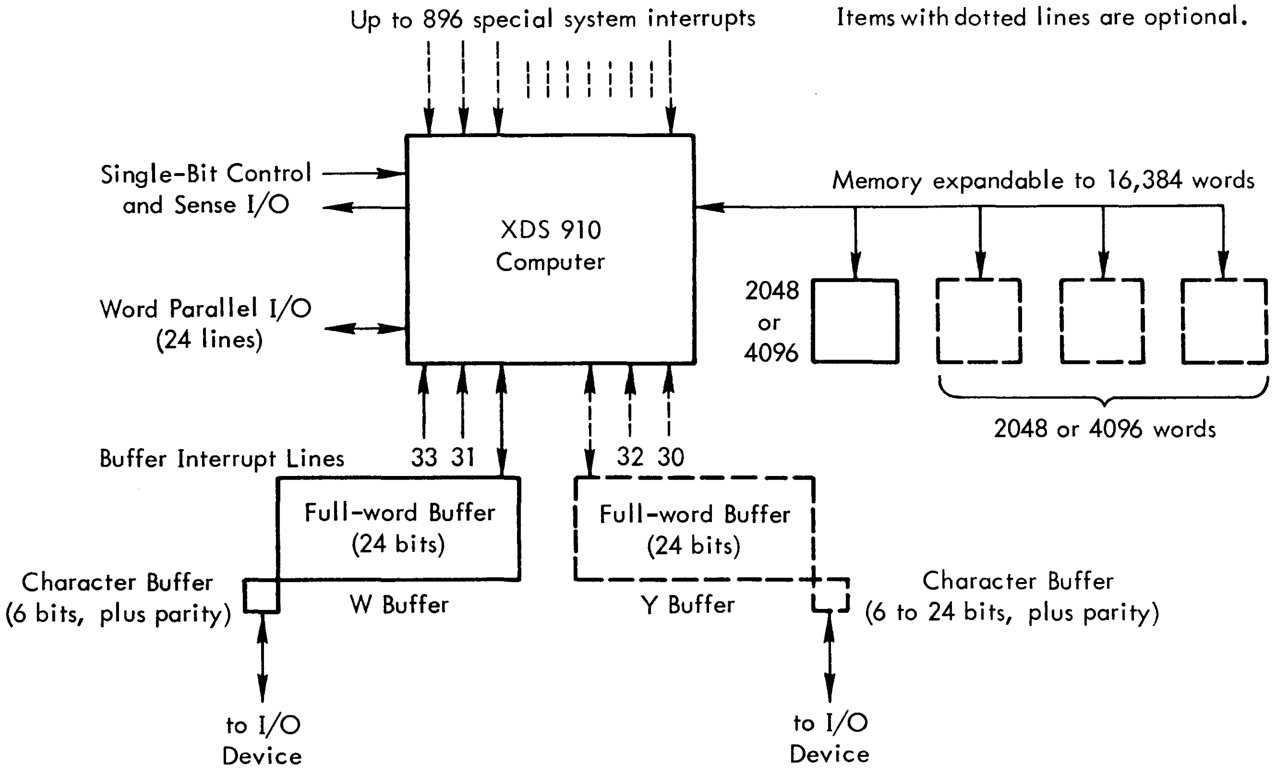

Up to 896 special system interrupts
Items with dotted lines are optional.
Single-Bit Control and Sense I/O
XDS 910 Computer
Word Parallel I/O (24 lines)
Memory expandable to 16,384 words
2048 or 4096
2048 or 4096 words
Buffer Interrupt Lines
33
31
32
30
Full-word Buffer (24 bits)
W Buffer
Character Buffer (6 bits, plus parity)
Full-word Buffer (24 bits)
Y Buffer
Character Buffer (6 to 24 bits, plus parity)
to I/O Device
to I/O Device

Figure 1. XDS 910 Computer Configuration

XDS 910 REGISTERS

The 910 Central Processor contains eight arithmetic and control registers. Four of the registers are available to the programmer and four are not.

REGISTERS AVAILABLE TO THE PROGRAMMER

The A, B, X, and P registers (see Figure 2, heavy lines) are available to the programmer for arithmetic, comparison, test, branch, and program control operations.

The 24-bit A register is the main accumulator for arithmetic operations.

The 24-bit B register is used as an extension of the A register. It contains the less significant portion of double-precision numbers.

The 24-bit X register is used to hold the index value in address modification. Indexing operations with the 14 least significant bits (address portion) of the X register, provide an indexing capability of up to 16,384 words.

The 14-bit P register (Program Counter) contains the memory address of an instruction before and during the time the instruction is being executed. Unless otherwise specified by the program (with a branch, skip, or EXECUTE instruction), the contents of the P register are incremented by 1 after each instruction is executed.

REGISTERS NOT AVAILABLE TO THE PROGRAMMER

The S, C, O, and M registers (see Figure 2, light lines) are not directly available to the programmer, but they are used by the 910 Central Processor to implement instruction execution.

The 14-bit S register contains the address of the memory location to be accessed for instructions or data.

The C register is a 24-bit arithmetic and control register. All instructions and data obtained from memory are brought into the C register for decoding. Address modification and parity generation/detection take place in the C register. Also, all input/output operations are routed through the C register.

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

The 25-bit M register (24-bit word, plus parity bit) contains each computer word as it is accessed from memory. Whenever memory is accessed, the contents of the M register are copied back into memory, thus assuring nondestructive readout of data and instructions.

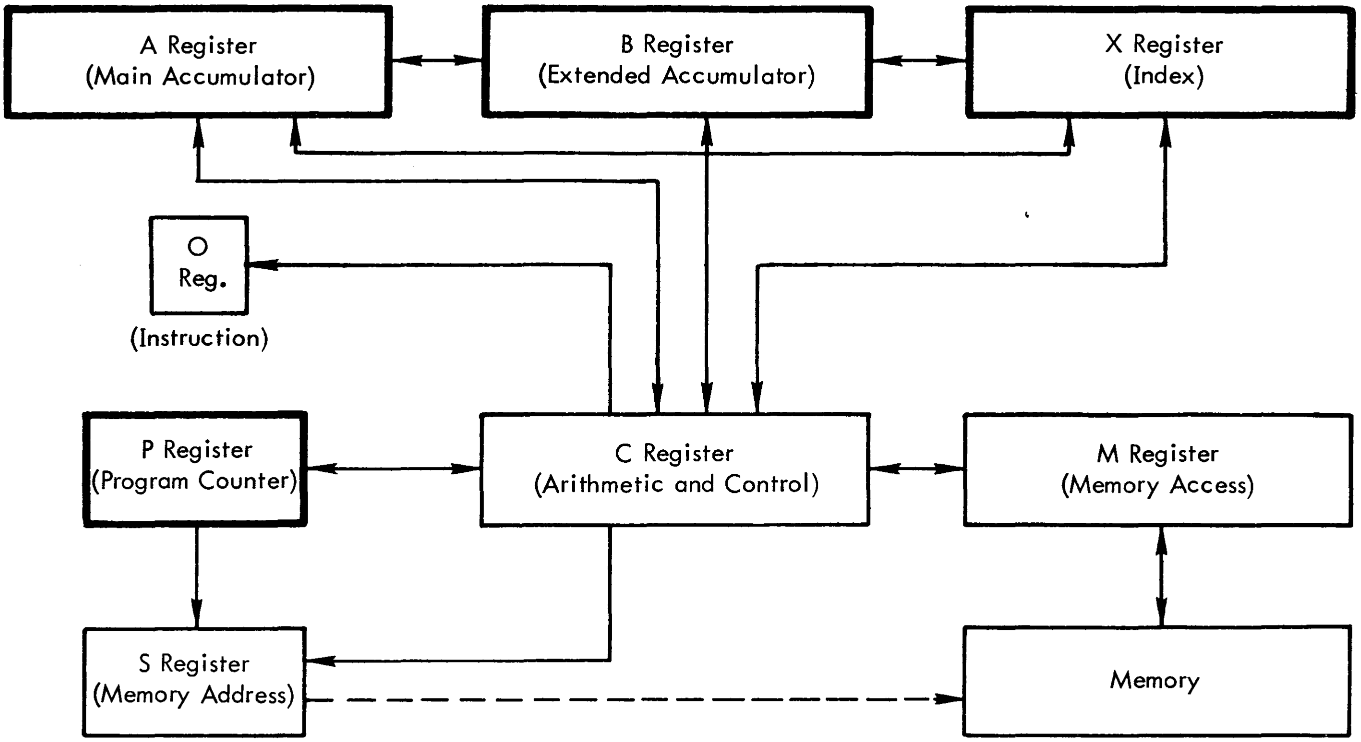


Figure 2. Basic Register Flow Diagram

XDS 910 MEMORY

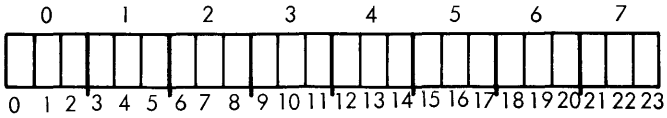
The basic XDS 910 memory consists of one random access, 2048- or 4096-word magnetic core module with a word size of 24 bits, plus parity. Additional 2048- or 4096-word memory modules are available. The Central Processor and the input/output buffers can directly address all memory. Addresses for memory words extend from octal location 00000 through 03777 (2K memory), 00000 through 07777 (4K memory), 00000 through 17777 (8K memory), or 00000 through 37777 (16K memory). The memory in a 16K system is a "wrap-around" or circular memory where the next location after 37777 is 00000. An attempt to read from a location whose address is not available causes zeros to be read. An attempt to store into such a location essentially results in a "no-op" operation, with the next instruction in sequence being executed. Thus, a program can use this property to determine the memory size of the machine within which it is operating.

Before accessing a memory word, the computer checks the power to ensure that the entire read/write cycle can be successfully completed. If it detects a power loss, the computer halts. Special logic (optional) may be included that prevents loss of information due to transient power failure or manual power shutoff.

The computer automatically generates even parity or checks for it during each read/write cycle. Setting a control panel parity switch causes the computer to halt automatically in case of parity error detection.

MEMORY WORD FORMATS

An XDS 910 Computer word is 24 binary digits (bits) long.

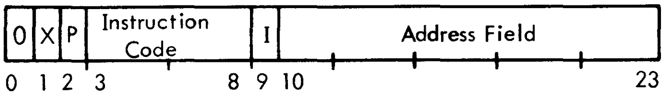


These bits are numbered (as shown above) from the left, or most significant end of the word, to the right, or least significant end of the word. All references to bit positions or bit numbers use this numbering scheme (e.g., bit 9 refers to bit position 9).

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. Octal digits are also numbered in the same general manner as individual bits, with octal 0 being the most significant digit and octal 7 the least. Octal 3, for example refers to bits 9, 10, and 11.

INSTRUCTION WORD FORMAT

The computer instruction word format is:



Bit position 0 is not used by the central processor decoding logic.

Bit position 1 contains the index register bit (X).

Bit positions 2 through 8 contain the instruction code field which determines the operation to be performed. The Programmed Operator feature in the 910 uses bit position 2; this bit position is also part of the "tag" field (bit positions 0 through 2).

Bit position 9 contains the indirect address bit (I).

Bit positions 10 through 23 contain the address field, which usually represents the location of the operand called for by the instruction code.

The following coding examples use standard META-SYMBOL format in expressing instructions. This format is

```
LDA    A,T
```

where:

LDA is a representative mnemonic instruction code,

A is a representative address, and T is a 1-digit octal integer that represents the tag field.

To express indirect addressing (that is, a 1-bit in the indirect address position), the programmer prefixes an asterisk to the address field:

```
LDA    *A,T
```

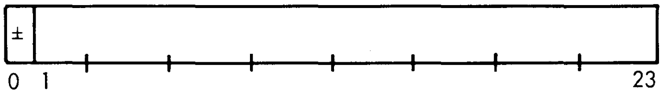
The interpretation of the tag field (bit positions 0 through 2) integer, T, is

Tag Field Integer T	Interpretation
0 (or blank)	No relative address, no index, no Programmed Operator
1	Programmed Operator
2	Index
3	Programmed Operator and Index
4	Relative address
5	Programmed Operator and relative address
6	Both relative address and index
7	Programmed Operator, index, and relative address

Three-letter Programmed Operator mnemonics (they have octal instruction codes 100-177) are usually used to denote Programmed Operators. The high-order 1-bit, in combination with tags of 0, 2, 4, 6, results in tags of 1, 3, 5, and 7, respectively. Programmed Operators are discussed further in this section under "Special Characteristics," and in Appendix E.

FIXED-POINT FORMAT

Fixed-point data words have the format



Numbers held in this format are 8-digit octal numbers, with the sign incorporated as the "leading bit," bit position 0, in the most significant octal digit. Thus, negative numbers have a 1 in bit position 0 and positive numbers have a 0 in bit position 0.

The memory holds fixed-point numbers as 23-bit fractions with an assumed binary point to the left of bit position 1. A full-word binary number has an equivalent precision of over six decimal digits. The range of values of a fixed-point number is from -1 to less than +1.

Programmers sometimes consider fixed-point numbers as integers, with the binary point to the right of bit position 23. The range of integer values is from -8,388,608 to +8,388,607 (-2^{23} to $+2^{23}-1$).

When performing computations with fixed-point numbers, the program must scale the values to keep them within the capacity of the computer registers, and align binary points so as to arrive at correct results.

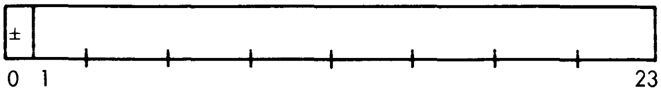
The memory holds negative, fixed-point numbers in two's complement form and the computer operates arithmetically on these numbers using a two's complement number system. See Appendix B for a discussion of two's complement arithmetic.

FLOATING-POINT FORMAT

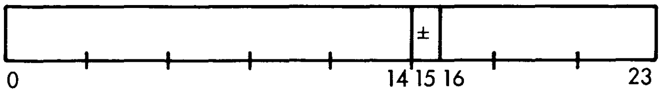
XDS offers standard Programmed Operator subroutines for performing double- and single-precision, floating-point arithmetic. Standard floating-point number formats are described below.

Double-Precision, Floating-Point Format

Most significant word



Least significant word



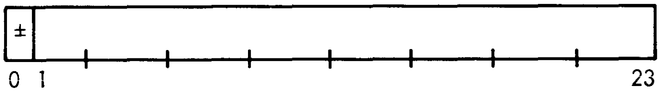
The fractional portion of a double-precision, floating-point number is a 39-bit proper fraction, with the leading bit being the sign bit and the assumed binary point being just to the left of the most significant magnitude bit (bit 1 of the upper word). The floating-point exponent is a 9-bit integer, with the leading bit being the sign. Standard routines operate on both fraction and exponent 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 x $2^{\pm E}$.

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

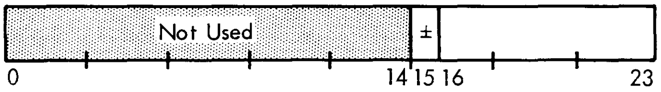
Standard Programmed Operators assume that the more significant word is in the A register, or stored in memory location M+ 1, and that the less significant word is in the B register, or stored in memory location M.

Single-Precision, Floating-Point Format

Fractional word



Exponent word



The fractional portion of a single-precision, floating-point number is a 24-bit proper fraction, with the leading bit being the sign and the assumed binary point being just to the left of the most significant magnitude bit. The floating-point exponent is a 9-bit integer with a leading sign bit. Standard routines operate on both fraction and exponent in two's complement form.

Single-precision, floating-point numbers have over six decimal digits of precision and a decimally equivalent exponent range of 10^{-77} to 10^{+77}.

Standard Programmed Operators assume that the fractional word is in A, or stored in memory location M+ 1, and that the exponent word is in B, or stored in memory location M. When entering a standard Programmed Operator routine, bits 0-14 of the exponent word are ignored.

SPECIAL CHARACTERISTICS

Certain computer features simplify programming and provide significant economies in memory utilization and program running time.

ADDRESS MODIFICATION

Address modification is accomplished through indexing and indirect addressing, used singly or in combination. In both indexing and indirect addressing, the computer performs address modification after bringing the instruction from memory but before executing it. The instruction remains in memory in its original form. The result of the address modification forms the "effective address" of the instruction operand.

Indexing

The computer contains an index (X) register for address modification. The use of this register to modify the address in an instruction does not increase instruction execution time.

If bit position 1 of an instruction contains a 1, the computer adds the contents of bits 10 through 23 of the X register to the contents of the address field of the instruction prior to execution. This addition does not retain any overflow or carry beyond the most significant address bit.

The computer's instruction set provides instructions for modifying and testing the X register and for transfering information between the X register and memory.

Indirect Addressing

The indirect address bit is in bit position 9 of the instruction. This bit position determines whether or not the computer uses indirect addressing with the instruction being executed.

A 0 in bit position 9 of an instruction causes the computer to use the contents of the address field (bit positions 10-23 of the instruction) as the 14-bit address requested by the instruction. A 1 in the index bit position causes the computer to add the contents of the X register to this address to form the effective address.

A 1 in bit position 9 of an instruction causes the computer to decode the contents of the location, accessed as described above, as if it were an instruction without an instruction code; that is, the computer's address logic reinitiates address decoding, using the word specified by the instruction.

For example, the instruction ADD 01000 causes the computer to obtain a word from location 01000 (assume it contains 00001005) and add it to the contents of the accumulator (A register). However, if the instruction

ADD *01000 is given, the computer obtains the word in location 01000, decodes the address it contains (01005), and adds the contents of location 01005 to the accumulator. If the word in location 01000 also has a 1 in the indirect address bit, the process of decoding is reiterated. Indirect addressing to as many levels as specified adds one cycle time to each instruction cycle time, for each level of indirect addressing performed.

If the instruction (or any subsequent word treated as an instruction) also calls for indexing, the contents of the index register are added to the address field of the instruction before indirect addressing occurs.

Examples: Indexing and Indirect Addressing

The octal instruction code for LOAD A register (LDA), used in the examples is 76. Parentheses denote "contents of."

Location	Contents	Effect
X register	00000001	
01000	00001001	
01001	00041002	
01002	00001003	
01003	00000002	
02000	0 76 01000	(1000) = 00001001 → A
02001	2 76 01000	(1000 + 1) = (1001) = 00041002 → A
02002	0 76 41000	((1000)) = (1001) = 00041002 → A
02003	2 76 41000	((1000 + 1)) = ((1001)) = (41002) = ((1002)) = (1003) = 00000002 → A

PROGRAMMED OPERATORS

Programmed Operators permit subroutines to be used in a program by giving a single "calling" instruction of the same mnemonic form as built-in machine instructions. The computer interprets the codes 0100–0177 as special instructions and transfers to a subroutine uniquely determined by each code. The computer records the return address at location 00000 so that program continuity is maintained. By means of indirect addressing through location 00000, the subroutine can gain access to the address of the calling instruction.

Programmed Operator subroutines are assigned three-letter, mnemonic designations in the same manner as built-in machine instructions described in Section 2. A program can use up to 64 Programmed Operators at any one time; however, since Programmed Operators are programmer-specified, the programmer can select alternate sets or subsets of the 64 Programmed Operators from program to program, or from section to section of the same program. The total number of Programmed Operators is without limit, but it is inconvenient to use more than 64 in one program. Other computers in the XDS 900 Series maintain compatibility among symbolic instructions through use of Programmed Operators. Mnemonic designations are identical in all computers. For example, while the designation "FLA" (for FLOATING ADD) refers to a built-in machine instruction in one computer, it may refer to a Programmed Operator subroutine in another. This technique preserves the one-to-one instruction relationship; programs written for one 900 Series Computer can be executed on any other computer in the series.

A more detailed discussion and a list of standard XDS Programmed Operator subroutines are in Appendix E.

OVERFLOW

An overflow detector in the computer makes it possible to recognize erroneous arithmetic operations that occur during the execution of a program. The OVERFLOW indicator on the control panel is set whenever any of the following conditions occur:

1. The result of an addition or subtraction cannot be contained within the A register.
2. A left-shift operation changes the contents of bit position 0 of the A register.
3. The MULTIPLY STEP instruction is executed with -1 in the effective memory location, 100 in bit positions 21 through 23 of the B register, and the contents of the A register divided by 2 is zero.

If the OVERFLOW indicator is set, it remains set until the appropriate reset instruction is executed. Section 2 contains instructions to reset or test and reset the state of the OVERFLOW indicator.

The only instruction affected by the state of the OVERFLOW indicator is OVERFLOW TEST (OVT), which skips if OVERFLOW is reset. Thus, if desired, the state of the OVERFLOW indicator can be ignored.

To determine whether a particular program instruction causes overflow, reset the OVERFLOW indicator before executing the instruction; then test the OVERFLOW indicator. An instruction that may be used to set overflow is RETURN BRANCH (BRR). The instruction BRR $, 4 (where $ is the location of the BRR) "branches" to the next location and sets the OVERFLOW indicator.

The execution of Programmed Operator, closed, and interrupt subroutines automatically preserves the status of the OVERFLOW indicator. In executing a Programmed Operator instruction, the computer automatically places the status of the OVERFLOW indicator in bit position 0 of location 00000 and resets the OVERFLOW indicator. The instruction MARK PLACE AND BRANCH (BRM) places the status of the OVERFLOW indicator in bit position 0 of the effective memory location and does not disturb the OVERFLOW indicator.

The instruction RETURN BRANCH (BRR) automatically merges the contents of the OVERFLOW indicator with the contents of bit position 0 of the effective memory location and places the result in the OVERFLOW indicator. Section 2 contains a description of the branch instructions.

SUBROUTINE EXECUTION

The XDS 910 Computer makes it possible to execute three kinds of subroutines:

1. Normal closed subroutine where the input parameters are specified in appropriate registers such as the A register
2. Interrupt subroutine that is entered as the result of an interrupt
3. Programmed Operator subroutine

A program enters a normal closed subroutine via a MARK PLACE AND BRANCH (BRM) instruction; BRM automatically stores the contents of the program counter (P register) and the status of the OVERFLOW indicator in the branch-to location. The P register value is normally the location of the BRM instruction. A RETURN BRANCH (BRR) instruction accomplishes the return to the main program; the BRR adds one to the stored P register value and transfers control to that location. See Section 2, Branch Group, for a description of the branch instructions.

Interrupt subroutines are closed subroutines, initiated by the detection of program-controlling interrupts, that automatically cause the appropriate interrupt subroutine to be entered. An interrupt causes normal program execution to be suspended and control to be transferred to a fixed location corresponding to that interrupt. The location normally contains a BRM instruction with the address of the interrupt servicing subroutine. When the BRM is executed, it automatically stores the current contents of the P register and the OVERFLOW indicator, in the branch-to location. The BRM then transfers control to the branch-to location+1. (When an interrupt occurs, the instruction process is completed, and control is transferred to the appropriate BRM without disturbing P. The value stored from P, therefore, is the address of the instruction to which program control should return after the interrupt is serviced by the interrupt subroutine). A BRANCH UNCONDITIONALLY (BRU) instruction with indirect addressing (through the branch-to location of the subroutine) returns control to the main program at the completion of the subroutine. BRU indirect also clears the interrupt from the active state. Note that this differs from the normal closed subroutine return that uses the BRR (stored P value + 1 ⟶ P).

2. MACHINE INSTRUCTIONS

INTRODUCTION

This section describes XDS 910 instructions in functional groups. Lists of instructions in functional, numerical, and alphabetical order are given in Appendix F, pages A-21, A-26, and A-29 respectively.

A diagram representing the format of the instruction accompanies the description of each instruction. Preceding each diagram is the mnemonic code and name that identifies the instruction. Within the diagram, the letter X in bit position 1 indicates that indexing can be used with the instruction, the letter I in bit position 9 indicates that indirect addressing can be used with the instruction, and the letter M in the address field indicates that the instruction obtains an operand from memory.

If bit position 1 of the instruction diagram contains a 0, indexing cannot be used with the instruction; if bit position 9 of the instruction diagram contains a 0, indirect addressing cannot be used with the instruction. Some instructions are shown with octal numbers in the address field; these instructions do not require an operand from memory, but use the address field to extend the operation code of the instruction.

The following statements apply to the instruction decriptions:

> Parentheses denote "contents of". For example, "(A)" means "contents of the A register."
>
> Subscripted characters identify inclusive bit positions. For example, "$(B)_{18-23}$" means "the contents of bit positions 18 through 23 of the B register.
>
> The contents of computer words and registers are expressed as octal-coded binary numbers; all other octal numbers used in this manual contain a leading zero, but decimal numbers do not. Thus, 0200 = 200_8 and 200 = 200_{10}.

The term "effective memory location" refers to the location in memory from which the operand is taken at the conclusion of all indirect addressing and indexing. This term is sometimes shortened to "effective location." It is the location whose address is the effective address. The term "effective operand" means the contents of the effective memory location.

The term "set" means "place a 1-bit in the contents of" a computer word, or "turn on" an indicator. "Reset" means "place a zero in the contents of" a computer word, or "turn off" an indicator, or "clear to zero".

The interrupt system can interrupt the program at the end of any instruction, except INCREMENT INDEX AND BRANCH (BRX) and ENERGIZE OUTPUT M (EOM).

Instruction timing is given in terms of memory cycles, where each cycle is 8 microseconds, including the time required for fetching the instruction and all operands. Indexing does not change the timing of any instruction, but each level of indirect addressing used adds one additional memory cycle to the instruction timing given.

LOAD/STORE INSTRUCTIONS

LDA LOAD A

0	X	0	76	I	M
0	1	2	3 8	9	10 23

LDA loads the contents of the effective memory location into the A register; the contents of the effective memory location are not affected.

Affected: (A) Timing: 2

STA STORE A

0	X	0	35	I	M
0	1	2	3 8	9	10 23

STA stores the contents of the A register in the effective memory location; the contents of the A register are not affected.

Affected: (M) Timing: 3

LDB LOAD B

0	X	0	75	I	M
0	1	2	3 8	9	10 23

LDB loads the contents of the effective memory location into the B register; the contents of the effective memory location are not affected.

Affected: (B) Timing: 2

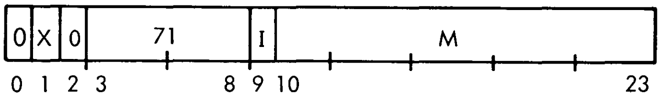
STB STORE B

0	X	0	36	I	M
0	1	2	3 8	9	10 23

STB stores the contents of the B register in the effective memory location; the contents of the B register are not affected:

Affected: (M) Timing: 3

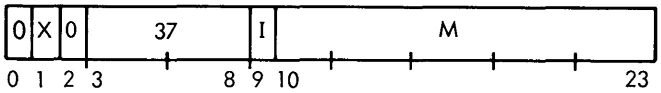
LDX LOAD INDEX



LDX loads the entire 24-bit contents of the effective memory location into the index register; the contents of the effective memory location are not affected.

Affected: (X) Timing: 2

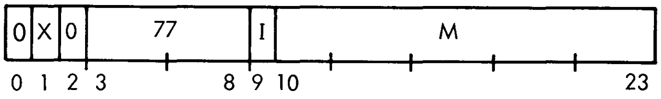
STX STORE INDEX



STX stores the entire 24-bit contents of the index register in the effective memory location; the contents of the index register are not affected.

Affected: (M) Timing: 3

EAX COPY EFFECTIVE ADDRESS INTO INDEX REGISTER



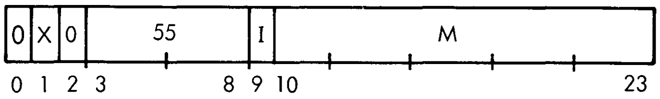
EAX copies the address of the effective memory location into bit positions 10 - 23 of the index (X) register; the ten most significant bits of the X register and the contents of the effective memory location are not affected.

The addressing process for this instruction operates as in a load instruction, except that instead of obtaining the contents of the effective .memory location, the effective memory address is the operand. For example, if EAX is executed with zeros in bit positions 1 and 9, the actual bit configuration in the address field of EAX is copied into bit positions 10 - 23 of the X register.

Affected: $(X)_{10-23}$ Timing: 2

ARITHMETIC INSTRUCTIONS

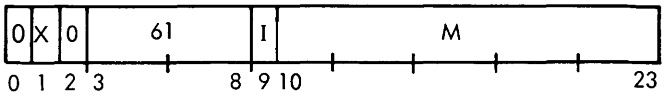
ADD ADD MEMORY TO A



This instruction adds the contents of the effective memory location to the contents of the A register and places the result in A. If both numbers are of the same sign but the sign of the result in the A register is opposite, overflow has occurred and the computer has set the OVERFLOW indicator.

Affected: (A), Of Timing: 2

MIN MEMORY INCREMENT

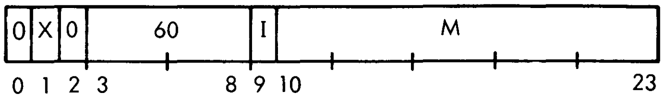


MIN increases the contents of the effective memory location by one, and places the resulting sum in the same location. The contents of the A register do not change.

Overflow occurs only when the contents of M are 37777777 before execution. In this case, 40000000 is the result in M.

Affected: (M), Of Timing: 3

MDE MEMORY DECREMENT

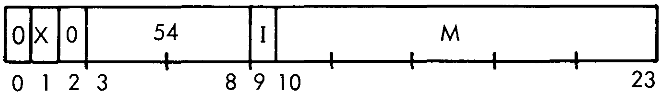


MDE decreases the contents of the effective memory location by one and places the resulting difference in the same location. The contents of the A register do not change.

An overflow occurs if the initial contents of memory are 40000000. The result inmemory in this case is 37777777.

Affected: (M), Of Timing: 3

SUB SUBTRACT MEMORY FROM A

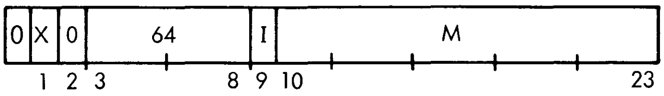


SUB subtracts the contents of the effective memory location from the A register and places the result in the A register.

