CONTROL DATA
924 COMPUTER
REFERENCE MANUAL

924 INSTRUCTIONS

			Page
ADD	Add	14	2-17
ADL	Add Logical	45	2-24
AJP	A Jump	22	2-36, 40
ALS	A Left Shift	05	2-27
ARS	A Right Shift	01	2-26
ATI	A to Index	26	2-16
CMA	Complement A	52 1	2-25
CMQ	Complement Q	52 2	2-25
DVI	Divide	25	2-19
ENA	Enter A	10	2-15
ENI	Enter Index	50	2-16
ENQ	Enter Q	04	2-15
EQS	Equality Search	64	2-33
EXF	External Function	74	3-3
IJP	Index Jump	55	2-38
INA	Increase A	11	2-21
INI	Increase Index	51	2-21
INQ	Increase Q	56	2-21
ISK	Index Skip	54	2-31
LAC	Load A Complement	13	2-11
LDA	Load A	12	2-11
LDL	Load Logical	44	2-24
LDQ	Load Q	16	2-11
LIL	Load Index	53	2-13
LLS	AQ Left Shift	07	2-26
LQC	Load Q Complement	17	2-11
LRS	AQ Right Shift	03	2-26
MEQ	Masked Equality	66	2-34
MTH	Masked Threshold	67	2-34
MUI	Multiply	24	2-18
PTS	Pattern Search	63	2-34
QJP	Q Jump	23	2-37, 40

			Page
QLS	Q Left Shift	06	2-27
QRS	Q Right Shift	02	2-26
QTI	Q to Index	27	2-16
RAD	Replace Add	70	2-29
RAO	Replace Add One	72	2-29
RSB	Replace Subtract	71	2-29
RSO	Replace Subtract One	73	2-29
SAL	Substitute Address	61	2-13
SBL	Subtract Logical	46	2-24
SCA	Scale A	34	2-28
SCL	Selective Clear	41	2-23
SCM	Selective Complement	42	2-22
SCQ	Scale AQ	35	2-28
SIL	Store Index	57	2-13
SKH	Skip High	30	2-31
SKL	Skip Low	31	2-31
SLJ	Selective Jump	75	2-37, 40
SLS	Selective Stop	76	2-37, 40
SSH	Storage Shift	37	2-30
SSK	Storage Skip	36	2-30
SST	Selective Set	40	2-22
SSU	Selective Substitute	43	2-24
STA	Store A	20	2-12
STL	Store Logical	47	2-24
STQ	Store Q	21	2-12
SUB	Subtract	15	2-17
TAL	Tally	62	2-20
THS	Threshold Search	65	2-33
UJP	Unconditional Jump	32	2-37
URJ	Unconditional Return Jump	33	2-41
XAQ	Interchange A and Q	52 0	2-12
XEC	Execute	60	2-41

CONTROL DATA 924 COMPUTER

REFERENCE MANUAL

168c
REV 3/63

Minor Revision (March, 1963)
This edition, publication 168c, is a minor revision but does not obsolete publication 168b. Pages changed in this edition are marked with a dot in the upper outside corner.

Address comments concerning this manual to:

Control Data Corporation
Technical Publications Department
501 Park Avenue
Minneapolis 15, Minnesota

RECORD OF CHANGE NOTICES

C. N. NO.	DATE ORIGINATED	DATE ENTERED	INITIALS	REMARKS

CONTENTS

GLOSSARY

APPENDIX SECTION

FIGURES

Chapter 1. Description

Chapter 3. Input-Output

Chapter 4. Operation

TABLES

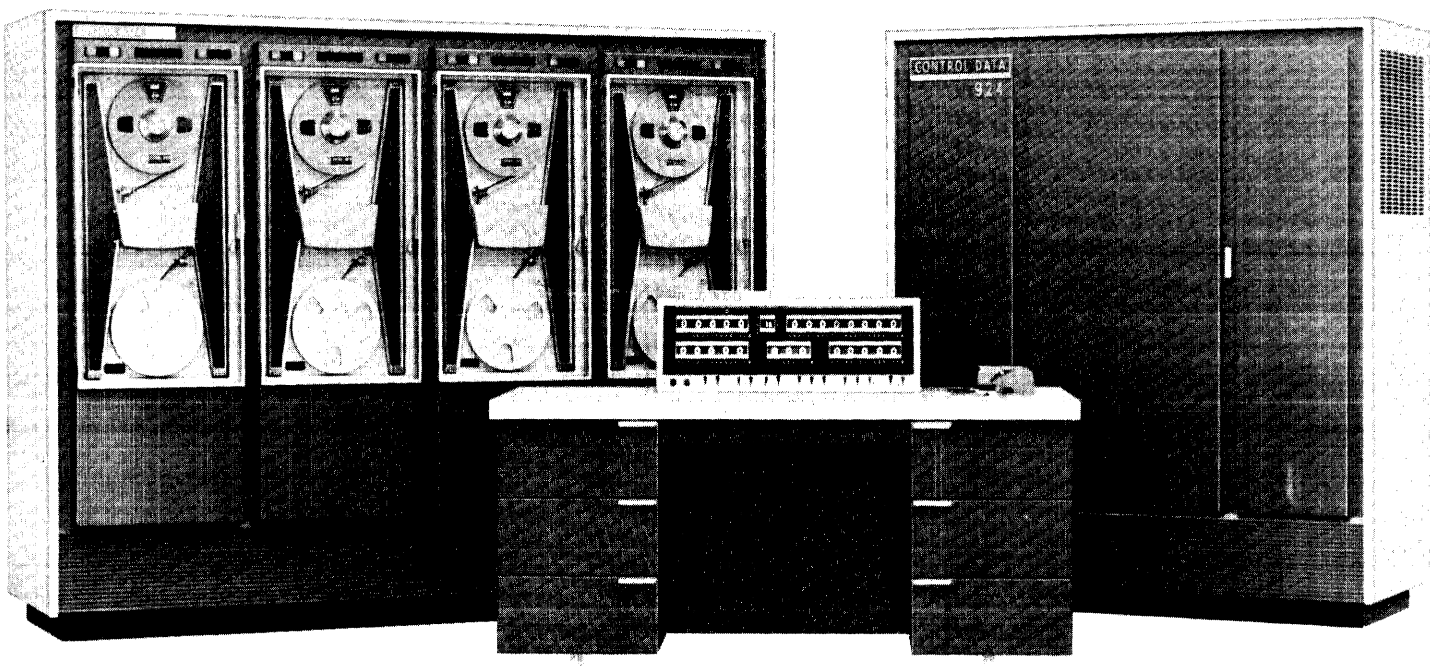
Chapter 1. Description

Chapter 3. Input-Output

Chapter 4. Operation

1

Description


CONTROL DATA
924

The CONTROL DATA* 924 Computer is a general purpose digital computer designed to solve both business and scientific problems. The reliability of high-speed transistor amplifier circuits and the efficiency of parallel operations are combined in the 924 to produce exceedingly fast computation and transfer speeds. Greater speed, reliability, and efficiency coupled with modular construction and large storage capacity results in an extremely versatile and powerful computer having highly flexible systems applications.

PHYSICAL DESCRIPTION

The basic 924 installation consists of two units; the console and main computer cabinet. An installation may also include, as optional equipment, any input-output devices capable of communication with the 160 and/or 1604 computer such as the 1607 Magnetic Tape System, the 1610 Punched Card Control Unit, the 1612 High Speed Line Printer and the 161 Typewriter.

The console contains the operator's display panel, controls, paper tape punch and paper tape reader.

The main cabinet holds the four printed circuit card chassis; a part of the core storage assembly is held in each of two chassis. The chassis are hinged for easy access (figure 1-1). Cabinet dimensions, to the nearest inch, are: Height 5 feet 8 inches, Depth 2 feet 2 inches, Length 5 feet 2 inches, and Weight approximately 1430 pounds.

---

* Registered trademark of Control Data Corporation

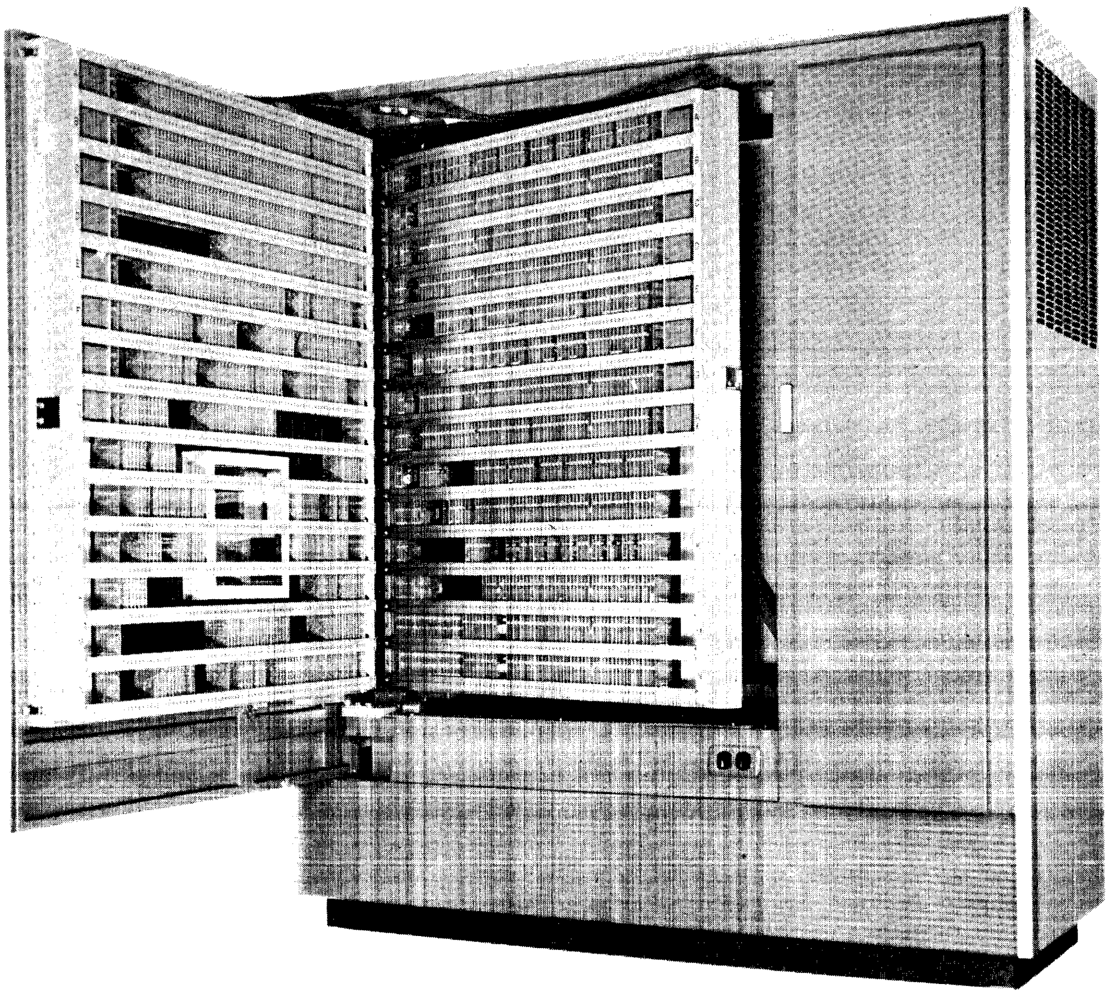


Figure 1-1. Main Cabinet Interior

924 CHARACTERISTICS

Stored-program, general-purpose digital computer

Parallel mode

24-bit word, 1 instruction per word

Single-address logic

Operation code	6 bits
Index designator	3 bits
Execution address	15 bits

Six index registers

Indirect addressing

Magnetic core storage
8, 192 24-bit words

Two independent 4096 word banks alternately phased

5. 3 μsec effective cycle time (representative program)

6. 4 μsec total cycle time

1. 8 μsec access time

Input-output
- Three 48-bit buffer input channels, compatible with 1604 peripheral equipment, 12 lower-order bits compatible with 160 peripheral equipment.
- Three 48-bit buffer output registers, compatible with 1604 peripheral equipment, 12 lower-order bits compatible with 160 peripheral equipment.

Program interrupt

Console, includes:
- Photoelectric paper tape reader
- Paper tape punch
- Register contents displayed in octal

Flexible 64 instructions
- Fixed point arithmetic
- Logical and masking operations
- Indexing
- Input-output
- Conditional and unconditional
- Jumps and stops
- Multiple precision capability (accumulator and auxiliary register operate as a single double-length register)
- Storage searching

Binary arithmetic
- Modulus 2^{24}-1 (one's complement)
- Parallel addition in 1. 2 μsec without access

Real-time clock

Completely solid-state
- Diode logic
- Transistor amplifiers
- Magnetic core storage

Small size
- Less than 400 square feet of floor space required

Low power consumption

Optional features
- Increase standard 8, 192 words of magnetic core storage to 16, 384 or 32, 768 words

LOGICAL DESCRIPTION

The computer performs calculations and processes data in a parallel binary mode through the step-by-step execution of individual instructions which are stored internally along with the data.

Functionally, computer operation may be divided into four major sections. INPUT-OUTPUT provides communication between the computer and the external equipment; ARITHMETIC performs the arithmetic and logical operations required for executing instructions; STORAGE provides internal storage for data and instructions; and CONTROL coordinates and sequences all operations for executing an instruction by obtaining the instruction from storage and translating it into commands for the other sections.

The registers in the computer are identified by letters. The operational registers (table 1-1) usually hold the end result of an operation. Their contents are displayed on the console and may be manually changed by the operator. The arithmetic properties of these registers are detailed in table 1-2. The transient registers used in formulating the result are secondary registers. They are not displayed and cannot be manually changed.

STORAGE SECTION

The magnetic core storage section of the 924 Computer provides high-speed, random access storage for 8,192 words. It consists of two independent storage units each with a capacity of 4096 words. These units operate together during the execution of a stored program and thus are considered as one 8,192 word storage system.

A word is 24 bits in length and is used as a 24-bit instruction or a 24-bit operand (data word). The location of each word in storage is identified by an assigned number or address. When a word is taken (read) from or entered (written) into storage, a reference is made to the storage address which holds the word. All odd storage addresses are located in one storage unit; all even addresses in the other.

The cycle time, or time for a complete storage reference, is 6.4 microseconds. Since the storage cycles of the two sections overlap one another in the execution of a program, the average effective cycle time for random addresses is about 5.3 microseconds.

TABLE 1-1. OPERATIONAL REGISTERS OF THE COMPUTER

Register	Function
A	Arithmetic
Q	Auxiliary Arithmetic
B^1 through B^6	Index registers (six)
P	Program Address
U	Program Control

TABLE 1-2. ARITHMETIC PROPERTIES OF REGISTERS

Register	No. of stages	Modulus	Complement Notation*	Arithmetic	Result
A	24	$2^{24}-1$	one's	subtractive	signed**
Q	24	$2^{24}-1$	one's		signed
P	15	2^{15}	two's	additive	unsigned
U	15	2^{15}	two's	subtractive	unsigned

* Refer to Appendix

**The result of an arithmetic operation in A satisfies $A \leq 2^{23}-1$ since A always is treated as a signed quantity. When the result in A is zero, it is always represented by 000...00 except when 111...11 is added to 111...11. In this case, the result is 111...11 (negative zero). If an instruction calls for subtracting positive zero (000...00) from negative zero, the computer complements the minuend and adds so that the actual operation is the addition of negative zero to negative zero and the result is negative zero.

INPUT-OUTPUT

The input-output section of the computer handles the flow of information to and from the computer. Prior to executing a program, the data and instructions which comprise the program (input) are loaded into computer storage. After computation is completed, the results (output) are transmitted from storage to an external equipment. All information is transmitted in 24-bit words.

The computer communicates with external equipment through six independent buffer channels which provide for the normal exchange of data (figure 1-2).

Input:	Channel 1	Output:	Channel 2
	Channel 3		Channel 4
	Channel 5		Channel 6

The input and output buffer channels are paired, channels 1 and 2, channels 3 and 4, and channels 5 and 6. Every external equipment is connected to one of these pairs. It is possible to connect as many as five different equipments to any given pair of channels. All six buffer channels may be concurrently transmitting information. However, only one external equipment can use any one buffer channel at any given instant.

In the 924 computer, input-output operations are independent of the main computer program. When data is transmitted, the main computer program initiates an automatic cycle which buffers data to and from computer storage. The main computer program then continues while the actual buffering of data is carried out independently and automatically.

This process of asynchronous input-output operations will be termed a buffer. Buffer transmissions employ independent access to computer storage so that computation continues while the external equipment is loading or unloading information from computer storage. The rate of exchange is, in most cases, dictated by the external equipment.

ARITHMETIC SECTION

The arithmetic section of the 924 computer consists of two operational registers, A and Q, and several secondary registers.