If both numbers are of the same sign after the contents of the effective address have been complemented for addition but the sign of the result in the A register is opposite, an overflow has occurred and the computer has set the OVERFLOW indicator.

Affected: (A), Of Timing: 2

MUS MULTIPLY STEP



The sign of A temporarily extends two bit positions to the left if the OVERFLOW indicator is reset. If the OVERFLOW indicator is set, the two bits extended are zeros. Then the contents of the memory location determined by the effective address are added to or subtracted from the A register, based on the contents of the three low-order bits of the B register. The arithmetic operation performed takes place according to the following table:

B_{21}	B_{22}	B_{23}	Arithmetic Operation
0	0	0	None
0	0	1	(A) + 2(M) ⟶ A
0	1	0	(A) + 2(M) ⟶ A
0	1	1	(A) + 4(M) ⟶ A
1	0	0	(A) - 4(M) ⟶ A
1	0	1	(A) - 2(M) ⟶ A
1	1	0	(A) - 2(M) ⟶ A
1	1	1	None

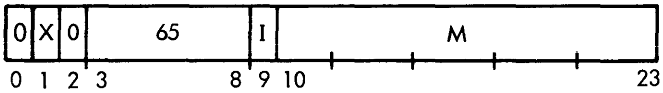
The computer then shifts the result in the double-length AB register two bit positions to the right.

The OVERFLOW indicator is set if (M) is -1, the contents of B_{21-23} were 100, and (A)/2 was originally zero. Otherwise, the OVERFLOW indicator is reset. Various multiply subroutines (such as Programmed Operators) use this instruction. Twelve MUS instructions can be repeated to provide a complete multiplication of the form (M) x (B) ⟶ AB. Prior to execution of the first step, the multiplier must be in the B register, the A register cleared, the double-length AB register shifted left one, and the OVERFLOW indicator turned off.

Affected: (AB), Of Timing: 2

The Programmed Operator subroutine MULTIPLY (MUL) requires 248 μsec for a full multiplication.

DIS DIVIDE STEP



DIS shifts the contents of the double-length AB register left one bit position and copies the complement of A_0 into B_{23}. If $(A_0) = (M_0)$, the contents of the memory location determined by the effective address are subtracted from the A register. If $(A_0) \neq (M_0)$, the contents of the memory location determined by the effective address are added to the A register.

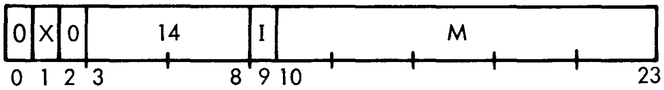
The Programmed Operator divide subroutines use this instruction.

Affected: (AB) Timing: 2

The Programmed Operator subroutine DIVIDE (DIV) requires 888 μsec for a full division. The subroutine provides a corrected remainder of the same sign as the original A register.

LOGICAL INSTRUCTIONS

ETR EXTRACT



ETR performs a logical "AND" between corresponding bits of the A register and the effective memory location and places the result in A. This instruction performs the operation bit by corresponding bit according to the following:

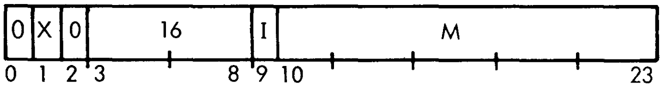
(A)	(M)	Result in A
0	0	0
0	1	0
1	0	0
1	1	1

Affected: (A) Timing: 2

Example: ETR M

	Before Execution	After Execution
(A) =	64231567	00231400
(M) =	00777600	00777600

MRG MERGE



MRG performs a logical "Inclusive OR" between corresponding bits of the A register and the effective memory location and places the result in A. This instruction performs the operation, bit by corresponding bit, as follows:

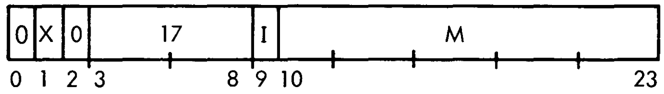
(A)	(M)	Result in A
0	0	0
0	1	1
1	0	1
1	1	1

Affected: (A) Timing: 2

Example: MRG M

	Before Execution	After Execution
(A) =	06446254	06746756
(M) =	02340712	02340712

EOR EXCLUSIVE OR



EOR performs a logical "Exclusive OR" between corresponding bits of the A register and the effective memory location and places the result in A. This instruction performs the operation bit by corresponding bit, as follows:

(A)	(M)	Result in A
0	0	0
0	1	1
1	0	1
1	1	0

Affected: (A) Timing: 2

Example: EOR M

	Before Execution	After Execution
(A) =	34165031	44112010
(M) =	70077021	70077021

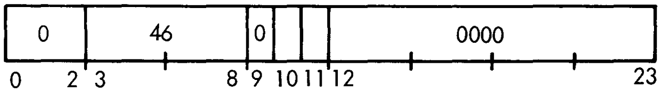
The proper memory word configuration logically inverts selected bit positions of the A register. If all "ones" appear in the memory word, a one's complement of A results.

Example: EOR M

	Before Execution	After Execution
(A) =	10357211	67420566
(M) =	77777777	77777777

REGISTER CHANGE INSTRUCTIONS

RCH REGISTER CHANGE



RCH performs the following operations upon the contents of the A and B registers, depending on the values of bit positions 10 and 11 of the instruction word:

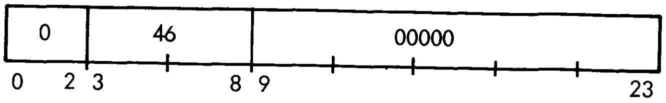
10	11	Function
0	0	Exchange A and B (XAB)
0	1	Copy B into A, clear B (BAC)
1	0	Copy A into B, clear A (ABC)
1	1	Clear A and B (CLR)

Indirect addressing and indexing do not apply to register change instructions.

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

XAB EXCHANGE A AND B

RCH 0

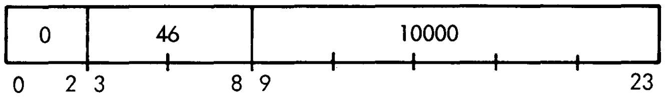


XAB copies the contents of the A register into the B register and simultaneously copies the contents of the B register into the A register.

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

BAC COPY B INTO A, CLEAR B

RCH 010000

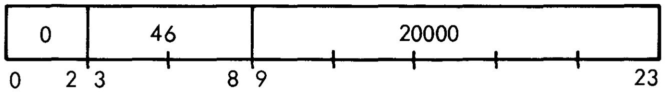


BAC copies the contents of the B register into the A register and simultaneously clears the B register to zero.

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

ABC COPY A INTO B, CLEAR A

RCH 020000

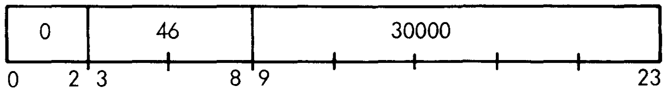


ABC copies the contents of the A register into the B register and simultaneously clears the A register to zero.

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

CLR CLEAR AB

RCH 030000



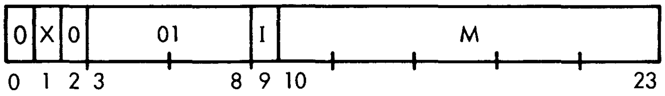
CLR clears the contents of both the A and B registers to zero.

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

BRANCH INSTRUCTIONS

Branch instructions conditionally or unconditionally change the course of the program by altering the contents of the program counter (P register). The programmer should note that these instructions branch to locations determined by the effective address; this means that the branch can operate with all levels of indirect and indexed addressing.

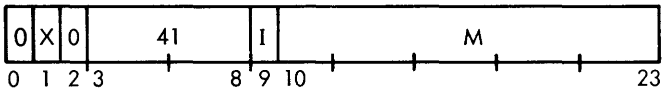
BRU BRANCH UNCONDITIONALLY



BRU takes the next instruction from the location determined by the effective address. A BRU instruction with an indirect address bit equal to 1 clears the highest priority interrupt level then active, in addition to branching to the effective location.

Affected: (P) Timing: 1

BRX INCREMENT INDEX AND BRANCH



BRX increments the contents of the entire X register by 1. If the resultant X register value contains a 1-bit in bit position 9, the computer transfers control to the effective location; if not, it takes the next instruction in sequence.

If a BRX instruction is indexed, any transfer of control is to the effective address determined by the value of the index before it is incremented. However, the test for transfer is based on the incremented value of the X register, just as if the BRX instruction were not indexed.

The 9 most significant bits of the X register (bits 0 through 8) have no effect on the execution of the instruction, but may be affected by it.

If a branch occurs, an interrupt cannot occur following the execution of this instruction.

Affected: (X), (P) Timing: 1 if branch
2 if no branch

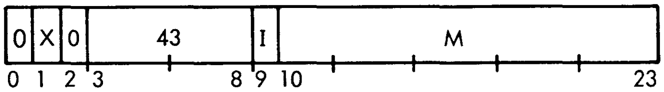
Example:

Location	Contents	Instruction	(X register)
00777	2 35 01500	STA 01500, 2	77777776
01000	0 41 01006	BRX 01006	77777777
01001	2 76 02000	LDA 02000, 2	77777777
01006	0 41 01001	BRX 01001	00000000
01007	2 76 02100	LDA 02100, 2	00000000

The execution of these instructions is in the following order, as given by their locations:

00777
01000
01006
01007

BRM MARK PLACE AND BRANCH



BRM is normally used to enter subroutines where a return to the main program is desired after the subroutine has been completed.

BRM stores the contents of the P register (normally the address of the BRM instruction itself) in the effective memory location (subroutine entry location) and transfers control to that location plus one (first instruction of subroutine). BRM also stores the status of the OVERFLOW indicator in bit 0 of the effective location. The contents of bits 1-9 of the effective location are cleared to zeros.

When a BRM stored in an interrupt location is executed (as the result of an interrupt) P contains the location of the next program instruction that would have been executed if the interrupt had not intervened. It is this "return location" instead of the BRM's own location that is stored in this instance. Information about the interrupt system is given in Section 3.

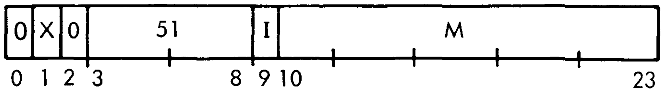
Affected: (M), (P) Timing: 2

Example: BRM 0522

Location	Contents
01517	0 43 00522

	OVERFLOW Indicator	Location 0522	P Register
Before execution:	1 (on)	---	01517
After execution:	1 (on)	40001517	00523

BRR RETURN BRANCH



BRR is normally used to return to the main program after completion of a subroutine in conjunction with MARK PLACE AND BRANCH (BRM) except in interrupt subroutines (see Section 3).

BRR copies the contents of the effective memory location (subroutine entry location) into an internal register and increments the contents by one. The instruction then stores the least significant 14 bits of the result in the P register. (The P register contains the address of the next instruction to be executed.) It also performs a logical OR between bit 0 of the effective memory location and the OVERFLOW indicator and places the result in the OVERFLOW indicator. There is no change in the contents of the effective memory location.

Affected: Of, (P) Timing: 2

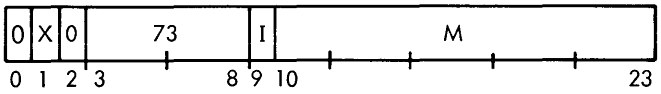
Example: BRR 02000

Location	Contents
02100	0 51 02000
02000	0 00 03220

If the computer executes the instruction in location 02100, it takes the next instruction from location 03221. Location 02000 still contains 0 00 03220.

TEST AND SKIP INSTRUCTIONS

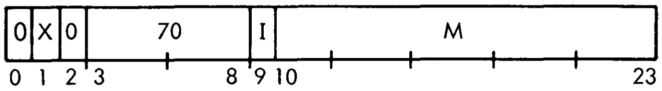
SKG SKIP IF A GREATER THAN MEMORY



SKG algebraically compares the contents of the A register with the contents of the effective memory location. If the contents of A are greater than the contents of the effective location, the computer skips the next instruction in sequence and executes the following instruction. If the contents of A are less than or equal to the contents of the effective location, the computer executes the next instruction in sequence. SKG alters neither A nor memory.

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

SKM SKIP IF A EQUALS MEMORY ON B MASK



SKM compares designated bit positions of the A register with corresponding bit positions in the effective memory location. If the specified bits in A are identical to those in the effective memory location, the computer skips the next instruction in sequence after SKM and executes the following instruction. If the specified bits are not identical, the computer executes the next instruction in sequence after SKM.

The programmer selects the bit positions to be compared by placing 1-bits in the corresponding bit positions of the B register and 0-bits in the remaining bit positions of B.

SKM considers the contents of A, B, and the effective location to be unsigned, 24-bit, nonnumeric quantities, and does not alter them.

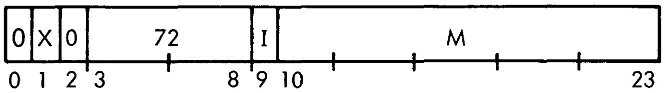
Affected: (P) Timing: 2 if no skip
3 if skip

Example: SKM M

(A)	(B)	(M)
00043007	00177000	57643240

Since SKM compares bit positions 8-14 only (as determined by (B), and (A) = (M) in these positions, a skip occurs. Note that if (B) = 0, a skip occurs regardless of (A) and (M).

SKA SKIP IF A AND MEMORY DO NOT COMPARE ONES



SKA compares the contents of the A register, bit by bit, with the contents of the effective memory location. If the A register and the effective location do not both have 1-bits in any corresponding bit positions, the computer skips the next instruction in sequence after SKA and executes the following instruction. If the A register and the effective location do have at least one pair of 1-bits in corresponding bit positions, the computer executes the next instruction in sequence after SKA.

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

(A)	(M)	Result
0	0	0
0	1	0
1	0	0
1	1	1

If the result produces a 1-bit in any bit position, a skip does not occur.

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

Different configurations of the memory word result in a wide variety of conditional skip 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

Contents of A Register	Effect
40000000	Skip if Memory is Positive
77777777	Skip if Memory = 0
00000001	Skip if Memory is Even

SKN SKIP IF MEMORY NEGATIVE

```
|0|X|0|     53     |I|            M             |
 0 1 2 3          8 9 10                       23
```

If the contents of the effective memory location are negative, i.e., if $(M_0) = 1$, the computer skips the next instruction in sequence after SKN and executes the following instruction. If the contents of the effective location are positive or zero, the computer executes the next instruction in sequence after SKN.

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

SHIFT INSTRUCTIONS

The shift instructions operate on the contents of the A and B registers and offer a complete facility for right and left shifting, cycling, and normalizing the contents of these two registers. The A and B registers, in combination, form a double-length register whose double-length contents can be shifted, cycled, or normalized. This double-length register is named "AB".

When the contents of the AB register shift right, bits from bit position 23 of the A register shift into bit position 0 of the B register. When the AB register shifts left, bits from bit position 0 of the B register shift into bit position 23 of the A register.

Two shift instructions allow the 48-bit contents of the AB register to be "cycled" right or left. When the contents of the AB register cycle, the bits that shift from one end of the one register copy into the other end of the other register.

Shift instructions use the instruction code to determine the direction of shift (66 = right; 67 = left); bits 10 and 11 of the effective address determine the method of shifting as follows:

Octal Position	(Bits 10, 11)	Octal Value	Function
	00	0	AB Shift
3	10	2	AB Cycle
	01	1	Normalize (left only)

Indexing of a direct address shift instruction affects only bits 18-23 of the address field. It is thus possible to index the number of shifts without affecting bits 10 and 11, which control the method of shifting. During indirect addressing, the full 14 bits of the address field are used in the address computation; thus, only the shift instructions RSH and LSH should be indirectly addressed, with bits 10 and 11 of the effective address determining the method of shifting.

When the computer interprets a shift instruction, bit positions 18 through 23 of the effective address of the instruction determine the amount of the shift. The computer treats these six bits as an unsigned count. If the initial count is equal to zero, no shifting occurs. Once the shift begins, the count is reduced by one for each position shifted until it reaches zero. The count C in the following instructions indicates the number of places to be shifted.

Shift timing is calculated as follows, where N is the number of places shifted.

Timing in Cycles	Number of Places Shifted
2+N	N=0, 0, 1, 2, 3, ... 48

RSH RIGHT SHIFT AB

```
|0|X|0|     66     |I|0|0|     0      |     C      |
 0 1 2 3          8 9 10 11 12       17 18         23
```

RSH shifts the contents of the AB register right the number of places specified in bits 18 through 23 of the effective address. The bit in the sign position of A does not shift; its value is copied into the vacated bit positions of the shifted number. The bit in the sign position of B shifts. Bits shifted out of A_{23} shift into B_0. Bits shifting past position B_{23} are lost.

This instruction may be used to perform scaling of floating-point numbers by use of indexing, where the difference of exponents is in the X register as a positive quantity.

Affected: (AB) Timing: 2 + N

Example: RSH 18 (0 66 00022)

		Before Execution	After Execution
(A)	=	45261237	77777745
(B)	=	27651260	26123727

RCY RIGHT CYCLE AB

```
|0|X|0|     66     |0|1|0|     0      |     C      |
 0 1 2 3          8 9 10 11 12       17 18         23
```

RCY shifts the contents of the AB register right the number of places specified in bits 18 through 23 of the effective address. The bit in the sign position of B shifts like any other bit in B. Bits shifting out of A_{23} shift into B_0; bits from B_{23} go into A_0. Thus, the computer treats the double-length register as if it were circular and cycles it onto itself so that no bits are lost.

Affected: (AB) Timing: 2 + N

Example: RCY 15 (0 66 20017)

		Before Execution	After Execution
(A)	=	76543210	34567765
(B)	=	01234567	43210012

LSH LEFT SHIFT AB

0	X	0	67	I	0	0	0	C
0	1	2	3 8	9	10	11	12 17	18 23

LSH shifts the contents of the AB register left the number of places specified in bits 18 through 23 of the effective address. Bits shift left through the sign position of A, but when a bit, different in value from the original sign, shifts into the sign position, the computer sets the OVERFLOW indicator. Bits shifting out of B_0 go into A_{23}. Bits shifting past position 0 in A are lost. Zeros fill the vacated bit positions on the right end of the B register.

Affected: (AB), Of Timing: 2 + N

Example: LSH 18 (0 67 00022)

		Before Execution	After Execution
(A)	=	46712370	70641327
(B)	=	64132711	11000000

LCY LEFT CYCLE AB

0	X	0	67	0	0	0	0	C
0	1	2	3 8	9	10	11	12 17	18 23

LCY shifts the contents of the AB register left the number of places specified in bits 18 through 23 of the effective address. Bits in the sign positions of A and B shift like any other bits in the number. Bits shifting out of B_0 shift into A_{23}; bits shifting out of A_0 shift into B_{23}. The computer treats the double-length register as if it were circular and cycles it onto itself. It loses no bits.

Affected: (AB) Timing: 2 + N

Example: LCY 9 (0 67 20011)

		Before Execution	After Execution
(A)	=	76543210	43210012
(B)	=	01234567	34567765

NOD NORMALIZE AND DECREMENT

0	X	0	67	0	0	1	00	C
0	1	2	3 8	9	10	11	12 17	18 23

NOD shifts the contents of the AB register left until (1) a bit appears in position 1 of A that is not equal to the bit in the sign position of A, or (2) until C shifts occur. The computer keeps count of the number of places shifted by decrementing the contents of the X register each time a shift occurs. If, in the attempt to normalize, shifting exceeds 48 places, the contents of the AB register were initially zero. In this case, the computer continues shifting until the shift count C reduces to zero. Zeros fill the vacated positions of AB.

The number C, in address bit positions 18 through 23 of the instruction, is an upper limit for the number of left shifts. The programmer must ensure that C is sufficiently large to permit a complete normalization.

Affected: (AB), (X) Timing: 2 + R where R is the resultant number of shifts

Example: NOD 24 (0 67 10030)

		Before Execution	After Execution
(A)	=	00004632	23153705
(B)	=	76124035	20164000
(X)	=	00000000	77777765

CONTROL INSTRUCTIONS

HLT HALT

0	X	0	00	I	M
0	1	2	3 8	9	10 23

When the computer executes this instruction, it halts computation and lights the HALT indicator in the console. Before halting, the computer increments the P register and brings the next instruction into the C register to be displayed. To resume automatic computation, the operator must set the RUN-IDLE-STEP switch to IDLE, then back to RUN.

The computer then executes the next instruction, according to the P register.

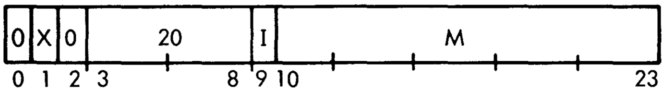
Indirect addressing and indexing do not apply to this instruction, nor does the instruction access memory.

When the computer executes HLT, all internal computation ceases at the end of the present instruction being executed. If an input/output operation is in progress, it continues until completed. Computation automatically resumes with the occurrence of a program interrupt, if the RUN-IDLE-STEP switch is still in the RUN position and the interrupt system is enabled.

The HALT light turns off when the RUN-IDLE-STEP switch is set to IDLE, or when an interrupt occurs.

Affected: Halt flip-flop Timing: 1

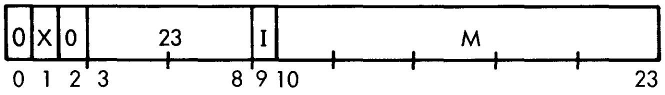
NOP NO OPERATION



Executing NOP does not affect the A register, B register, X register, or memory. Indirect addressing and indexing do not apply to this instruction, nor does the instruction access memory.

Affected: None Timing: 1

EXU EXECUTE



EXU 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 computer executes the contents of the effective location and then executes the next instruction in sequence following the EXU.

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

If the contents of the effective memory location are a skip instruction, then, depending on the skip decision, program control returns 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 repeats, with the normal return being the initial EXU location plus one. This process can cascade indefinitely. (See Figure 9 in Appendix D.)

Affected: Determined by executed instruction Timing: 1 + executed instruction

BREAKPOINT TESTS

This instruction tests the status of the BREAKPOINT switches singly or in any combination. If a tested BREAKPOINT switch is reset (off), the computer skips the next instruction in sequence and executes the following instruction. If the tested BREAKPOINT switch is set (on), the computer executes the next instruction in sequence.

Mnemonic	Name of Instruction	Octal Configuration
BPT 1 (SKS 020400)	BREAKPOINT 1 Test	0 40 20400
BPT 2 (SKS 020200)	BREAKPOINT 2 Test	0 40 20200
BPT 3 (SKS 020100)	BREAKPOINT 3 Test	0 40 20100
BPT 4 (SKS 020040)	BREAKPOINT 4 Test	0 40 20040

If more than one BREAKPOINT switch is specified in the test, the computer skips the next instruction if any of the specified switches are reset, but does not skip the next instruction if all of the specified switches are set. Thus, the instruction BPT 1,3 (0 40 20500) causes the computer to skip the next instruction unless switches 1 and 3 are both set (2 and 4 are ignored in this case).

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

OVERFLOW INSTRUCTIONS

OVT OVERFLOW INDICATOR TEST AND RESET
(SKS 020001) 0 40 20001

This instruction tests the status of the OVERFLOW indicator. If the indicator is on, the computer executes the next instruction in sequence, and turns the indicator off (clears to zero). If the indicator is off, the computer skips the next instruction in sequence and executes the following instruction.

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

ROV RESET OVERFLOW
(EOM 020001) 0 02 20001

ROV unconditionally resets the OVERFLOW indicator (clears to zero).

Affected: Of Timing: 1

FLOATING-POINT OPERATIONS

XDS Programmed Operator subroutines perform in either single or double precision modes. Double precision operation permits results with an accuracy of approximately 11 decimal digits. Single precision operation permits faster execution times, with approximately seven decimal digits of accuracy.

The standard XDS Programmed Operators assume that the most significant word is in A, or stored in location M + 1, while the less significant word is in B, or memory location M; see Section 1, Floating-Point Format.

DOUBLE-PRECISION, FLOATING-POINT OPERATIONS

Programmed Operators that perform double-precision, floating-point operations use a fractional number of 39 bits (38 bits plus sign) and an exponent of nine bits (eight plus sign). Numbers have the fraction equal to 11 decimal digits plus sign and the multiplier as high as $10^{\pm 77}$.

The Programmed Operator subroutines that perform double-precision, floating-point operations are:

Mnemonic	Name	Function	Approximate Execution Time
FLA	Floating Add	Floating (A, B) + (M + 1, M) ⟶ A, B	896 μsec
FLS	Floating Subtract	Floating (A, B) - (M + 1, M) ⟶ A, B	1016 μsec
FLM	Floating Multiply	Floating (A, B) x (M + 1, M) ⟶ A, B	1696 μsec
FLD	Floating Divide	Floating (A, B) ÷ (M + 1, M) ⟶ A, B	1872 μsec

SINGLE PRECISION, FLOATING-POINT OPERATIONS

Programmed Operators that perform single precision, floating-point operations use a fractional number of 24 bits (23 bits plus sign) and an exponent of nine bits (eight bits plus sign). Numbers have the fraction equal to six decimal digits plus sign and the exponent as high as $10^{\pm 77}$.

The Programmed Operator subroutines that perform single precision, floating-point operations are:

Mnemonic	Name	Function	Approximate Execution Time
FSA	Floating Add, Single-Precision	Floating (A) + (M + 1) ⟶ A Exponent in B, M	432 μsec
FSS	Floating Subtract, Single-Precision	Floating (A) - (M + 1) ⟶ A Exponent in B, M	472 μsec
FSM	Floating Multiply, Single-Precision	Floating (A) x (M + 1) ⟶ A Exponent in B, M	464 μsec
FSD	Floating Divide, Single Precision	Floating (A) ÷ (M + 1) ⟶ A Exponent in B, M	792 μsec

See Appendix E for a detailed discussion of the Programmed Operator feature.

3. INTERRUPT SYSTEM

XDS 900 Series Computers contain a priority interrupt system that provides added program control of input/output operations, aids in programming simultaneous input/output and compute operations, and also 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 W buffer, real-time clock, power fail-safe can cause the interruption of programs being executed by the computer by transmitting interrupt pulses (such as end-of-word signals or clock pulses) to interrupt levels in the computer. Appendix D contains a diagram of a portion of the XDS 910 Interrupt System.

PRIORITY ASSIGNMENT

All interrupt devices used with a specific computer installation are assigned unique, numbered priority levels (see Table 1) identified by octal numbers from 030 through 077 and 0200 through 01777, with the higher-priority interrupt levels having a smaller number (except for the optional power fail-safe interrupt levels, which have the highest priority). Interrupt levels 030-077 are XDS optional hardware interrupt levels, are normally reserved for up to 40 special-purpose interrupt devices, and are always added in pairs. The W buffer End-of-Word and End-of-Transmission interrupts (levels 031 and 033) are a standard feature of the 910 Computer; all other interrupt levels are optional, and are added according to the requirements of individual installations. The optional levels 0200-01777 are special systems interrupt levels, which may be added in any number up to 896, for general-purpose interrupts.

INTERRUPT LEVEL OPERATION

Each interrupt level (as shown in Figure 3) has three distinct operating states – INACTIVE, WAITING, and ACTIVE. In the inactive state, the level has not received a pulse from its assigned interrupt device. When the pulse is received and the level is armed, the Waiting flip-flop (ff) is unconditionally set to produce a steady signal; the waiting state begins. If no higher-priority level is in the active state, the signal from the Waiting ff sets the interrupt ff. At the end of the execution cycle of the instruction currently being executed, the interrupt is acknowledged by the computer, and the instruction in the memory location with the same octal number as the interrupt level (e.g., location 036) is executed without affecting the program counter. Normally, when the instruction in the interrupt address is executed, the level Active ff is set; the active state begins. The Interrupt ff is also cleared at this time, and all lower-priority interrupt levels are prevented from becoming active. (This allows interrupt levels to be arranged in the order of their importance and/or need for servicing.)

SUBROUTINE INTERRUPT

If the interrupt level is a "subroutine" interrupt, the instruction in the interrupt address is normally a BRANCH AND MARK PLACE (BRM) instruction to a servicing subroutine which ends in a BRANCH UNCONDITIONALLY (BRU) instruction, indirectly 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 interrupt instruction) in the first location of the servicing subroutine.

Table 1. Interrupt Levels (Arranged in Order of Priority)

Address	Description	Address	Description
030-077	XDS HARDWARE INTERRUPT LEVELS	034-035 040-077	Other optional, Special-purpose interrupts
036	POWER ON (Optional) Always armed and always enabled	0200-01777	SPECIAL SYSTEM INTERRUPTS (Optional) Enabled by EIR. Always armed, or may be armed selectively by AIR (provided that the Arm Interrupts Control Unit is present as a part of the computer)
037	POWER OFF (Optional) Always armed and always enabled	0200-0217	Group 1 (00 in control word address field)
030	Y BUFFER END-OF-WORD (Optional) Armed if enabled by EIR	0220-0237	Group 2 (01 in control word address field)
031	W BUFFER END-OF-WORD (Standard) Armed if enabled by EIR	. . .	
032	Y BUFFER END-OF-TRANSMISSION (Optional) Armed if enabled by EIR		
033	W BUFFER END-OF-TRANSMISSION (Standard) Armed if enabled by EIR	01760-01777	Group 56 (67 in control word address field)

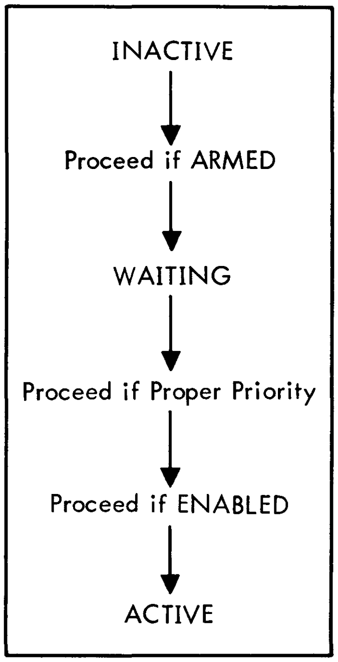


Figure 3. Interrupt Arm-Enable Response

The BRU instruction, indirectly addressed, returns program control to the next instruction in sequence in the interrupted program, and clears the interrupt level (resets both the level Waiting ff and the level Active ff). The interrupt level is now back in the inactive state. An interrupt-servicing subroutine is also interrupted whenever a higher-priority interrupt level becomes active. This process may be repeated indefinitely and, as each subroutine is processed and its interrupt level cleared, program control is returned to the subroutine interrupted by the higher-priority interrupt. If a RETURN BRANCH (BRR) instruction is used at the end of an interrupt-servicing subroutine, the interrupt level is not cleared and program control is not returned to the proper location. Also, a BRU with indirect addressing within an interrupt-servicing subroutine prematurely clears the interrupt level.

SINGLE-INSTRUCTION INTERRUPT

If the interrupt level is a single-instruction interrupt level, the instruction in the interrupt address is executed and the interrupt level is automatically cleared (provided that the instruction requires a timing of two or more cycles), and the computer executes the next instruction in sequence after the instruction at which the interrupt occurred. 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 076 (and location 076 contains the instruction MIN 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 030-077 and any of the interrupt levels 0200-01777 may be "subroutine" or "single-instruction" interrupts, as required. If the instruction in a single-instruction interrupt level memory location is a branch and the branch should occur, the interrupt is cleared but there is no automatic return to the interrupted program, and no record is kept of the contents of the program counter, when the branch is executed.

NON-INTERRUPTABLE INSTRUCTIONS

If an INCREMENT INDEX AND BRANCH (BRX) instruction is being executed and the branch should occur, the computer can not acknowledge an interrupt until the instruction to which the BRX branches is executed. Also, if an ENERGIZED OUTPUT M(EOM) instruction is being executed, the computer can not acknowledge the interrupt until the instruction following the EOM is also executed.

INTERRUPT ARM/ENABLE RESPONSE

Two control features are available to the programmer concerning the interrupt system - Arm and Enable. As shown in Figures 3 and 11, an interrupt level proceeds to the waiting state only if it is "armed" when the interrupt pulse is sent to it. If the level is "disarmed", the pulse does not pass through the AND gate in front of the interrupt level, and the pulse is not "remembered" by the interrupt level. Once in the waiting state, the interrupt level remains in the waiting state as long as any higher-priority interrupt level is already in the active state. Also, an interrupt level proceeds to the active state only if it is "enabled." If the interrupt level is "disabled," the steady signal from the Waiting ff does not pass through the AND gate in front of the Interrupt ff.

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 consideration, and will always cause a program interrupt if an interrupt device pulses its interrupt line. This type of interrupt is always considered armed and enabled, and cannot be disarmed or disabled, except by rewiring the computer. Some of the XDS optional hardware interrupts (levels 030-077) such as power fail-safe, and any of the special systems interrupts (levels 0200-01777) may be of this type, depending on the requirements of a particular installation. All other XDS optional hardware interrupt levels (030-077 are armed, disarmed, enabled, and disabled by means of the computer control console and/or the program being executed. The control of special system interrupts (0200-01777) is discussed separately.

XDS OPTIONAL HARDWARE INTERRUPTS

These interrupt levels are both armed and enabled if the INTERRUPT ENABLED indicator on the computer control console (see Section 5) is turned on, and are both disarmed and disabled if the indicator is turned off. (Interrupt levels considered always armed and enabled are not affected by this indicator.) Whenever the START button on the computer control console is pressed, all interrupt levels and the Interrupt ff are cleared, and the INTERRUPT ENABLED indicator is turned off. Thereafter, the indicator may be controlled by the operator with the INTERRUPT ENABLE switch and/or the program being executed.

INTERRUPT ENABLE SWITCH

Whenever this switch is manually held in the ENABLE position, the INTERRUPT ENABLED indicator is unconditionally turned on. Any controllable interrupt levels receiving an interrupt pulse while the indicator is on proceed to the waiting state. Any in the waiting state proceed to the active state as soon as their priority allows. When the switch is released, it automatically returns to the COMPUTER position.

If any interrupts occur during the time the INTERRUPT ENABLE switch is in the ENABLE position, and the Enable ff is in the reset state when the switch is released, the INTERRUPT ENABLED indicator is turned off, interrupt levels 030-077 are disarmed and disabled, and interrupt levels 0200-01777 are disabled. However, any interrupt levels in the active state are processed until cleared. If the Enable ff is set when the switch is released, the indicator remains on and the interrupt levels remain armed and enabled.

INTERRUPT ENABLE/DISABLE INSTRUCTIONS AND TESTS

Two machine instructions are used to set and reset the Enable ff, and two instructions are used to test the status of the INTERRUPT ENABLED indicator.

EIR ENABLE INTERRUPTS
EOM 020002 0 02 20002

EIR unconditionally sets the Enable ff and turns on the INTERRUPT ENABLED indicator. If any interrupt levels are in the waiting state, the one with the higher priority is acknowledged, and proceeds to the active state. Interrupt levels 030-077 remain armed and enabled (and interrupt levels 0200-01777 remain enabled) as long as the Enable ff is set, regardless of the position of the INTERRUPT ENABLE switch.

Affected: INTERRUPT ENABLED Timing: 1

DIR DISABLE INTERRUPTS
EOM 020004 0 02 20004

DIR unconditionally resets the Enable ff. Also, if the INTERRUPT ENABLE switch is in the COMPUTER position, the INTERRUPT ENABLED indicator is turned off, interrupt levels 030-077 are disarmed and disabled, and interrupt levels 0200-01777 are disabled. Any subsequent interrupt pulses to levels 030-077 are not "remembered" until the indicator is again turned on by the switch or by execution of an EIR. However, if the switch is in the ENABLE position when DIR is executed, the instruction resets the Enable ff, but does not turn off the indicator and does not disarm or disable the interrupt levels. Thus, the switch may be used to override a DIR, but never on EIR. When the switch is released after a DIR has been executed, the indicator is turned off, interrupt levels 030-077 are disarmed and disabled, interrupt levels 0200-01777 are disabled, but any interrupts in the active state are processed until cleared.

Affected: INTERRUPT ENABLED Timing: 1

IET INTERRUPT ENABLED
(Skip if Interrupt System Enabled)
SKS 020004 0 40 20004

If the INTERRUPT ENABLED indicator is on, the computer skips the next instruction in sequence and executes the following instruction.

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

IDT INTERRUPT DISABLED TEST
(Skip if Interrupt System Disabled)
SKS 020002 0 40 20002

If the indicator is on, the computer executes the next instruction in sequence. If the indicator is off, the computer skips the next instruction in sequence and executes the following instruction.

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

Note: EIR and DIR are EOM's in the Internal Control mode, and IET and IDT are SKS's in the Internal Test mode (see Section 4, Primary Input/Output Instructions).

END-OF-WORD/END-OF-TRANSMISSION INTERRUPT OPERATIONS

A program can use the W and Y buffers as single-word, direct, program-controlled, input/output buffers. Special input/output instructions (EOM's in the buffer control mode, see Section 4) control this type of operation. In the buffer control mode, the program can specify that interrupts occur as each word is transmitted from the buffer to the peripheral device (on output), or as soon as the buffer is filled from the peripheral device (on input). This is the end-of-word interrupt. The program can also specify that an end-of-transmission occurs (on input) when the buffer detects a terminal signal such as end-of-record from a magnetic tape unit, card reader, paper tape reader, etc. During both input and output operations, this interrupt occurs when the peripheral device being used in the transmission disconnects from the buffer. The buffer is then ready for another input/output operation.

End-of-word and end-of-transmission interrupts can also control input/output transmission when the program is operating in the block transmission or "interlaced" mode (optional feature). In this mode, an end-of-transmission interrupt also occurs when the buffer has sent a specified number of words from memory to a peripheral device (or when the buffer has read a specified number of words into memory from a peripheral device, as well as when the buffer detects an end-of-record signal). Since the buffer automatically controls input/output in the interlaced mode, the end-of-word interrupts are not generated while the buffer is in this mode of transmission. See Section 4, Interlaced Block Transmission, for terminal input/output conditions during interlace control.

SPECIAL SYSTEMS INTERRUPTS

Interrupt levels 0200-01777 are optional, general-purpose interrupts that are added in groups of 16 according to the requirements of a particular computer system, and may be any desired combination of subroutine and/or single-instruction interrupts. If the optional Arm Interrupt Control Unit is not present as a part of the computer, these interrupts are always armed (cannot be disarmed, except by rewiring the computer) and any interrupt pulse entering the interrupt level unconditionally sets the level to the waiting state. However, these interrupts are enabled only if the INTERRUPT ENABLED indicator is on, and are disabled if the indicator is off (see Interrupt Arm-Enable Response).

ARM INTERRUPTS (OPTIONAL)

If the optional Arm Interrupt Control Unit is present as a part of the computer, interrupt levels 0200-01777 must be armed (and/or disarmed) in groups of sixteen (i.e., interrupt levels 0200-0217, 0220-0237, etc.), and only by a specific combination of the two instructions ARM INTERRUPTS (AIR) and PARALLEL OUTPUT (POT) and a control word. These interrupt levels are enabled if the INTERRUPT ENABLED indicator is on, and disabled if the indicator is off. Also, these interrupts are initially disarmed and disabled when the START button on the computer control console is pressed.

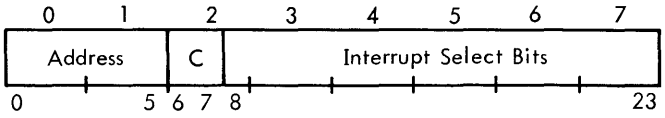
AIR ARM INTERRUPTS
EOM 020020 0 02 20020

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 (POT instructions are discussed in Section 4). The instruction sequence AIR, POT must be used for each group of interrupt levels; otherwise, an unpredictable operation occurs. These instructions have no effect on the INTERRUPT ENABLED indicator and the Enable ff, and the Control Unit is not affected by the indicator or the Enable ff.

Affected: Arm Interrupt Control Unit Timing: 1

CONTROL WORD

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



The Address field (bits 0-5) identifies which group of interrupts is being addressed (e.g., an Address field of octal 00 identifies interrupt levels 0200-0217). The C field (bits 6 and 7) specifies whether the interrupt levels selected by bits 8-23 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 field (e.g., 0200, 0220, etc.). Bit position 23 represents the highest-numbered (lowest-priority) level within the group.

The C field control functions are:

Bit Positions 6	7	Octal Value	Function
0	0	0	Not used
0	1	2	Arm only those interrupt levels that are selected by a 1 in bit positions 8-23. (Interrupt levels represented by a zero in bit position 8-23 are not affected.)
1	0	4	Disarm only those interrupt levels that are selected by a zero in bits 8-23. (Interrupt levels represented by a 1 in bit positions 8-23 are not affected.)
1	1	6	Arm all interrupt levels selected by a 1 and disarm those levels selected by a zero.

Example:

The following partial program enables the entire interrupt system, arms interrupt levels 0210-0227, disarms levels 0230-0237, but does not change the current state (armed or disarmed) of levels 0200-0207.

Location	Instruction	Address	Comments
	EIR		Enable entire interrupt system (turns INTERRUPT ENABLED 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.
	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 arms level 0210-0217. If any of levels 0200-0207 are already armed or disarmed they remain so.
CW2	01777400		This control word arms levels 0220-0227 and disarms levels 0230-0237, regardless of their previous state.

4. INPUT/OUTPUT SYSTEM

INTRODUCTION

The XDS 910 has a comprehensive input/output system to complement its high internal processing speed and versatile instructions. This 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 the individual characteristics of a wide variety of devices, yet it leaves a high degree of input/output control to the programmer.

This system is capable of the following types of input/output:

1. Buffered input/output of data words, each under direct program control.
2. Input/output of blocks of characters or words time-shared with memory and multiplexed with computation using "interlaced" buffers.
3. Direct parallel input/output of up to 24 bits of information to and from external static registers under program control.
4. Single-bit input/output, such as equipment on/off status, sense switches, and pulsing and sensing of special devices.

A buffer assembles and disassembles data words as they are transmitted between core memory and the peripheral equipment. The buffer maintains control of operations such as characters per word transmitted and direction of peripheral operation (as in magnetic tape forward/reverse).

The W buffer, standard equipment in the computer, performs input/output of data words, each under direct program control. On output, the buffer transmits words in 6-bit characters, the number of characters per word – 1, 2, 3, 4 – being under program control. On input, the buffer receives words in 6-bit characters with the number of characters per word being under program control. The system may include the Y buffer, identical in function to the W buffer, as a second input/output buffer. Additionally, the Y buffer may contain the facility for input/output of 24-bit words (no character assembly/disassembly).

Each buffer can control as many as 30 input/output devices and automatically handles character, word assembly and disassembly, and input/output parity detection and generation.

Both buffers are bidirectional and can communicate with 6-bit character devices (and word devices of up to 24 bits for the Y buffer). For character-oriented devices, the program specifies the number of characters to be contained in each word during the transmission.

Each buffer may have an "interlace" associated with it. Interlace allows input/output of blocks of data words with buffer-to-memory and memory-to-buffer word transmission being completely automatic and multiplexed with computation.

When in use, a buffer interlace controls the transfer of the data words going through the associated 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 interlace itself controls input/output termination during interlaced operation.

An interlaced buffer uses the memory logic of the central processor to facilitate input and output of data words. The transfer of each word between a buffer and memory requires two memory cycles. During this time, computation stops in the central processor. The Y buffer has priority over the W buffer for the use of the word input/output logic. Any interlaced buffer has priority over the central processor for memory access.

W AND Y BUFFER REGISTERS DESCRIPTION

Figure 4 contains a block diagram of a buffer and the functional control of information between the buffer, the memory, and the external devices. See Appendix D for a functional flow diagram of the W buffer.

Each of the 30 devices that can be attached to a buffer has a unique, two-digit, octal address by which it is chosen for an input/output operation. To choose the peripheral device, the program loads the proper unit address into the 6-bit Unit Address Register (UAR). This address selects both the device and, if appropriate, the function to be performed. Placing a unit address in the UAR "connects" the peripheral unit addressed to the buffer and the buffer becomes "active." When the UAR contains a zero address, or any time that a terminal or initial condition clears the contents of UAR, the buffer is "inactive," it is "ready" to buffer ready tests, and it is not connected to a peripheral unit.

The Word Assembly Register (WAR) and the Single Character Register (SCR) comprise the active portion of a buffer. The word assembly register, a 24-bit, word-sized buffer, contains the word of data actively being transmitted during an input or output operation. During input, 6-bit characters (plus parity) come into the single character register where the buffer assembles them, one at a time, into the WAR. Depending on the number of

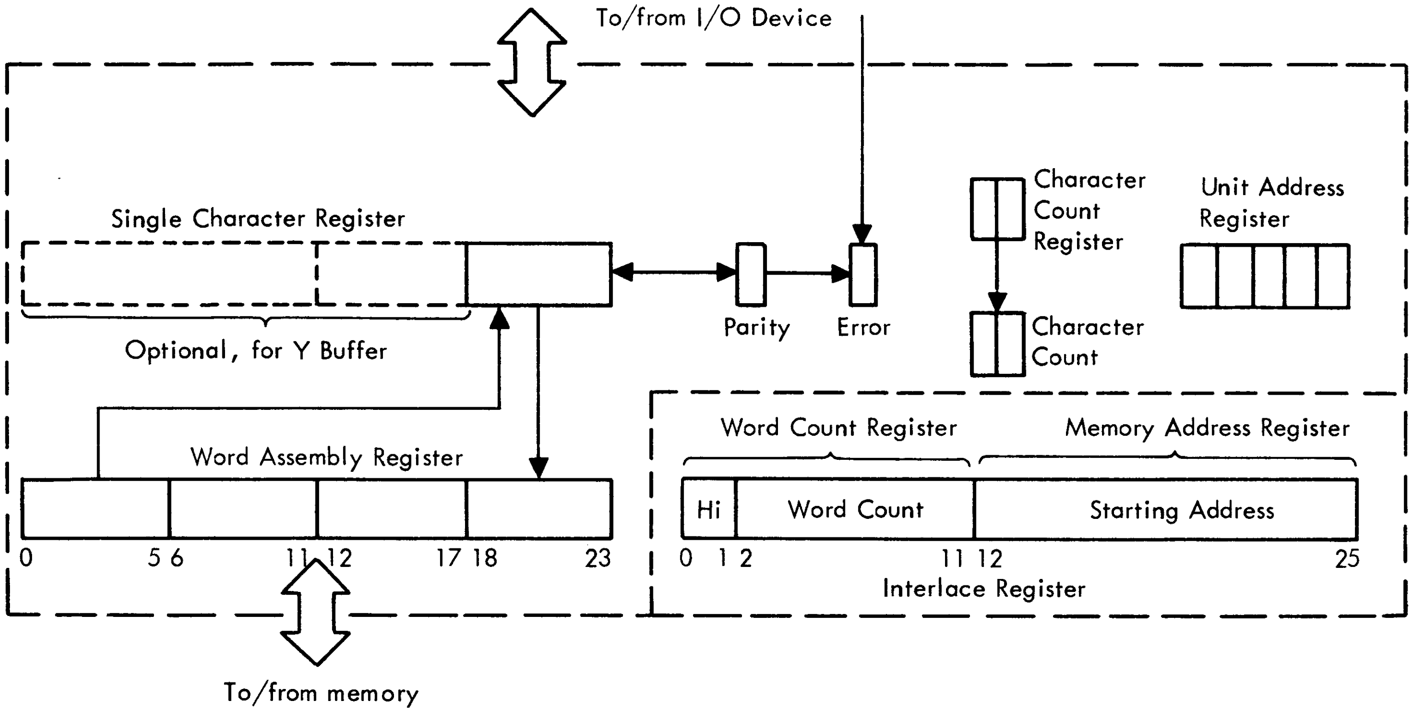
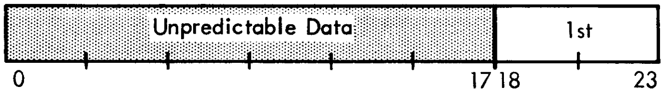


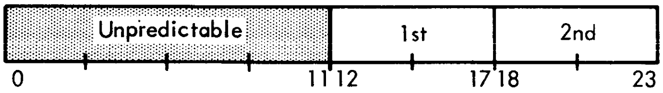
Figure 4. XDS 910 W(Y) Buffer

characters per word specified, the word assembled during input has the form shown below. In each case, the unfilled character positions contain unpredictable data.

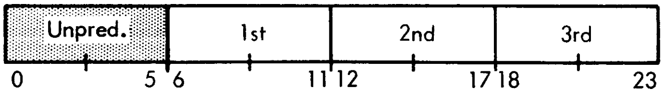
One character per word



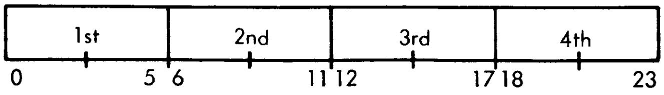
Two characters per word



Three characters per word



Four characters per word



A word assembled during a single-word operation is placed into memory by a WIM instruction. Under interlace control, the interlaced buffer automatically places the word, when assembled, into memory.

During output, words come from memory into the WAR where the buffer disassembles them into the SCR, one 6-bit character at a time. Depending on the characters per word mode specified, the buffer transmits the 6-bit characters (with generated parity) as follows:

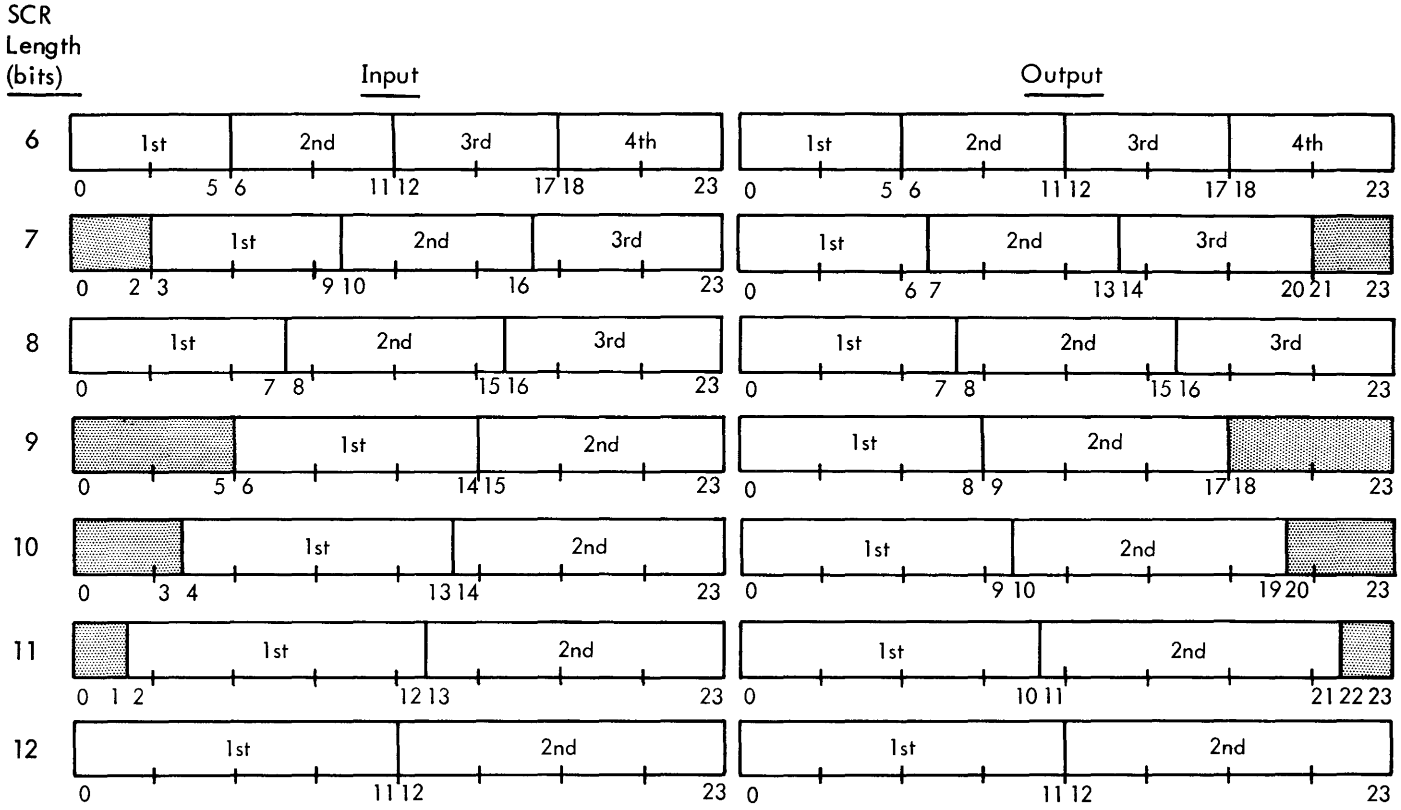
Function	Mode
Output one character from bits 0 through 5	One character per word
Output two characters from bits 0 through 5, 6 through 11	Two characters per word
Output three characters from bits 0-5, 6-11, 12-17	Three characters per word
Output four characters from bits 0-5, 6-11, 12-17, 18-23	Four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

After each character transfer, the word in the WAR shifts left six bits to be ready for the next transfer until those characters needed from each word are used. When required, a new word containing the next character(s) comes to the WAR.

Y BUFFER CHARACTER ASSEMBLY (OPTIONAL)

The Y buffer can have a single character register of one specified size from 6 bits to 24 bits in length. Using character assembly, a Y buffer inputs and outputs words according to the available option shown in Table 2.

Table 2. Y Buffer Character Assembly Options



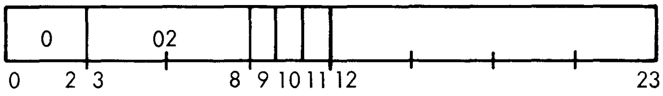
No character assembly/disassembly occurs with Single Character Register sizes of 13 bits and larger; the buffer treats the contents of the Single Character Register as a truncated word.

INTERLACE REGISTERS (OPTIONAL)

A buffer interlace contains two working registers, the Word Count Register (WCR) and the memory Address Register (MAR). In the set-up sequence – EOM, POT – for an interlaced input/output operation, the POT instruction transmits to the interlace a data word made up of the word count (that is, length of the block) and the starting address of the block. See "Interlaced Block Transmission" and "Programming the Interlace Register," pages 33 and 34.

PRIMARY INPUT/OUTPUT INSTRUCTIONS

EOM ENERGIZE OUTPUT M



ENERGIZE OUTPUT M (EOM) is a multipurpose instruction that operates in four distinct modes with many functional configurations. The modes are buffer control, input/output control, internal control, and system control. In the third and fourth modes, EOM initiates and controls non-buffer operations such as special systems transmissions. Each of the frequently used EOM instruction configurations has a mnemonic recognized by the standard assembler, META-SYMBOL (see "Standard Buffer EOM Instructions," page 28).

The setting of two bits (10, 11) within the instruction format determines the different modes for the operation of EOM:

Bit Positions		
10	11	Mode
0	0	Buffer Control
0	1	Input/Output Control
1	0	Internal Control
1	1	System Control

An EOM in the buffer control mode operates essentially as a setup or preparation facility for data transmissions or other peripheral activities using the buffer. EOM in this mode specifies the buffer to be used, the peripheral unit on that buffer, the operation, and the desired character format. EOM also details to the buffer and its "connected" peripheral unit the use of BCD or binary data transmission, the allowance or not of a leader (as in paper tape), and the direction of operation (as in forward or reverse directions for magnetic tape). Execution of such an EOM also connects the specified peripheral unit to the buffer. EOM in this mode can also alert the interlace, which is the optional, automatic, buffer control for input/output.

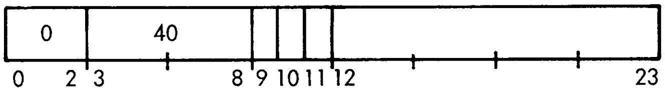
An EOM in the input/output mode directs peripheral devices to perform nontransmitting operations such as rewind magnetic tape and upspace the printer. It can alert peripheral devices that a PARALLEL INPUT (PIN) or PARALLEL OUTPUT (POT) instruction follows. This EOM can also give an extension of the word count from 10 to 12 bits (for transmission of blocks of from 1024 to 4095 words). Without disturbing the associated buffer, this EOM can also set up the interlace.

An EOM in the internal control mode enables and disables the interrupt system. The internal control EOM can prepare the system for the selective arming and disarming of the system interrupt levels. See Section 3, Interrupt System.

An EOM in the system control mode is specially coded for a given installation and system. Addressing capability is 15 bits or 32,768 combinations for these special system designations.

If an interrupt occurs during the execution of an EOM in any mode, no acknowledgement can occur until execution of the instruction following the EOM has been completed.

SKS SKIP IF SIGNAL NOT SET



SKS is a multipurpose test instruction used for testing the states and responses of input/output buffers and their attached peripheral devices as well as for testing internal and external indicators. SKS is a "skip class" instruction providing a decision and transfer capability to all buffers, devices, indicators, and systems that require it. It operates in four distinct modes: special internal test, input/output unit test, internal test, and special system test. In the input/output unit test mode, the SKS tests buffer-oriented, input/output functions. Each of the frequently used SKS instruction configurations has a mnemonic recognized by META-SYMBOL (see "Standard Buffer SKS Instructions, " page 29).

The setting of two bits (10, 11) within the instruction format determines the different modes of operation of SKS:

Bit Positions		
10	11	Mode
0	0	Special Internal Test
0	1	Input/Output Unit Test
1	0	Internal Test
1	1	Special System Test

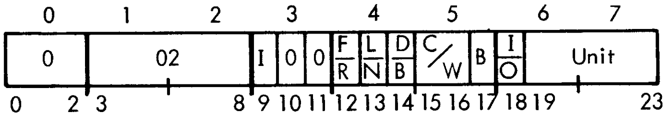
In the input/output unit test mode, SKS tests peripheral devices directly. These include testing indicators in a magnetic tape unit such as beginning of tape, end of tape, file-protect ring present, and end-of-file.

In the internal test mode, SKS tests whether or not the interrupt system is enabled or disabled, whether a breakpoint switch is set, and whether Overflow is set. Other configurations of this SKS also perform tests for buffer ready and for buffer error.

In the special internal and special system test modes, SKS tests signals of special configuration as the specific systems require.

BUFFER CONTROL EOM

The ENERGIZE OUTPUT M (EOM) in the buffer control mode addresses and connects the specified buffer and selects the desired unit address. The detailed instruction format is:



Bit Designation	Octal Position	Octal Value	Function
002	0-2	002	Bit positions 0 through 8 contain the instruction code for EOM.
I	3	4	A 1-bit in position 9 alerts the buffer interlace.
00	3	0	Bit positions 10 and 11 contain the indicator for the buffer control mode.
F/R	4	0 4	Bit position 12 specifies the direction in which the peripheral device will operate. A 0 specifies the forward direction, and should always be used for devices incapable of operating in reverse. A 1 (octal 4) specifies the reverse direction.
L/N	4	0 2	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 (octal 2) specifies a start without leader, and should always be used for a device incapable of producing leader.

Bit Designation	Octal Position	Octal Value	Function
D/B	4	0 1	Bit position 14 specifies the character format. A 0 specifies BCD format; a 1 specifies Binary format.
C/W	5	0 2 4, 6	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 (octal 0), two by 01 (octal 2), three by 10 (octal 4) and four by 11 (octal 6).

Bit Designation	Octal Position	Octal Value	Function
B	5	0 1	Bit position 17 specifies the buffer to be controlled. A 0 specifies the W buffer; a 1 specifies the Y buffer.
I/O	6	0 4	Bit position 18 specifies the direction of transmission. A 0 specifies input; a 1-bit specifies output.
UNIT	6-7		Bit positions 19-23 specify the peripheral device to be used in the input/output operation. (See Table 3 for unit address codes for peripheral devices.)