The A register (accumulator) is the principal arithmetic register. Some of the more important functions of A are:

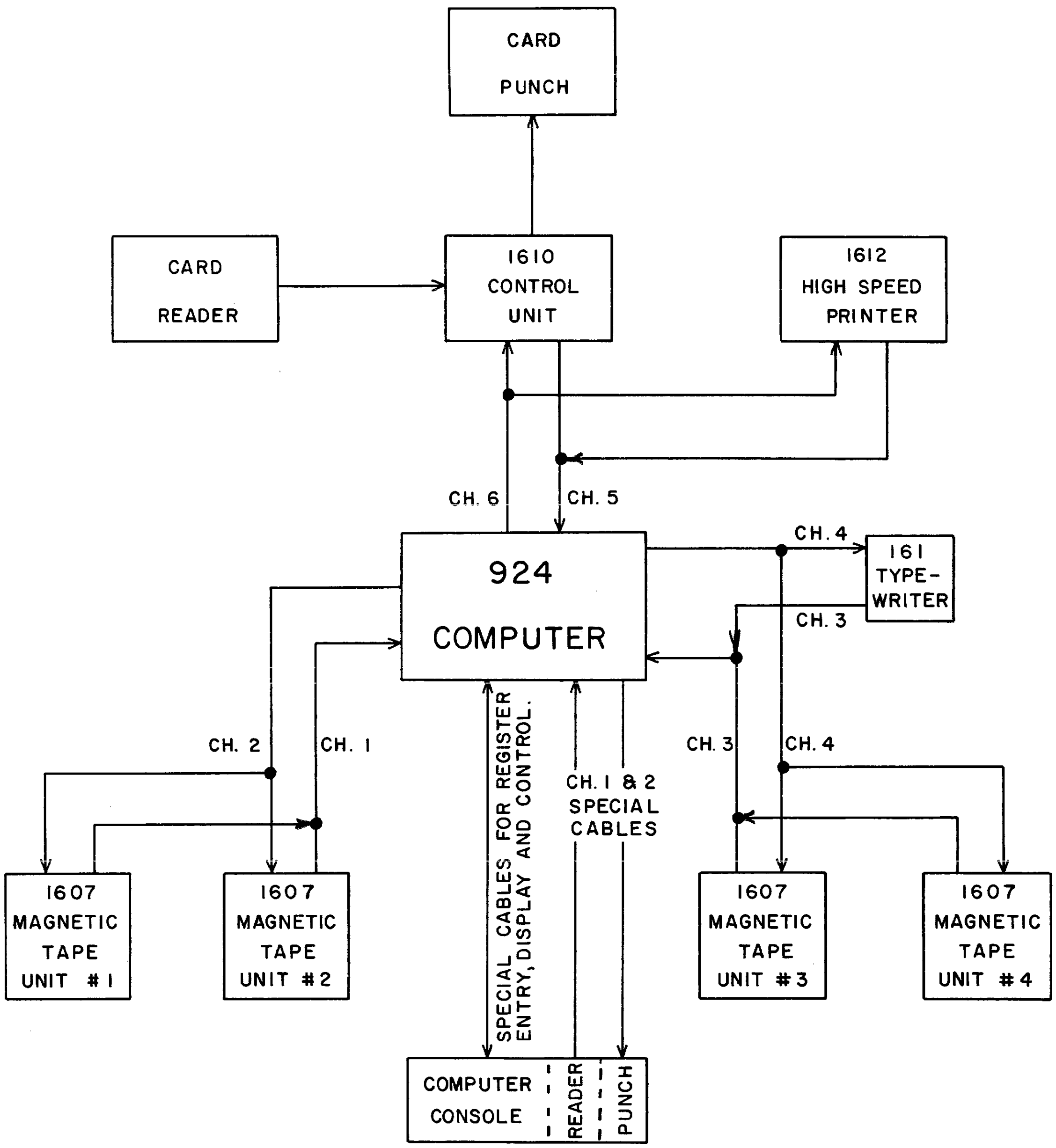


Figure 1-2. 924 Input-Output System

1) All arithmetic and logical operations use the A register in formulating a result. The A register is the only register with provisions for adding its contents and the contents of a storage location or another register.

2) Shifting - A may be shifted to the right or left separately or in conjunction with Q. Right shifting is open-ended; the lowest bits are discarded and sign extended. Left shifting is circular; the highest order bit appears in the lowest order stage after each shift; all other bits move one place to the left.

3) Control for conditional instructions - A holds the word which conditions jump and search instructions.

The Q register is an auxiliary arithmetic register and is generally used in conjunction with the A register. The principal functions of Q are:

1) Providing temporary storage of contents of A while A is used for another arithmetic operation.

2) Forming a double-length register, AQ or QA.

3) Shifting to the right or left, separately or in conjunction with A.

Both A and Q may load (ATI, QTI) or be loaded from (ENA B 0, ENQ B 0) any of the six index registers without the use of storage references. Similarly, the contents of A and Q may be interchanged to permit the use of Q as an auxiliary accumulator. (XAQ)

CONTROL SECTION

The control section directs the operations required to execute instructions and to exchange data with external equipment. It also establishes the timing relationships needed to perform the operations in the proper sequence.

The control section acquires an instruction from storage, interprets it, and sends the necessary commands to other sections. The composition of a 924 instruction is shown below.

Instruction Format

f (6 bits)	b (3 bits)	m, y, or k or unused (15 bits)
operation code	index designator	base execution address

Each of the 64 instructions has an operation code which specifies the operation to be performed. This code is usually written as a 2 octal digit numeric code or as a mnemonic code (see appendix).

The index designator generally is used for address modification; it specifies one of the six index registers whose contents are to be added to the base execution address. The index designator may also condition jump and stop instructions, specify indirect addressing, or further distinguish between two unlike instructions having the same operation code.

The execution address may be used as an address of an operand, m; as an operand, y; or as a shift count, k.

The eight operational registers in the control section are P, U and B^1 through B^6.

The program address counter (P) is a two's complement additive register. It provides program continuity by generating in sequence the storage addresses which contain the individual instructions. Usually at the completion of each instruction, the count in P is advanced by one to specify the address of the next instruction.

The program control register, U, holds an instruction while it is executed. After executing an instruction, an exit, jump exit or skip exit is performed. An exit advances the count in P by one and executes the next instruction specified by the contents of P. A jump exit executes the instruction at the storage location specified by the execution address of the jump instruction. The execution address is, in this case, entered into P and specifies the starting location of a new sequence of instructions. A skip exit advances the count in P by two, bypassing the next sequential instruction and executing the following one.

Each of the six index registers, B^1 through B^6, provides storage for quantities which are used in a variety of ways, depending on the instruction. The B registers have no provisions for arithmetic operations. In the majority of instructions the B registers hold quantities to be added to the execution address. All address modifications are performed in A. For search instructions, the contents of a B register indicate the number of words to be searched. This quantity is transmitted to U, and reduced one count for each word searched.

2

Description of Instructions

WORD FORMAT

A computer word consists of 24 bits and may be interpreted as a 24-bit instruction or data word. Each instruction is composed of three parts or codes: operation code, index designator, and execution address.

bit [redacted] 23 ... bit 00

Operation (Function) Code f	Index Designator b	Execution Address m, y, k
6 bits	3 bits	15 bits

Code	Range	
Operation f	01 - 76_8	Specifies the operation to be performed. Codes 00, 52.3-52.7 and 77 are interpreted as faults which stop computer operation.
Execution Address m, y, k	00000 through 77777_8	Used in one of three ways: 1) as a shift count k 2) as an operand address, m 3) as an operand, y
Index Designator b	0 1-6 7	No address modification Relative address modification Specifies the index designator whose contents are to be added to the execution address. (Refer to jump and stop instructions for exceptions.) Indirect addressing

Execution Address

The base execution address may be used as: (1) a shift count, k; (2) an operand, y; (3) an address of an operand, m, in storage. The execution address may also be modified or left unmodified depending on the index designator. If unmodified, the address is represented by the lower-case symbol k, y, or m; if the address is modified the symbols are capitalized. The following examples point out the relationship between the unmodified and modified execution address.

The modified shift count K is represented by:

1) $K = k + (B^b)$ where:
   - K = modified shift count
   - k = unmodified shift count (execution address)
   - (B^b) = contents of index register b.

   If the index designator = 0, then K = k.

The modified operand Y is represented by:

2) $Y = y + (B^b)$ where:
   - Y = modified operand
   - y = unmodified operand (execution address)
   - (B^b) = contents of index register b.

   If the index designator = 0, then Y = y.

The modified operand address M is represented by:

3) $M = m + (B^b)$ where:
   - M = modified address of operand
   - m = unmodified address of operand (execution address)
   - (B^b) = contents of index register b.

   If the index designator = 0, then M = m. Note that (3) is the only case in which the execution address is interpreted as an address of an operand.

Address Modification

The three possible modes of address modification are identified by the index designators as follows:

1) b = 0 No Address Modification. In this mode the execution address is interpreted without modification; nothing is added to or subtracted from it. (Direct addressing.)

2) b = 1-6 Relative Address Modification. In this mode the execution address is modified and is equal to the initial execution address plus the contents of the designated index register. One's complement arithmetic is used in determining the modified execution address.

3) b = 7 Indirect Addressing. In this mode a storage reference is made to the location designated by the execution address. The lower-order 18 bits of the 24-bit word are read from storage and interpreted as the b designator (3 bits) and execution address (15 bits) of the present instruction. The new index designator may refer to any one of the three modes.

Examples:

1) No Address Modification

f b m
LDA 0 address

This instruction is interpreted as load accumulator from the storage location designated by the sum of the execution address and the contents of the specified index register, B^b. Since b = 0, no index register is designated and m specifies the storage location whose contents are loaded into A.

2) Relative Address Modification

f b m
LDA 6 address $(B^6) = 00001_8$

In this example, the accumulator is loaded from the storage location designated by the execution address plus the contents of index register 6. Therefore, the contents of the storage location named by the execution address plus 00001_8 is loaded into the accumulator. $M = m + (B^b)$.

3) Indirect Addressing

Current Instruction = LDA 7 00100 (f b m)

(00100) = STA 6 00200

$(B^6) = 00001_8$

When the b designator of the current instruction is 7, the mode is indirect addressing. The lower 18 bits of the contents of the storage location designated by the execution address, 00100, are read from storage into the U register where they are interpreted as the index designator and execution address of the current instruction.

The index designator is inspected again and because it is not 0 or 7 the relative address mode exists. (Note that the new index designator could reference any one of the three modes of address modification.) The execution address, 00200, plus the contents of B^6, 00001_8 specify the storage location whose contents will be loaded into the accumulator. $M = 00200_8 + (00001_8) = 00201_8$

Sequential Execution of Instructions

Example:

	f	b	m
(00300) =	LDA	0	00310
(00301) =	ADD	1	00210
	$(B^1) = 00101_8$		
(00302) =	. . . .		

The P register holds address 00300 (an even lowest bit indicates the address of the program step is in the even storage unit). The storage reference is initiated; the 24-bit word is read from address 00300 and entered into U . Computer operation is now dependent upon the interpretation of the 24-bit instruction in U.

The operation code, LDA, and the index designator, 0, are translated. The function of the instruction, LDA, is to load the A register with the contents of the designated storage location. Because the index designator is 0, the execution address is not modified. The translation of the operation code initiates the sequence of the commands which execute the instruction and the operand in address 00310 is loaded into A.

The contents of P are increased by one and the next instruction is read from storage address 00301. This instruction is now translated in U. The ADD instruction causes the quantity in storage location M to be added to the contents of the A register. Since the index designator is not 0 or 7, the contents of the index register are added to the execution address to form M. $M = m + (B^b) = 00210_8 + 00101_8 = 00311_8$. The contents of storage address 00311 are added to the contents of the A register completing the instruction. The contents of the P register are again increased by one and the instruction at address 00302 is read from storage and executed.

INSTRUCTIONS

The 64 computer instructions are described on the following pages, (EXF instructions are discussed in detail in chapter three). The title line contains the mnemonic code and format, name, and average execution time of the instruction. Abbreviations and symbols are defined as follows:

A	A register (accumulator)
A_n	The binary digit in position n of the A register
$\rightarrow$	Transmit to
b	Index designator
B^b	Designated index register
Exit	Proceed to next instruction
j	The condition designator for jump and stop instructions
k	Unmodified shift count
K	Modified shift count. $K = k + (B^b)$
m	Unmodified operand address
M	Modified operand address. $M = m + (B^b)$
( )	Contents of a register or storage location
( )'	One's complement contents of a register or storage location
( )f	Final contents of a register or storage location
( )i	Initial contents of a register or storage location
Q	Auxiliary arithmetic register
y	Unmodified operand
Y	Modified operand. $Y = y + (B^b)$

Instruction Execution Time

The time needed to execute an instruction varies from application to application because of the following factors.

If consecutive storage references are made to the same storage unit (even-even or odd-odd) the read access time from storage will be maximized.

If the base execution address is to be modified, the instruction execution time will be increased if consecutive instructions and operand addresses are used. There will be no increase in execution time if non-consecutive addresses in the same bank are used for instructions and operand words.

If indirect addressing is specified, at least one additional reference will be needed to complete the instruction. (The new index designator may itself specify indirect addressing.)

If an input-output request exists, the request will, in most cases, be processed before the next instruction is executed. (Refer to chapter three.)

If a storage reference is made at the end of the preceding instruction, execution of the next instruction may be delayed.

The instruction execution times listed on the following pages were compiled by averaging the times for a long list of the same instructions. The list was arranged for typical values of the factors.

ORDER OF INSTRUCTIONS

Mnemonic Code	Name	Timing*
FULL-WORD TRANSMISSION		
LDA	LOAD A	9.9
LAC	LOAD A COMPLEMENT	9.9
LDQ	LOAD Q	9.9
LQC	LOAD Q COMPLEMENT	9.9
STA	STORE A	9.8
STQ	STORE Q	9.8
XAQ	INTERCHANGE A AND Q	6.2
ADDRESS TRANSMISSION		
LIL	LOAD INDEX	9.3
SIL	STORE INDEX	9.8
SAL	SUBSTITUTE ADDRESS	9.8
ENA	ENTER A	7.0
ENQ	ENTER Q	7.0
ATI	A TO INDEX	7.0
QTI	Q TO INDEX	7.0
ENI	ENTER INDEX	6.2
FULL-WORD ARITHMETIC		
ADD	ADD	9.9
SUB	SUBTRACT	9.9
MUI	MULTIPLY	20.4 + .8n*
DVI	DIVIDE	38.0
TAL	TALLY	14.8

---

* Timing is average execution time in usec

* n = Number of ones in multiplier

ORDER OF INSTRUCTIONS

ADDRESS ARITHMETIC

INA	INCREASE A	7.0
INI	INCREASE INDEX	7.0
INQ	INCREASE Q	7.0

LOGICAL

SST	SELECTIVE SET	9.9
SCM	SELECTIVE COMPLEMENT	9.9
SCL	SELECTIVE CLEAR	9.9
SSU	SELECTIVE SUBSTITUTE	9.9
LDL	LOAD LOGICAL	9.9
ADL	ADD LOGICAL	9.9
SBL	SUBTRACT LOGICAL	9.9
STL	STORE LOGICAL	9.8
CMA	COMPLEMENT A	6.2
CMQ	COMPLEMENT Q	6.2

SHIFTING

ARS	A RIGHT SHIFT	6.2 + .4s*
QRS	Q RIGHT SHIFT	6.2 + .4s*
LRS	AQ RIGHT SHIFT	6.2 + .4s*
LLS	AQ LEFT SHIFT	6.2 + .4s*
QLS	Q LEFT SHIFT	6.2 + .4s*
ALS	A LEFT SHIFT	6.2 + .4s*
SCA	SCALE A	3.8 + .4s*
SCQ	SCALE AQ	3.8 + .4s*

---

* s = Number of places shifted

ORDER OF INSTRUCTIONS

REPLACE

RAD	REPLACE ADD	15.9
RSB	REPLACE SUBTRACT	15.9
RAO	REPLACE ADD ONE	15.9
RSO	REPLACE SUBTRACT ONE	15.9

STORAGE SEARCH

EQS	EQUALITY SEARCH	6.9 + 5.2r*
THS	THRESHOLD SEARCH	6.9 + 6.0r*
MEQ	MASKED EQUALITY	6.9 + 6.0r*
MTH	MASKED THRESHOLD	6.9 + 6.0r*
PTS	PATTERN SEARCH	6.9 + 6.0r*

STORAGE TEST

SSK	STORAGE SKIP	9.2
SSH	STORAGE SHIFT	15.9

SKIP

ISK	INDEX SKIP	7.0
SKH	SKIP HIGH	6.4
SKL	SKIP LOW	6.4

JUMPS AND STOPS (Normal)

AJP	A JUMP	7.7
QJP	Q JUMP	7.7
SLJ	SELECTIVE JUMP	7.9
SLS	SELECTIVE STOP	7.9
UJP	UNCONDITIONAL JUMP	4.8
IJP	INDEX JUMP	7.0

*r = Number of additional words searched

ORDER OF INSTRUCTIONS

JUMPS AND STOPS (Return)

AJP	A JUMP	7.7
QJP	Q JUMP	7.7
SLJ	SELECTIVE JUMP	7.9
SLS	SELECTIVE STOP	7.9
URJ	UNCONDITIONAL RETURN JUMP	4.8
XEC	EXECUTE	8.2 + x usec for instruction at M

FULL-WORD TRANSMISSION

1) In Full-Word Transmission instructions, a 24-bit operand or data word is used in executing the instruction.