Table 3. Unit Address Codes

Octal Value	Input Function	Octal Value	Output Function
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 Input No. 1	66	Disc File Output No. 1
27	Disc File Input No. 2	67	Disc File 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

INPUT/OUTPUT CONTROL EOM

EOM in the I/O control controls various operations peculiar to a given device such as rewind tape, space paper, or skip to format channel on the printer. It also controls certain buffer functions such as Terminate Output.

0 1 2 3 4 5 6 7

0	02	I	0	1	Function	B	I/O	Unit
0 2	3 8	9	10	11	12 16	17	18	19 23

Bit Designation	Octal Position	Octal Value	Function
002	0-2	002	Bit positions 0 through 8 contain the instruction code for EOM.
I	3	4	A 1 in bit position 9 alerts the interlace attached to the buffer specified by bit 17.
01	3	1	Bit positions 10 and 11 specify I/O control mode of EOM.
Function	4-5		Bit positions 12-16 specify control peculiar to each peripheral device.
B	5	0 1	Bit position 17 specifies the buffer to be controlled. A 0 specifies the W buffer; a 1 specifies the Y buffer.
I/O	6	0 4	Bit position 18 specifies the direction of transmission. A 0 specifies input; a 1 specifies output.
UNIT	6-7		Bit positions 19-23 specify the peripheral device to be used in the input/output operation. (See Table 3 for I/O and unit address codes for peripheral devices.)

An I/O control EOM occurring between an interlace alert EOM and the POT sets the high-order interlace count bits equal to bit positions 22 and 23 of the EOM. When bits 18-23 are all zeros, a 1-bit in bit position 12 terminates outputs or converts magnetic tape inputs to "scan".

STANDARD BUFFER EOM INSTRUCTIONS

Several EOM function configurations have standard uses. These have standard assembler-type mnemonics and are separate instructions.

ALC 0 ALERT W BUFFER
EOM 050000

0	02	50000
0 2	3 8	9 23

ALC 0 alerts the W buffer interlace. This instruction does not disturb the buffer in any way, except that the end-of-word interrupt is inhibited until the interlaced I/O operation terminates.

ALC 1 ALERT Y BUFFER
EOM 050100 0 02 50100

Affected: W(or Y) Interlace Timing: 1

DSC 0 DISCONNECT W BUFFER
EOM 0

0	02	00000
0 2	3 8	9 23

DSC 0 disconnects the W buffer. This instruction unconditionally sets the unit address register to 00 regardless of whether or not the buffer is currently addressing a device. DSC disconnects any device connected to the buffer. This instruction unconditionally makes the buffer ready and clears the error indicator.

DSC 1 DISCONNECT Y BUFFER
EOM 0100 0 02 00100

Affected: W(or Y) Buffer, Error Indicator Timing: 1

TOP 0 TERMINATE OUTPUT ON W BUFFER
EOM 014000

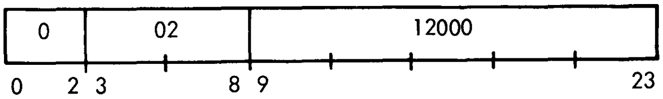
0	02	14000
0 2	3 8	9 23

The execution of this instruction causes the buffer to disconnect when the buffer delivers the last specified character of the word in the word assembly register to the peripheral device. TOP must always be used to terminate a noninterlaced buffer output operation.

TOP 1 TERMINATE OUTPUT ON Y BUFFER
EOM 014100 0 02 14100

Affected: W(or Y) Buffer Timing: 1

ASC 0 ALERT TO STORE ADDRESS IN W BUFFER
EOM 012000



This instruction (available with the Memory Interlace Option) alerts a PINable interlace so that the PIN instruction that follows can store the contents of the memory address register. This instruction affects the buffer in no other way. See Direct Parallel Instructions, this section, for a detailed discussion of PIN.

ASC and PIN determine the current status of the W buffer. The two instructions are written together:

```
ASC   0
PIN   M
```

When the program executes these two instructions, the effective memory location designated by the PIN instruction contains:

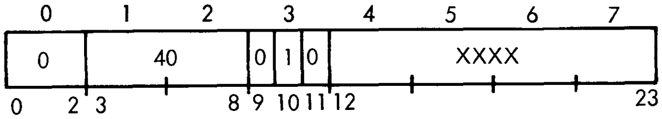
Bit Positions	Contents
0 through 8	Zero
9 through 23	Contents of W buffer's Memory Address Register

ASC 1 ALERT TO STORE ADDRESS IN Y BUFFER
EOM 012100 0 02 12100

Affected: W(or Y) Interlace Timing: 1

INTERNAL TEST SKS

The SKIP IF SIGNAL NOT SET (SKS) in the internal test mode tests the indicators in the selected buffer. The instruction format is:



Bit Designation	Octal Position	Octal Value	Function
040	0-2	040	Bit positions 0 through 8 contain the instruction code for SKS
010	3	2	Bit positions 10 and 11 contain the internal test mode selection
XXXX	4-7		Bit positions 12 through 23 specify the test. Each of these tests cause the computer to skip when the test condition is true. Selecting more than one test causes the computer to skip only when skip condition exists on all of the selected tests.

STANDARD BUFFER SKS INSTRUCTIONS

BRTW W BUFFER READY TEST
SKS 021000 0 40 21000

BRTY Y BUFFER READY TEST
SKS 022000 0 40 22000

If the buffer is ready to accept a new input/output instruction, the computer skips the next instruction in sequence and executes the following instruction. If the buffer is active, or in the process of disconnecting a peripheral unit, the computer executes the next instruction in sequence.

BETW W BUFFER ERROR TEST
SKS 020010 0 40 20010

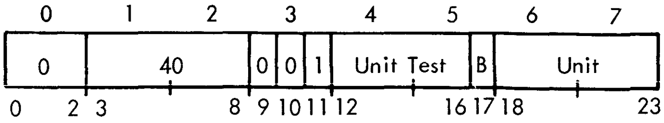
BETY Y BUFFER ERROR TEST
SKS 020020 0 40 20020

BETW (BETY) tests the error detector in the selected buffer. If the error detector has not been set, the computer skips the next instruction in sequence and executes the following instruction. If the error detector has been set, the computer executes the next instruction in sequence.

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

INPUT/OUTPUT UNIT TESTS

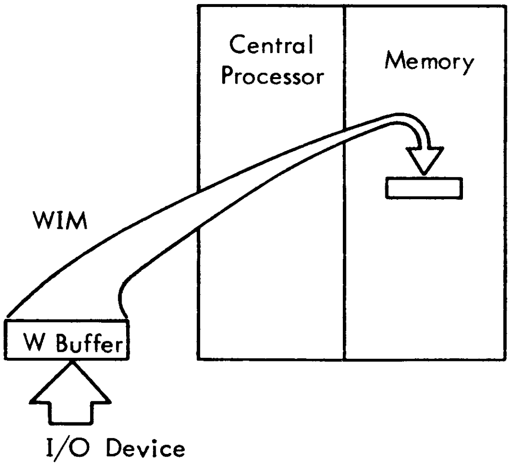
SKS in the input/output test mode tests the condition of the peripheral devices in the system directly. The individual test condigurations for SDS peripheral devices are contained in Section 6.



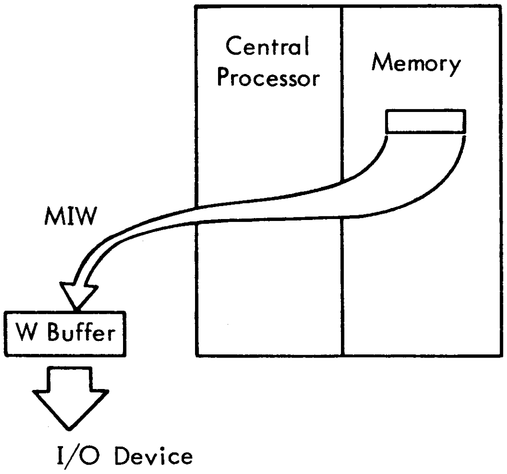
Bit Designation	Octal Position	Octal Value	Function
040	0-2	040	Bit positions 0 through 8 contain the instruction code for SKS
	3	1	Bit position 9 contains zero. Bit positions 10 and 11 contain the I/O unit test mode selection
Unit Tests	3-4		Bit positions 12 through 16 select the test.
B	5	0, 1	Bit position 17 specifies the buffer to be activated. A 0 specifies the W buffer; a 1 the Y buffer.
Unit Address	6-7		Bit positions 18 through 23 specify the unit address. (See Table 3)

SINGLE-WORD TRANSMISSION

Using the W and Y buffers, a program can transmit data words between memory and peripheral devices under the direct control of single instructions. To accomplish this, the program first alerts the buffer and the peripheral device with an energize or "alert" instruction; then the program performs the direct control transmission. One of the configurations of the multipurpose instruction, ENERGIZE OUTPUT M (EOM), alerts the buffer. The program performs the direct control operations with two instructions associated with each buffer. For the W buffer, W INTO MEMORY (WIM) causes a word from a peripheral transmission to be taken from the W buffer and placed directly in the specified memory location without disturbing any internal registers.



MEMORY INTO W (MIW) causes a word to be taken from a specified memory location and placed in the W buffer to be output to the currently operating peripheral device connected to the buffer.



YIM and MIY instructions function in a similar manner for the Y buffer. The general test instruction, SKIP IF SIGNAL NOT SET (SKS), provides the facility for testing error indicators and/or for testing various peripheral device indicators.

MIW MEMORY INTO W BUFFER WHEN EMPTY

0	X	0	12	I	M
0	1	2	3 … 8	9	10 … 23

MIW transfers the contents of the effective memory location into the W buffer. The central processor waits until the buffer is empty and ready to accept the data word.

The buffer must be connected to the desired peripheral device by a previous buffer control EOM instruction that selects the buffer, the unit address, and all appropriate control functions and connects the buffer to the desired peripheral device.

Affected: W Buffer Timing: 2 + wait

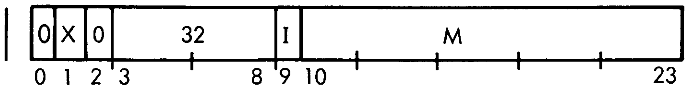
MIY MEMORY INTO Y BUFFER WHEN EMPTY

0	X	0	10	I	M
0	1	2	3 … 8	9	10 … 23

MIY transfers the contents of the effective memory location into the Y buffer. The central processor waits until the buffer is empty and ready to accept the data word.

Affected: Y Buffer Timing: 2 + wait

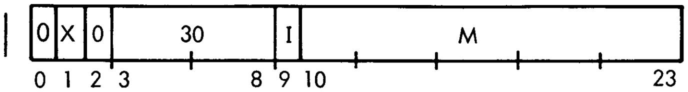
WIM W BUFFER INTO MEMORY WHEN FULL



WIM transfers the contents of the W buffer into the effective memory location. The central processor waits until the buffer is full and ready to deliver the data word.

Affected: (M) Timing: 3 + wait

YIM Y BUFFER INTO MEMORY WHEN FULL



YIM transfers the contents of the Y buffer into the effective memory location. The central processor waits until the buffer is full and ready to deliver the data word.

Affected: (M) Timing: 3 + wait

SINGLE-WORD OPERATIONS

Using the buffer control EOM and input or output instructions (MIW, WIM, etc.), data words transfer between the buffer and memory under direct program control. Between data words, the computer waits until the buffer is ready to perform the transfer. The cause of this delay is normally that the buffer is actively transmitting or receiving the previously requested data word.

The interrupt system allows the program to connect the device to be used in the transfer, to enable the interrupt, and then to continue processing in the main program; this eliminates the central processor tie-up. When the buffer empties or fills during the transfer from or to memory, the End-of-Word interrupt to the corresponding interrupt location notifies the program that the buffer is ready. A service routine is entered via a BRANCH AND MARK PLACE (BRM) instruction in the appropriate interrupt location processing the interrupt. This routine contains the instruction (MIW or WIM, for example) that can execute immediately without computer tie-up.

An input/output operation is started by executing EOM (see Primary Input/Output Instructions). EOM also copies the number of characters per word into the character counter (CC), clears the error switch (E) and, for input, sets the single character register (SCR) to receive inputs. As soon as the first character arrives, it is stored in the SCR. In addition, a signal is sent from the source of the data to the buffer, indicating that the SCR is loaded. With the arrival of this signal, which closes the input data loop, the SCR is connected serially to the word assembly register (WAR) and a 6-bit circular left shift takes place through the WAR. This process transfers the contents of the SCR into the least significant bits of the WAR. When the character is transferred to the SCR, a seventh parity bit is checked by the buffer. If a parity error occurs, the parity bit is set and the control panel ERROR indicator is lighted. Note that such an error does not stop the computer. Facilities are provided in the computer that allow the program to interrogate and reset this bit at a later time.

As soon as the contents of the SCR have been shifted into the WAR the contents of the character counter are decremented by one. If the new contents of this counter are still greater than zero, the buffer will wait for the next character. When the next character arrives, the SCR and the WAR repeat the shifting operation described above. The information that was in the 6 right-hand bit positions of the WAR are shifted left 6 bits, and the new contents of the SCR are shifted into the 6 right-hand positions of the WAR. When the character counter reaches zero, it generates an interrupt pulse to the computer, that forces the computer into a subroutine to handle the word of information held in the word assembly register of the buffer, providing that the interrupt system is enabled (see Section 3).

An alternate method may be used with punched paper tape and magnetic tape. In these cases, a gap on the tape generates a second interrupt which signifies the end of a block. During the input process the main program may ascertain when the transfer of information is complete by testing the status of the buffer. If the buffer is ready, the transfer is complete and another input/output operation may be initiated.

The input/output system outputs data in a manner exactly analogous to input. Parity in this case is generated rather than checked. As noted previously, a closed-loop synchronizing system is used to assure that no data is lost. During input this takes the form of a signal from the input source indicating that the data has been sent to the character buffer. During output, a signal from the output device is used to time the loading of each character into the character register.

Data can be lost in one of two ways. During input, the WAR and the SCR can both be full (if the program is not correct) at the time that another character is ready to be entered. If, for example, the source of data is a magnetic tape unit, information cannot be delayed and will be lost. Similarly, during output, if data has not been loaded into the WAR by the time all previous data has been used, a position on the magnetic tape will be empty. In either case the computer detects the occurrence of the error and lights the ERROR indicator.

The W buffer can operate up to a maximum frequency of 62,500 characters per second under computer control.

Input Termination

When the end-of-record signal is detected by the W buffer, the buffer automatically disengages from the device, generates an End-of-Transmission Interrupt, and the buffer is then ready for another operation. The buffer logic is reset, except that the state of error indicator is maintained and the last word of the input will still be 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 an End-of-Transmission Interrupt is generated, the buffer is disconnected, and zeros are automatically forced into the least significant positions of the last word. The End-of-Word interrupt is not generated for this partial word. The partial word can be stored in memory by a WIM 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 W buffer is either the last group of characters from the input device, if they were not previously transferred to memory by a WIM instruction, or zeros if the last group of characters had been transferred to memory. In either case, it is safe to issue one WIM instruction after the buffer has disengaged without hanging up the computer.

When the buffer automatically disengages, an End-of-Transmission interrupt is actuated if the interrupt system is enabled. A special SKS instruction (BRTW or BRTY) to test the buffer for ready is provided if the interrupt system is not being employed.

Scan

During input from magnetic tape the program may convert the operation to a scan by a proper EOM instruction (see Magnetic Tape Input/Output). When the W buffer is operating in the scan mode, character after character is shifted into the W buffer but the buffer never signals that it is full. When the end-of-record (gap) is detected, an End-of-Word interrupt is generated while the tape is still moving. If a read or scan command is sent to the tape within 500 microseconds, the reading or scanning continues. If no command is given in this interval, the tape will stop, generate an End-of-Transmission interrupt, and disconnect the buffer. When the End-of-Word interrupt occurs, the last four characters of the record scanned (not zeros) are held in the W buffer.

Output Termination

When the computer has transferred the last word of information of an output record to the W (or Y) buffer, it should follow this with a TERMINATE OUTPUT (TOP) instruction. The last word in the buffer will be properly sent to the output device and then the buffer will automatically disengage from the device, maintaining only the status of the error indicator. In the case of magnetic tape, a tape gap is generated before the tape unit is disengaged. After the output unit is disengaged, the buffer is ready for another operation. If the interrupt system is enabled, an End-of-Transmission interrupt signal is generated.

INTERLACED BLOCK TRANSMISSION

Using the W and Y buffers, a program can transmit blocks of data to and from core storage under buffer interlace control. (See Figure 5.)

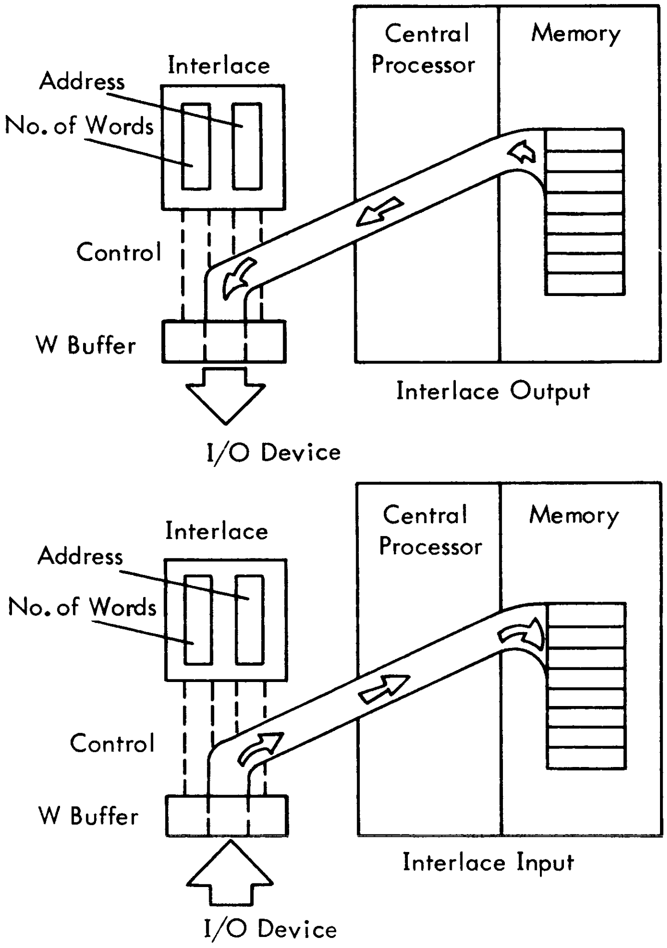


Figure 5. Interlace Output and Input

The interlace control, which is optional equipment, uses the W buffer previously described and, additionally, a 26-bit interlace register. This register is divided into two parts, a 12-bit word count register (WCR) and a 14-bit memory address register (MAR). The register is loaded with a 24-bit word taken from the computer's memory by execution of POT instruction which loads the 14-bit MAR and 10 bits of the WCR. The two high-order bits of the WCR permit the counter to reach 4095. If these bits are required, they are set by an EOM (I/O control mode) instruction. An entire block of information may now be copied into or read out of memory without interfering with other activities of the computer.

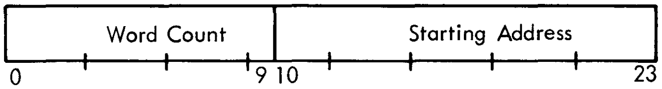
The WCR holds the number of words in a block of an output operation. For input the WCR holds the maximum number of words that the computer will accept. The MAR contains the initial address into which information is to be placed, or from which information is to obtained. Because of it's 14 bit capacity, the MAR permits access to any memory location.

As with single-word operations, EOM initiates the input/output operation. Information flows one character at a time into the single character register (SCR), the characters are shifted into the word assembly register (WAR), the character counter (CC) is decremented each time a new character is entered, and parity errors are sensed. When the character counter reaches zero an interrupt is not initiated. Instead, a number of other events occur. At the end of the memory cycle during which the buffer's character counter has reached zero, the computer is halted. The contents of the WAR are stored in the memory location specified by the present contents of the MAR portion of the interface register. After this is accomplished, the computer is started again, the MAR is incremented by one, and the WCR is reduced by one. All of these operations require 16 microseconds. It should be noted that during this interval the buffer's SCR can be accepting new information for the next word. The interlace operation occurs regardless of the HALT indicator status and regardless of the position of the RUN-IDLE-STEP switch.

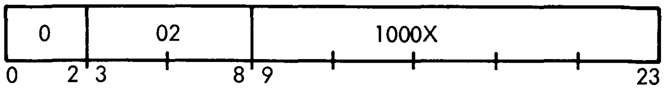
After the next word has been assembled, the character counter will have gone to zero again. This word will be stored in the new address specified by the MAR portion of the interlace register: this procedure will continue until the interlace register's WCR reaches zero or, on input, until a gap or end of record is encountered.

The system outputs data in a manner exactly analogous to input and, similarly, requires 16 microseconds per word.

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 0-9 of the data word, and is transmitted to bit positions 2-11 of the interlace register (WCR), providing for input/output of up to 1023 (01777) words. The starting address of the input/output operation is right-justified in bit positions 10-23 of the data word, and is transmitted to bit positions 12-25 of the interlace register (MAR), providing addressing of memory locations 0 through 037777 (16,385). For word counts of more than 1023 words, a second EOM is used between the alert EOM and the POT instruction, and has the following format:



Bits 22 and 23 of this EOM contain the 2 most-significant bits of the word count, and are transmitted to bit positions 0 and 1 of the WCR. The word-count capability is thus extended from 1023 (01777) to 2047 (03777), 3071 (05777), or 4095 (07777) words by using the values 1, 2, or 3, respectively, for X.

PROGRAMMING THE INTERLACE REGISTER

Before the execution of the POT instruction which engages the interlace, the appropriate interlace register must be "alerted". The EOM instruction that alerts the interlace may be ALC (which alerts the interlace register without otherwise affecting the buffer) or it may be the same buffer control EOM that connects the buffer to the I/O device, with a 1-bit in bit position 9 of the EOM (coded with an asterisk prefixed to the operand field of the instruction).

The instruction following the alert EOM (buffer control or ALC) may be another EOM to set the two most-significant bits of the WCR. It is important that no other instructions be programmed between the alert EOM and the POT instruction

The next instruction must be the POT instruction, which sets the least-significant portion of the WCR, sets the MAR, and engages the interlace. If the buffer has not been previously started with a buffer control EOM, it can be started after the POT instruction without disturbing the interlace. Thus, either of the following sequences can be used to initiate an interlaced buffer operation.

1.	RPT *0, 1, 4 (buffer control)	Alert the interlace, connect the buffer to the device and start the operation at 4 characters word
	EOM 01000X	Transmit the contents of bit positions 22 and 23 of this instruction to bits 0 and 1 of the WCR. (If this instruction is omitted, bits 0 and 1 of the WCR will be zero.)
	POT WORD	Transmit the contents of bit positions 0-9 of location WORD to bit positions 2-11 of the WCR, transmit the contents of bit positions 10-23 of location WORD to bit positions 12-25 of the interlace register (MAR), and inhibit the End-of-Word interrupt.

2. EOM * (or ALC)	Alert the interlace.
EOM 01000X	Set 2 high-order bits of WCR.
POT WORD	Set 10 low-order bits of WCR, set MAR, and inhibit the End-of-Word interrupt.
RPT 0, 1, 4	Connect the buffer to device number 1 and start the operation of 4 characters per word

The second method allows the computer operator to manually single step through the sequence of instructions.

Terminating Interlaced Inputs

If the word count is equal to the number of words in the input record, both the interlace and the W buffer disengage simultaneously, the End-of-Transmission interrupt signal is actuated if the interrupt system is enabled, and the End-of-Word interrupt returns to normal operation. If the word count is greater than the number of words in the input record, the interlace register and the buffer (UAR) both automatically disengage when the end-of-record is detected. The End-of-Transmission interrupt is generated, but the last characters received will not be stored in memory if the number of characters in the record is not an integral multiple of the number of characters per word specified by the buffer control EOM. This last, partial word may be stored with a WIM without "hanging up" the computer.

If the word count is less than the number of words in the record, the interlace register word count reaches zero. Then the interlace automatically disengages, but the W buffer continues to input. When the next word is ready in the W buffer, an End-of-Word interrupt signal is actuated. If the interrupt system is enabled, the interrupted program can then store that word and following words into memory by single-word transmission; or it can set up the interlace again without disturbing the buffer and let the interlace register transfer the word and following words to memory; or the program can disconnect the buffer (DSC) or skip the remainder of the record (SRC or SRR). If the remainder of the record is ignored-, the buffer continues to input characters into the SCR (resulting in a character rate error) until an end of record signal disconnects the buffer.

Terminating Interlaced Outputs

When the interlace register is controlling an output operation, the terminate output command is automatically sent by the interlace register to the W buffer after the word count reaches zero. When the buffer has properly sent out the last word, the buffer and the interlace automatically disengage, an End-of-Transmission interrupt signal is generated, and the End-of-Word interrupt returns to normal, single-word operations.

DIRECT PARALLEL INPUT/OUTPUT

EOM and SKS instructions control parallel input/output operations like they control single-word buffer operations.

Two instructions, PARALLEL OUTPUT (POT) and PARALLEL INPUT (PIN), cause any word in core memory to be presented in parallel at a peripheral connector, or, inversely, cause signals sent to a peripheral 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 peripheral device involved in the input/output operation. This signal tells the device to send its data word as soon as it is operational. When a device 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 data word into a specified memory location without disturbing any arithmetic registers. The computer waits 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 peripheral 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 peripheral device. The computer waits during the execution of POT until it receives the ready signal from the external device.

Special system requirements demand that complete words of control information or data be transferred between the central processor and the special external devices. The PIN or POT preceded by the activating EOM satisfies this requirement. The EOM alerts the system device by specific address and the PIN or POT transfers the required word. Thus, the EOM and PIN/POT operate in all special systems as they do in interlaced-W buffer/standard peripheral equipment operations. That is, the EOM activates and alerts the special device and the PIN/POT transfers 24 bits to or from the effective memory location specified. To avoid a possible computer hang-up, the SKS instruction can test the ready signal of the special device after the EOM but prior to the PIN/POT. If the ready signal from the external device sets one of the priority interrupts, the 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 persists through the POT or PIN execution, no computer hang-up occurs.

PIN PARALLEL INPUT

0	X	0	33	I	M
0	1	2	3 8	9	10 23

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

Affected: (M) Timing: 4 + wait

POT PARALLEL OUTPUT

0	X	0	13	I	M
0	1	2	3 8	9	10 23

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

Affected: External Device Timing: 3 + wait

SINGLE-BIT INPUT/OUTPUT

EOM and SKS instructions also perform single-bit input/output control and testing for special or standard devices. One configuration of EOM transmits a single signal of approximately 8 microseconds duration to an external connector and also provides the connector with a 15-bit address for the destination of this signal. SKS tests for the presence of a similar signal on an external connector and skips accordingly.

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

Execution of an EOM (system mode) causes an 8-microsecond signal to be transmitted to one of a possible 32,768 signal destinations.

EOM ENERGIZE OUTPUT M

*	02	*	1	1	
0 2	3 8	9	10	11	12 23

Bit positions 3 through 8 contain the EOM instruction code.

Bit positions 10 and 11 contain the system mode indicator.

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

Bit positions 0, 1, and 9 are reserved for special system address bits.

Affected: Special Device Timing: 1

Execution of an SKS (system test mode) causes the 15-bit address field of the SKS instruction to be presented to the collection of special system devices. If the addressed external device is supplying a set signal to the central processor, the next instruction in sequence from the SKS is executed. If there is no signal, the next instruction in sequence is skipped and the following instruction is executed.

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

*	40	*	1	1	
0 2	3 8	9	10	11	12 23

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

5. CONTROL CONSOLE

The basic 910 Computer system includes a console for operator control. This console contains a control panel (see Figure 6) and may contain an input/output typewriter (see Section 6, Peripheral Equipment). The control panel, which is connected directly to the central processor, contains switches for control of operation, and illuminating indicators for visual display. See Appendix C for computer operating procedures.

CONTROLS

POWER

The POWER switch turns the computer power system on or off, and is lighted in the on condition.

REGISTER

This four-position, rotary switch selects the internal register to be displayed in REGISTER DISPLAY. The selectable registers are:

C register (arithmetic and control)
A register (main accumulator)
B register (extended accumulator)
X register (index)

RUN-IDLE-STEP

This three-position, toggle switch has two stationary positions (RUN and IDLE) and a spring-loaded, momentary position (STEP). With this switch in the RUN position, instruction execution occurs automatically at computer speed. When this switch is placed in the IDLE position, the computer "idles" immediately after obtaining an instruction from memory. If at the same time, the REGISTER switch is in position C, the contents of the C register (the next instruction to be executed) is shown in REGISTER DISPLAY. Moving the switch to STEP causes the computer to execute the current contents of the C register, load the C register with the next instruction in sequence, and automatically return to an idle state. The STEP switch must be allowed to return to IDLE before it can be activated again to execute the next instruction.

START

This switch initializes the control section of the computer: it resets the W and Y buffers; clears the P register, OVERFLOW indicator, MEMORY PARITY error indicator, clears all interrupt levels, disables the interrupt system; and sets a HALT (00) instruction in the C register. For proper operation, operator should set the RUN-IDLE-STEP switch to IDLE and the REGISTER SELECT switch to C before pressing this switch.

HOLD

Placing the HOLD switch in the up position causes the current contents of the program counter to be held and prevents it from counting. At this time, the operator can insert instructions into the C register and execute them without stepping the program counter (P register).

FILL

Raising and releasing this switch automatically reads one word (4 characters) from Paper Tape Reader 1 into memory location 00002, simultaneously sets the X register to the value 77777771 (-7), and executes the instruction in location 00002. A short program (called a "bootstrap" then loads and executes itself without further action by the operator. The bootstrap program loads a binary tape of any length into any portion of memory.

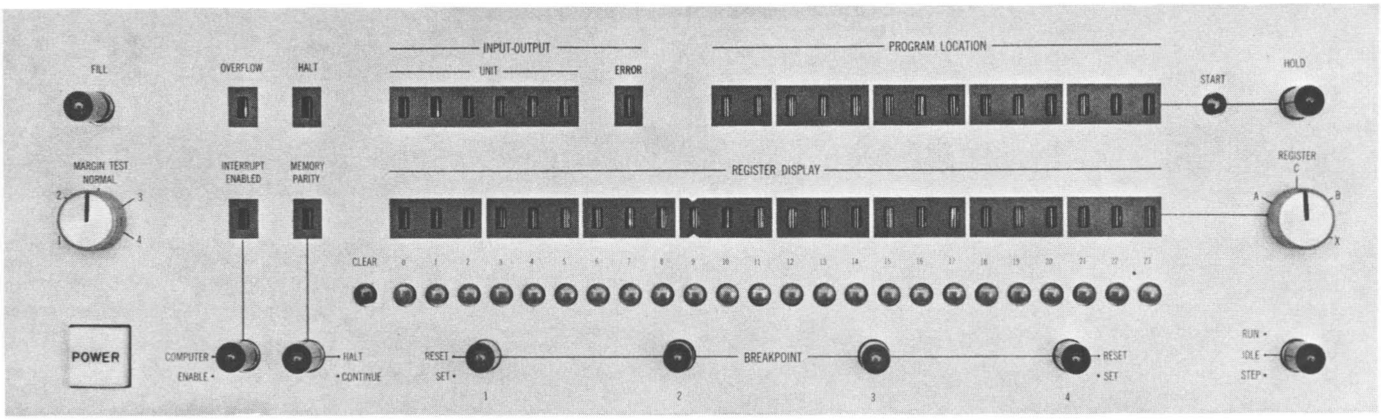


Figure 6. XDS 910 Control Panel

Bootstrap Program:

Location	Instruction
0002	WIM 012, 2
0003	BRX 02
0004	LDX 011
0005	WIM 0, 2
0006	SKS 021000
0007	BRX 05
0010	(First instruction)
0011	(Starting address with indirect address tag)

Appendix C, page A-11, "Fill Procedure", specifies how to load a program using the FILL switch.

MEMORY PARITY

If an operand, instruction, or access from memory encounters a parity error, MEMORY PARITY lights. When the switch below the indicator is in the HALT position, the computer enters the idle state whenever a memory parity error occurs. Setting the switch to CONTINUE clears the MEMORY PARITY indicator and the computer continues normal operation. If the switch is in the CONTINUE position when a memory parity error occurs, the computer ignores the error.

INTERRUPT ENABLED

This indicator is on if the interrupt system is enabled, and is off if the system is disabled. The switch below this indicator allows the operator to enable the interrupt system. In the ENABLE position, the switch enables the interrupt system regardless of program operations; in the COMPUTER position, the switch allows the program to enable or disable the interrupt system. The switch is stationary in the COMPUTER position and momentary in the ENABLE position.

BREAKPOINT

The BREAKPOINT switches are externally controlled, internally testable program switches. Breakpoint test instructions test them.

CLEAR

Pressing this button clears the REGISTER DISPLAY indicators and register selected by the REGISTER switch.

DISPLAYS

REGISTER DISPLAY

This display consists of 24 binary indicators that show the contents of the register selected by the REGISTER switch. With the RUN-IDLE-STEP switch in IDLE, pressing the CLEAR pushbutton clears the selected register to all zeros. By pressing the pushbuttons beneath selected indicators, the operator may enter any desired configuration of 1-bits into the selected register. If the operator clears or changes this display, the actual contents of the selected register change identically.

PROGRAM LOCATION

This display consists of 14 binary indicators that show the current contents of the P register (Program Counter). When the RUN-IDLE-STEP switch is in IDLE, the indicators in this display contain the memory address of the next instruction to be executed. This display (and thus the P register) may be changed by entering a BRU and an address into the C register with the set buttons and then executing the BRU instruction.

HALT

This indicator displays the current status of the Halt flip-flop. If the computer executes a HALT (HLT) instruction, the Halt flip-flop is set and the HALT indicator is turned on. Placing the RUN-IDLE-STEP switch in IDLE clears the Halt flip-flop and turns off the HALT indicator.

INPUT-OUTPUT

The UNIT indicators display the current contents of the W buffer unit address register, which is the 6-bit unit address code for the peripheral device currently connected to the W buffer (see Table 3). The ERROR indicator displays the current status of the W buffer error flip-flop. When the ERROR indicator is on, it indicates that an error has occurred during the previous input/output operation. The indicators can be cleared by pressing START.

OVERFLOW

This display shows the status of the OVERFLOW indicator.

6. PERIPHERAL EQUIPMENT

This section describes some of the input/output devices that can be attached to a buffer, specifies the EOM and SKS instruction 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 the W buffer 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 the W buffer at 4 characters/word, on unit 1.

RKB 0,1,4 READ KEYBOARD
EOM 02601 0 02 02601

This instruction connects the typewriter to the buffer, turns on the typewriter (lights the input light) and initializes the buffer to assemble 4 characters/word.

When a typewriter input operation immediately follows typewriter output, the program must allow 40 milliseconds (5000 computer cycles) after the buffer disconnects before executing RKB. Otherwise, the last character transmitted to the typewriter may reappear as the first character read back into the buffer.

TYP 0,1,4 WRITE TYPEWRITER
EOM 02641 0 02 02641

This instruction connects the typewriter to the buffer, turns on the typewriter, and initializes the buffer to output 4 characters/word.

TERMINATING TYPEWRITER INPUT/OUTPUT

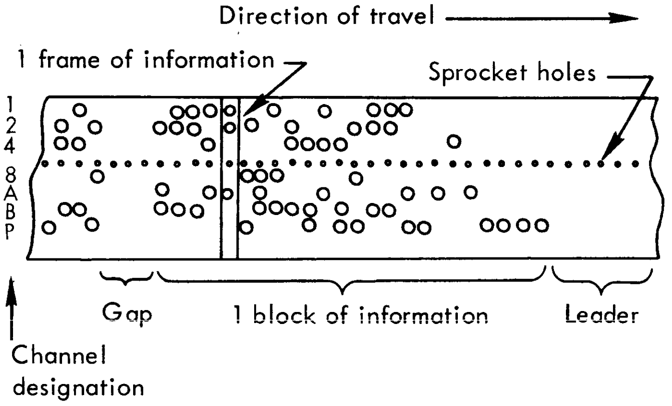
Since the typewriter is not a record-oriented device, it provides no terminating signals. Thus, the program must disconnect the typewriter at the end of an input with a DISCONNECT CHANNEL (DSC) instruction. If typewriter output is accomplished using interlace, the interlace control automatically terminates output (clears the unit address register in the buffer). If single-word transmission is used for typewriter output, the program must terminate the output operation with a TERMINATE OUTPUT (TOP) instruction. If the buffer unit address register is not cleared after a typewriter input or output, the W BUFFER READY TEST (BRTW) 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 buffer the error flip-flop in the buffer is set and the ERROR indicator on the control panel is turned on.

PAPER TAPE INPUT/OUTPUT

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 the first block of tape 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 in 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 takeup spool.)
3. Place the tape actuator in the RUN position.

Example: Typewriter Output then Input

This routine types out the message: PROGRAM, then returns the carriage to await the input of a single character. Input terminates with a carriage return typed by the operator; thehousekeeping operations necessary to determine when the carriage return has been input are not given.

Location	Instruction	Address	Comments
TYPE2	TYP	0,1,4	Connect W buffer to typewriter 1
	MIW	MSSGE	Output first word of message.
		MSSGE+1	The central processor "hangs up" on this instruction until the fourth character from the preceding instruction has cleared the W buffer into the single character buffer for output. This MIW executes filling the W buffer with the contents of location MSSGE+1.
	TOP	0	Terminate output when W buffer system is clear.
	BRTW BRU	 $-1	The program "hangs up" here until the buffer transmits the last character. The symbol $ is an assembler expression which means "the current setting of the location counter at assembly time" – the memory location of this instruction.
	LDX	WAIT	Load the X register with the contents of location WAIT (00043073), which furnishes a 1-bit in bit position 9 and a count (-2500 in two's complement form) in bit positions 10-23.
	NOP		No operation
	BRX	$-1	This instruction causes the instructions NOP and BRX to be executed 2500 times, providing a 5000-cycle (40-millisecond) delay in the program.
	RKB	0,1,1	Connect typewriter 1 to the W buffer for input at one character per word. The octal configuration is 0 02 02601.
	WIM	KEYWD	The computer "hangs up" on this instruction until a character enters the W buffer from the keyboard; then the word in the buffer is placed into location KEYWD. The input character is in bit positions 18 through 23 of KEYWD. Bit positions 0 through 17 are unpredictable.

At this point, the word in KEYWD is placed elsewhere in memory and the routine returns to the WIM above. When executed, a test is made to determine if the new input character is the carriage return code. Indexing and/or indirect addressing can be used with the WIM to facilitate input. When the carriage return is detected, the following is exeucted.

Location	Instruction	Address	Comments
	DSC	0	This instruction disconnects the W buffer by immediately clearing the unit address register to zero. The octal configuration of this EOM is 0 02 00000.
	BRU	OUT	Return to main program.
MSSGE	47514627 51214452	These two locations contain the octal configuration of the message.	

RPT 0,1,4 READ PAPER TAPE
EOM 02604 0 02 02604

This instruction connects the paper tape reader to the W buffer, 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 the first frame of the gap between blocks, generates an End-of-Transmission signal, and disconnects from the buffer (clears the buffer 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 3700 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, DSC must be executed within 0.3 millisecond (approximately 40 computer cycles) after the last character is read. Otherwise, characters continue to enter the buffer and a character rate error occurs. (The program may also store the unwanted remainder of the record in an unused portion of memory. When the reader disconnects, after reading the last character, an End-of-Transmission interrupt occurs if the interrupt system is enabled.)

ERROR CONDITIONS

If a parity or character rate error occurs during a paper tape read operation, the buffer error flip-flop is set and the ERROR indicator on the computer control panel is turned on.

Example: Read a block of binary tape of any length using the W buffer without interlace or interrupts.

This routine reads data into memory, at 4 characters/word, starting at location 02000.

Location	Instruction	Address	Comments
READ	PZE		Reserve the entry location. This routine assumes that the address of the last program instruction executed before the routine was entered will be stored in bits 10-23 of the PZE instruction.
	LDX	START	Load the X register with the contents of location START, which contains the starting address for storage.
	RPT	0,1,4	Initiate the paper tape read on the W buffer with paper tape reader 1, at 4 characters/word.
IN	WIM	0,2	Transfer the contents of the W buffer into the location specified by bits 10-23 of the X register.
	BRTW		Test for buffer readiness. If the paper tape reader has sensed a blank frame, the unit address is cleared, and the computer will skip the next instruction and execute the following instruction. If the paper tape reader has not sensed a blank frame, the buffer is still connected to the paper tape reader, and the computer will execute the next instruction.
	BRX	$-2	Add 1 to the contents of the X register and take the next instruction from location IN.
	BRR	READ	Return program control to the main program. The address of the last word stored in memory is contained in the X register.
START	00042000		This location (as set up by the main program) contains the starting address 02000 for storage of the paper tape information, as well as a 1 in bit 9 (0004) to provide for incrementing the X register.

If the number of characters on the tape is an integral multiple of 4, the last word stored in memory is all zeros. If the number of characters is not an integral multiple of 4, the last word stored in memory contains zeros in the least significant portion of the word.

Example: Read paper tape of known length

This routine reads a 64-character block from paper tape into memory beginning at location 02000. The routine uses the 4-character/word mode, making the input 16 words; it uses the optional, W buffer interlace. The routine is a closed subroutine using interrupts.

Location	Instruction	Address	Comments
RDPT	PZE		This assembler instruction reserves the entry location.
	CLR STA	SWICH	These two instructions clear an "input-finished" indicator.
	EIR		Enable the interrupt system.
	RPT	*0, 1, 4	This instruction connects paper tape reader 1 to the W buffer, specifies 4-characters/word mode and alerts the interlace (the *). The octal configuration for this EOM is 0 02 42604.
	POT	REED	This instruction transmits the word count and starting address to the interlace. The input operation begins.
	BRR	RDPT	This instruction branches back to the main program while the paper tape input is in progress.
REED	01002000		This location contains the starting location 02000 in bit positions 10-23 and the word count 16(020) in bit positions 0-9 (right justified).

When the buffer has transmitted the 16 words into memory, the End-of-Transmission (I2W) interrupt transfer to location 033 occurs, executing the instruction in that location.

Location	Instruction	Address	Comments
033	BRM	FNISH	This instruction branches to the End-of-Transmission processing portion of this routine.
FNISH	PZE		This assembler instruction reserves the entry location from the interrupted program.
	MIN	SWICH	An operation-finished switch is set.
	BRU	*FNISH	This instruction branches back to the main program and clears interrupt level 033 from the active state (* means indirect address bit set).

PAPER TAPE PUNCH

The paper tape punch is used primarily for punching programs and/or data to be loaded back into memory later. 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 punched manually by depressing the FEED button until the desired amount of leader is produced.

The punch tape instructions to follow are coded for the W buffer, 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 buffer, starts the punch motor (if not already on), and initializes the buffer to output 4 characters/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 tape punch continues to punch as long as it receives characters from the buffer, regardless of the infrequency of transmission. The punch operates at 60 characters per second, asynchronously. If the buffer does not supply characters 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 buffer unit address register is not cleared, and the computer will not skip the next instruction when BRTW is subsequently executed. If the punch operation is accomplished under interlace control, a TERMINATE OUTPUT (TOP) instruction is automatically generated when punch operations are completed. 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. Alternatively, 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 buffer error flip-flop is set and the ERROR indicator on the computer control panel is turned on.

Example: Punch Paper Tape

This routine punches a block of eight words from locations 02000 through 02007. A 1-inch gap precedes and follows the block. The routine is a closed subroutine that does not use interrupts; the interrupt system is disabled. The routine uses the index register and does not restore it.

Location	Instruction	Address	Comments
PUNCH	PZE		Reserve the subroutine entry location.
	BRTW		W buffer ready?
	BRU	$-1	W buffer not ready.
	LDX	COUNT	Load the X register with the contents of location COUNT. COUNT is assumed to contain 07777770 (-8)
	PTL	0,1,4	This instruction connects the W buffer to paper tape punch 1 and specifies 4-character/word mode. The instruction asks for leader to be punched. The octal configuration for this EOM is 0 02 00644.
BACK	MIW	02010,2	This instruction transfers to the W buffer the word in location 02010 modified by the index register (2010 + (X))→ buffer . Then the next instruction in sequence is executed. The routine returns to this instruction and "hangs up" until the W buffer is free and can receive a new word of data.
	BRX	BACK	Increment the X register, then test for negative result. If negative, transfer to BACK, if positive (or zero), execute the next instruction.
	TOP	0	This instruction is executed in one cycle and then the next instruction following TOP is executed. The execution of TOP causes the W buffer to be disconnected when the last character has been shifted out of the buffer and transmitted out of the single character register.
	BRTW		Output completed?
	BRU	$-1	Output not complete
	EXU	$-6	This instruction causes the PTL instruction to be executed, producing a 1-inch gap at the end of the block.
	EXU	$-4	This instruction causes the TOP instruction to be executed, disconnecting the buffer after the blank frames have been punched.
	BRR	PUNCH	Return to the main program.

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 7, 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 (rows 12-3) of column 1 form the first character (with the most significant bit in row 12), the bottom 6 rows (rows 4-9) form the next character (with the most significant bit in row 4). The first character in column 1 enters bit positions 0-5 of the first computer word; the second character of column 1 enters positions 6-11, and so on. Thus, a single card may represent up to 160 characters (40 words at 4 characters/word) in binary format.

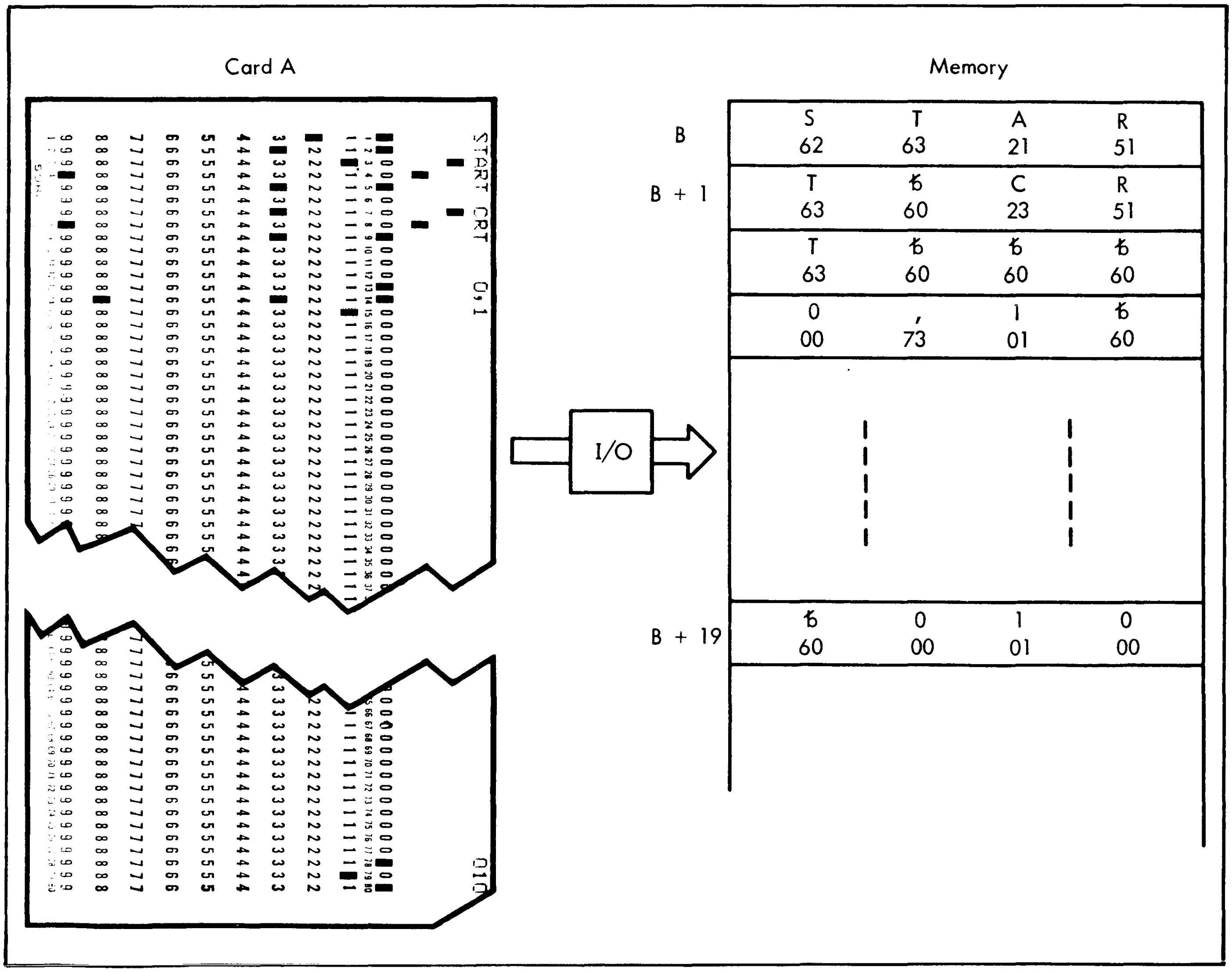
CARD READER

Before beginning 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 in 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 buffer (BRTW or BRTY)
   b. Test card reader (see 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



B	S 62	T 63	A 21	R 51
B + 1	T 63	ƀ 60	C 23	R 51
	T 63	ƀ 60	ƀ 60	ƀ 60
	0 00	, 73	1 01	ƀ 60
B + 19	ƀ 60	0 00	1 01	0 00

Figure 7. Card Read into Memory in Hollerith

Example: Read Hollerith Card

This program reads one card in Hollerith format under interrupt control, using single-word transmission.

```
Location   Instruction   Address    Comments

CARDR      PZE                      Reserve a location for subroutine entry.
           EIR                      Enable interrupt system.
           CLR                      Clear A and B registers.
           STA           DONE       Store zeros in (clear) location DONE.
           BRTW                     Skip the next instruction if the W buffer is ready.
           BRU           $-1        Computer hangs up in this loop until the W buffer is ready.  (A BRM to
                                    a routine that alerts the operator could be used instead.)
           CRT           0,1        Skip the next instruction if card reader 1 on the W buffer is ready to
                                    feed and read.
           BRU           $-1        Computer hangs up in this loop until the card reader is ready.  (Here
                                    too, BRM to an operator-alert routine may appear instead.)
           RCD           0,1,4      Connect card reader 1 to the W buffer, start a feed and read cycle,
                                    and assemble 4 characters/word in Hollerith format.
           BRR           CARDR      Branch back to main program.
```

The reader reads characters into the buffer. When the buffer is full, an End-of-Word interrupt is initiated at location 031.

```
031        BRM           RETRN
           .
           .
           .
RETRN      PZE                      Reserve the return entry location from the main program.
           WIM           *READ      Transfer the word in the W buffer into the location specified by the con-
                                    tents of location READ.
           MIN           READ       Increment the input location by one.
           BRU           *RETRN     Branch back to the main program and clear interrupt level 031.
```

When the reader reads the last four characters from the card and reaches the end of the card, the End-of-Transmission interrupt to location 033 occurs (the End-of-Word interrupt is inhibited).

```
033        BRM           LAST       W buffer is disconnected.
           .
           .
           .
LAST       PZE                      Reserve location for subroutine entry.
           BETW                     Skip the next instruction if End-of-Transmission interrupt was not be-
                                    cause of a read or feed check, or if no error has occurred.
           BRM           ERR        Branch to error-servicing subroutine.
           MIN           DONE       Increment contents of location DONE by 1.  Location DONE may be
                                    inspected by the main program to verify successful completion of the
                                    read operation.
           BRU           *LAST      Branch back to main program and clear interrupt level 033.
```

(or 160 binary-coded) characters to the buffer, generates an end-of-record signal, and waits for the next EOM. The card reader instructions to follow are coded without interlace, using the W buffer at 4 characters/word, for unit number 1.

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.

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 an SDS internal code. This mode can read up to 80 characters (20 words at 4 characters/word) from a single card.

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

This instruction causes the reader to stop transmission of characters to the buffer. The remaining characters are not checked for validity, but a read check, feed check, or end-of-record condition still cause an End-of-Transmission interrupt and disconnect the card reader from the buffer.

Card Reader Tests

The card reader tests to follow are coded for the W buffer, using unit number 1.

CRT 0,1 CARD READER READY TEST
(Skip if Card Reader Ready)
SKS 012006 0 40 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 First Column)
SKS 014006 0 40 14006

This test determines if the first column is about to be read by the card reader. Since the elapsed time between execution of a card reader EOM and reading of the first column is approximately 85 milliseconds (10,625 computer cycles), this test allows the computer to perform other operations in the interval. If FCT is executed less than 1.2 milliseconds (approximately 150 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 Card Reader not at End of File)
SKS 011006 0 40 11006

This test determines if an end-of-file (EOF) condition exists for the card reader. This condition exists when the hopper is empty and the EOF ON indicating switch is on (lighted). (When an end-of-file condition exists, the END OF FILE indicator is also lighted.) 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 read motor Turn on FEED CHECK indicator Turn on NOT READY indicator Set error flip-flop in buffer (test with BETW) Disconnect card reader from buffer (clear unit address register) Generate End-of-Transmission interrupt signal
2. Read malfunction	Turn on READ CHECK indicator (other responses are identical to feed malfunction)

Condition	Card Reader Response
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-Transmission interrupt signal Disengage card reader motor Wait for new EOM

When reading cards in the single-word mode of transmission, a W BUFFER READY TEST (BRTW) should be issued before each WIM to ensure that the card reader has not become disconnected (read or feed check). Otherwise, the computer will hang up on the WIM 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 indicators 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 (lights) and the card hopper is empty, the end-of-file condition can be 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 (lights) whenever a malfunction occurs in the read station during a read cycle.

VALIDITY CHECK   This indicator turns on (lights) 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 the EOF ON and END OF FILE.

CARD PUNCH

Before starting a card punch operation, the 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 buffer (BRTW or BRTY)
   b. Test card punch (see 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 an entire card image to the card punch coupler 12 times for each card. A card image consists of 80 characters of Hollerith-coded information or 160 characters of binary-coded information. Before each row is punched, the coupler examines the card image and forms an appropriate row image which it loads into the buffer. 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 the W buffer 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 Hollerith format. A transmission of 80 characters (20 words at 4 characters/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 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 interlace outputs.

Example: Punch Hollerith Card

This program punches one card in Hollerith mode. It is a closed subroutine that uses interlace and interrupts. The contents of location COUNT counts the 12 times the program presents the card image to the punch.

Location	Instruction	Address	Comments
CARDP	PZE		Reserve the entry location for this subroutine.
	EIR		Enable the interrupt system.
	LDA STA	CARDP ENTR2	LDA and STA place the location of the last program instruction executed into location ENTR2.
	MIN	ENTR2	Add 1 to the stored contents of ENTR2, forming a return address to the main program.
	LDA STA	NEG 12 COUNT	These two instructions initialize a memory location to be used as a row counter.
GETRW	BRTW		Test the W buffer for a ready condition.
	BRU	$-1	This instruction is executed if the buffer is not ready.
	CPT	0,1	Test card punch 1 on the W buffer for a ready condition.
	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 also be placed here.
	PCD	*0,1,4	This instruction is executed if the punch is ready. It alerts the W buffer with interlace, connects card punch 1 to the W buffer, and starts a card moving toward the punch station. Four characters per word and Hollerith format are specified. The octal configuration of this instruction is 0 02 02646.
	POT	PNCH	Transmits the word count and starting address to the buffer.
	BRU	*ENTR2	Branch back to the main program (and clear the interrupt on subsequent returns to GETRW).
PNCH	01202000		The word in PNCH specifies that 20 words will be output from memory beginning in location 02000.
COUNT	77777764		Note that the card image must be sent to the buffer a total of 12 times to punch a card.
NEG 12	77777764		

The execution of the main program continues while the interlaced buffer performs the output. When the buffer finishes with the output for punching one row, an interrupt occurs at interrupt level 033, the End-of-Transmission location for the W buffer.

033	BRM	ENTR2	
	.		
	.		
ENTR2	PZE		Reserve a location for routine entry.
	MIN SKN	COUNT COUNT	Increase the contents of COUNT (the row counter) by 1 until a zero value results; if more rows are to be punched, the next instruction is skipped and the following instruction is executed.
	BRU	*ENTR2	Return control to the main program and clear the interrupt.
	BRU	GETRW	Transfer to the initiate-punch EOM.

Card Punch Tests

The card punch tests that follow are coded for the W buffer, using unit number 1.

PBT 0,1 PUNCH BUFFER TEST
(Skip if Punch Buffer Ready)
SKS 012046 0 40 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 40 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 but the card does not feed properly (or the punch buffer is not loaded at punch time), the error flip-flop in the buffer is set.

LINE PRINTER

XDS 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 with a rotating character drum and a bank of 132 print hammers. The drum rotates 56 characters, in lines of 132 each, pas the hammer bank. Upon command from the computer, the print hammers selected by the buffer drive the paper against the ribbon and onto the appropriate character typeface as it passes the print position. The characters are transmitted sequentially to the printer buffer for storage before printing. A program-controlled format tape loop provides fixed (or preselected) line space control. Upspacing of 1 to 7 lines, as well as page control (upspacing to line positions designated on the format loop), 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 8, for XDS line printers consists of eight switches and indicators.

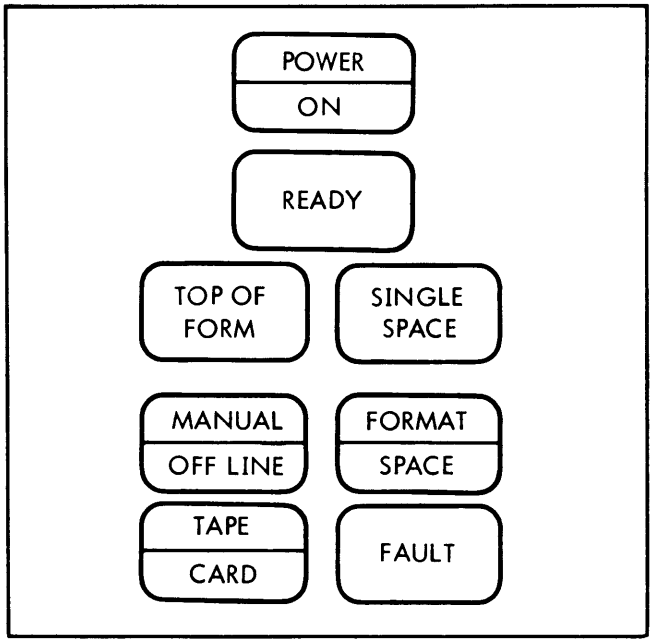


Figure 8. Printer Control Indicator Lights and Switches

POWER/ON This switch is an alternate action switch. The computer must be turned on for this switch to be 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.

READY When the printer is initially turned on, this 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.

TOP OF FORM   Pressing TOP OF FORM causes the printer to position paper according to format 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.

SINGLE SPACE   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.

FAULT   This 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[†]   This control is a combination of one switch and two independent indicators. The computer or the operator may initiate off-line operation, which is indicated as being in process 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 lighted and remains lighted until the switch is again pressed. OFF-LINE is normally reset when the end-of-file is detected from the input unit. Pressing READY also resets OFF-LINE, that is by switching the printer from the "ready" to the"not ready" state.

FORMAT/SPACE[†]   This dual indicator 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.

TAPE/CARD[†]   This dual-indicator 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 the buffer and specifies a character transmission of 4 characters per word.