2) All modes of address modification apply to the full-word transmission instructions (except XAQ).

3) With the exception of XAQ, one storage reference is made during the execution of full-word transmission instructions. If indirect addressing is designated, at least two storage references are made. No storage reference is made during the execution of the XAQ instruction.

LDA b m Load A 9.9 us

Replaces the contents of A with a 24-bit operand contained in storage location M. The contents of M remain unchanged.

LAC b m Load A complement 9.9 us

Replaces the contents of A with the complement of a 24-bit operand contained in storage location M. The contents of M remain unchanged.

LDQ b m Load Q 9.9 us

Replaces the contents of Q with a 24-bit operand contained in the storage location M. The contents of M remain unchanged.

LQC b m Load Q Complement 9.9 us

Replaces the contents of Q with the complement of a 24-bit operand contained in storage location M. The contents of M remain unchanged.

STA b m Store A 9.8 us

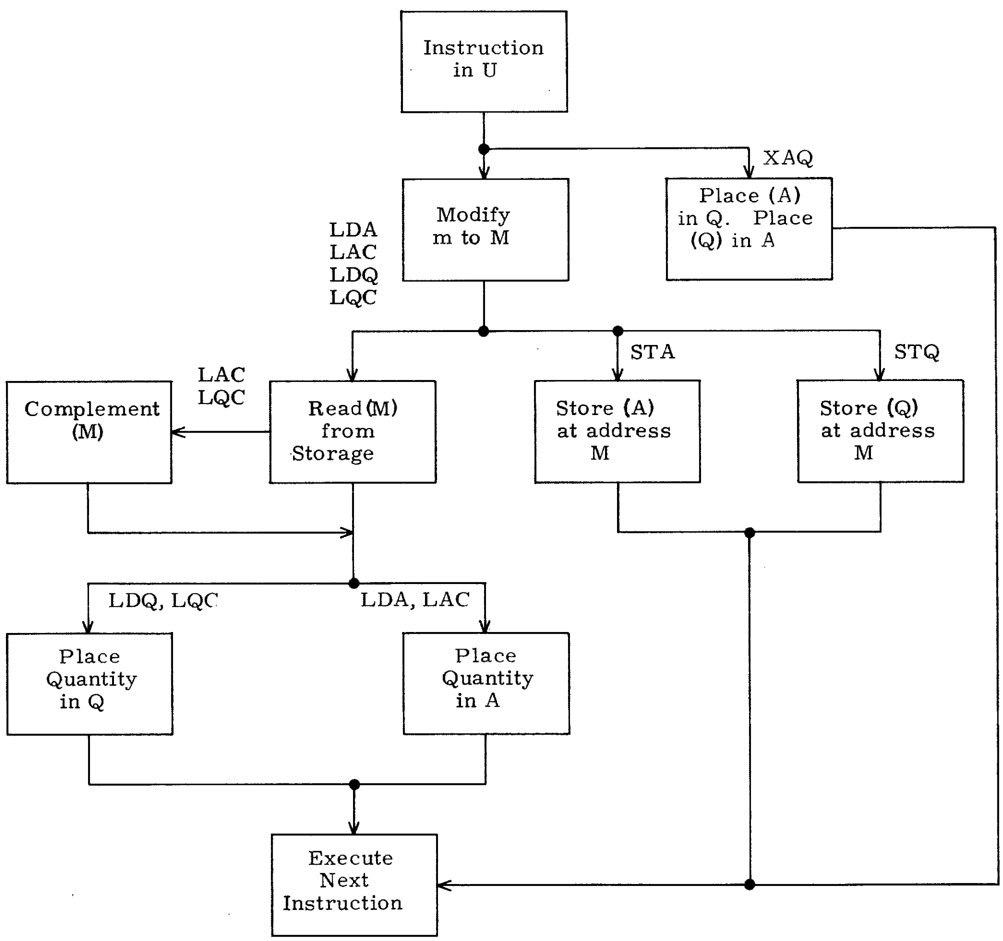
Replaces the contents of the designated storage location, M, with the contents of A. The initial contents of A remain unchanged.

STQ b m Store Q 9.8 us

Replaces the contents of the designated storage location, M, with the contents of Q. The initial contents of Q remain unchanged.

XAQ 0 m Interchange A and Q* 6.2 us

Places the contents of the A register into the Q register and at the same time places the contents of the Q register into the A register. The result of this instruction is that the contents of the A and Q registers are interchanged. The m portion of this instruction is not used.



LDA, LAC, LDQ, LQC, STA, XAQ, and STQ

---

* If the index designator does not have the indicated value, the instruction will not be executed.

ADDRESS TRANSMISSION

1) In the Address Transmission instructions, only the address portion (the lower 15 bits) of the word is used.

2) One storage reference is made during LIL, SIL, and SAL instructions. If indirect addressing is designated, at least two storage references are required. If indirect addressing is designated in the ENI instruction, one storage reference is required. No storage references are made for the remaining instructions.

3) Address modification applies to the SAL, ENQ and ENA instructions only.

LIL b m Load Index 9.3 us

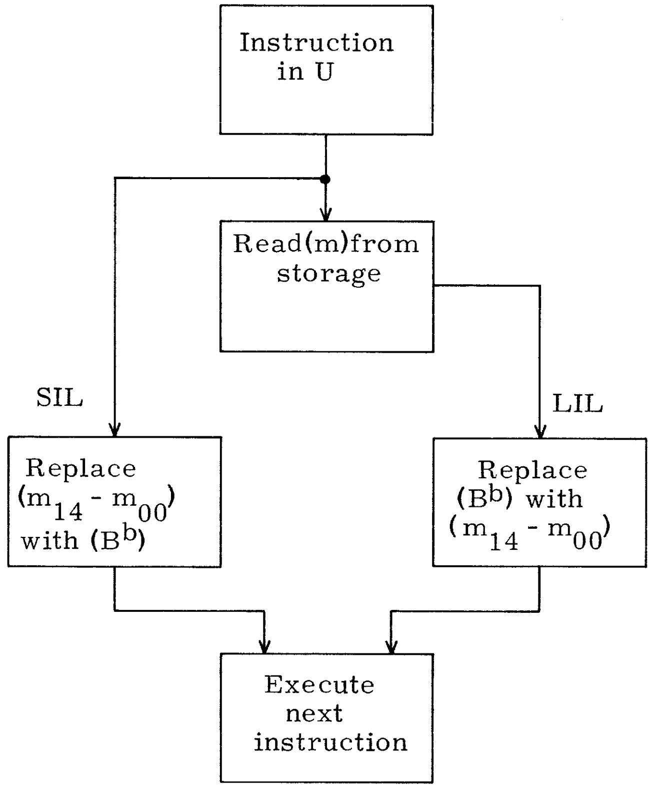
Replaces the contents of the designated index register with the address portion of storage location m. If b = 0 this instruction becomes a pass (do nothing) instruction. Initial contents of m remain unchanged.

SIL b m Store Index 9.8 us

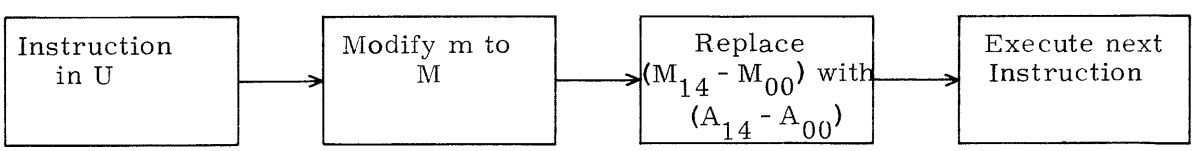
Replaces the address portion of storage location m with the contents of the designated index register. The remaining bits of the word in storage remain unchanged. If b = 0, the address portion of m is set to all 1's. Initial contents of B^b remain unchanged; initial contents of m are changed.

SAL b m Substitute Address 9.8 us

Replaces the address portion of M with the lower order 15 bits of A. Remaining bits of M are not modified and the initial contents of A are unchanged.



LIL and SIL



SAL

ENA b y Enter A 7.0 us

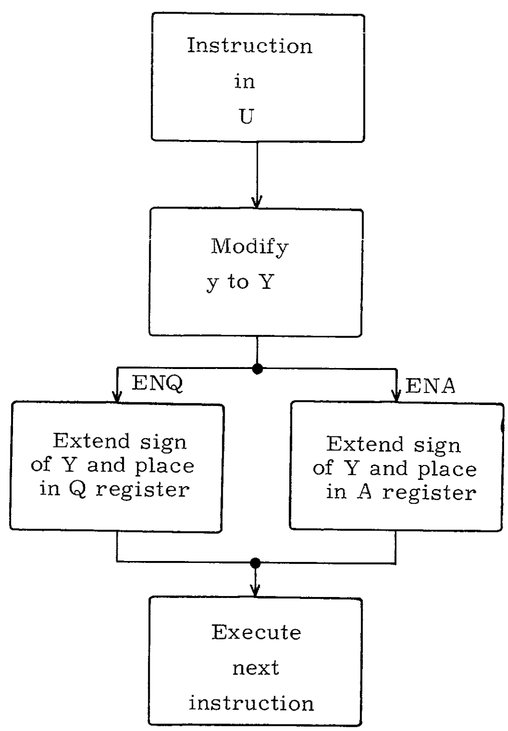
The 15-bit operand, Y, is entered into the A register and its highest order bit (sign bit) is extended in the remaining 9 bits. The largest positive 15-bit operand that can be entered into A is 37777_8 (2^{14}-1) and the "0" sign bit will be duplicated in each of the upper 9 bits. Negative zero will be formed in A if:

1) $(B^b) = 77777_8$ and $y = 77777_8$ or
2) $b = 0$ and $y = 77777_8$.

ENQ b y Enter Q 7.0 us

The 15-bit operand, Y, is entered into Q and its highest order bit (sign bit) is extended in the remaining 9 bits. The largest positive 15-bit operand that can be entered into Q is 37777_8 (2^{14}-1) and its "0" sign bit will be duplicated in each of the upper 9 bits. Negative zero will be formed in Q if:

1) $(B^b) = 77777_8$ and $y = 77777_8$ or
2) $b = 0$ and $y = 77777_8$.



ENA and ENQ

ATI b m A to Index 7.0 us

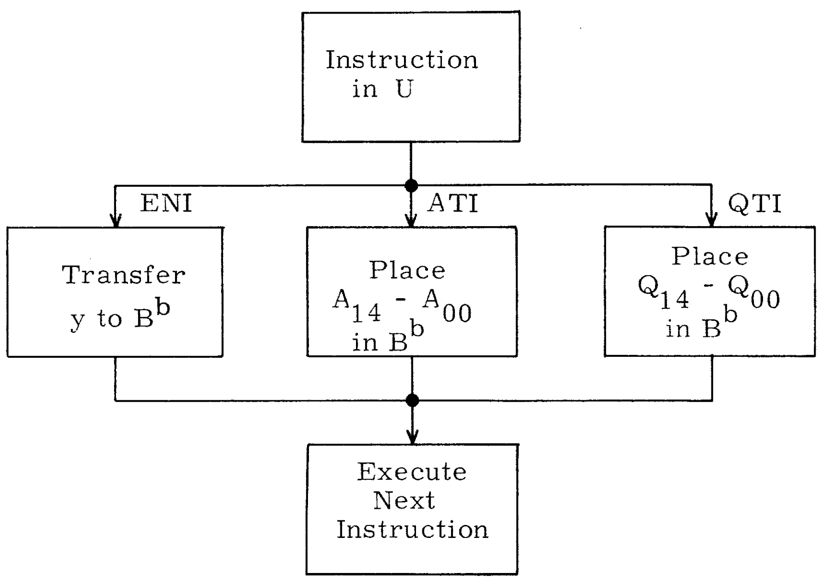
This instruction transmits the lower order 15 bits of the A register to the index register specified by the index designator. The address portion of the instruction, m, is not used in this instruction. The contents of the A register are unchanged by this instruction.

QTI b m Q to Index 7.0 us

This instruction transmits the lower order 15 bits of the Q register to the index register specified by the index designator. The address portion of the instruction, m, is not used in this instruction. The contents of the Q register are unchanged as a result of the instruction.

ENI b y Enter Index 6.2 us

Replaces (B^b) with the operand y. If b = 0, this instruction becomes a pass or do nothing instruction.



ATI, QTI, and ENI

FULL-WORD ARITHMETIC

1) In Full-Word Arithmetic instructions, a 24-bit operand is used in executing each instruction. The TAL instruction is unique in this category in that an operand, as such, is not used. The TAL instruction examines a 24-bit quantity in the A register.

2) All modes of address modification apply to these instructions except TAL.

3) One storage reference is made for each instruction except TAL unless indirect addressing is designated. In this case, at least two references are made.

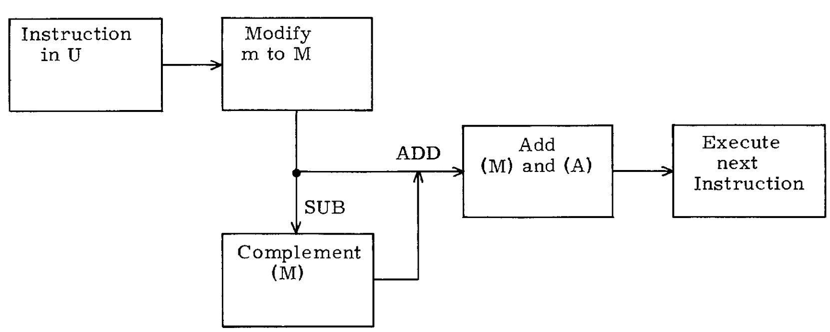
4) If the capacity of the A register $\pm (2^{23}-1)$ is exceeded during the execution of the instructions an arithmetic overflow fault is produced. When executing the DVI instruction, if the result exceeds the capacity of the A register $\pm (2^{23}-1)$ a divide fault is produced. (Refer to appendix.)

ADD b m Add 9.9 us

Adds a 24-bit operand obtained from storage location M to (A). A negative zero may be produced by this instruction if (A) and (M) are initially negative zero. The contents of storage address M remain unchanged.

SUB b m Subtract 9.9 us

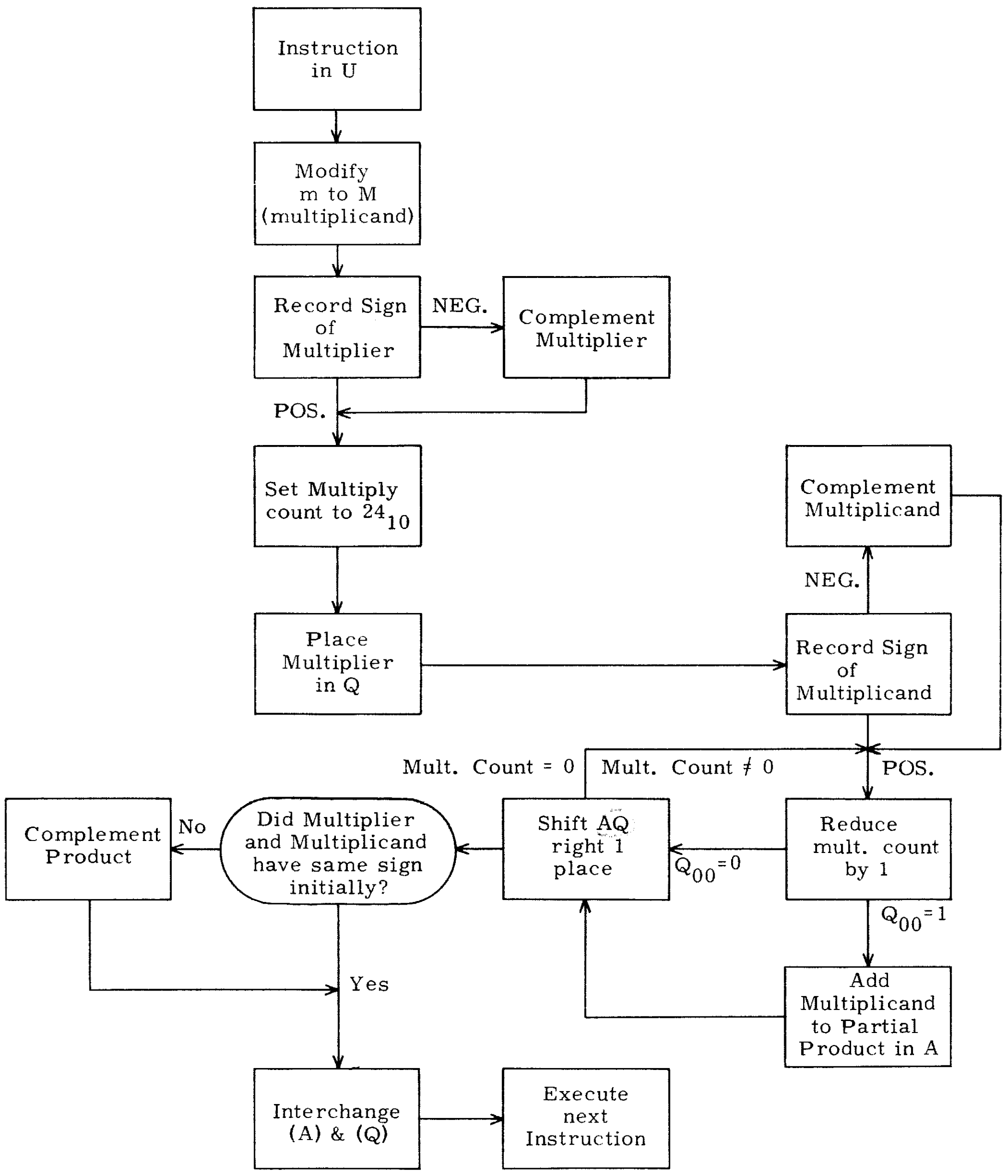
Obtains a 24-bit operand from storage location M and subtracts it from the initial contents of A. A negative zero will be produced if the initial contents of A are negative zero and that of storage location M are positive zero. The contents of address M remain unchanged.