This instruction is followed by the transmission of up to 132 characters. The characters are printed left-justified on the page if the character count is less than 132; unused character positions appear as blanks on the right side of the printed page. If the character count is more than 132, the printer produces an undetectable error.

The following control instructions are coded for the W buffer using unit number 1:

POL 0,1   PRINTER OFF-LINE
EOM 010260   0 02 10260

This instruction places the printer off-line and initiates an off-line print operation. The selected input device for the printer (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 (3125 cycles)

Upspace more than 1 line: add 10 milliseconds (1250 cycles) for each additional line.

[†] 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.

LINE PRINTER TESTS

The line printer tests that follow are coded for the W buffer, using unit number 1.

PFT 0,1 PRINTER FAULT TEST
(Skip if no Printer Fault)
SKS 011060 0 40 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 40 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 new PSC or PSP. A dummy PLP instruction may be issued between two space instructions (PSC or PSP). This dummy instruction will provide the timing required. PRT may be used after the dummy PLP instruction 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 40 14060

This instruction tests the printer for paper position. If the paper is positioned at the end of page (defined 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).

OFF-LINE PRINTING

The facility for off-line printing is an optional feature allowing the line printer to produce printed records from card or magnetic tape sources without computer attention. Character transmission proceeds directly from the source to the printer, and the buffer 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 of 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, which interprets the first character of each line image as a format control character (see Table 4) and performs the indicated function before printing the line.

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 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.
5. Press MANUAL/OFF switch.
6. Press READY switch on, which initiates actual data transfer.

Table 4. 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
40	-	Do not upspace
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

Example: Print Two Lines

This program prints two lines at the top of a page with a single upsapace between. Assume that the printer is ready or is becoming ready after a print operation. The program is a closed subroutine for printer 1 on the W buffer; interrupts are not used (the interrupt system is assumed to be disabled).

```
Location  Instruction  Address      Comments

PRINT     PZE                       Save this location for subroutine entry.

          LDX          = -33        Load index with -33, the length of a line image in words. (The =
                                    is a META-SYMBOL literal statement)

          BRTW                      W buffer ready?

          BRU          $-1          W buffer not ready.

          PRT          0, 1         Test for printer ready. If not ready, execute the next instruction.
                                    If ready, skip the next instruction.

          BRU          $-1          Not ready, return to the test.

          PSC          0, 1, 1      This instructs the printer to move paper to top of the page.

          PLP          0, 1, 4      Connect line printer 1 to the W buffer; specify 4 character/word mode.

          MIW          LINE1+33,2   Output 1 word from image for line 1.

          BRX          $-1          Repeat until transmission of line image is completed.

          TOP          0            Terminate output after the last character is transmitted

          LDX          = -33        Reload index with -33.

          PRT          0,1
          BRU          $-1          Wait for printer to become ready after printing first line.

          PSP          0, 1, 1      Upspace printer 1 line.

          PLP          0, 1, 4      Connect line printer 1 to the W buffer; specify 4 character/word mode.

          MIW          LINE2+33,2   Output 1 word from image for line 2.

          BRX          $-1          Repeat until transmission for line image is complete.

          TOP          0            Terminate output after the last character is transmitted

          BRR          PRINT        Branch back to main program.
```

Printing Off-Line Under Computer Control

The procedure for computer control of off-line printing is:

1. Turn the equipment on and prepare the desired input device for operation
2. Select desired input device by means of the TAPE/ CARD switch.
3. Select either the FORMAT or SPACE mode.
4. Press the READY switch on.
5. Under program control, test the tape or card unit and the line printer for "ready" condition.
6. 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. Upon termination of an off-line operation, a physically connected off-line printer system return on-line, provided the MANUAL indicator is off. 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 XDS 910 Computer 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 Mylar side 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 positioned 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 recorded at 200 bits-per-inch density after the end-of-reel marker is sensed.

Characters are recorded on tape in seven parallel channels. A change in the magnetic flux in a channel is used to record a 1-bit for a given character position. No change in magnetic flux indicates a 0-bit. Six of the channels are used for information; the seventh 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 tape are recorded without alteration. Data also can be recorded in binary-coded decimal (BCD) mode using even parity. In this mode, characters from the tape 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. 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. This procedure permits more than one file of information to be written on a single tape.

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 buffer even though the record itself is correct. As a general rule, the error indicator should be ignored after a 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 an advisable programming practice 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 that might arise through subsequent tape repositioning.

MAGNETIC TAPE UNIT TESTS

The magnetic tape unit tests that follow are coded for the W buffer, with n being the number (0-7) of the magnetic tape unit.

TRT 0,n TAPE READY TEST
(Skip if Tape Unit not Ready)
SKS 01041n 0 40 1041n

Tape unit n is tested for not ready. If the tape is ready, the next instruction in sequence is executed; if the tape is not ready, the next instruction in sequence is skipped and the following instruction is executed.

A tape is not ready:

> if there is no physical unit set to the logical unit number being tested.
>
> If the selected unit is not in the automatic mode, or if the tape is in motion for any operation.

FPT 0,n FILE PROTECT TEST
(Skip if Tape not File Protected)
SKS 01401n 0 40 1401n

Tape unit n is tested for the presence of a file-protect ring. If the file-protect ring is inserted, the next instruction in sequence is skipped and the following instruction is executed; if the file-protect ring is not inserted, the next instruction in sequence is executed. The skip will not occur if there is no logical unit n on the buffer. This instruction should be used before any write operation to determine whether it is possible to perform the operation.

BTT 0,n BEGINNING OF TAPE TEST
(Skip if not Beginning of Tape)
SKS 01201n 0 40 1201n

Tape unit n is tested for being positioned at the beginning of the tape. If the tape is not positioned on the load-point marker, the next instruction in sequence is skipped and the following instruction is executed. If the tape is 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 buffer.

ETT 0,n END OF TAPE TEST
(Skip if not End of Tape)
SKS 01101n 0 40 1101n

Tape unit n is tested for being positioned at 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 an 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 backward over the end-of-reel marker. The skip will not occur if there is no logical unit n on the buffer.

DT2 0,n DENSITY TEST, 200 BPI[†]
(Skip if Tape Unit not at 200 BPI)
SKS 01621n 0 40 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 Tape Unit not at 556 BPI)
SKS 06116n 0 40 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 Tape Unit not at 800 BPI)
SKS 01721n 0 40 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 at End/of File)
SKS 013610 0 40 13610

The tape control unit is tested to determine whether or not a tape under its control encountered an end-of-file during the last read or scan operation. If end-of-file

[†] These instructions apply only to 41.7-kc and 96-kc magnetic tape systems.

has not been encountered, the next instruction in sequence is skipped and the following instruction is executed. If end-of-file has been 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 No Tape Gap Signal)
SKS 012610 0 40 12610

The tape control unit is tested to determine whether or not a tape under its control is in motion in the gap following a record. If so, the computer executes the next instruction in sequence; if not, the computer skips the next instruction in sequence and executes the following instruction. This instruction applies 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. During this time, the test instruction does not cause the computer to 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 causes the computer to skip the next instruction.

MAGPAK TEST
(Skip if Tape Unit not MAGPAK)
SKS 01021n 0 40 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, 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 buffer. 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 is given to a tape unit on that buffer. 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 last word of the input area as a 17170000 word.

Once a record has been written on tape, it cannot be assured that any records previously written which follow the new record can be read. This means that a record in the middle of a file cannot be updated or rewritten it it is desired to read the records that follow it. Any errors detected either by the buffer (in character parity) or by the control unit (longitudinal parity) sets the error indicator in the buffer. 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 recorded 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 is used to start the tape in reverse. When the buffer 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 buffer shifts the characters through its word assembly register, but does not consider a word complete until a tape gap is encountered. When the gap is reached, buffer uses the last four characters in the word assembly as the only word read from the record. When scanning in reverse 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 logical 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. 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 with the interrupt system enabled. When the beginning of the record is reached, the first word of the record is assembled into the buffer and the end-of-word interrupt occurrs. A WIM instruction stores the word in memory, and the program checks the word against a search key. If they agree, then the program need only wait for the buffer 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 buffer to become inactive.

RTB 0,n,4 READ TAPE IN BINARY
EOM 0361n 0 02 0361n

Tape unit n is started in a Binary read mode.

RDT 0,1,4 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 Without Interlace

This program reads one record in BCD from magnetic tape unit 1 on the W buffer. The program is a subroutine that uses interrupts. The tape is not at the beginning or end of tape. The end-of-record from the tape determines the number of words to be read in.

```
Location  Instruction  Address   Comments

MTRWOI    PZE                    Reserve a location for subroutine entry.

          TRT          0,1       Test for tape unit ready. If ready, the next instruction in sequence
                                 is executed.

          BRU          $+2       This instruction branches around the not ready exit.

          BRU          NOTRDY    This instruction branches to an assumed routine that determines what
                                 is not ready.

          RTD          0,1,4     This instruction addresses the W buffer, connects magnetic tape unit 1 to
                                 it, specifies 4 characters per word, and specifies the BCD mode. This
                                 EOM has the octal configuration 0 02 02611.

          BRR          MTRWOI    Return to main program.
```

Four characters come from the tape and go into the W buffer. When the buffer fills, the End-of-Word interrupt to location 031 occurs.

```
031       BRM          NXTWRD

NXTWRD    PZE

          WIM          *REED     Transfer the word received in the W buffer into the location speci-
                                 fied by the contents of location REED.

          MIN          REED      Increment the contents of REED for the next data input location.

          BRU          *NXTWRD   Branch back to the main program (and clear the interrupt).
```

When the last four characters come from the tape, the End-of-Word interrupt occurs as usual; at some later time when the tape stops, the End-of-Transmission interrupt to location 033 occurs.

```
033       BRM          LAST      When this interrupt occurs, the W buffer disconnects.

LAST      PZE                    Reserve the entry location.

          BETW                   Test for an error on the W buffer.

          BRM          ERR       Branch to an assumed error routine.

          BRU          *LAST     Transfer back to the main program and clear the interrupt.
```

Example: Read Magnetic Tape with Interlace

This program reads one record from magnetic tape unit 1 on the W buffer. The program is a subroutine that uses the End-of-Transmission interrupt. The tape is not at the beginning or the end of the tape. The routine uses the interlaced W buffer.

```
Location  Instruction  Address   Comments

READT     PZE                    Reserve a location for the subroutine entry.

          TRT          0, 1      This instruction tests for ready on magnetic tape unit 1 on the W
                                 buffer. If magnetic tape unit 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 test is 0 40 10411.

          BRU          $+2       Skip one instruction

          BRU          $-2       Branch 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 activates the W buffer, connects it to magnetic tape
                                 unit 1 and starts tape motion. The instruction also alerts the inter-
                                 lace. The 4-characters/word and BCD modes are specified. The
                                 octal configuration is 0 02 02611.

          POT          REDTP     Transmit the word count and starting address to the buffer.

          BRR          READT     Branch back to the main program and clear the interrupt.
           .
           .
           .
REDTP     06202000               The word in REDTP 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 cause a
                                 031 interrupt to indicate more data in the buffer.
```

The main program continues while the buffer performs the input operation. When finished, the End-of-Transmission interrupt goes to location 033.

```
033       BRM          COMPL     This instruction in interrupt location 033 branches and marks to
                                 COMPL to finish the read operation.
           .
           .
           .
COMPL     PZE                    Reserve a location for the subroutine entry.

          BETW                   Test the W buffer for error. If an error is detected, the next instruc-
                                 tion in sequence is executed. If not, the next one is skipped and the
                                 following instruction is executed. The octal configuration of this
                                 instruction is 0 40 20010.

          BRM          ERST      This instruction branches to an assumed routine to re-read the block
                                 a number of times and, if the error continues, to notify the operator.

          BRU          *COMPL    Return control to the main program and clear interrupt level 033.
```

MAGNETIC TAPE UNIT CONTROLS

The following instructions are used for controlling magnetic tape units. These instructions are EOMs in the input/output control mode.

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. This instruction does not affect the buffer in any way.

RTS 0 CONVERT READ TO SCAN
EOM 014000 0 02 14000

The tape unit currently on the buffer 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 buffer is instructed to skip the remainder of the record being read. This instruction applies only to 41.7-kc and 96-kc magnetic tape systems.

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 terminaton 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 buffer that gap has been reached. If no further write instruction is received within one millisecond, the tape is stopped and disconnected from the buffer.

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 erased and might 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 so addressed, 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 correcting 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 appeared 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 buffer 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 instructions to the tape units specify start-without-leader. Since the tape unit generates leader on all write operations automatically, 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. The magnetic tape write instructions to follow are coded for magnetic tape unit n on the W buffer without interlace, using the 4 characters/word mode:

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.

Example: Write Magnetic Tape

This program writes one record on magnetic tape unit 1. The program is a closed subroutine that uses interrupts and the W buffer with interlace.

Location	Instruction	Address	Comments
WRITE	PZE		Reserve a location for the subroutine entry.
	TRT	0,1	Test whether or not magnetic tape unit 1 on the W buffer is ready. The octal configuration is 0 40 10411.
	BRU	$+2	This instruction branches two locations ahead. This instruction is executed if the magnetic tape unit is ready.
	BRU	$-2	This instruction is executed if the tape unit is not ready. Alternately, a BRM to a time loop with the facility to inform the operator that the tape unit will not become ready can be placed here.
	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 40 14011.
	BRM	OPER	Branch and mark to an assumed routine that calls the operator and instructs him to insert file-protect ring on magntetic tape unit 1.
	WTD	*0,1,4	This instruction connects magnetic tape unit 1 to the W buffer, alerts the interlace, specifies BCD transfer mode, and starts the tape moving. Four-characters/word mode is specified. The octal configuration of the instruction is 0 02 04651.
	POT	A	Transmit starting address and word count to the buffer.
	BRR	WRITE	Branch back to the main program.
	• • •		
A	06202000		The word in A specifies that 100 words will be transmitted from memory, beginning with location 02000.

The main program continues while the buffer performs the output. When finished, the End-of-Transmission interrupt goes to interrupt location 033.

033	BRM	FAST	This instruction branches and marks at location FAST.
	• • •		
FAST	PZE		This instruction reserves the entry location.
	BETW		Test for an error occurring during the write operation.
	BRM	ERR	Exit to an assumed error routine.
	BRU	*FAST	Return to the main program and clear the interrupt.

APPENDIX A CONVERSION TABLES

XDS CHARACTER CODES

Characters		XDS	Card	Magnetic Tape
		Internal		BCD Code
Typewriter	Printer	Code	Code	on Tape
∅	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	12③
# 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	12-0④	72
	.	33	12-8-3	73
⌑ or)	)	34	12-8-4	74
[	[	35	12-8-5	75
<	<	36	12-8-6	76
‡ Stop	‡⑤	37②	12-8-7	77

Characters		XDS	Card	Magnetic Tape
		Internal		BCD Code
Typewriter	Printer	Code	Code	on Tape
-	-	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	11-0④	52
S	S	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
ƀ	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	77②	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 to 2777 (Octal)	1024 to 1535 (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 to 3777 (Octal)	1536 to 2047 (Decimal)

Octal-Decimal Integer Conversion Table

4000 to 4777 (Octal)	2048 to 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 to 5777 (Octal)	2560 to 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

APPENDIX B TWO'S COMPLEMENT ARITHMETIC

XDS 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 XDS systems, the sign bit is in the first bit position to the left of the most significant magnitude bit. Thus, if an XDS 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	1 010 001 = $21_8 = 17_{10}$ 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 C COMPUTER OPERATING PROCEDURES

The following are recommended control console operations to accomplish common computer functions.

TURN COMPUTER ON

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Press POWER switch.

LOAD PROGRAM WITH FILL SWITCH

1. Insert the program in Paper Tape Reader 1 (the initial portion of the tape is the bootstrap program).
2. Set the RUN-IDLE-STEP switch to IDLE.
3. Press START switch.
4. Set the RUN-IDLE-STEP switch to RUN.
5. Raise and release the FILL switch.

LOAD PROGRAM WITH LOADING SYSTEM

Refer to the operating procedures furnished with the particular assembler, compiler, monitor, diagnostic, or utility system being used.

EXECUTE PROGRAM

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to C.
3. Press CLEAR and enter a BRU to the program starting location into REGISTER DISPLAY, using the set buttons. Format of the instruction is

   $$000\ \underbrace{000\ 001}_{\text{BRU}}\ \underbrace{0xx\ xxx\ xxx\ xxx\ xxx}_{\text{Program starting location}}$$

4. Set the RUN-IDLE-STEP switch to RUN. The computer then executes the BRU and continues instruction execution at computer speed. Or, set the RUN-IDLE-STEP switch to STEP and release the switch. The computer executes the BRU and returns to the idle state with the contents of the first instruction of the program displayed in REGISTER DISPLAY, and the address of the first instruction of the program displayed in PROGRAM LOCATION. The operator may continue to cause the computer to execute instructions in this manner by repeatedly setting the RUN-IDLE-STEP switch to STEP, allowing the switch to return to IDLE each time. This process is called "stepping" instructions.

INSPECT MEMORY CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to C.
3. Press CLEAR and enter a BRU to the memory location to be examined into REGISTER DISPLAY, using the set buttons. Format of the instruction is

   $$000\ \underbrace{000\ 001}_{\text{BRU}}\ \underbrace{0xx\ xxx\ xxx\ xxx\ xxx}_{\text{Memory location}}$$

4. Set the RUN-IDLE-STEP switch to STEP and release the switch. PROGRAM LOCATION now contains the 14-bit address of the location to be inspected and REGISTER DISPLAY contains the 24-bit contents of the location.
5. To inspect other memory locations, repeat steps 3 and 4 above.

MODIFY MEMORY CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to A.
3. Press CLEAR and enter the desired configuration into the A register, using the set buttons below REGISTER DISPLAY.
4. Set the REGISTER switch to C.
5. Enter 035 XXXXX into REGISTER DISPLAY, using the set buttons. (035 is the octal instruction code for STORE A and XXXXX is the octal address of the memory location to be changed.
6. Set the RUN-IDLE-STEP switch to STEP and release the switch. The computer executes the STORE A instruction and returns to the idle state.

INSPECT/MODIFY REGISTER CONTENTS

1. Set the RUN-IDLE-STEP switch to IDLE.
2. Set the REGISTER switch to the desired register (A, B, C, or X). The contents of the selected register are immediately displayed in REGISTER DISPLAY and may be changed by pressing CLEAR and inserting a new configuration with the set buttons.
3. Set the REGISTER switch back to C before placing the RUN-IDLE-STEP switch into RUN or STEP.

CLEAR HALT CONDITION

1. Set the RUN-IDLE-STEP switch to IDLE. The Halt flip-flop is now reset.
2. To continue with the displayed instruction, set the RUN-IDLE-STEP switch to RUN (for automatic operation) or to STEP (for single-stepping).

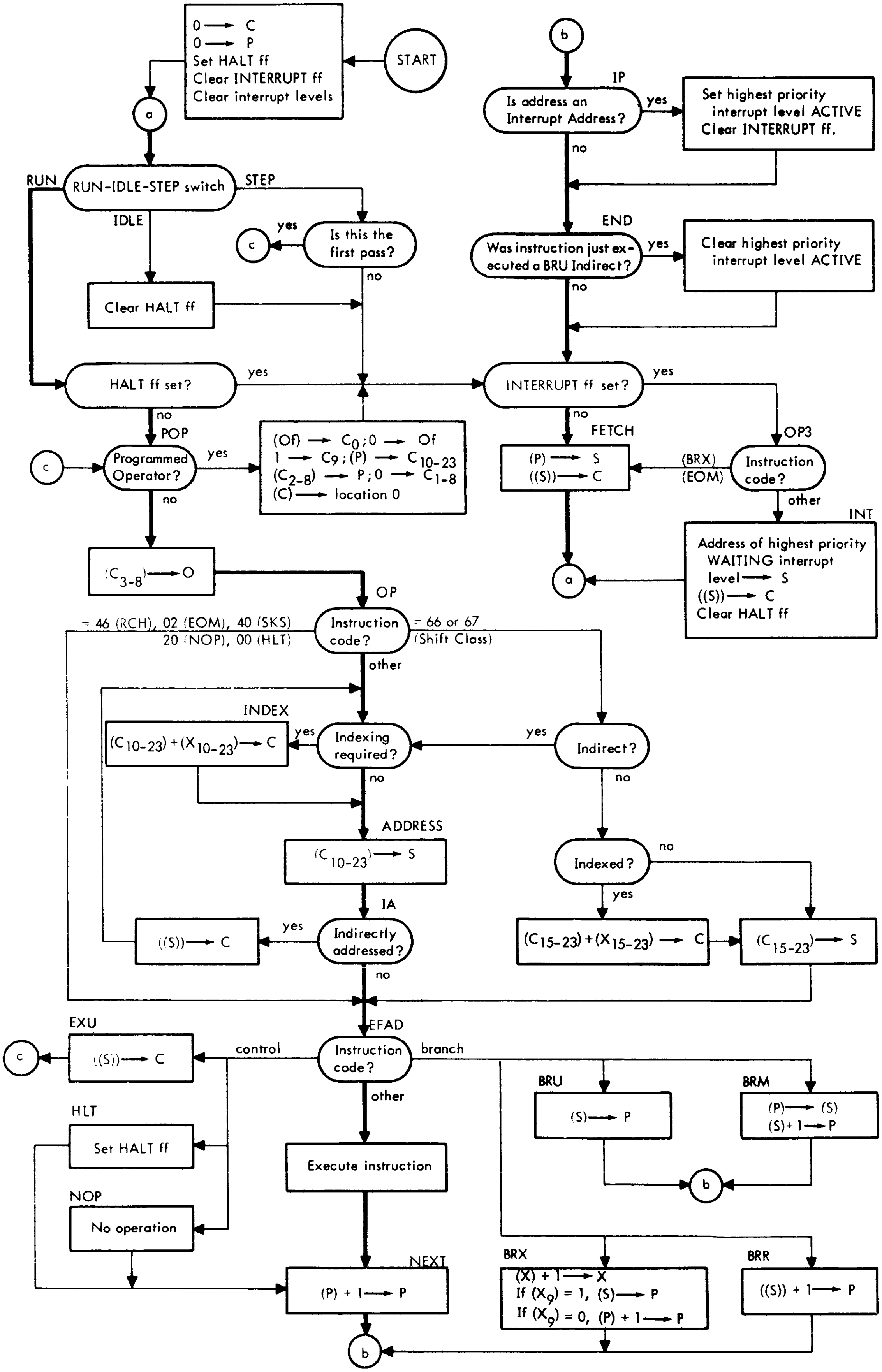


Figure 9. Instruction Execution Diagram

APPENDIX D DETAILED MACHINE FUNCTIONS

INSTRUCTION EXECUTION

Figure 9 is intended to show the major relationships between certain operating and program conditions during instruction execution, but does not necessarily correspond to actual computer operations. The following are considered:

START switch
RUN-IDLE-STEP switch
HALT ff (flip-flop)
Programmed Operators
Indexing
Indirect addressing
Control and branch instructions
Subroutine interrupts

Figure 9 assumes that the START switch has been pressed, a program is being executed, and an instruction is in the C register. The following paragraphs provide additional explanations of the functions performed at various steps in the instruction execution cycle. The labels below correspond to the labels that appear in Figure 9.

POP If bit 2 of the instruction is a 1, the instruction is a Programmed Operator. See Appendix E for a detailed discussion of this feature.

OP If bit 2 of the instruction is a zero, the O register contains the 6-bit code for the operation to be performed. Shift and cycle instructions require special address modification, and some other instructions (REGISTER CHANGE, ENERGIZE OUTPUT M, TEST/SKIP, and HALT) do not allow address modification.

INDEX If the instruction is indexed (a 1 in bit 1 of the instruction word), add the address field of the X register to the address field of the C register.

ADDRESS Copy the address field of the C register into the S register.

IA If the instruction operand is indirectly addressed (a 1 in bit 9 of the instruction word), load the C register with the contents of the S register, and go back to check for further address modification.

EFAD After all address modification, the S register contains the address of the effective operand of the operation to be performed.

NEXT For most instructions, the P register is incremented near the end of the execution cycle, in preparation for accessing the next instruction. However, since branch instructions operate directly on the P register, the effect of these instruction is shown. Also shown are control instructions. (Note that the EXU instruction loops back to OP after the effective operand is copied into the C register.)

IP If the instruction just executed was at an interrupt address (single instruction or subroutine), set the interrupt level ACTIVE and clear the INTERRUPT ff.

END If the instruction just executed was a BRU indirect (or was a single-instruction interrupt), clear the highest priority interrupt level in the active state.

FETCH If the INTERRUPT ff is not set, copy the next instruction into the C register and return to ⓐ.

OP3 If the instruction just executed was a BRX or EOM, wait until the next instruction is executed before going to INT.

INT Copy the address of the highest priority WAITING interrupt level into the S register, copy the contents of the memory location specified by the S register into the C register, clear the HALT ff, and return to ⓐ.

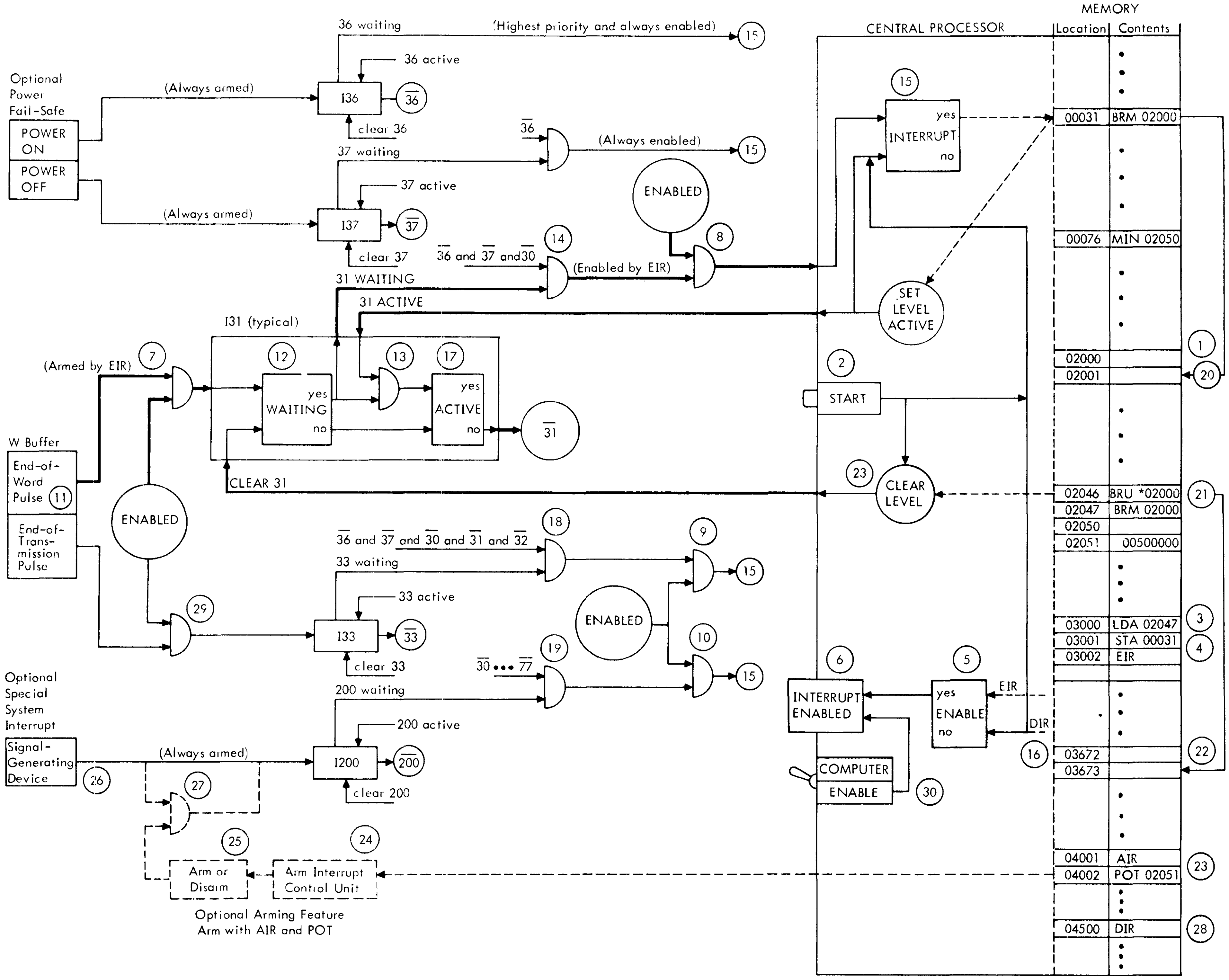


Figure 10. Priority Interrupt System Diagram

TYPICAL INTERRUPT CYCLE

Figure 10 is intended to show the progress of a typical interrupt cycle, and does not necessarily reflect actual circuitry. The circled numbers in the paragraphs below refer to specific portions of Figure 10.

① Program (02000-04500) is loaded into memory.

② START button is pressed, clearing all interrupts, Arm Interrupt Control Unit ㉔, Enable ff⑤, and the Interrupt ⑮. Program is entered by means of a BRU 03000 placed in C and RUN-IDLE-STEP switch placed in RUN.

③ Instructions 03000 and 03001 store the entrance to servicing subroutine in address of W buffer End-of-Word interrupt level (00031)

④ EIR in location 03002) sets Enable ff⑤, turns on INTERRUPT ENABLED indicator⑥, arms gate⑦, and enables gates⑧,⑨, ⑩, etc.

⑪ W buffer transmits end-or-word pulse through gate⑦to set the level Waiting ff ⑫. If the INTERRUPT ENABLED indicator had been off, then the pulse at gate⑦would have been lost.

⑫ The Waiting ff presents a steady signal at gates ⑬ and ⑭; the interrupt level is now in the waiting state, and remains in the waiting state until cleared by a BRU indirect or by the START switch. The waiting state is not affected by a DISABLE INTERRUPTS (DIR) instruction.

⑭ If no higher-priority interrupts (i.e., 36, 37 and 30) are in the active state, the signal passes through the priority gate ⑭ and through the ENABLE gate⑧to set the Interrupt ff ⑮. If the INTERRUPT ENABLED indicator had been off, the signal would not pass through gate⑧ and the interrupt level would remain in the waiting state.

⑯ Assuming that the Interrupt ff is set during the execution cycle of the instruction in 03672, and the instruction is not a BRX or EOM, the S register is set to 031, BRM 02000 in that location is brought out to the C register and executed, with the following results:

a. The contents of the Program Counter (03673) are placed in bits 10-23 of location 02000.

b. An Interrupt Active pulse is transmitted to gate ⑬ and the level Active ff ⑰ is set. Lower-priority interrupts are inhibited at priority gates ⑱, ⑲, etc.

c. The Interrupt ff is cleared, allowing interrupt levels 36, 37 and 30 to interrupt the servicing subroutine for level 31.

d. Program control is transferred to the second location within the servicing subroutine (location 02001).

㉑ At the end of the servicing subroutine (location 02046), execution of BRU *02000 causes the following:

a. The contents of location 02000 (the address at which the program should resume) are placed back into the Program Counter.

b. A clear-interrupt pulse is transmitted to the interrupt level to clear the Waiting ff and the Active ff. The steady signal (31) is now presented at all lower-priority interrupt levels and they may now interrupt the program.

c. Program control is transferred to the next instruction in sequence after the instruction at which the interrupt occurred.

㉒ The interrupted program continues at location 03673.

㉓ Instructions in locations 04001 and 04002 instruct the Arm Interrupt Control Unit㉔to set a group of arming flip-flops (52) to allow an interrupt pulse from a signal-generating device ㉖ to pass through an arming gate ㉗ to the interrupt level 0200.

㉘ DIR (instruction in location 04500) resets the Enable ff, disarms gates⑦, ㉙, etc., disables gates⑧,⑨, ⑩, etc., and turns off the INTERRUPT ENABLE INDICATOR.

㉚ The INTERRUPT ENABLE switch causes an interrupt enabled condition when it is manually held in the ENABLE position - regardless of the state of the Enable ff.

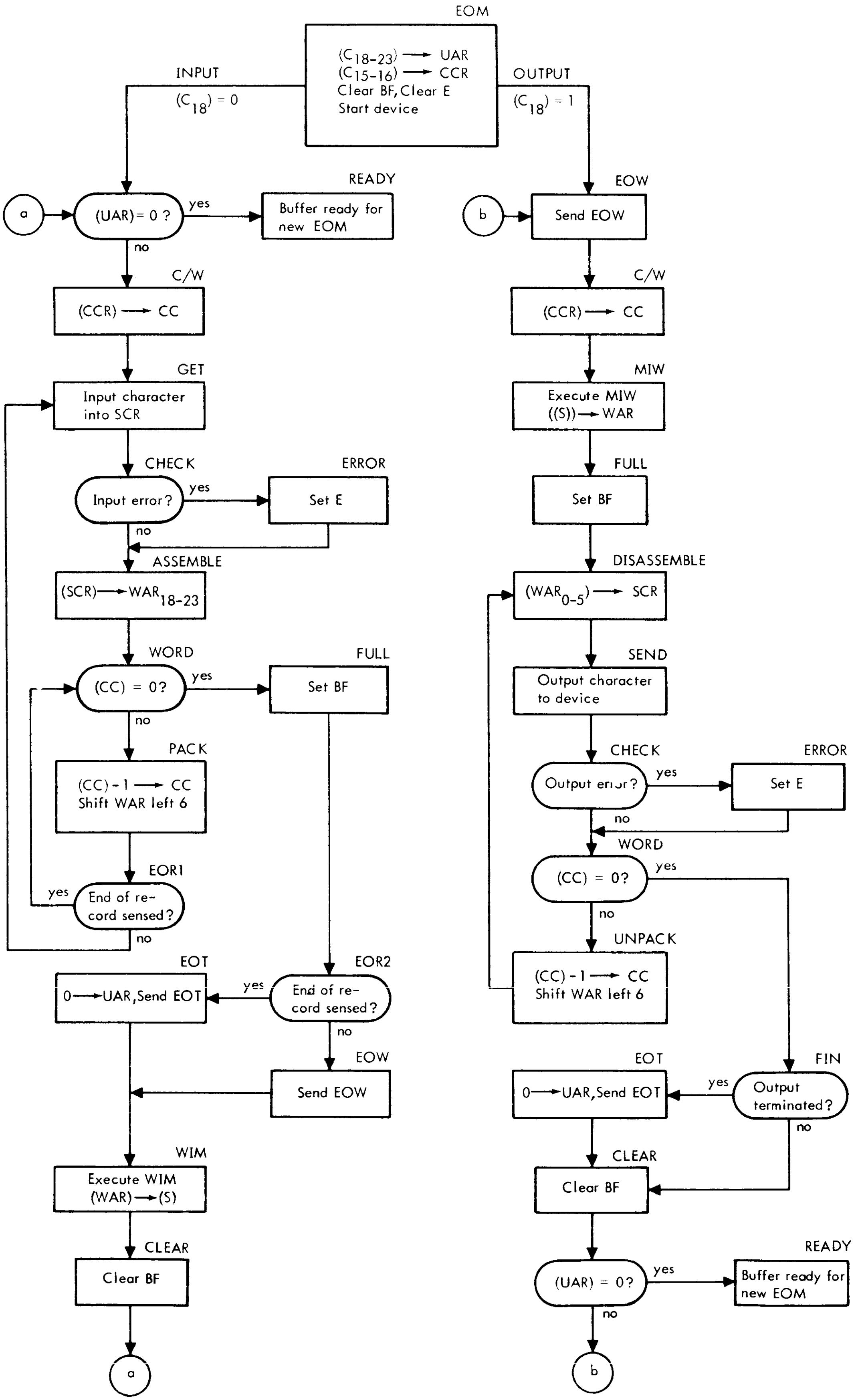


Figure 11. Buffer Operation, Single-Word Transmission

BUFFERED INPUT/OUTPUT

SINGLE WORD TRANSFER

Figure 11 shows the major relationships between certain buffer conditions in input/output operations. The following paragraphs refer to Figure 11 and assume the W buffer is being used in the single-word mode of operation, and that buffer interrupts are enabled. (Refer also to Figure 4, page 24.)

EOM Execution of a buffer control EOM:

1. Places bits 18-23 of the EOM into the buffer unit address register (UAR)
2. Places bits 15 and 16 of the EOM (characters per word) into a character count register (CCR)
3. Clears the buffer full (BF) indicator
4. Clears the buffer error (E) indicator
5. Starts the device specified by the unit address. The contents of bits 18-23 of the EOM are displayed on the control panel in UNIT ADDRESS. Bit 18 of the EOM (the first bit of the 6-bit unit address code) specifies input or output.

INPUT

READY If the UAR contains all zeros, the buffer is currently disconnected and is ready for a buffer control EOM; this will cause a skip if a W BUFFER READY TEST (BRTW) is executed.

C/W Accept a character from the peripheral device specified by the UAR into the single character register (SCR).

CHECK Was there a parity error?

ERROR Set the ERROR indicator on the control panel. This will cause a skip if W BUFFER ERROR TEST (BETW) is executed when the indicator is set.

ASSEMBLE The contents of the SCR are copied into bits 18-23 of a 24-bit word assembly register (WAR).

WORD If the number of characters/word specified by the EOM have been assembled, set BF and go to EOR2.

PACK Decrease CC by 1, and shift WAR 6 places left to make room for new input.

EOR1 If no end-of-record is sensed by the input device, go back to GET for the next character. If an end-of-record is sensed, go back to WORD until CC = zero. Thus, if the last word in an input record does not contain the specified count of characters/word, zeros fill the least-significant portion of that word.

FULL Set BF; word is ready to be stored in memory.

EOR2 If an end-of-record is sensed by the input device, go to EOT. If not, go to EOW (or COUNT).

EOT Clear UAR (disconnect buffer), and generate the End-of-Transmission (EOT) interrupt (I2W).

EOW Generate the End-of-Word (EOW) interrupt (I1W).

WIM Computer executes W BUFFER INTO MEMORY (WIM) instruction.

CLEAR Clear the BF indicator in preparation for the next input word.

OUTPUT

EOW Same as for input.

C/W Same as for input.

MIW Computer executes MEMORY INTO W BUFFER (MIW) instruction.

FULL Same as for input.

DISASSEMBLE Copy the contents of bits 0-5 of the WAR into the SCR.

SEND Transmit the contents of the SCR to the device specified by the UAR.

CHECK Same as for input.

ERROR Same as for input.

WORD Same as for input.

UNPACK Decrease CC by 1, shift WAR left 6 places for new character output.

FIN Has output been terminated with a TERMINATE OUTPUT (TOP) instruction?

EOT Same as for input.

CLEAR Same as for input.

READY Same as for input.

INTERLACE CONTROL

Figure 12 shows the automatic operations of the buffer during interlaced transmission. The following paragraphs refer to Figure 12 and assume that an interlaced I/O operation has been initiated by a buffer control EOM.

INTERLACE

The word count is stored in the word count register (WCR), the starting address is stored in the memory address register (MAR), and the automatic interlace control begins.

EOR

If an end-of-record signal has been received by the buffer from the peripheral device, the I/O operation is terminated.

FIN

If the specified number of words have been processed, the I/O operation is terminated.

TRANSMIT

During input, the contents of the WAR are copied into the memory location specified by the contents of the MAR. During output, the contents of the memory location specified by the contents of the MAR are copied into the WAR. Words are assembled or disassembled as described in Single-Word Transmission.

NEXT

The contents of the MAR are incremented by 1 and the contents of the WCR are decremented by 1.

EOT

If an end-of-record signal has been received, the unit address register (UAR) is cleared and an End-of-Transmission (EOT) interrupt signal is transmitted to the EOT interrupt level. If the Interrupt System is enabled, a program interrupt occurs.

OUTPUT

When the specified number of words have been transmitted to the peripheral device, the buffer automatically terminates the output operation, clears the UAR, and transmits the EOT interrupt signal.

INPUT

When the specified number of words have been stored in memory, the interlace is disengaged and the buffer returns to the single-word mode of transmission, as depicted in Figure 11.

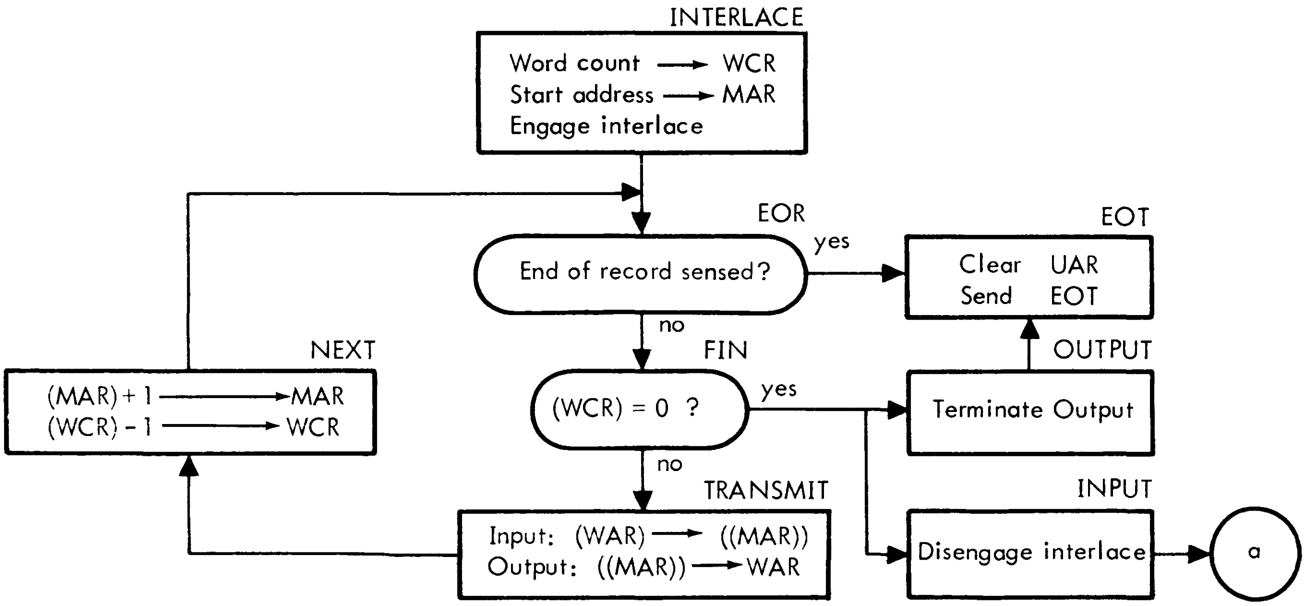


Figure 12. Buffer Operation, Interlaced Transmission

APPENDIX E PROGRAMMED OPERATORS

The XDS Programmed Operator (POP) feature enables a programmer to code a subroutine call with a single instruction, just as if the subroutine were a machine instruction. The XDS Programmed Operator feature uses the operation code to indicate the transfer address. When the computer detects a 1 in bit position 2 of an instruction, bit positions 3 through 8 are not interpreted as a normal instruction, but instead are treated as the 6 low-order bits of an address to which the computer transfers control. Thus, the operand address field is free to designate an address for use by the subroutine. There are 64 (decimal) locations (100_8 through 177_8) to which a transfer may occur. These locations constitute a linkage table; they normally contain appropriate unconditional branch (BRU) instructions to maintain the communication link between the POP code and the subroutine being called by it.

When the computer detects the POP code, the location of the POP code (that is, the contents of the P register) is preserved in location 0. Also, the state of the OVERFLOW indicator is preserved in bit position 0 of location 0 and the OVERFLOW indicator is reset. Thus, the normal BRR instruction may be used to leave the POP subroutine and return to the main program.

To allow access to an operand in the main program by the POP subroutine, bit position 9 (the indirect address bit) of location 0 is unconditionally set to 1. In this manner, when the subroutine refers indirectly to location 0, the indirect addressing is perpetuated one more level.

By judicious use of the programmed operator principle, a one-to-one program correspondence may be maintained between XDS 900 Series Computers. For example, XMA is a 930 machine instruction; its function may be simulated on the XDS 910 by a programmed operator. Thus, the main program requires the same number of instructions for either the XDS 910 or 930.

Another advantage of the Programmed Operator is the ability to change the arithmetic mode of a program without recoding the arithmetic portions of the program. For example, if the programmer codes all arithmetic instructions as programmed operators, he could simply change the arithmetic subroutine package and, hence, the arithmetic mode of the main program.

In summary, the following operations take place when the computer detects a Programmed Operator:

1. $(Of) \longrightarrow 0_0$ (preserve status of OVERFLOW indicator)
2. $0 \longrightarrow O_f$ (reset OVERFLOW indicator)
3. $0 \longrightarrow 0_{1-8}$ (clear bits 1-8 of location 0)
4. $1 \longrightarrow 0_9$ (insert indirect address bit)
5. $(P) \longrightarrow 0_{10-23}$ (save P register for return address)
6. $(C)_{2-8} \longrightarrow P$ (branch to location indicated in POP code)

A library of Programmed Operator subroutines is available which greatly extends the XDS 910 instruction repertiore (see page A-20). Each subroutine is identified by a unique mnemonic and represents an available instruction that may be used in preparing 910 programs.

Up to 64 Programmed Operator instructions may be used in any one program. The program loading system automatically organizes the interconnection between POP instructions and their corresponding subroutines. Each POP mnemonic is converted to an octal code of 100 to 177. A memory location from 0100 through 0177, corresponding to each POP code, is then loaded with an unconditional branch to the corresponding subroutine.

Example: XMA is a Programmed Operator (POP code 162) that exchanges the contents of the A register with the contents of the effective address of XMA. The contents of the B and X registers are not permanently affected by this subroutine.

Location	Instruction	Effective Address	(Effective Address)[†]	(Location 0)[†]	(Of)[†]	(A Register)[†]	(B Register)[†]
01341	LSH 6			00053601	Set	25252525	01234500
01342	XMA 02163	02163	77777777	40041342	Reset	25252525	01234500
00162	BRU 0300	00300	07500306			25252525	01234500
00300	STB 0306	00306	01234500			25252525	01234500
00301	LDB *0	02163	77777777			25252525	77777777
00302	STA *0	02163	25252525			25252525	77777777
00303	XAB					77777777	25252525
00304	LDB 0306	00306	01234500		Reset	77777777	01234500
00305	BRR 0	00000	40041342	40041342	Set	77777777	01234500
01343	Continue in main program						

[†] Contents after execution of the instructions.

PROGRAMMED OPERATOR INSTRUCTIONS

Mnemonic	Name
ADM	Add A to M
ATD	Arctangent of A – Double-Precision, Fixed-Point
ATF	Arctangent of A – Floating-Point
ATN	Arctangent of A – Single-Precision, Fixed-Point
BDD	Binary to Decimal Conversion – Double-Precision, Fixed-Point
BDF	Binary to Decimal Conversion – Double-Precision, Floating-Point
BFS	Binary to Decimal Conversion – Single-Precision, Floating-Point
BID	Binary to Decimal Conversion – Single-Precision, Fixed-Point
CAB	Copy A into B
CAX	Copy A into Index
CBA	Copy B into A
CBX	Copy B into Index
CLA	Clear A
CLB	Clear B
CNA	Copy Negative into A
COS	Cosine of A – Single-Precision, Fixed-Point
CSD	Cosine of A – Double-Precision, Fixed-Point
CSF	Cosine of A – Floating-Point
CXA	Copy Index into A
CXB	Copy Index into B
DBD	Decimal to Binary Conversion – Double-Precision, Fixed-Point
DBF	Decimal to Binary Conversion – Double-Precision, Floating-Point
DFS	Decimal to Binary Conversion – Single-Precision, Floating-Point
DIB	Decimal to Binary Conversion – Single-Precision, Fixed-Point
DIV	Divide
DPA	Double-Precision Add – Fixed-Point
DPD	Double-Precision Divide – Fixed-Point
DPM	Double-Precision Multiply – Fixed-Point
DPN	Double-Precision Negate – Fixed-Point
DPS	Double-Precision Subtract – Fixed-Point
DSQ	Double-Precision Square Root – Fixed-Point

Mnemonic	Name
EXF	Exponential of A – Single-Precision Floating-Point
EXP	Exponential of A – Single-Precision, Fixed-Point
FFF	Fixed-Floating Format Conversion
FLA	Floating Add – Double-Precision
FLD	Floating Divide – Double-Precision
FLM	Floating Multiply – Double-Precision
FLN	Floating Negate – Double-Precision
FLS	Floating Subtract – Double-Precision
FSA	Floating Add – Single-Precision
FSD	Floating Divide – Single-Precision
FSM	Floating Multiply – Single-Precision
FSN	Floating Negate – Single-Precision
FSQ	Floating Square Root – Double-Precision
FSS	Floating Subtract – Single-Precision
LDE	Load Exponent
LDP	Load – Double-Precision
LGF	Logarithm of A – Floating-Point
LOG	Logarithm of A – Fixed-Point
LQP	Load – Quadruple-Precision
LRSH	Logical Right Shift A and B
LTP	Load – Triple-Precision
MUL	Multiply
SIN	Sine of A – Single-Precision, Fixed-Point
SKB	Skip if M and B do not Compare Ones
SKD	Difference Exponents and Skip
SKE	Skip if A Equals M
SKR	Reduce M, Skip if Negative
SND	Sine of A – Double-Precision, Fixed-Point
SNF	Sine of A – Floating-Point
SQR	Square Root of A –Single-Precision, Fixed-Point
STD	Store – Double-Precision
STE	Store Exponent
STP	Store – Triple-Precision
STQ	Store – Quadruple-Precision
XEE	Exchange Exponents
XMA	Exchange M and A
XXA	Exchange Index and A
XXB	Exchange Index and B

APPENDIX F INSTRUCTION LISTS

XDS 910 INSTRUCTION LIST — FUNCTIONAL CATEGORIES

Designation	Instruction Code	Name	Function	Timing	Page
LOAD/STORE					
LDA	76	LOAD A	(M) → A	2	8
STA	35	STORE A	(A) → M	3	8
LDB	75	LOAD B	(M) → B	2	8
STB	36	STORE B	(B) → M	3	9
LDX	71	LOAD INDEX	(M) → X	2	9
STX	37	STORE INDEX	(X) → M	3	9
EAX	77	COPY EFFECTIVE ADDRESS INTO INDEX REGISTER	Effective address → X	2	9
ARITHMETIC					
ADD	55	ADD M TO A	(A) + (M) A	2	9
MIN	61	MEMORY INCREMENT	(M) + 1 → M	3	9
MDE	60	MEMORY DECREMENT	(M) - 1 → M	3	9
SUB	54	SUBTRACT M FROM A	(A) - (M) → A	2	9
MUS	64	MULTIPLY STEP	Add or subtract (M) from (A) based on B_{21-23}, 1/4 (AB) → AB		
DIS	65	DIVIDE STEP	2 (AB) → AB $\overline{A_0} \rightarrow B_{23}$ If $(A_0) = (M_0)$, (A)-(M) → A; If $(A_0) \neq (M_0)$, (A)+(M) → A	2	10
LOGICAL					
ETR	14	EXTRACT	(A) AND (M) → A	2	10
MRG	16	MERGE	(A) OR (M) → A	2	10
EOR	17	EXCLUSIVE OR	$(M)\overline{(A)}$ OR $\overline{(M)}(A)$ → A	2	11
REGISTER CHANGE					
RCH	46	REGISTER CHANGE			
CLR	0 46 30000	CLEAR AB	0 → AB	1	11
XAB	0 46 00000	EXCHANGE A AND B	(A) ↔ (B)	1	11
BAC	0 46 10000	COPY B INTO A, CLEAR B	(B) → A, 0 → B	1	11
ABC	0 46 20000	COPY A INTO B, CLEAR A	(A) → B, 0 → A	1	11
BRANCH					
BRU	01	BRANCH UNCONDITIONALLY	Effective Address → P	1	12

Designation	Instruction Code	Name	Function	Timing	Page
BRX	41	INCREMENT INDEX AND BRANCH	$(X) + 1 \longrightarrow X$ If (X) negative, Effective Address $\longrightarrow P$ If (X) positive, $(P)+1 \longrightarrow P$	 1 2	 12
BRM	43	MARK PLACE AND BRANCH	$(P) \longrightarrow M;\ M + 1 \longrightarrow P$	2	12
BRR	51	RETURN BRANCH	$(M) + 1 \longrightarrow P$	2	12
TEST/SKIP					
SKG	73	SKIP IF A GREATER THAN M	If $(A) \leq (M),\ (P) + 1 \longrightarrow P$ If $(A) > (M),\ (P) + 2 \longrightarrow P$	2 3	 13
SKM	70	SKIP IF A = M ON B MASK	If $(B)(A) \neq (B)(M),\ (P) + 1 \longrightarrow P$ If $(B)(A) = (B)(M),\ (P) + 2 \longrightarrow P$	2 3	 13
SKN	53	SKIP IF M NEGATIVE	If $(M) \geq 0,\ (P) + 1 \longrightarrow P$ If $(M) < 0,\ (P) + 2 \longrightarrow P$	2 3	 14
SKA	72	SKIP IF M AND A DO NOT COMPARE ONES ANY-PLACE	If $(A)(M) \neq 0,\ (P) + 1 \longrightarrow P$ If $(A)(M) = 0,\ (P) + 2 \longrightarrow P$	2 3	 13
SKS	40	SKIP IF SIGNAL NOT SET	If signal = 1, $(P) + 1 \longrightarrow P$ If signal = 0, $(P) + 2 \longrightarrow P$	1 2	 26
SHIFT					
RSH	0 66 000XX	RIGHT SHIFT AB	(AB) shift right N places	2 + N	14
RCY	0 66 200XX	RIGHT CYCLE AB	(AB) cycled right N places	2 + N	14
LSH	0 67 000XX	LEFT SHIFT AB	(AB) shift left N places	2 + N	15
LCY	0 67 200XX	LEFT CYCLE AB	(AB) cycled left N places	2 + N	15
NOD	0 67 100XX	NORMALIZE AND DECREMENT X	(AB) left and $(X) - 1 \longrightarrow X$ until $A_0 \neq A_1$, or N shifts	2 + N	15
CONTROL					
HLT	00	HALT	Halts computation	1	15
NOP	20	NO OPERATION	--------	1	16
EXU	23	EXECUTE	Instruction in M is performed, (P) is unchanged	1	16
BREAKPOINT TESTS					
BPT 1	0 40 20400	BREAKPOINT 1 TEST	Test BREAKPOINT switch 1	1, 2	16
BPT 2	0 40 20200	BREAKPOINT 2 TEST	Test BREAKPOINT switch 2	1, 2	16
BPT 3	0 40 20100	BREAKPOINT 3 TEST	Test BREAKPOINT switch 2	1, 2	16
BPT 4	0 40 20040	BREAKPOINT 4 TEST	Test BREAKPOINT switch 4	1, 2	16
OVERFLOW					
OVT	0 40 20001	OVERFLOW INDICATOR TEST AND RESET	Test OVERFLOW indicator	1, 2	16
ROV	0 02 20001	RESET OVERFLOW	Turn off OVERFLOW indicator	1	16

Designation	Instruction Code	Name	Function	Timing	Page
INTERRUPT					
EIR	0 02 20002	ENABLE INTERRUPT SYSTEM		1	23
DIR	0 02 20004	DISABLE INTERRUPT SYSTEM		1	23
IET	0 40 20004	INTERRUPT ENABLED TEST	Skip if Interrupt System enabled	1, 2	23
IDT	0 40 20002	INTERRUPT DISABLED TEST	Skip if Interrupt System disabled	1, 2	23
AIR	0 02 20020	ARM INTERRUPTS		1	21
BUFFER					
ALC 0	0 02 50000	ALERT W BUFFER		1	28
ALC 1	0 02 50100	ALERT Y BUFFER		1	28
DSC 0	0 02 00000	DISCONNECT W BUFFER		1	28
DSC 1	0 02 00100	DISCONNECT Y BUFFER		1	28
TOP 0	0 02 14000	TERMINATE OUTPUT ON W BUFFER		1	28
TOP 1	0 02 14100	TERMINATE OUTPUT ON Y BUFFER		1	28
INTERLACE STORING CONTROL					
ASC 0	0 02 12000	ALERT TO STORE ADDRESS IN W BUFFER		1	29
ASC 1	0 02 12100	ALERT TO STORE ADDRESS IN Y BUFFER		1	29
INPUT/OUTPUT					
MIW	12	M INTO W BUFFER WHEN EMPTY	(M) → W	2 + wait	30
MIY	10	M INTO Y BUFFER WHEN EMPTY	(M) → Y	2 + wait	30
WIM	32	W BUFFER INTO M WHEN FULL	(W) → M	3 + wait	31
YIM	30	Y BUFFER INTO M WHEN FULL	(Y) → M	3 + wait	31
POT	13	PARALLEL OUTPUT	(M) → Unit M in Parallel	3 + wait	35
PIN	33	PARALLEL INPUT	(Unit M) → M in Parallel	4 + wait	35
EOM	02	ENERGIZE OUTPUT M	8 µsec Pulse to Point(s) Addressed	1	25
SKS	40	SKIP IF SIGNAL NOT SET	If signal = 1, (P) + 1 → P If signal = 0, (P) + 2 → P	1 2	26
BETW	0 40 20010	W BUFFER ERROR TEST	Skip if No W Buffer Error	1, 2	29
BETY	0 40 20020	Y BUFFER ERROR TEST	Skip if No Y Buffer Error	1, 2	29
BRTW	0 40 21000	W BUFFER READY TEST	Skip if W Buffer Ready	1, 2	29
BRTY	0 40 22000	Y BUFFER READY TEST	Skip if Y Buffer Ready	1, 2	29

Designation	Instruction Code	Name	Function	Timing	Page
TYPEWRITER					
RKB 0, 1, 4	0 02 02601	READ KEYBOARD		1	38
TYP 0, 1, 4	0 02 02641	WRITE TYPEWRITER		1	38
PAPER TAPE					
RPT 0, 1, 4	0 02 02604	READ PAPER TAPE		1	40
PTL 0, 1, 4	0 02 00644	PUNCH PAPER TAPE WITH LEADER		1	41
PPT 0, 1, 4	0 02 02644	PUNCH PAPER TAPE WITH NO LEADER		1	41
PUNCHED CARD					
CRT 0, 1	0 40 12006	CARD READER READY TEST	Skip if Card Reader Ready	1, 2	45
FCT 0, 1	0 40 14006	FIRST COLUMN TEST	Skip if Column about to be Read	1, 2	45
CFT 0, 1	0 40 11006	CARD READER END-OF-FILE TEST	Skip if Card Reader Not at End of File	1, 2	45
RCD 0, 1, 4	0 02 02606	READ CARD DECIMAL (Hollerith)		1	45
RCB 0, 1, 4	0 02 03606	READ CARD BINARY			45
SRC 0, 1	0 02 12006	SKIP REMAINDER OF CARD		1	45
CPT 0, 1	0 40 14046	CARD PUNCH READY TEST	Skip if Card Punch Ready	1, 2	48
PBT 0, 1	0 40 12046	PUNCH BUFFER TEST	Skip if Punch Buffer Ready	1, 2	48
PCD 0, 1, 4	0 02 02646	PUNCH CARD DECIMAL (Hollerith)		1	46
PCB 0, 1, 4	0 02 03646	PUNCH CARD BINARY		1	46
MAGNETIC TAPE					
TRT 0, n	0 40 1041n	TAPE READY TEST	Skip if Tape Unit Not Ready	1, 2	53
FPT 0, n	0 40 1401n	FILE PROTECT TEST	Skip if Tape Unit Not File Protected	1, 2	53
BTT 0, n	0 40 1201n	BEGINNING OF TAPE TEST	Skip if Tape Unit Not at Beginning of Tape	1, 2	53
ETT 0, n	0 40 1101n	END OF TAPE TEST	Skip if Tape Unit Not at End of Tape	1, 2	53
	0 40 1021n	MAGPAK TEST	Skip if Tape Unit Not MAGPAK	1, 2	53
DT2 0, n	0 40 1621n	DENSITY TEST, 200 BPI	Skip if Tape Unit Not at 200 BPI Density	1, 2	53
DT5 0, n	0 40 1661n	DENSITY TEST, 556 BPI	Skip if Tape Unit Not at 556 BPI Density	1, 2	53
DT8 0, n	0 40 1721n	DENSITY TEST, 800 BPI	Skip if Tape Unit Not at 800 BPI Density	1, 2	53