ADD and SUB

MUI b m Multiply 20.4 + .8n*

Forms a 48-bit product from two 24-bit operands. The multiplier must be loaded into A prior to execution of the instruction. The execution address specifies the storage location of the multiplicand. The product is contained in QA as a 48-bit quantity. The operands are considered as integers and therefore the binary point is assumed to be at the lower-order (right hand) end of the A register.



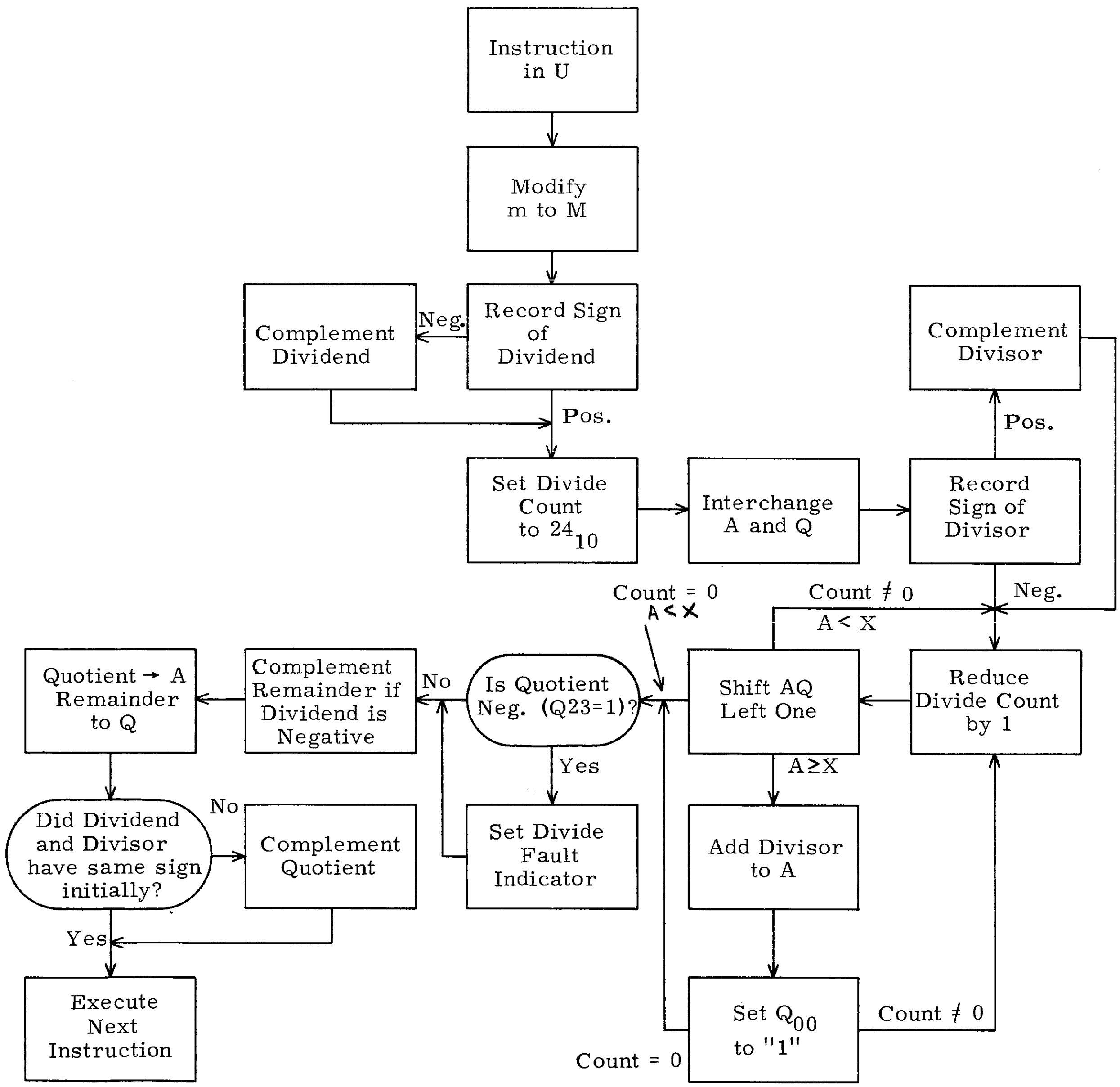
MUI

---

* n = Number of ones in multiplier

DVI b m Divide 38.0 us

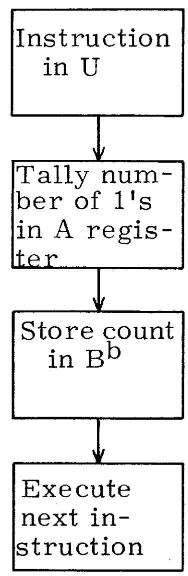
Divides a 48-bit integer dividend by a 24-bit integer divisor. The 48-bit dividend must be formed in the QA register prior to executing the instruction. If a 24-bit dividend is loaded into A, the sign of the dividend in A must be extended throughout Q. The 24-bit divisor is read from the storage location specified by the execution address. The quotient is formed in A and the remainder is left in Q at the end of the operation. Dividend and remainder have the same sign.



DVI

TAL b m Tally 14.8 us

This instruction counts the number of ones in the A register and stores the count in the index register specified by the designator b; the contents of the A register remain unchanged. The m portion of the instruction is not used.



TALLY

ADDRESS ARITHMETIC

1) In the Address Arithmetic instructions, only the lower 15-bits (or the address portion) of the operand or data word are used.

2) All modes of address modification apply to the INA and INQ instructions. In executing the INI instruction, indirect addressing may be used.

3) No storage reference is made during these instructions unless indirect addressing is designated. In this case, at least one storage reference is made.

INA b y Increase A 7.0 us

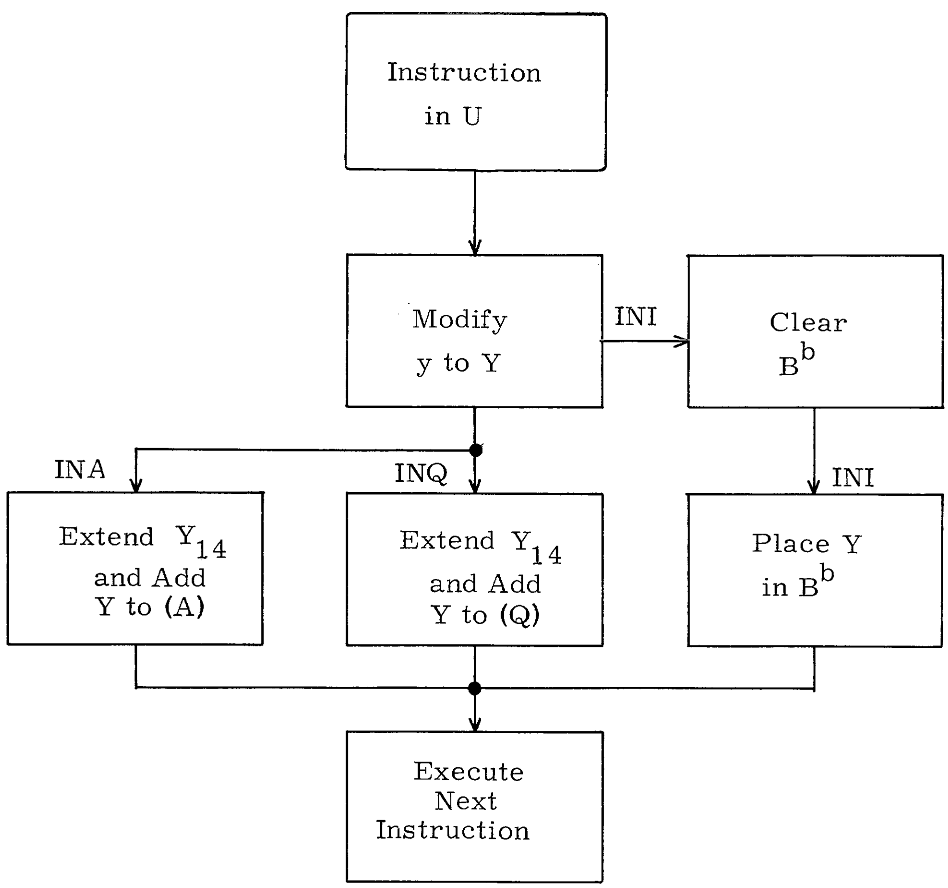
Adds Y to A. The 15-bit operand Y with its highest order bit is extended in the remaining 9 bits and added to A. Y is thus considered a 15-bit signed operand, with sign extended.

INI b y Increase Index 7.0 us

Increases (B^b) by the operand y. If the b designator is zero, this instruction becomes a pass or do nothing instruction.

INQ b y Increase Q 7.0 us

Adds Y to Q. The 15-bit operand Y, with its highest order bit extended in the remaining 9 bits, is added to Q. Y is thus considered as a 15-bit signed operand, with sign extended.



INA, INI, and INQ

LOGICAL

1) All modes of address modification apply to these instructions except CMA and CMQ.

2) The LDL, ADL, SBL and STL instructions achieve their result by forming a logical product. A logical product is a bit by bit multiplication of two binary numbers.

$$0 \times 0 = 0 \qquad 1 \times 0 = 0$$
$$0 \times 1 = 0 \qquad 1 \times 1 = 1$$

3) A logical product is used, in many cases, to select specific portions of an operand for entry into another operation. For example, if only a specific portion of an operand in storage is to be added to (A), as the operand passes through the exchange register (X, a secondary register) it is subjected to a mask comprised of a predetermined pattern of "0's" and "1's". Forming the logical product of (X) and the mask causes X to retain the original contents only in those stages which have corresponding "1's" in the mask. When only the selected bits remain in X, the instruction proceeds to conclusion.

SST b m Selective Set 9.9 us

Sets the individual bits of A to "1" where there are corresponding "1's" in the word at storage location M. "0" bits in the word at storage location M do not modify the corresponding bits in A. In a bit by bit comparison of (A) and (M) there are four possible combinations of bits.

1)	2)	3)	4)
$(A)_i = 1$	$(A)_i = 1$	$(A)_i = 0$	$(A)_i = 0$
$(M)_i = 1$	$(M)_i = 0$	$(M)_i = 1$	$(M)_i = 0$
$(A)_f = 1$	$(A)_f = 1$	$(A)_f = 1$	$(A)_f = 0$
$(M)_f = 1$	$(M)_f = 0$	$(M)_f = 1$	$(M)_f = 0$

SCM b m Selective Complement 9.9 us

Individual bits of A are complemented where there are corresponding "1's" in the word at storage location M. If the corresponding bits at M are "0's" the associated bits of A remain unchanged.

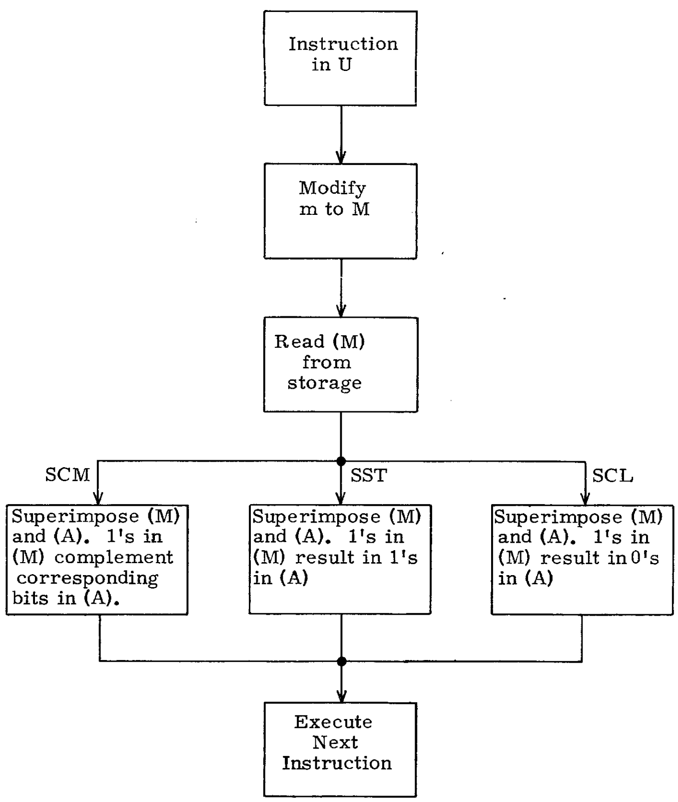
1)	2)	3)	4)
$(A)_i = 1$	$(A)_i = 1$	$(A)_i = 0$	$(A)_i = 0$
$(M)_i = 1$	$(M)_i = 0$	$(M)_i = 1$	$(M)_i = 0$
$(A)_f = 0$	$(A)_f = 1$	$(A)_f = 1$	$(A)_f = 0$
$(M)_f = 1$	$(M)_f = 0$	$(M)_f = 1$	$(M)_f = 0$

SCL b m Selective Clear 9.9 us

Clears individual bits of A where there are corresponding "1's" in the word at storage location M. If the corresponding bits at M are "0's" the associated bits of A remain unchanged.

In a bit by bit comparison of (A) and (M) there are four possible combinations of bits.

1)	2)	3)	4)
$(A)_i = 1$	$(A)_i = 1$	$(A)_i = 0$	$(A)_i = 0$
$(M)_i = 1$	$(M)_i = 0$	$(M)_i = 1$	$(M)_i = 0$
$(A)_f = 0$	$(A)_f = 1$	$(A)_f = 0$	$(A)_f = 0$
$(M)_f = 1$	$(M)_f = 0$	$(M)_f = 1$	$(M)_f = 0$



SCM, SST, and SCL

SSU b m Selective Substitute 9.9 us

Substitutes selected portions of an operand at storage address M into the A register where there are corresponding "1's" in the Q register (mask). The portions of A not masked by "1's" in Q are left unmodified.

LDL b m Load Logical 9.9 us

Loads A with the logical product of Q and the designated storage location, M.

ADL b m Add Logical 9.9 us

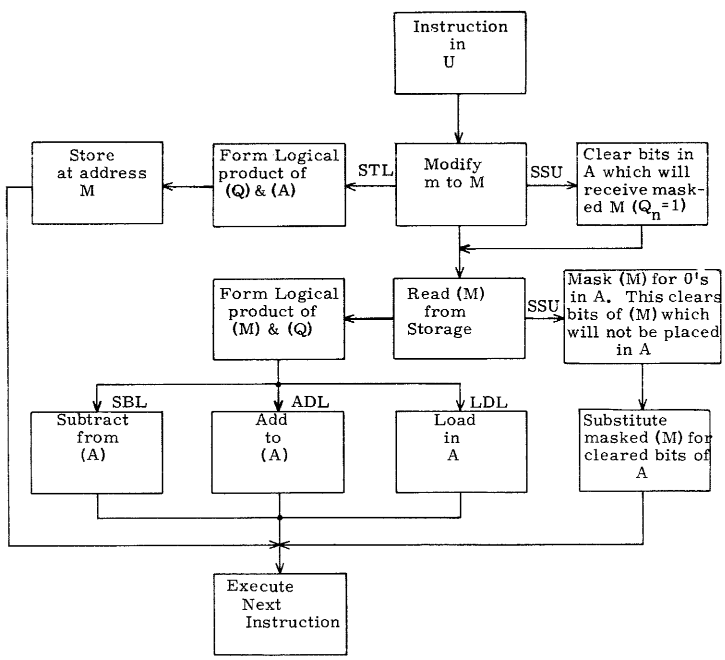
Adds to A the logical product of Q and the quantity in location M. Once the logical product is formed addition follows normal rules.

SBL b m Subtract Logical 9.9 us

Subtracts from A the logical product of the Q register and the quantity in storage location M. When the logical product is formed, the subtraction proceeds in the normal manner.

STL b m Store Logical 9.8 us

Replaces the bits in location M with the logical product of Q and A registers. Neither (A) or (Q) are modified.