Designation	Instruction Code	Name	Function	Timing	Page
TFT 0	0 40 13610	TAPE END-OF-FILE TEST	Skip if Tape Unit Not at End of File	1, 2	53
TGT 0	0 40 12610	TAPE GAP TEST	Skip if Tape Unit Not in Gap	1, 2	54
WTB 0, n, 4	0 02 0365n	WRITE TAPE IN BINARY		1	59
WTD 0, n, 4	0 02 0265n	WRITE TAPE IN DECIMAL (BCD)		1	59
EFT 0, n, 4	0 02 0367n	ERASE TAPE FORWARD		1	59
ERT 0, n, 4	0 02 0767n	ERASE TAPE IN REVERSE		1	59
RTB 0, n, 4	0 02 0361n	READ TAPE IN BINARY		1	54
RTD 0, n, 4	0 02 0261n	READ TAPE IN DECIMAL (BCD)		1	54
RTS 0	0 02 14000	CONVERT READ TO SCAN		1	57
SFB 0, n, 4	0 02 0363n	SCAN FORWARD IN BINARY		1	55
SFD 0, n, 4	0 02 0263n	SCAN FORWARD IN DECIMAL (BCD)		1	55
SRB 0, n, 4	0 02 0763n	SCAN REVERSE IN BINARY		1	55
SRD 0, n, 4	0 02 0663n	SCAN REVERSE IN DECIMAL (BCD)		1	55
SRR 0	0 02 13610	SKIP REMAINDER OF RECORD ON TAPE IN READ OPERATION		1	57
REW 0, n	0 02 1401n	REWIND		1	57
PRINTER					
PRT 0, 1	0 40 12060	PRINTER READY TEST	Skip if Printer Ready	1, 2	50
EPT 0, 1	0 40 14060	END OF PAGE TEST	Skip if Printer Not at End of Page	1, 2	50
PFT 0, 1	0 40 11060	PRINTER FAULT TEST	Skip if No Print Fault	1, 2	50
POL 0, 1	0 02 10260	PRINTER OFF-LINE		1	49
PSC 0, 1, n	0 02 1n460	PRINTER SKIP TO FORMAT CHANNEL n		1	49
PSP 0, 1, n	0 02 1n660	PRINTER SPACE n LINES		1	49
PLP 0, 1, 4	0 02 02660	PRINT LINE PRINTER		1	49

XDS 910 INSTRUCTION LIST — NUMERICAL ORDER

Instruction Code	Designation	Name	Page
00	HLT	HALT	15
01	BRU	BRANCH UNCONDITIONALLY	12
02	EOM	ENERGIZE OUTPUT M	26
0 02 00000	DSC 0	DISCONNECT W BUFFER	28
0 02 00100	DSC 1	DISCONNECT Y BUFFER	28
0 02 00644	PTL 0,1,4	PUNCH PAPER TAPE WITH LEADER	41
0 02 02601	RKB 0,1,4	READ KEYBOARD	38
0 02 02604	RPT 0,1,4	READ PAPER TAPE	38
0 02 02606	RCD 0,1,4	READ CARD DECIMAL (HOLLERITH)	45
0 02 0261n	RTD 0,n,4	READ TAPE IN DECIMAL (BCD)	45
0 02 0263n	SFD 0,n,4	SCAN FORWARD IN DECIMAL (BCD)	55
0 02 02641	TYP 0,1,4	WRITE TYPEWRITER	38
0 02 02644	PPT 0,1,4	PUNCH PAPER TAPE WITH NO LEADER	41
0 02 02646	PCD 0,1,4	PUNCH CARD DECIMAL (HOLLERITH)	46
0 02 0265n	WTD 0,n,4	WRITE TAPE IN DECIMAL (BCD)	59
0 02 02660	PLP 0,1,4	PRINT LINE PRINTER	49
0 02 03606	RCB 0,1,4	READ CARD BINARY	45
0 02 0361n	RTB 0,n,4	READ TAPE IN BINARY	54
0 02 0363n	SFB 0,n,4	SCAN FORWARD IN BINARY	55
0 02 03646	PCB 0,1,4	PUNCH CARD BINARY	45
0 02 0365n	WTB 0,n,4	WRITE TAPE IN BINARY	55
0 02 0367n	EFT 0,n,4	ERASE TAPE FORWARD	59
0 02 0663n	SRD 0,n,4	SCAN REVERSE IN DECIMAL (BCD)	55
0 02 0763n	SRB 0,n,4	SCAN REVERSE IN BINARY	55
0 02 0767n	ETT 0,n,4	ERASE TAPE IN REVERSE	59
0 02 10260	POL 0,1	PRINTER OFF-LINE	49
0 02 12000	ASC 0	ALERT TO STORE ADDRESS IN W BUFFER	29
0 02 12006	SRC 0,1	SKIP REMAINDER OF CARD	45
0 02 12100	ASC 1	ALERT TO STORE ADDRESS IN Y BUFFER	29
0 02 13610	SRR 0	SKIP REMAINDER OF RECORD	57
0 02 14000	RTS 0	CONVERT READ TO SCAN	57
0 02 14000	TOP 0	TERMINATE OUTPUT ON W BUFFER	28
0 02 1401n	REW 0,n	REWIND	57
0 02 14100	TOP 1	TERMINATE OUTPUT ON Y BUFFER	28
0 02 1n460	PSC 0,1,n	PRINTER SKIP TO FORMAT CHANNEL n	49
0 02 1n660	PSP 0,1,n	PRINTER SPACE n LINES	49

Instruction Code	Designation	Name	Page
0 02 20001	ROV	RESET OVERFLOW	16
0 02 20002	EIR	ENABLE INTERRUPT SYSTEM	20
0 02 20004	DIR	DISABLE INTERRUPT SYSTEM	20
0 02 20020	AIR	ARM INTERRUPTS	21
0 02 50000	ALC 0	ALERT W BUFFER	28
0 02 50100	ALC 1	ALERT Y BUFFER	28
10	MIY	M INTO Y BUFFER WHEN EMPTY	30
12	MIW	M INTO W BUFFER WHEN EMPTY	30
13	POT	PARALLEL OUTPUT	35
14	ETR	EXTRACT	10
16	MRG	MERGE	10
17	EOR	EXCLUSIVE OR	11
20	NOP	NO OPERATION	16
23	EXU	EXECUTE	16
30	YIM	Y BUFFER INTO M WHEN FULL	31
32	WIM	W BUFFER INTO M WHEN FULL	31
33	PIN	PARALLEL INPUT	35
35	STA	STORE A	8
36	STB	STORE B	8
37	STX	STORE INDEX	9
40	SKS	SKIP IF SIGNAL NOT SET	9
0 40 1021n		MAGPAK TEST	54
0 40 1041n	TRT 0, n	TAPE READY TEST	53
0 40 11006	CFT 0, 1	CARD READER END-OF-FILE TEST	45
0 40 1101n	ETT 0, n	END OF TAPE TEST	53
0 40 11060	PFT 0, 1	PRINTER FAULT TEST	50
0 40 12006	CRT 0, 1	CARD READER READY TEST	45
0 40 1201n	BTT 0, n	BEGINNING OF TAPE TEST	53
0 40 12046	PBT 0, 1	PUNCH BUFFER TEST	48
0 40 12060	PRT 0, 1	PRINTER READY TEST	50
0 40 12610	TGT 0	TAPE GAP TEST	54
0 40 13610	TFT 0	TAPE END-OF-FILE TEST	53
0 40 14006	FCT 0, 1	FIRST COLUMN TEST	45
0 40 1401n	FPT 0, n	FILE PROTECT TEST	53
0 40 14046	CPT 0, 1	CARD PUNCH READY TEST	48
0 40 14060	EPT 0, 1	END OF PAGE TEST	50
0 40 1621n	DT2 0, n	DENSITY TEST, 200 BPI	53
0 40 1661n	DT5 0, n	DENSITY TEST, 556 BPI	53

Instruction Code	Designation	Name	Page
0 40 1721n	DT8 0,n	DENSITY TEST, 800 BPI	53
0 40 20001	OVT	OVERFLOW INDICATOR TEST AND RESET	16
0 40 20002	IDT	INTERRUPT DISABLED TEST	20
0 40 20004	IET	INTERRUPT ENABLED TEST	20
0 40 20010	BETW	W BUFFER ERROR TEST	29
0 40 20020	BETY	Y BUFFER ERROR TEST	29
0 40 20040	BPT 4	BREAKPOINT NO. 4 TEST	16
0 40 20100	BPT 3	BREAKPOINT NO. 3 TEST	16
0 40 20200	BPT 2	BREAKPOINT NO. 2 TEST	16
0 40 20400	BPT 1	BREAKPOINT NO. 1 TEST	16
0 40 21000	BRTW	W BUFFER READY TEST	29
0 40 22000	BRTY	Y BUFFER READY TEST	29
41	BRX	INCREMENT INDEX AND BRANCH	12
43	BRM	MARK PLACE AND BRANCH	12
0 46 00000	XAB	EXCHANGE A AND B	11
0 46 10000	BAC	COPY B INTO A, CLEAR B	11
0 46 20000	ABC	COPY A INTO B, CLEAR A	11
0 46 30000	CLR	CLEAR AB	11
51	BRR	RETURN BRANCH	12
53	SKN	SKIP IF M NEGATIVE	14
54	SUB	SUBTRACT M FROM A	9
55	ADD	ADD M TO A	9
60	MDE	MEMORY DECREMENT	9
61	MIN	MEMORY INCREMENT	9
64	MUS	MULTIPLY STEP	9
65	DIS	DIVIDE STEP	10
0 66 000XX	RSH	RIGHT SHIFT AB	14
0 66 200XX	RCY	RIGHT CYCLE AB	14
0 67 000XX	LSH	LEFT SHIFT AB	15
0 67 100XX	NOD	NORMALIZE AND DECREMENT X	15
0 67 200XX	LCY	LEFT CYCLE AB	15
70	SKM	SKIP IF A = M ON B MASK	13
71	LDX	LOAD INDEX	9
72	SKA	SKIP IF MA AND A DO NOT COMPARE ONES	13
73	SKG	SKIP IF A GREATER THAN M	13
75	LDB	LOAD B	8
76	LDA	LOAD A	8
77	EAX	COPY EFFECTIVE ADDRESS INTO INDEX	9

XDS 910 INSTRUCTION LIST — ALPHABETICAL ORDER

Designation	Instruction Code	Name	Page
ABC	0 46 20000	COPY A INTO B, CLEAR A	11
ADD	55	ADD M TO A	9
AIR	0 02 20020	ARM INTERRUPTS	21
ALC 0	0 02 50000	ALERT W BUFFER	28
ALC 1	0 02 50100	ALERT Y BUFFER	28
ASC 0	0 02 12000	ALERT TO STORE ADDRESS IN W BUFFER	29
ASC 1	0 02 12100	ALERT TO STORE ADDRESS IN Y BUFFER	29
BAC	0 46 10000	COPY B INTO A, CLEAR B	11
BETW	0 40 20010	W BUFFER ERROR TEST	29
BETY	0 40 20020	Y BUFFER ERROR TEST	29
BPT 1	0 40 20400	BREAKPOINT NO. 1 TEST	16
BPT 2	0 40 20200	BREAKPOINT NO. 2 TEST	16
BPT 3	0 40 20100	BREAKPOINT NO. 3 TEST	16
BPT 4	0 40 20040	BREAKPOINT NO. 4 TEST	16
BRM	43	MARK PLACE AND BRANCH	12
BRR	51	RETURN BRANCH	12
BRTW	0 40 21000	W BUFFER READY TEST	29
BRTY	0 40 22000	Y BUFFER READY TEST	29
BRU	01	BRANCH UNCONDITIONALLY	12
BRX	41	INCREMENT INDEX AND BRANCH	12
BTT 0, n	0 40 1201n	BEGINNING OF TAPE TEST	53
CFT 0, 1	0 40 11006	CARD READER END-OF-FILE TEST	45
CLR	0 46 30000	CLEAR AB	11
CPT 0, 1	0 40 14046	CARD PUNCH READY TEST	48
CRT 0, 1	0 40 12006	CARD READER READY TEST	45
DIR	0 02 20004	DISABLE INTERRUPT SYSTEM	20
DIS	65	DIVIDE STEP	10
DSC 0	0 02 00000	DISCONNECT W BUFFER	28
DSC 1	0 02 00100	DISCONNECT Y BUFFER	28
DT2 0, n	0 40 1621n	DENSITY TEST, 200 BPI	53
DT5 0, n	0 40 1661n	DENSITY TEST, 556 BPI	53
DT8 0, n	0 40 1721n	DENSITY TEST, 800 BPI	53
EAX	77	COPY EFFECTIVE ADDRESS INTO INDEX	9
EFT 0, n, 4	0 02 0367n	ERASE TAPE FORWARD	59
EIR	0 02 20002	ENABLE INTERRUPT SYSTEM	20

Designation	Instruction Code	Name	Page
EOM	02	ENERGIZE OUTPUT M	26
EOR	17	EXCLUSIVE OR	11
EPT 0, 1	0 40 14060	END OF PAGE TEST	50
ERT 0, n, 4	0 02 0767n	ERASE TAPE IN REVERSE	59
ETR	14	EXTRACT	10
ETT 0, n	0 40 1101n	END OF TAPE TEST	53
EXU	23	EXECUTE	16
FCT 0, 1	0 40 14006	FIRST COLUMN TEST	45
FPT 0, n	0 40 1401n	FILE PROTECT TEST	53
HLT	00	HALT	15
IDT	0 40 20002	INTERRUPT DISABLED TEST	20
IET	0 40 20004	INTERRUPT ENABLED TEST	20
LCY	0 67 200XX	LEFT CYCLE AB	15
LDA	76	LOAD A	8
LDB	75	LOAD B	8
LDX	71	LOAD INDEX	9
LSH	0 67 000XX	LEFT SHIFT AB	15
MDE	60	MEMORY DECREMENT	9
MIN	61	MEMORY INCREMENT	9
MIW	12	M INTO W BUFFER WHEN EMPTY	30
MIY	10	M INTO Y BUFFER WHEN EMPTY	30
MRG	16	MERGE	10
MUS	64	MULTIPLY STEP	9
NOD	0 67 100XX	NORMALIZE AND DECREMENT X	15
NOP	20	NO OPERATION	16
OVT	0 40 20001	OVERFLOW INDICATOR TEST AND RESET	16
PBT 0, 1	0 40 12046	PUNCH BUFFER TEST	48
PCB 0, 1, 4	0 02 03646	PUNCH CARD BINARY	46
PCD 0, 1, 4	0 02 02646	PUNCH CARD DECIMAL (HOLLERITH)	46
PFT 0, 1	0 40 11060	PRINTER FAULT TEST	50
PIN	33	PARALLEL INPUT	35
PLP 0, 1, 4	0 02 02660	PRINT LINE PRINTER	49
PTL 0, 1, 4	0 02 00644	PUNCH PAPER TAPE WITH LEADER	41
POL 0, 1	0 02 10260	PRINTER OFF-LINE	49
POT	13	PARALLEL OUTPUT	35
PPT 0, 1, 4	0 02 02644	PUNCH PAPER TAPE WITH NO LEADER	41
PRT 0, 1	0 40 12060	PRINTER READY TEST	50
PSC 0, 1, n	0 02 1n460	PRINTER SKIP TO FORMAT CHANNEL n	49

Designation	Instruction Code	Name	Page
PSP 0, 1, n	0 02 1n660	PRINTER SPACE n LINES	49
RCB 0, 1, 4	0 02 03606	READ CARD BINARY	45
RCD 0, 1, 4	0 02 02606	READ CARD DECIMAL (HOLLERITH)	45
RCY	0 66 200XX	RIGHT CYCLE AB	14
REW 0, n	0 02 1401n	REWIND	57
RKB 0, 1, 4	0 02 02601	READ KEYBOARD	38
ROV	0 02 20001	RESET OVERFLOW	16
RPT 0, 1, 4	0 02 02604	READ PAPER TAPE	40
RCH	46	REGISTER CHANGE	11
RSH	0 66 000XX	RIGHT SHIFT AB	14
RTB 0, n, 4	0 02 0361n	READ TAPE IN BINARY	54
RTD 0, n, 4	0 02 0261n	READ TAPE IN DECIMAL (BCD)	54
RTS 0	0 02 14000	CONVERT READ TO SCAN	57
SFB 0, n, 4	0 02 0363n	SCAN FORWARD IN BINARY	55
SFD 0, n, 4	0 02 0263n	SCAN FORWARD IN DECIMAL (BCD)	55
SKA	72	SKIP IF M AND A DO NOT COMPARE ONES	13
SKG	73	SKIP IF A GREATER THAN M	13
SKM	70	SKIP IF A = M ON B MASK	13
SKN	53	SKIP IF M NEGATIVE	14
SKS	40	SKIP IF SIGNAL NOT SET	29
SRB 0, n, 4	0 02 0763n	SCAN REVERSE IN BINARY	55
SRC 0, 1	0 02 12006	SKIP REMAINDER OF CARD	45
SRD 0, n, 4	0 02 0663n	SCAN REVERSE IN DECIMAL (BCD)	55
SRR 0	0 02 13610	SKIP REMAINDER OF RECORD	57
STA	35	STORE A	8
STB	36	STORE B	8
STX	37	STORE INDEX	9
SUB	54	SUBTRACT M FROM A	9
TFT 0	0 40 13610	TAPE END-OF-FILE TEST	53
TGT	0 40 1261n	TAPE GAP TEST	54
TOP 0	0 02 14000	TERMINATE OUTPUT ON W BUFFER	28
TRT 0, n	0 40 1041n	TAPE READY TEST	53
TYP 0, 1, 4	0 02 02641	WRITE TYPEWRITER	38
WIM	32	W BUFFER INTO M WHEN FULL	31
WTB 0, n, 4	0 02 0365n	WRITE TAPE IN BINARY	59
WTD 0, n, 4	0 02 0265n	WRITE TAPE IN DECIMAL (BCD)	59
XAB	0 46 00000	EXCHANGE A AND B	11
YIM	30	Y BUFFER INTO M WHEN FULL	31

INDEX

XDS

Xerox Data Systems 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

Building Arts Bldg.
Suite G100
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

13701 Riverside Drive
Sherman Oaks, Calif. 91403
(213) 986-5510

*Wells Fargo Building
Suite 410
Del Amo Financial Center
21535 Hawthorne Blvd.
Torrance, Calif. 90503
(213) 542-5561

3333 South Bannock
Suite 400
Englewood, Colo. 80110
(303) 761-2645

320 Ward Avenue
Honolulu, Hawaii 96814
(808) 531-8257

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 Building
Suite 415
400 108th Avenue N.E.
Bellevue, Wash. 98004
(206) 454-3991

Midwestern Region

*International Tower Building
Suite 500
8550 West Bryn Mawr Ave.
Chicago, Illinois 60631
(312) 693-6060

Clausen Bldg., Suite 310
16000 W. Nine Mile Road
Southfield, Michigan 48075
(313) 353-7360

4410 Woodson Road
Suite 111
St. Louis, Missouri 63134
(314) 423-6200

One Erieview Plaza
13th Floor
Cleveland, Ohio 44114
(216) 522-1850

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

10 Perimeter Park
Suite 111
Atlanta, Georgia 30341
(404) 451-2755

Jefferson Bank Bldg.
Suite 720
3525 N. Causeway Blvd.
Metairie, Louisiana 70002
(504) 837-1515

4920 S. Lewis Avenue
Suite 103
Tulsa, Oklahoma 74105
(918) 743-7753

One Turtle Creek Village
Suite 212
Dallas, Texas 75219
(214) 528-6580

*2300 West Loop South
Suite 150
Houston, Texas 77027
(713) 623-0510

Eastern Region

10227 Wincopin Circle
Suite 716
Columbia, Maryland 21043
(301) 730-4900

20 Walnut Street
Wellesley Hills, Mass. 02181
(617) 237-2300

Brearley Office Building
190 Moore Street
Hackensack, N. J. 07601
(201) 489-0100

The Fortune Building
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) 643-4250

Kogerama Building
Suite 212
No. 1 Tidewater
Executive Center
Norfolk, Virginia 23502
(703) 497-6811

Washington (D.C.) Operations

*2351 Research Blvd.
Rockville, Maryland 20850
(301) 948-8190

Canada

864 Lady Ellen Place
Ottawa 3, Ontario
(613) 722-8387

1009 7th Avenue, S.W.
Calgary 2, Alberta
(403) 265-8134

280 Belfield Road
Rexdale 605, Ontario
(416) 677-8422

1901 North Service Road
Trans-Canada Highway
Dorval, P.Q.
(514) 683-3755

INTERNATIONAL OFFICES
& REPRESENTATIVES

European/African Headquarters

Scientific Data Systems
York House, Empire Way
Wembley, Middlesex
HA 9 OQB
England
(01) 903-4521, 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.
66, Route de Versailles
78-Louveciennes
Yvelines
951 86 00 (Paris area)

Israel

Elbit Computers Ltd.
Subsidiary of Elron
Electronic Industries Ltd.
88 Hagiborim Street
Haifa
6 4613

*Regional Headquarters

XDS 910 INPUT/OUTPUT INSTRUCTIONS

Buffer Instructions and Tests

BUFFER CONTROL EOM[†]

Mnemonic		Code	Name	Page
EOM	A	02	Energize Output M	26
ALC	0	0 02 50000	Alert W Buffer	28
DSC	0	0 02 00000	Disconnect W Buffer	28
TOP	0	0 02 14000	Terminate Output on W Buffer	28
ASC	0	0 02 12000	Alert to Store Address in W Buffer	29

DIRECT PARALLEL INPUT/OUTPUT

Mnemonic		Code	Name	Page
PIN	A, T	33	Parallel Input	35
POT	A, T	13	Parallel Output	35

INTERNAL TEST SKS

Mnemonic		Code	Name	Page
SKS	A	40	Skip if Signal Not Set	29
BRTW		0 40 21000	W Buffer Ready Test	29
BRTY		0 40 22000	Y Buffer Ready Test	29
BETW		0 40 20010	W Buffer Error Test	29
BETY		0 40 20020	Y Buffer Error Test	29

SINGLE-WORD DATA TRANSFER

Mnemonic		Code	Name	Page
MIW	A, T	12	Memory Into W Buffer	30
MIY	A, T	10	Memory Into Y Buffer	30
WIM	A, T	32	W Buffer Into Memory	31
YIM	A, T	30	Y Buffer Into Memory	31

Peripheral Device Instructions and Tests

PAPER TAPE[†]

Mnemonic		Code	Name	Page
RPT	0, 1, 4	0 02 02604	Read Paper Tape	40
PPT	0, 1, 4	0 02 02644	Punch Paper Tape Without Leader	41
PTL	0, 1, 4	0 02 00644	Punch Paper Tape With Leader	41

CARD[†]

Mnemonic		Code	Name	Page
RCB	0, 1, 4	0 02 03606	Read Card Binary	45
RCD	0, 1, 4	0 02 02606	Read Card Decimal (Hollerith)	45
SRC	0, 1	0 02 12006	Skip Remainder of Card	45
CRT	0, 1	0 40 12006	Card Reader Ready Test	45
FCT	0, 1	0 40 14006	First Column Test	45
CFT	0, 1	0 40 11006	Card Reader EOF Test	45
PCD	0, 1, 4	0 02 02646	Punch Card Decimal (Hollerith)	46
PCB	0, 1, 4	0 02 03646	Punch Card Binary	46
PBT	0, 1	0 02 12046	Punch Buffer Test	48
CPT	0, 1	0 02 14046	Card Punch Ready Test	48

LINE PRINTER[†]

Mnemonic		Code	Name	Page
PLP	0, 1, 4	0 02 02660	Print Line Printer	49
POL	0, 1	0 02 10260	Printer Off-Line	49
PSC	0, 1, n	0 02 1n460	Printer Skip to Format Channel n	49
PSP	0, 1, n	0 02 1n660	Printer Upspace n Lines	49
PFT	0, 1	0 40 11060	Printer Fault Test	50
PRT	0, 1	0 40 12060	Printer Ready Test	50
EPT	0, 1	0 40 14060	End of Page Test	50

TYPEWRITER[†]

Mnemonic		Code	Name	Page
RKB	0, 1, 4	0 02 02601	Read Keyboard	38
TYP	0, 1, 4	0 02 02641	Write Typewriter	38

MAGNETIC TAPE[†]

Mnemonic		Code	Name	Page
TRT	0, n	0 40 1041n	Tape Ready Test	53
FPT	0, n	0 40 1401n	File Protect Test	53
BTT	0, n	0 40 1201n	Beginning of Tape Test	53
ETT	0, n	0 40 1101n	End of Tape Test	53
DT2	0, n	0 40 1621n	Density Test, 200 BPI	53
DT5	0, n	0 40 1661n	Density Test, 556 BPI	53
DT8	0, n	0 40 1721n	Density Test, 800 BPI	53
TFT	0	0 40 13610	Tape End-of-File Test	53
TGT	0	0 40 12610	Tape Gap Test	54
SKS	01021n	0 40 1021n	MAGPAK Test	54
RTB	0, n, 4	0 02 0361n	Read Tape Binary	54
RTD	0, n, 4	0 02 0261n	Read Tape Decimal (BCD)	54
SFB	0, n, 4	0 02 0363n	Scan Forward Binary	55
SFD	0, n, 4	0 02 0263n	Scan Forward Decimal (BCD)	55
SRB	0, n, 4	0 02 0763n	Scan Reverse Binary	55
SRD	0, n, 4	0 02 0663n	Scan Reverse Decimal (BCD)	55
REW	0, n	0 02 1401n	Rewind	57
RTS	0	0 02 14000	Convert Read to Scan	57
SRR	0	0 02 13610	Skip Remainder of Record	57
WTB	0, n, 4	0 02 0365n	Write Tape Binary	57
WTD	0, n, 4	0 02 0265n	Write Tape Decimal (BCD)	57
EFT	0, n, 4	0 02 0367n	Erase Tape Forward	57
ERT	0, n, 4	0 02 0767n	Erase Tape in Reverse	57

Legend: A = address; *A indirect address; T = tag field; n = number ($0 \le n \le 7$)

[†] Mnemonics and octal codes given are for the W buffer, device number 1 (or n), and the 4 characters/word mode. Octal codes for the Y buffer are obtained by adding 0100 to the codes for these instructions.

XDS
Xerox Data Systems

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

Building Arts Bldg.
Suite G100
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

13701 Riverside Drive
Sherman Oaks, Calif. 91403
(213) 986-5510

*Wells Fargo Building
Suite 410
Del Amo Financial Center
21535 Hawthorne Blvd.
Torrance, Calif. 90503
(213) 542-5561

3333 South Bannock
Suite 400
Englewood, Colo. 80110
(303) 761-2645

320 Ward Avenue
Honolulu, Hawaii 96814
(808) 531-8257

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 Building
Suite 415
400 108th Avenue N.E.
Bellevue, Wash. 98004
(206) 454-3991

Midwestern Region

*International Tower Building
Suite 500
8550 West Bryn Mawr Ave.
Chicago, Illinois 60631
(312) 693-6060

Clausen Bldg., Suite 310
16000 W. Nine Mile Road
Southfield, Michigan 48075
(313) 353-7360

4410 Woodson Road
Suite 111
St. Louis, Missouri 63134
(314) 423-6200

One Erieview Plaza
13th Floor
Cleveland, Ohio 44114
(216) 522-1850

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

10 Perimeter Park
Suite 111
Atlanta, Georgia 30341
(404) 451-2755

Jefferson Bank Bldg.
Suite 720
3525 N. Causeway Blvd.
Metairie, Louisiana 70002
(504) 837-1515

4920 S. Lewis Avenue
Suite 103
Tulsa, Oklahoma 74105
(918) 743-7753

One Turtle Creek Village
Suite 212
Dallas, Texas 75219
(214) 528-6580

*2300 West Loop South
Suite 150
Houston, Texas 77027
(713) 623-0510

Eastern Region

10227 Wincopin Circle
Suite 716
Columbia, Maryland 21043
(301) 730-4900

20 Walnut Street
Wellesley Hills, Mass. 02181
(617) 237-2300

Brearley Office Building
190 Moore Street
Hackensack, N. J. 07601
(201) 489-0100

The Fortune Building
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) 643-4250

Kogerama Building
Suite 212
No. 1 Tidewater
Executive Center
Norfolk, Virginia 23502
(703) 497-6811

Washington (D.C.) Operations

*2351 Research Blvd.
Rockville, Maryland 20850
(301) 948-8190

Canada

864 Lady Ellen Place
Ottawa 3, Ontario
(613) 722-8387

1009 7th Avenue, S.W.
Calgary 2, Alberta
(403) 265-8134

280 Belfield Road
Rexdale 605, Ontario
(416) 677-8422

1901 North Service Road
Trans-Canada Highway
Dorval, P.Q.
(514) 683-3755

INTERNATIONAL OFFICES & REPRESENTATIVES

European/African Headquarters

Scientific Data Systems
York House, Empire Way
Wembley, Middlesex
HA 9 OQB
England
(01) 903-4521, 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.
66, Route de Versailles
78-Louveciennes
Yvelines
951 86 00 (Paris area)

Israel

Elbit Computers Ltd.
Subsidiary of Elron
Electronic Industries Ltd.
88 Hagiborim Street
Haifa
6 4613

*Regional Headquarters