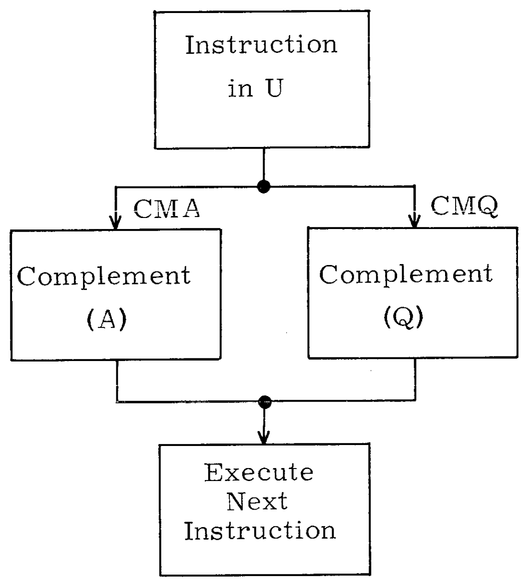
ADL, LDL, SBL, SSU, and STL

CMA 1 m Complement A* 6.2 us

This instruction complements the contents of the A register. The address portion of this instruction, m, is not used.

CMQ 2 m Complement Q* 6.2 us

This instruction complements the contents of the Q register. The address portion of this instruction, m, is not used.



CMA and CMQ

---

* If the index designator does not have the indicated value, the instruction will not be executed.

SHIFTING

1) All modes of address modification apply to all these instructions, except SCA and SCQ.

2) If the modified shift count, K, is greater than 63_{10}, a fault indicator is set. Regardless of the magnitude of count, however, the required number of shifts is executed. (K is reduced by one count for each shift executed and when K = 0, shifting stops.)

3) Shifting will be completed before an input, output or interrupt request will be processed. (See chapter three.)

ARS b k A Right Shift 6.2 + .4s*

Shifts contents of A to the right K places. The sign is extended and the lower bits are discarded. The largest practical shift count is 23_{10} since the register is now an extension of the sign bit.

QRS b k Q Right Shift 6.2 + .4s*

Shifts contents of Q to the right K places. The sign is extended and the lower bits are discarded. The largest practical shift count is 23_{10} since the register is now an extension of the sign bit.

LRS b k Long Right Shift 6.2 + .4s*

Shifts contents of AQ to the right K places as one 48-bit register. The A register is considered as the leftmost 24 bits and the Q register as the rightmost 24 bits. The sign of A is extended. The lower order bits of A replace the higher order bits of Q and the lower order bits of Q are discarded. The largest practical shift count is 47_{10} since AQ is now an extension of the sign of A.

LLS b k Long Left Shift 6.2 + .4s*

Shifts contents of AQ to the left K places, left circular, as one 48-bit register. The higher order bits of A replace the lower order bits of Q and the higher order bits of Q replace the lower order bits of A. The largest practical shift count 48_{10} returns AQ to its original state.

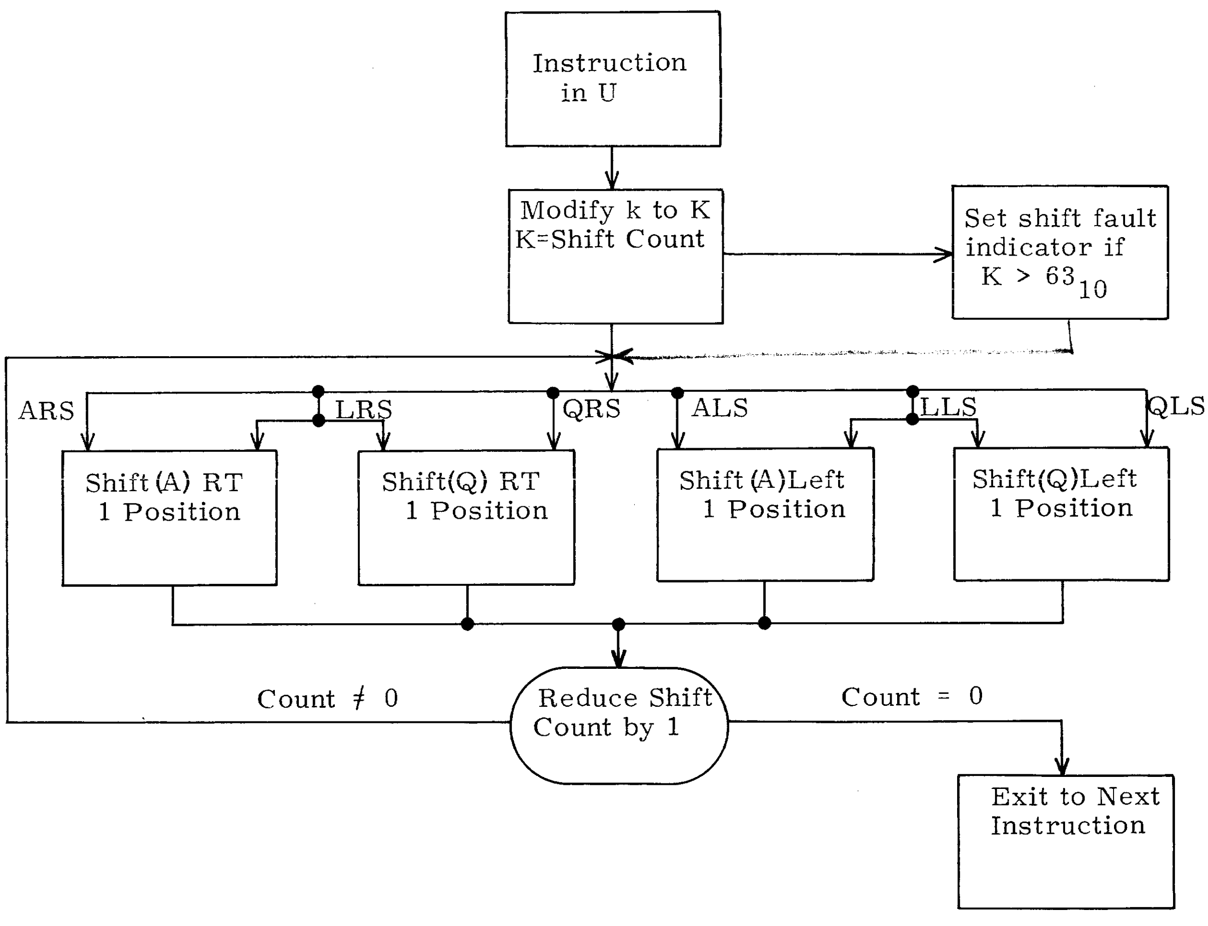
* s = Number of positions shifted

QLS b k Q Left Shift 6.2 + .4s*

Shifts contents of Q to the left K places, left circular. The higher order bits of Q replace the lower order bits. The largest practical shift count 24_{10} returns the register to its original state.

ALS b k A Left Shift 6.2 + .4s*

Shifts contents of A to the left K places, left circular. The higher order bits of A replace the lower order bits. The largest practical shift count 24_{10} returns the register to its original state.



Shift Instructions

In the SCA and SCQ instructions:

1) Address modification does not apply. Rather, the index register is used to preserve the scale factor.

2) If b = 0, scaling is executed but the scale factor is lost.

3) If b = 7, indirect addressing is used and at least one storage reference is made.

4) If the (A)i is already scaled or equal to positive or negative zero, k → B^b and scaling is not executed.

5) If the execution address is initially equal to 0, B^b is cleared.

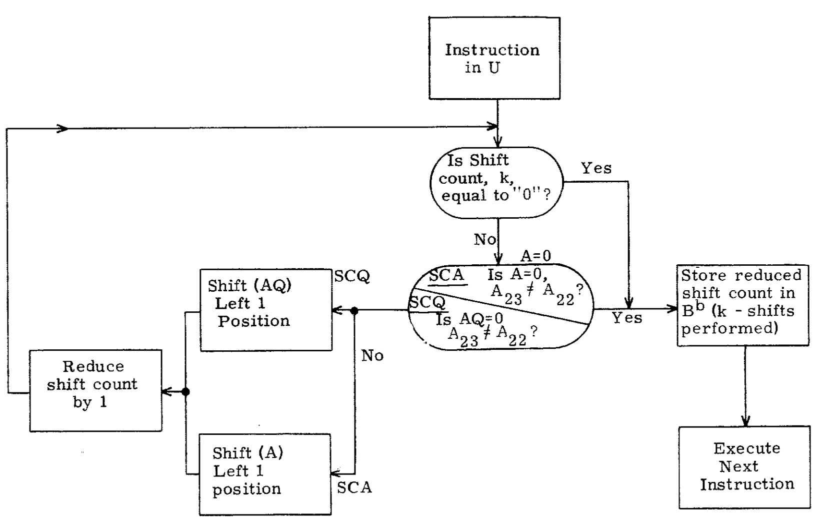
6) The shift fault indicator is not affected by these instructions.

SCA b k Scale A 3.8 + .4s*

Shifts A left circularly until the most significant digit is to the right of the sign bit or until k = 0. Shift count k is reduced by one for each shift and terminates when k = 0 or the most significant digit is to the right of the sign bit. Upon termination the count (scale factor) is entered in the designated index register.

SCQ b k Scale AQ 3.8 + .4s*

Shifts AQ left circularly until the most significant digit is to the right of the sign bit. Shift count k is reduced by one for each shift. Operation terminates when k = 0 or the most significant digit is to the right of the sign bit. Upon termination the count (scale factor) is entered in the designated index register.



SCA and SCQ

* s = Number of positions shifted

REPLACE

1) All modes of address modification apply to these instructions.

2) During the execution of the replace instructions, two storage references are made. If indirect addressing is designated, at least three references are made.

3) If the capacity of the A register $\pm(2^{23}-1)$ is exceeded during the execution of the following instructions, an arithmetic overflow fault is produced. (Refer to appendix.)

RAD b m Replace Add 15.9 us

Obtains a 24-bit operand from storage location M and adds it to the initial contents of A. The sum is left in A and is also transmitted to location M.

RSB b m Replace Subtract 15.9 us

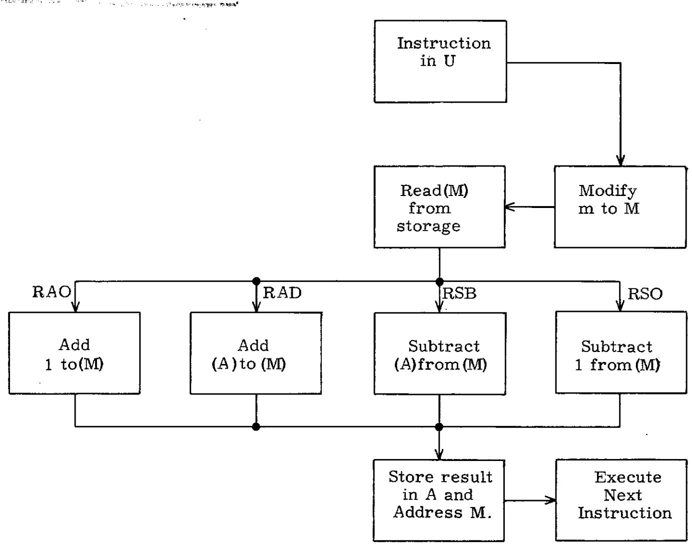
Subtracts (A) from (M) and places the result in both the A register and location M.

RAO b m Replace Add One 15.9 us

Replaces the operand in storage location M with its original value plus one. The result is also placed in A.

RSO b m Replace Subtract One 15.9 us

Replaces the operand in storage location M with its original contents minus one. The difference is also left in A.



Replace

STORAGE TEST

1) All modes of address modification may be used in these instructions.

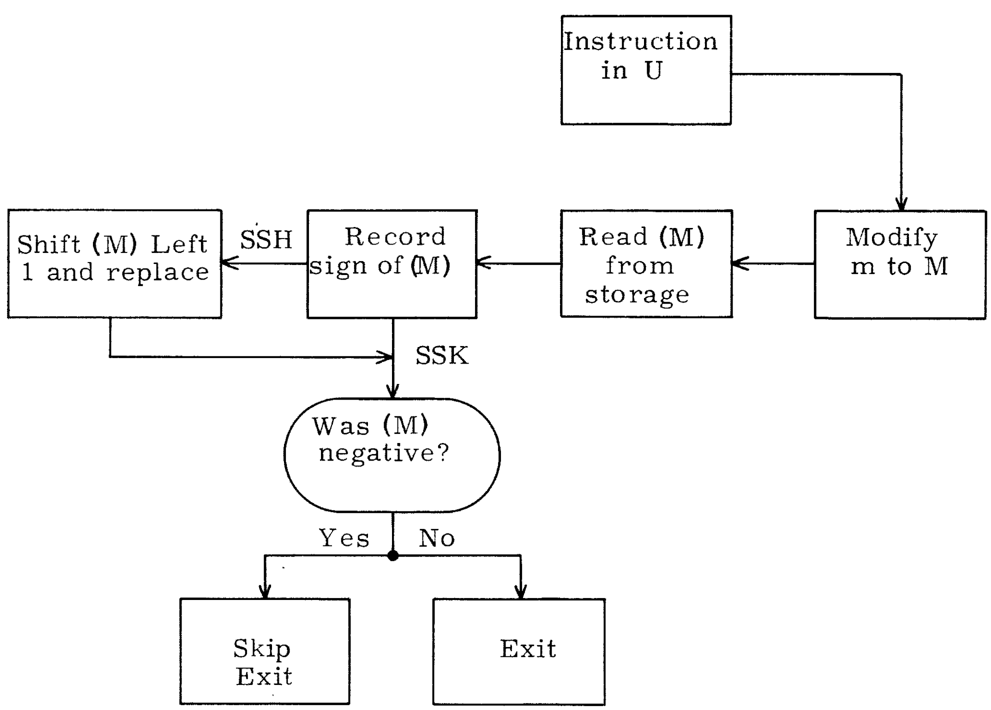
2) At least one storage reference is made unless indirect addressing is designated in which case at least two storage references are made.

SSK b m Storage Skip 9.2 us

Senses the sign bit of the operand in storage location M. If the sign is negative, a skip exit is taken. If the sign is positive, an exit is taken. The contents of the operational registers are left unmodified.

SSH b m Storage Shift 15.9 us

Senses the sign bit of the quantity in storage location M. If the sign bit is negative a skip exit is taken, and if the quantity is positive an exit is taken. In either case the quantity is shifted left circularly one bit before the exit. The contents of the operational registers are left unmodified.



SSH and SSK

SKIP

1) Indirect addressing is the only mode of address modification recognized by these instructions.

2) No storage reference is made unless indirect addressing is specified, in which case at least one reference will be made.

ISK b y Index Skip 7.0 μsec

Compares (B^b) with y. If the two quantities are equal, B^b is cleared and a skip exit is performed. If the quantities are unequal, (B^b) is increased one count in the U register and an exit is performed. Counting is performed in a two's complement subtractive counter, thus, it is possible to count through negative zero and positive zero.

SKH b y Skip High 6.4 μsec

Compares the quantity in the designated index register with the operand, y. If the quantity in the index register is greater than or equal to the quantity y, an exit is performed. If the quantity in the index register is less than the quantity y, an exit is performed. That is

if $(B^b) \geq y$ then Exit;
if $(B^b) < y$ then Skip Exit.

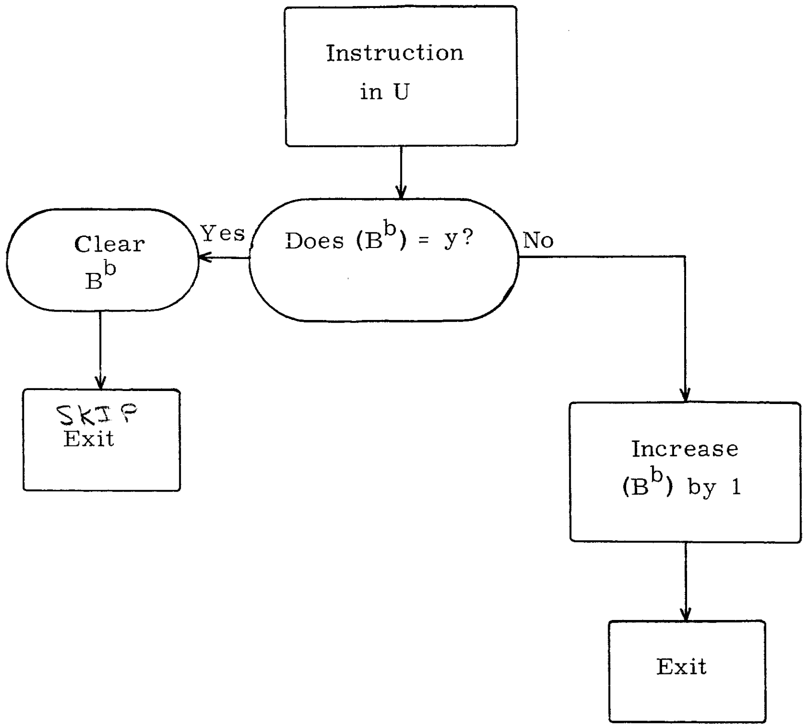
The contents of the index register are unchanged by this instruction.

SKL b y Skip Low 6.4 μsec

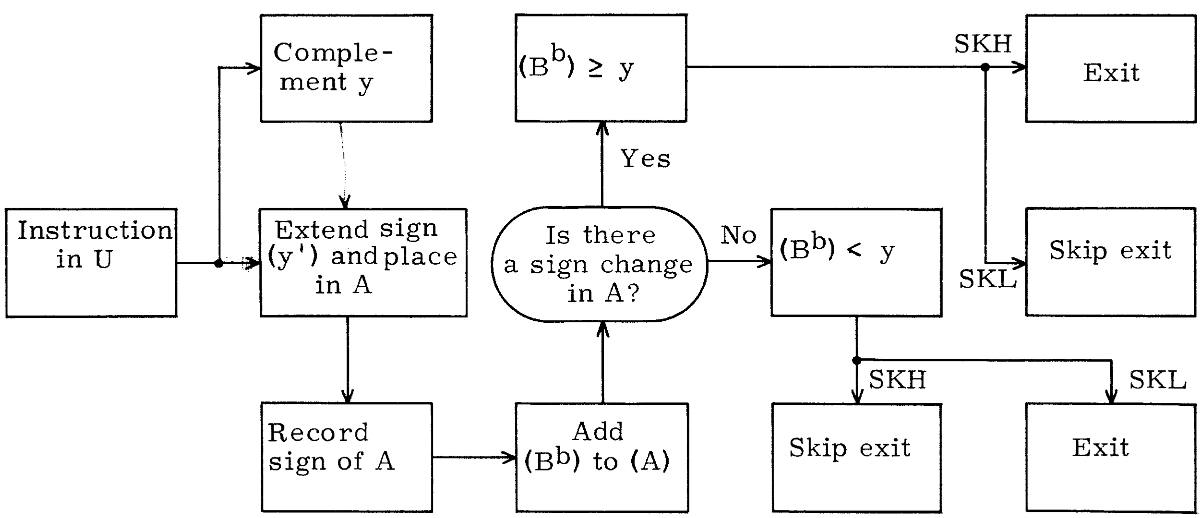
Compares the quantity in the designated index register with the operand, y. If the quantity in the index register is less than the quantity y, an exit is performed. If the quantity is greater than or equal to y, a skip exit is performed. That is,

if $(B^b) < y$ Exit,
if $(B^b) \geq y$ Skip Exit.

The contents of the index register are unchanged by this instruction.



ISK



SKH and SKL

STORAGE SEARCH

1) If b = 0 in the following instructions only the word at storage location m will be searched.

2) If b = 7, indirect addressing is used to obtain the execution address and b designator.

3) If $(B^b) = 0$, no search is made.

EQS b m Equality Search 6.9 + 5.2r*

Searches a list of operands to find one that is equal to (A). The number of items to be searched is specified by (B^b). These items are in sequential addresses beginning at location m. The search begins with the last address, $m + (B^b) - 1$. (B^b) is reduced one count for each word that is searched until an operand is found that equals (A) or until (B^b) equals zero. If the search is terminated by finding an operand that equals (A), a skip exit is made. The address of the operand satisfying this condition is given by the sum of m and the final contents of B^b. If no operand is found that equals (A), an exit is taken. Positive zero and minus zero are recognized as the same quantity.

THS b m Threshold Search 6.9 + 6.0r*

Searches a list of operands to find one that is greater than (A). The number of items to be searched is specified by (B^b). These items are located in sequential addresses beginning at location m. The search begins with the last address, $m + B^b - 1$. The content of the index register is reduced by one for each operand examined. The search continues until an operand is reached that is greater than (A) or until (B^b) is reduced to zero. If the search is terminated by finding an operand greater than the value in A, a skip exit is performed. The address of the operand satisfying the condition is given by the sum of m and the final contents of B^b. If no operand in the list is greater than the value in A, an exit is performed. In the comparison made here positive zero is considered as greater than minus zero.

---

* r = Number of additional words searched

MEQ b m Masked Equality 6.9 + 6.0r*

Searches a list of operands to find one such that the logical product of (Q) and (M) is equal to (A). This instruction, except for the mask in Q, operates in the same manner as an equality search.

MTH b m Masked Threshold 6.9 + 6.0r*

Searches a list of operands to find one such that the logical product of (Q) and (M) is greater than (A). Except for the mask in Q, this instruction operates in the same manner as the threshold search.

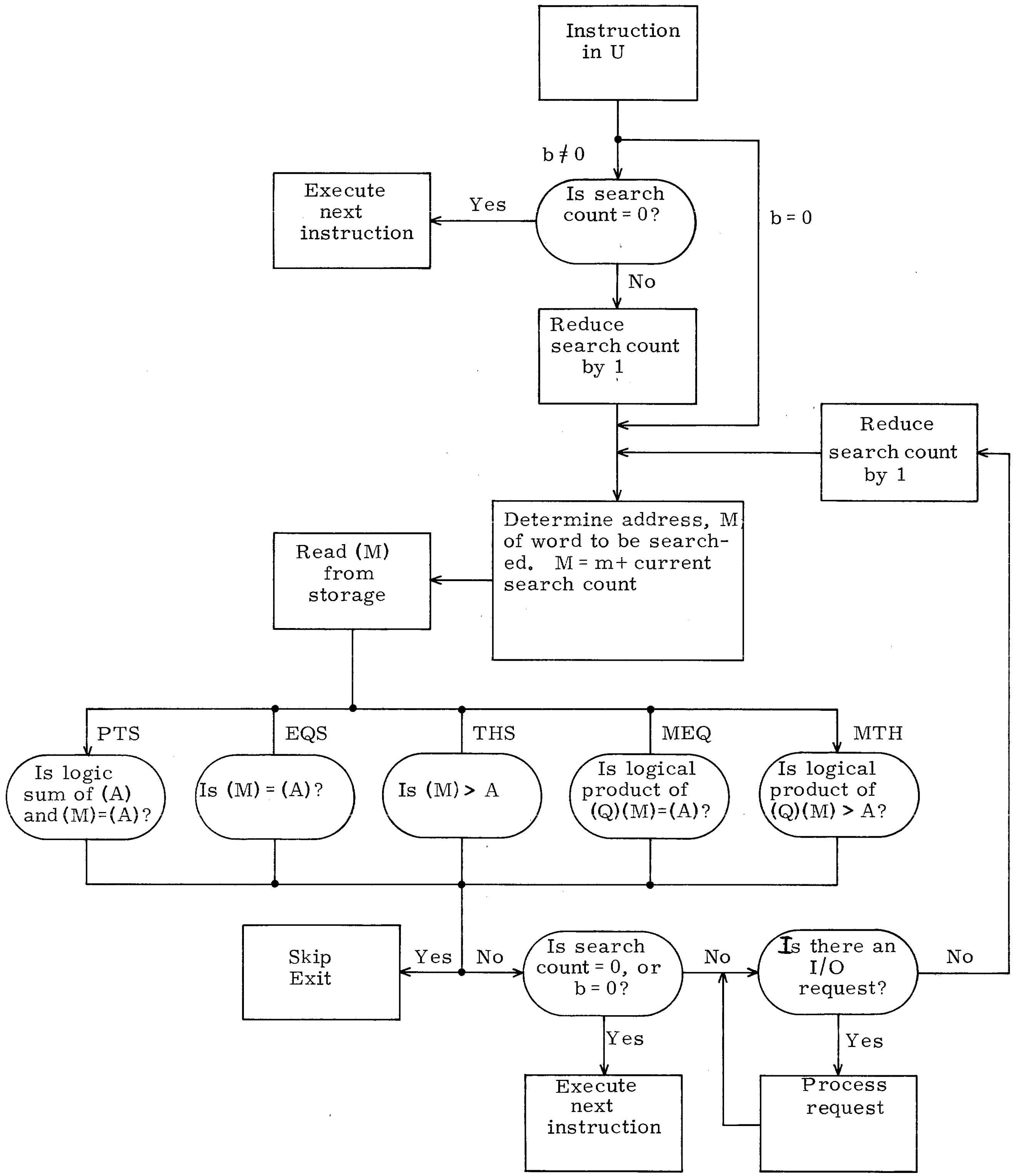
PTS b m Pattern Search 6.9 + 6.0r*

Searches a list of operands to find one such that the "inclusive or" of (A) and (M) is equal to (A). These items are in sequential addresses beginning at the location specified by m. The search begins with the last address, $m + B^b - 1$. (B^b) is reduced one count for each word that is searched until an operand is found that satisfies the criteria or until (B^b) equals zero. If the search is terminated by finding an operand which satisfies the condition, a skip exit is performed. The address of the operand satisfying the inclusive or condition is given by the sum of m and the final contents of B^b. If no operand in the list is found such that the inclusive or of (A) and (M) = (A), an exit is performed.

On a bit-by-bit basis, three combinations of the initial values of A and M meet the search criterion; one combination does not. The combinations are:

$A_i = 0$	$A_i = 1$	$A_i = 1$	$A_i = 0$
$M_i = 0$	$M_i = 0$	$M_i = 1$	$M_i = 1$
Search criterion met			Search criterion not met

* r = Number of additional words searched


Instruction in U
b ≠ 0
b = 0
Is search count = 0?
Yes
Execute next instruction
No
Reduce search count by 1
Reduce search count by 1
Determine address, M, of word to be searched. M = m+ current search count
Read (M) from storage
PTS
EQS
THS
MEQ
MTH
Is logic sum of (A) and (M)=(A)?
Is (M) = (A)?
Is (M) > A
Is logical product of (Q)(M)=(A)?
Is logical product of (Q)(M) > A?
Skip Exit
Yes
No
Is search count = 0, or b = 0?
No
Is there an I/O request?
No
Yes
Execute next instruction
Yes
Process request

Search

JUMPS AND STOPS

NORMAL

1) Address modification is used in the UJP instruction.

2) If indirect addressing is designated for the IJP or UJP instruction, at least one storage reference is required.

A jump instruction causes a current program sequence to terminate and initiates a new sequence at a different location in storage. The Program Address Register, P, provides the continuity between program steps and always contains the storage location of the current program step.

When a jump instruction occurs, P is cleared and a new address is entered. In most jump instructions, the execution address, m, specifies the beginning address of the new program sequence. The word at address m is read from storage, placed in U and the first instruction of the new sequence is executed.

Some of the jump instructions are conditional upon a register containing a specific value or upon the position of an operator's jump or stop key on the console. If the criterion is satisfied, the jump is made to location m. If it is not satisfied, the program proceeds in its regular sequence to the next instruction.

AJP j m A Jump 7.7 us

Jumps to m if the conditions of the A register specified by the jump designator j exist. If not, the next instruction is executed.

j = 0 jump if (A) = 0
j = 1 jump if (A) $\neq$ 0
j = 2 jump if (A) = +
j = 3 jump if (A) = -

When (A) is negative zero the interpretation is:

j = 0 The jump is executed because, in this case, negative zero is recognized as positive zero.
j = 1 The jump is not executed when (A) = + 0 or - 0.
j = 2 The jump is not executed because the sign bit is a "1".
j = 3 The jump is executed because the sign bit is a "1".

QJP j m Q Jump 7.7 us

Jumps to m if the condition of the Q register specified by the jump designator, j, exists. If not, the next instruction is executed.

j = 0 jump if (Q) = 0

j = 1 jump if (Q) $\neq$ 0

j = 2 jump if (Q) = +

j = 3 jump if (Q) = -

When (Q) is negative zero the AJP interpretation applies.

SLJ j m Selective Jump 7.9 us

Jumps to m if the condition of the jump keys specified by j exists. If not, the next instruction is executed.

j = 0 Jump unconditionally. (Does not reference jump key setting.)

j = 1 Jump if jump key 1 is set.

j = 2 Jump if jump key 2 is set.

j = 3 Jump if jump key 3 is set.

SLS j m Selective Stop 7.9 us

Stops at present step in the sequence if the condition of the stop key specified by j exists. If the stop condition exists, the stop is executed, and the jump is executed unconditionally when the ~~Run~~ START-Step Key is moved to the run or step position. If the stop condition is not satisfied, the jump is executed unconditionally.

j = 0 Stop unconditionally. (Does not reference stop key setting.)

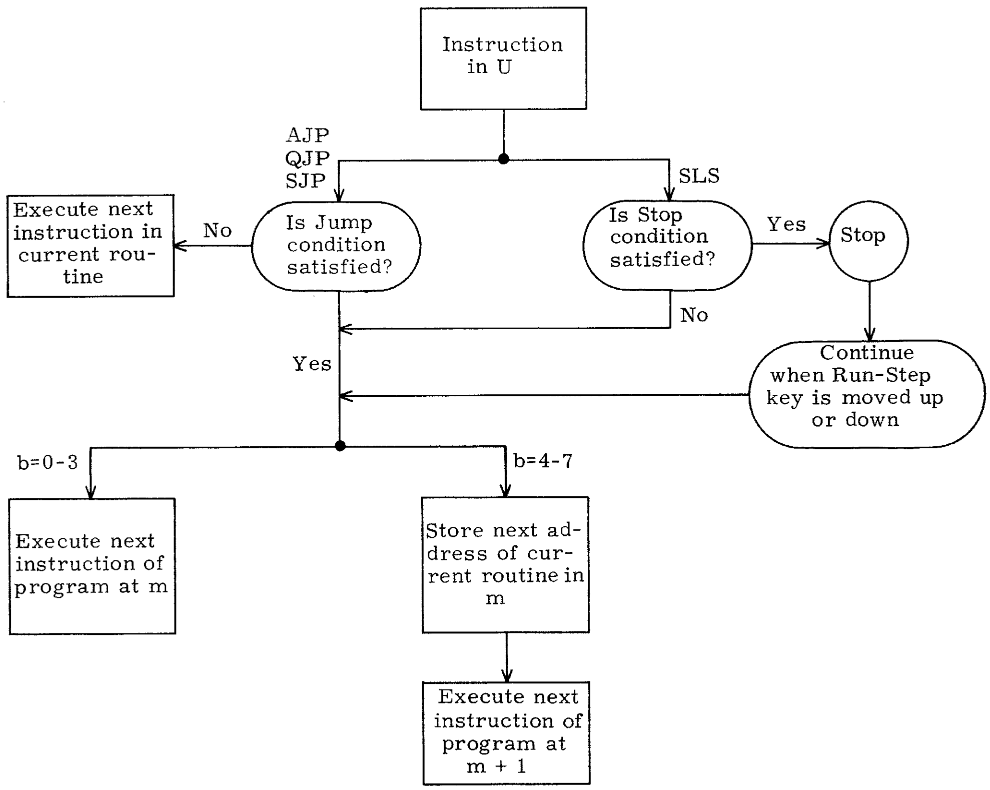
j = 1 Stop if stop key 1 is set.

j = 2 Stop if stop key 2 is set.

j = 3 Stop if stop key 3 is set.

UJP b m Unconditional Jump 4.8 us

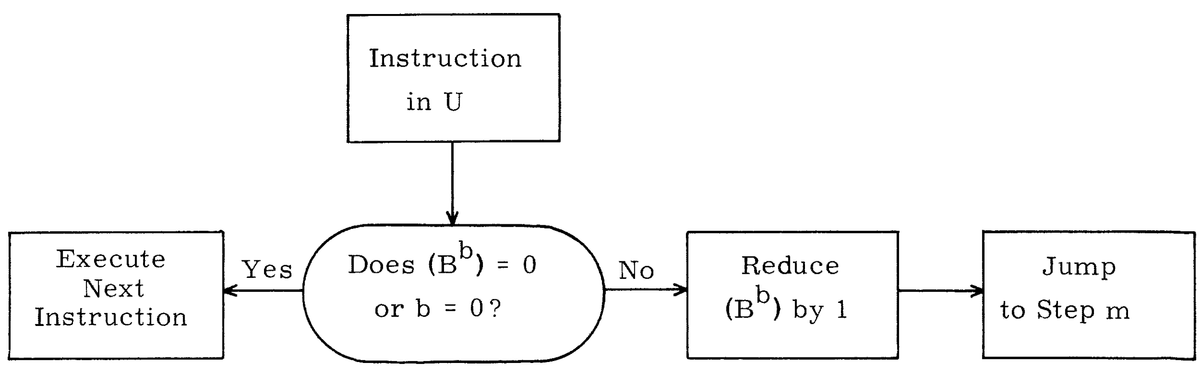
Jumps unconditionally to M. $M = m + (B^b)$. If b = 0, the jump is to m. If b = 7, indirect addressing will be used.



AJP, QJP, SJP, and SLS

IJP b m Index Jump 7.0 us

Examines (B^b). If this quantity is not zero, the quantity is reduced one count and a jump is executed to program step m. The counting operation is performed in a two's complement subtractive counter but negative zero is not generated because IJP terminates at positive zero. (See appendix.) If (B^b) is zero, the present program sequence is continued.



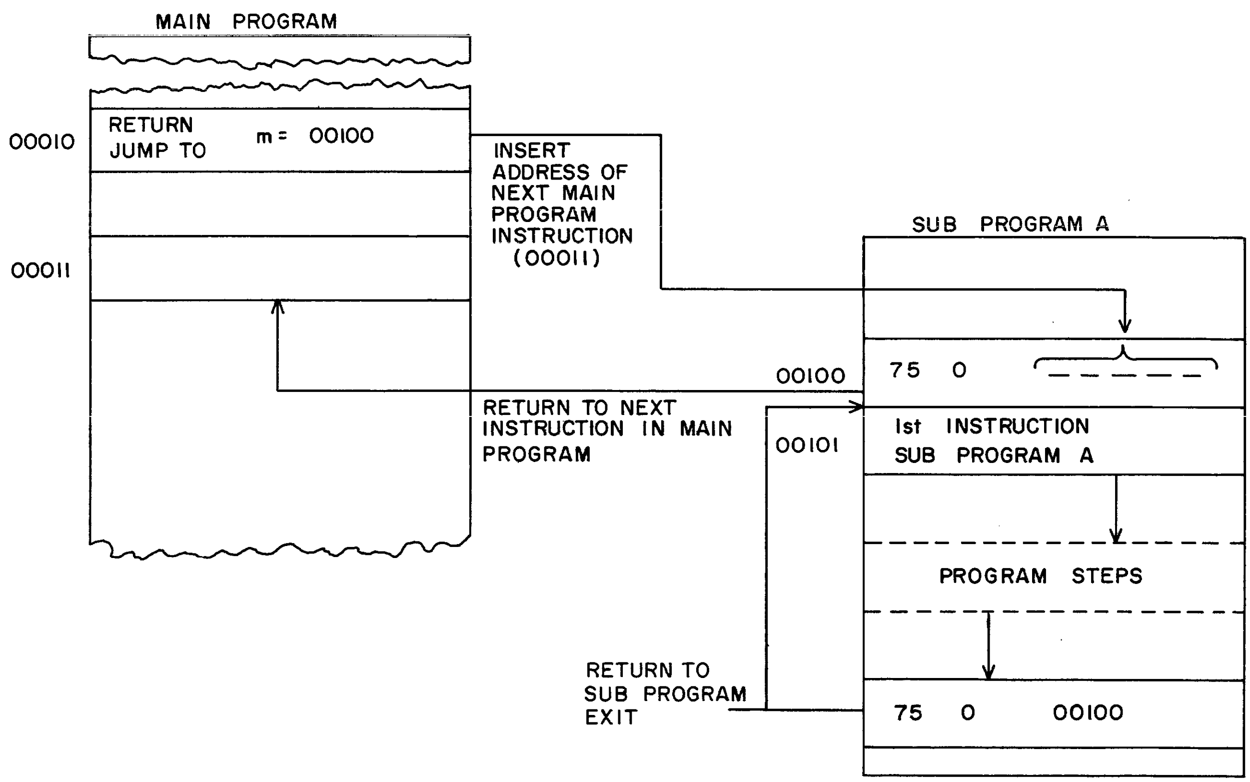
IJP

RETURN

1) Address modification applies only to URJ and XEC.

2) One storage reference is made.

A return jump begins a new program sequence at the address specified by m + 1, e.g., 00101. At the same time, the execution address of the first instruction in the new program sequence, 00100, is replaced with the address of the next program step in the main program (00011). The first instruction in a new program sequence is usually an unconditional jump which allows a return to the main program after completing the subprogram sequence.



Return Jump

AJP j m A Jump 7.7 us

Executes a return jump to storage location m if the condition of the A register specified by j exists. If not, the next instruction is executed.

- j = 4 Return jump if (A) = 0
- j = 5 Return jump if (A) ≠ 0
- j = 6 Return jump if (A) = +
- j = 7 Return jump if (A) = -

When (A) is negative zero the interpretation is:

- j = 4 The return jump is executed because, in this case, negative zero is recognized as positive zero.
- j = 5 The return jump is not executed when (A) = + 0 or - 0.
- j = 6 The return jump is not executed because the sign bit is a "1".
- j = 7 The return jump is executed because the sign bit is a "1".

QJP j m Q Jump 7.7 us

Executes a return jump to storage location m if the condition of the Q register specified by j exists. If not, the next instruction is executed.

- j = 4 Return jump if (Q) = 0
- j = 5 Return jump if (Q) ≠ 0
- j = 6 Return jump if (Q) = +
- j = 7 Return jump if (Q) = -

Note: If (Q) = negative zero, refer to the AJP instruction.

SLJ j m Selective Jump 7.9 us

Executes a return jump to storage location m on condition j where condition j represents the setting of the jump keys. If the condition is not satisfied, the next instruction is executed.

- j = 4 Return jump unconditionally. (Does not reference jump keys.)
- j = 5 Return jump if jump key 1 is set.
- j = 6 Return jump if jump key 2 is set.
- j = 7 Return jump if jump key 3 is set.

Note: The set position of a jump key is in the up position.

SLS j m Selective Stop 7.9 us

Stops on condition j and executes a return jump if the Run-Step key is moved in the run or step position. If the stop condition is not satisfied, the stop is not executed and the return jump is executed unconditionally.

- j = 4 Stop unconditionally. The return jump is executed when the Run-Step key is moved in either position.
- j = 5 Stop if stop key 1 is set. (Return jump.)
- j = 6 Stop if stop key 2 is set. (Return jump.)
- j = 7 Stop if stop key 3 is set. (Return jump.)

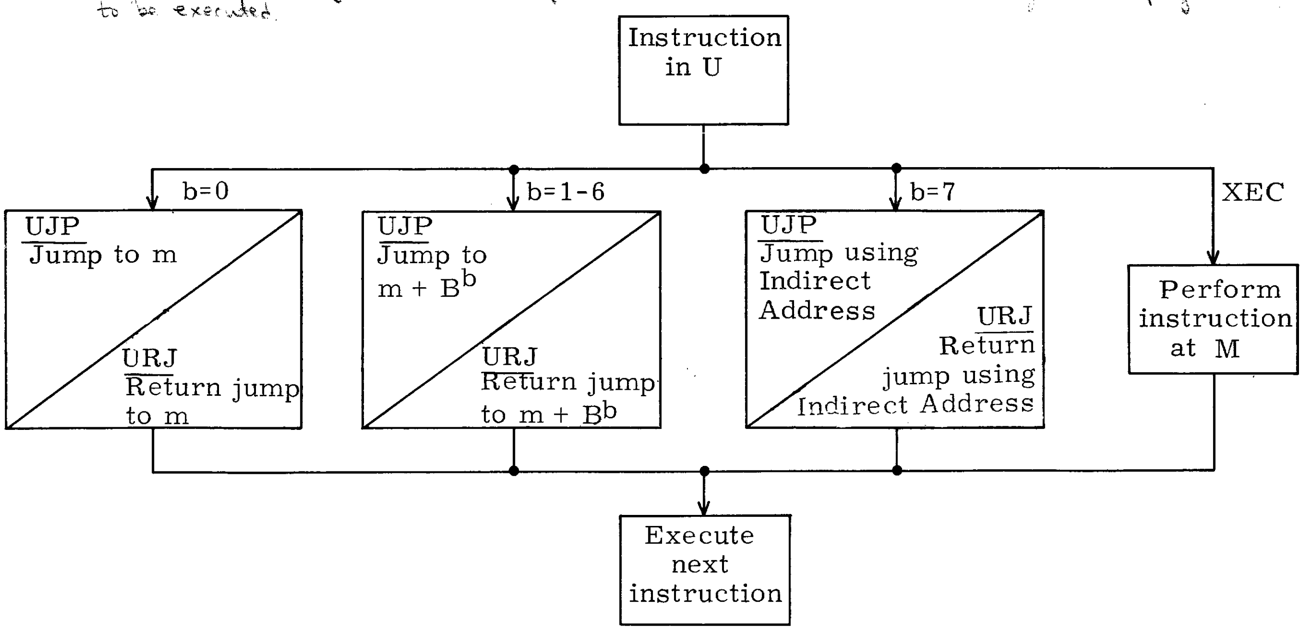
URJ b m Unconditional Return Jump 4.8 us

Return jumps unconditionally to M. $M = m + (B^b)$. If b = 0, the return jump is to m. If b = 7, indirect addressing will be used.

XEC b m Execute 8.2 + x us for inst. at M

Jumps to M. After executing M, the main sequence is continued unless M performs a jump. In this case, a new program is initiated at the location specified by the jump and a return is not effected to the main sequence. This instruction is effectively an indirect instruction, or a subroutine of a single instruction.

Program Address Register, P, is unchanged and instruction at M is brought into program control register to be executed.



UJP, URJ, and XEC

3

Input-Output

METHODS OF DATA EXCHANGE

The computer communicates with external equipment via six buffer channels. These channels provide for the parallel transmission of binary words to and from computer storage.

BUFFER CHANNELS

The six independent buffer channels are grouped in three pairs:

Input:	Channel 1	Output:	Channel 2
	Channel 3		Channel 4
	Channel 5		Channel 6

Every external equipment is connected to one of these pairs. All six buffer channels may be concurrently transmitting information. However, only one external equipment can use any one buffer channel at any given instant. The rate of data flow on the buffer channels is usually dependent on the operating speed of the external equipment. 8 μsecs/24 bit word - 3Mc bit rate

19.2 μsec/word - any channel alone
26.2 " " - interrupts only
38.4 " " interrupts + program steps

INITIATION AND CONTROL OF DATA EXCHANGE

The actual buffer (exchange) operation, although program initiated, proceeds under controls that are independent of the program. Information is buffered asynchronously with the execution of the program, thus, storage is permitted to serve its primary function of working with the arithmetic and control sections during the time a buffer is in progress.

Buffer Control Word

Information is buffered in blocks one word at a time. The current and terminal storage addresses of the block are located in adjacent storage locations called buffer control words. Each of the six buffer channels has two assigned storage addresses which hold the buffer control word for the current and terminal addresses of the block.

Special Address

00000	Initial start
00002	Channel 1 control (terminal address)
00003	Channel 1 control (current address)
00004	Channel 2 control (terminal address)
00005	Channel 2 control (current address)
00006	Channel 3 control (terminal address)
00007	Channel 3 control (current address)
00010	Channel 4 control (terminal address)
00011	Channel 4 control (current address)
00012	Channel 5 control (terminal address)
00013	Channel 5 control (current address)
00014	Channel 6 control (terminal address)
00015	Channel 6 control (current address)

Buffer Control Words

	24 (23)	14	00 BIT POSITION
00002	NOT USED	TERMINAL ADDRESS (LWA + 1)	
00003	NOT USED	STARTING ADDRESS (CURRENT ADDRESS)	

The terminal address is one greater than the last address to be used in the buffer. Prior to initiating a buffer operation, the terminating address must be entered into the even-addressed control word. The starting address is automatically entered into the word when the buffer is initiated by an EXF instruction.

In 1604 mode (48-bit transfer), current and terminal addresses must both be even (or odd), since incrementing of word register goes up by 2 rather than by 1.

The execution address of the EXF instruction is used to designate the buffer starting address in storage. This address is automatically recorded in the lower 15 bits of the appropriate odd-number special storage location. The terminal address (plus one) of the block of data must have been previously recorded, by the program, in the lower 15-bit portion of the appropriate even-numbered special storage location prior to the execution of the external function instruction.

External Function (EXF) Instructions

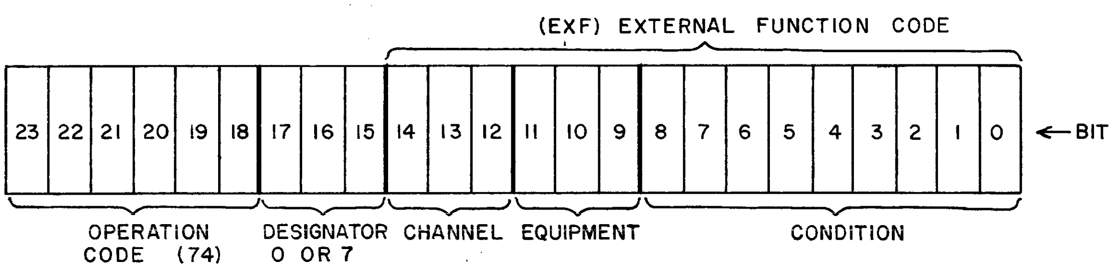
The EXF instructions initiate a buffer, sense for specified conditions, and select operations and equipment. EXF codes are listed at the end of the chapter.

There are three kinds of external instructions:

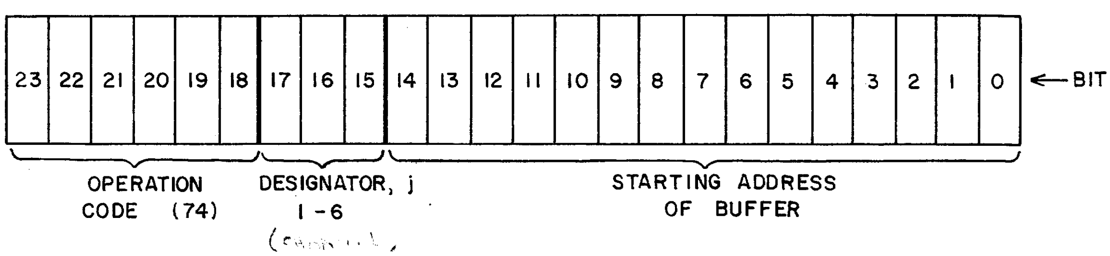
Select	74 0 XXXXX
Sense	74 7 XXXXX
Activate	74 j XXXXX

The composition of an external function instruction is shown below.

Select and Sense



Activate



The 74 0 (EXF Select) instructions select the external equipment which is to communicate with the computer and/or its mode of operation. The select instructions do not activate the buffer but, rather, establish initial operating conditions within the designated equipment so that information will be properly processed when the buffer is activated.

Three basic modes of communication can be selected by an EXF instruction: 1604 (48-bit), 160 (12-bit), and 24-bit. Selecting 1604 or 160 mode permits communication with any peripheral equipment.

No distinction is made between 24 and 48-bit transmissions on the control lines. However, when 24-bit mode is selected, a single storage word is read; two storage words are required for each 48-bit transmission.

When the 160 mode is selected, the computer appears as a 160 to peripheral equipments. In this mode a status request is accomplished by simply selecting the equipment and using the desired EXF code. The Status response will be placed on the input lines. A one word input buffer must be executed to obtain the response. Only the lower 12 bits are significant during this mode.

If an input or output operation is to follow, the computer program places the terminal address in the appropriate control address. An activate instruction follows and buffering proceeds in the normal manner except only the lower 12 bits of data are transmitted to or from computer storage.

The EXF 7 instructions sense the condition of an external equipment or the internal conditions (faults) of the computer. If the specified condition exists, a skip exit is performed; if not present, an exit is performed (Example 1).

Example 1	(00110)	74 7 00011
	(00111)	75 0 00007
	(00112)	53 1 00005

In this example, the translation of program step 00110 is skip on channel 1 inactive. If channel 1 is inactive the next instruction to be executed would be program step 00112, i.e., 53 1 00005. If channel 1 were active, however, program step 00111 would be the next instruction executed. In either case, the sensing of a condition in no way alters the condition.

The EXF j instructions activate buffer channel j where j equals 1-6. The execution address of the instruction, y, must designate the starting address of the buffer region in storage. These instructions are the only instructions which can initiate a buffer (information flow) between the computer and an external equipment.

The following steps should be completed prior to initiating a buffer operation.

1) Sense for: (a) channel inactive and (b) equipment ready.
2) Select the external equipment and its mode of operation.
3) Substitute the terminal address into the buffer control word.

An equipment is ready if there is no motion, that is, a transmission is not taking place.

A buffer channel is active while data is being buffered. A buffer channel is inactive if the previous input-output operation has been completed.

The buffer must be terminated (incremented starting address = terminating address) in order to inactivate the channel. This can be accomplished by setting the starting address equal to the terminating address and activating the channel (74 j instruction). This makes the channel inactive but no additional information is transmitted.

The only way to clear a synchronous or asynchronous print select for the printer is to activate and then deactivate the print buffer. This results in a blank line if paper advance has previously been selected. (Otherwise "double-ending" occurs on the

Buffer Scanner

Data exchange on each of the separate buffer channels is initiated by the program but proceeds under a control that is independent of the main program. This control is the buffer scanner. Order = channel 1, 3, 2, 6, interrupt request line, channel 5, channel 4

The buffer scanner prevents one external equipment from dominating all others by sequentially monitoring the six buffer channels and the interrupt line to sense when any of these require action. When one of the positions demands action, the scanner stops. One word is processed and the scanner is released to resume its monitoring. If none of the positions demand action, a full scan is made in 1.6 usec. If, however, all seven positions demand processing at the same time, the maximum time lapse for processing successive words on the same channel will be 175 usec.

Figure 3-1 shows the relationship between executing instructions and recognizing action requests (input-output requests) from the buffer channels or interrupt line. During the time an input-output request is being processed, the scanner is released and resumes monitoring. If the scanner halts (indicating a channel demands action) this request will be processed before the main program is allowed to continue. NOTE

when start-step switch is in "step" position, buffer scanner stops on the interrupt request line.

Clearing 924 External Equipment

When external equipment is used with the 924 in 924 or 1604 mode this equipment can be cleared in one of three ways. They are as follows:

1. Selecting another piece of 1604 type equipment.
2. Programming an EXF channel clear, i.e., 740 C0000 where C = channel number.
3. By an external master clear.

The last piece of 1604 type equipment that had been selected must be cleared before changing from 1604 mode to 160 mode on a given channel. An EXF channel clear can be used to accomplish this. The channel clear instruction can only be used in 924 or 1604 mode and must be programmed before selecting 160 mode.

It is not necessary to clear a selected piece of 160 equipment before going from 160 mode to 1604 or 924 mode.

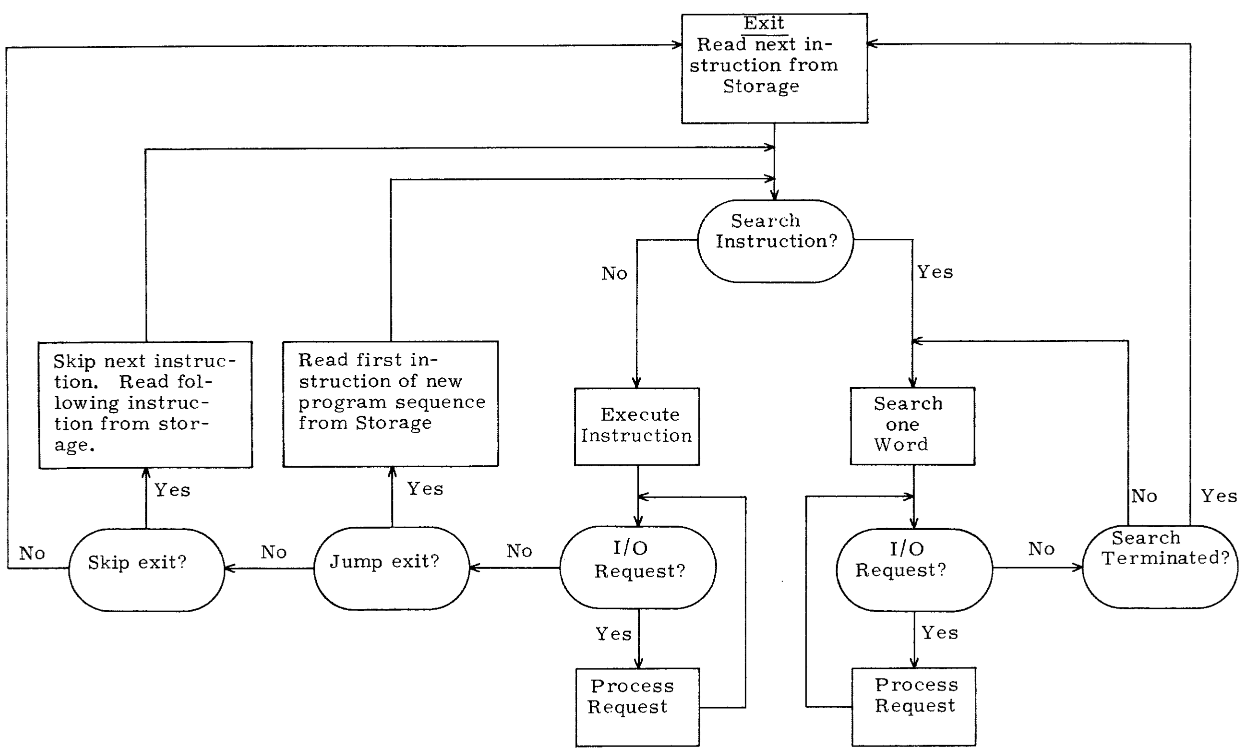


Figure 3-1. Instruction Searching

Interrupt

In each piece of external equipment as well as in parts of the internal computer control, certain conditions may arise which make it necessary that the main program be notified of their presence. The signal which notifies the computer of these conditions is called an interrupt and is program controlled. If an interrupt is desired when a particular condition arises, an external function select code (74.0 xxxxx) must have been previously programmed to permit an interrupt on that condition. Unless such selection is made, no interrupt is produced when the condition arises.

When an interrupt occurs, the main program is halted and a previously programmed routine of instructions (interrupt routine) is performed which must determine the cause of the interruption and take appropriate action. After completing these operations the interrupt routine must return to the main program. The main program resumes at the exact point from which the entrance to the interrupt routine was made.

If more than one source of interrupt has occurred simultaneously, the interrupt processing routine determines the order in which they are processed (not necessarily the order of their appearance).

The 16, 30 and 40 external interrupt signals* are processed on a priority basis. When the interrupt 30 line is activated the computer executes a return jump to address 30 (that is, it stores P at address 30 and jumps to address 31). When the interrupt 40 line is activated the 924 does a return jump to address 40. An interrupt from any other source will be designated as an interrupt 16 and the computer will do a return jump to address 16.

When an interrupt is recognized by the computer, the interrupt lockout is set to prevent recognition of other interrupts. Clearing of the interrupt lockout does not depend upon which interrupt the computer has recognized (16, 30, or 40). The only condition that clears the interrupt lockout is when the computer does a RNI (Read Next Instruction) from address 16, 30 or 40.

The interrupt lines are sampled and priority established when the scanner is locked in the interrupt position. If more than one interrupt line is activated, the lower numbered interrupt (i.e. 16) will be recognized first. If this line of higher priority becomes active again before the previous interrupt routine has been completed the computer will again recognize the line of higher priority.

Interrupts are not "stacked". If successive interrupts from the same piece of equipment occur before the first interrupt is processed (or cleared), there is no indication of how many may have occurred.

* The 30 and 40 interrupt signals originate from 160-A equipment.

Interrupt Routine

The interrupt is processed by performing a return jump to address 00016, 30, or 40. These are special addresses allocated for use as the entrance point to the interrupt routine and for the return from this routine to the main program.

In general, the interrupt routine (table 3-1) checks for all possible interrupt conditions by means of sense (74.7) instructions. After determining which selected condition caused the interrupt a jump is made to that portion of the routine which processes the interrupt.

TABLE 3-1. TYPICAL INTERRUPT ROUTINE

00016	75 0 ______ (next address of main program)	Exit to Main Program
00017	75 0 int00	Entrance to Interrupt Routine
int00	74 7 00131	Sense Overflow
int01	75 0 ovf00	
int02	74 7 11201	Sense Reader End of Tape
int03	75 0 ret00	
int04	74 7 ______	
int05	75 0 ______	
ovf00	______	Process Overflow
ovf01	______	
	______	
	______	
	74 0 00070	Clear Arithmetic Faults
ovf___	75 0 00016	Exit to Main Program
ret00	______	Process Reader End of Tape
ret01	______	
	______	
	______	
	74 0 10000	Remove Interrupt Signal
	74 0 11200	Remove Interrupt Selection
ret___	75 0 00016	Exit to Main Program

When an interrupt signal occurs, it remains on until turned off by a select (74.0) instruction from the computer. Sensing an interrupt does not remove the interrupt signal nor does it remove the interrupt selection. It is therefore mandatory that the routine turn off the interrupt signal so that upon exiting from the routine the signal will not immediately cause another interrupt. NOTE

In the Process Reader Not Ready (retxx) portion of the table, the interrupt signal is removed by a 74 0 10000 (clear all channel one selections). The interrupt selection is removed by a 74 0 11200 (No Interrupt on Reader Not Ready). Note that, if desired, the interrupt selection need not be removed as evidenced by the ovfxx section of the routine.

Real Time Clock

Incorporated into the interrupt circuitry of the 924 computer is a real time clock source which generates a pulse once every 100 milliseconds. This pulse will produce an interrupt only if a 74 0 01000 (select real time interrupt) instruction is programmed.
Once in the interrupt routine to process this interrupt, it must either be reset by 74001000 or deselected (74001001) so that an exit can be made through location 16

CONSOLE INPUT-OUTPUT EQUIPMENT

Two input-output devices mounted on the console are standard equipment with the 924 computer. A Teletype high speed punch and a high speed tape reader provide for the processing of perforated paper tape. The console input-output units communicate with the central computer via buffer channel 1 (input) and 2 (output). Other input-output units may share these channels but console input-output units cannot use any of the other channels.

Data may be transmitted between the console equipments and the computer in either the character or the assembly mode. In the character mode a 7 or 8-bit character is buffered one character at a time. This character occupies the lowest bit positions of a 24-bit word, the upper bits are "0's".

In the assembly mode the 24-bit word (consisting of four 6-bit characters) is buffered. During an input buffer in the assembly mode four successive characters assembled into a 24-bit word are sent to the computer. The first character occupies the upper 6 bits of the word; the last character occupies the lower-order 6 bits. For an output buffer in the assembly mode a 24-bit word from the computer is disassembled into four characters beginning with character 3, which is the upper 6 bits of the word.

PAPER TAPE READER

The paper tape reader enters information stored on punched paper tape into the computer. The reader, which is always connected to channel 1, operates at a maximum rate of 350 characters (frames) per second; the time interval between successive characters from the reader is 3.3 milliseconds. Selecting the Reader, 74 0 112xx, automatically selects 24-bit mode. Any other mode selection for this channel after the 74 0 112xx Select, but prior to the Activate (74 1 xxxxx) is illegal. (A 160 or 1604 mode Select clears the reader.)

Manual controls for the reader are on the control panel of the console. When the Reader Mode switch is set to Assembly, the tape is positioned at the first frame of the first word (load point); when it is set to Character, the tape moves ahead one frame.

Information is stored on paper tape in seven or eight levels. A frame, which is across the width of the tape, can store seven (or eight) bits (figure 3-2). The sprocket or feed holes between levels 3 and 4 generate signals to time and control the reading of the tape.

In the assembly mode, level 7 is used as a control rather than an information level. The first of the four characters in a word is indicated by a hole in the control level.

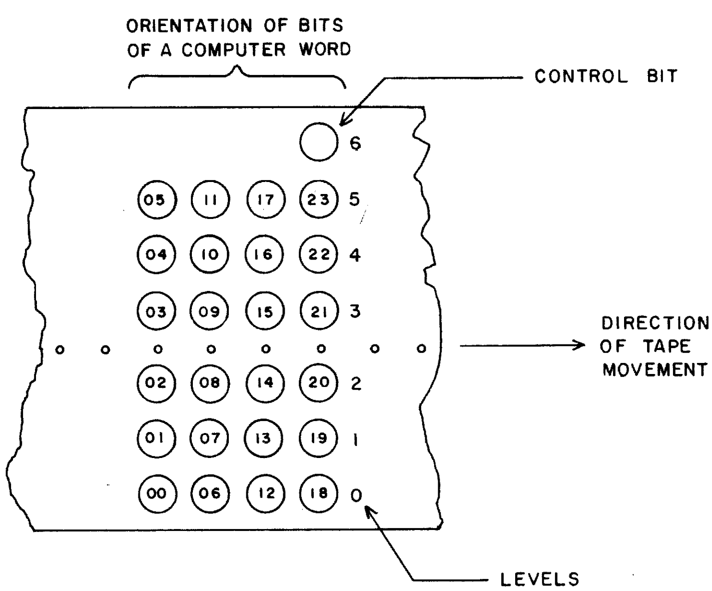


Figure 3-2. Seven Level Punched Paper Tape

Reader tape motion stops on any one of three conditions:

1) When buffer operation terminates (assembly or character mode).
2) When the load point in assembly mode is reached.
3) If a 7th level punch in the assembly mode is absent 4 characters from the last 7th level punch indicating the end of information or faulty tape.

The end-of-tape indicator is set on any of three conditions:

1) Absence of a 7th level every 4th character in the assembly mode.
2) On a computer Master Clear.
3) By a 74 0 1121X instruction. This instruction is used to indicate the end of information in the character mode.

After reading all information on the tape in the assembly mode, tape motion stops and the end-of-tape indicator is set because the 7th level control bit is missing. In the character mode, however, a 74 0 1121X instruction must be programmed to set the end-of-tape indicator. The state of the end-of-tape indicator, regardless of the mode of operation, may be used to determine if all information on the paper tape has been read.

Reader Load Mode

The reader has an auto-load feature which enables the loading of paper tape with a minimum of effort. The following steps are necessary to load in paper tape by this method.

1) External master clear the computer. ("clear" switch up)
2) Master clear the computer. ("clear" switch down)
3) Set Mode switch to Sweep (down).
4) Turn reader on.
5) Position tape in reader.
6) Put reader Mode switch momentarily in Assembly or Character mode.
7) Put initial address of buffer in the P register.
8) Put Start/Step switch in Start* position.

---

* One address (Assembly mode) or one character will be loaded in if the Start/Step switch is put in Step instead of Start. The U register will display this address or character.

The reader will stop upon absence of a 7th level in Assembly mode and will stop upon absence of tape in Character mode.

The computer must be master cleared (external and internal) after using Reader Load mode. Data will be stored at consecutive addresses starting from the initial address that was put into the P register.

PAPER TAPE PUNCH

The punch which prepares paper tape output is always connected to buffer channel 2. Nominal operating rate is 110 characters per second. In character mode, 7-bit characters are punched; in assembly mode, 6-bit characters are punched, with a 7th level every four characters.

On the punch, the feedout lever provides for punching out leader. A micro-switch is mounted near the roll of paper tape that supplies the punch. When the supply is low, the switch opens and provides an out-of-tape indication which may be sensed by a 74 ∇ 21200 instruction.

INTERNAL EXF SELECT INSTRUCTIONS

74 0 000C0	Interrupt on Channel C Inactive	Selects interrupt when channel C is inactive. C = 1 - 6
000C1	Remove Selection Above	Interrupt on channel C inactive selection removed
00070	Clear Arithmetic Faults	Removes arithmetic fault indication and turns off arithmetic fault background lights on console
00100	Interrupt on Arithmetic Faults	Selects interrupt on occurrence of any arithmetic faults
00101	Remove Selection Above	Interrupt on arithmetic faults selection removed
01000	Select Real-Time Interrupt	Produces an interrupt pulse every 100 milliseconds
01001	Remove Selection Above	Stops process of producing interrupt pulses by removing real time interrupt selection

BUFFER MODE INSTRUCTIONS

74 0 04010 - Select 24-bit mode for channel 1 and 2
04011 - Select 160 (12-bit) mode for channel 1 and 2
04012 - Select 1604 (48-bit) mode for channel 1 and 2
04020 - Select 24-bit mode for channel 3 and 4
04021 - Select 160 (12-bit) mode for channel 3 and 4
04022 - Select 1604 (48-bit) mode for channel 3 and 4
04030 - Select 24-bit mode for channel 1 and 2, 3 and 4
04031 - Select 160 (12-bit) mode for channel 1 and 2, 3 and 4
04032 - Select 1604 (48-bit) mode for channel 1 and 2, 3 and 4
04040 - Select 24-bit mode for channel 5 and 6
04041 - Select 160 (12-bit) mode for channel 5 and 6
04042 - Select 1604 (48-bit) mode for channel 5 and 6
04050 - Select 24-bit mode for channel 1 and 2, 5 and 6
04051 - Select 160 (12-bit) mode for channel 1 and 2, 5 and 6
04052 - Select 1604 (48-bit) mode for channel 1 and 2, 5 and 6
04060 - Select 24-bit mode for channel 3 and 4, 5 and 6
04061 - Select 160 (12-bit) mode for channel 3 and 4, 5 and 6
04062 - Select 1604 (48-bit) mode for channel 3 and 4, 5 and 6
04070 - Select 24-bit mode for channel 1 and 2, 3 and 4, 5 and 6
04071 - Select 160 bit mode for channel 1 and 2, 3 and 4, 5 and 6
04072 - Select 1604 bit mode for channel 1 and 2, 3 and 4, 5 and 6

CHANNEL CLEAR INSTRUCTIONS

74 0 C0000 Clear all channel C selections
Clears all previous equipment selections made on channel C.

INTERNAL EXF SENSE INSTRUCTIONS

74 7 000C0 Skip on Channel C Active

Skip Exit if channel C is active

000C1 Skip on Channel C Inactive

Skip Exit if channel C is inactive

ARITHMETIC FAULT INSTRUCTIONS

Refer to appendix. II (page 2)

74 7 00110 Skip on Divide Fault

00111 Skip on No Divide Fault

00120 Skip on Shift Fault

00121 Skip on No Shift Fault

00130 Skip on Overflow Fault

00131 Skip on No Overflow Fault

CONSOLE EXF SELECT INSTRUCTIONS

PAPER TAPE READER

74 0 11200 Reader and No Interrupt on End-of-Tape

Selects reader.* Reader will not move until buffer is activated.

The interrupt on End-of-Tape selection cleared.

---

*End-of-tape indicator is cleared when reader mode of operation is manually selected; Reader Ready light turns on.

11210 Set End-of-Tape Indicator

Sets end-of-tape indicator*

Clears the Interrupt on End-of-Tape Selection

Primarily used in controlling the reader in character mode

11220 Reader and Interrupt on End-of-Tape

Selects reader

Selects interrupt on end-of-tape **

An external master clear or a 74 0 11200 (P. T. Reader and no interrupt on end-of-tape) clears interrupt on end-of-tape

PAPER TAPE PUNCH

74 0 21200 Punch Assembly Mode

Selects punch

Turns on punch motor

Selects assembly mode

21210 Punch Character Mode

Selects punch

Turns on punch motor

Selects character mode

21240 Turn Punch Motor Off

Turns punch motor off

---

* Reader Ready light turns off and paper tape stops moving when end-of-tape indicator is set.

** The lack of a Mod 4 and a 7th level in assembly mode or a 74 0 11210 (set end-of-tape indicator) will set end-of-tape indicator.

CONSOLE EXF SENSE INSTRUCTIONS

PAPER TAPE READER

74 7 11200 Skip on Reader, End-of-Tape
Skip exit if end-of-tape indicator is set

11201 Skip on Reader, No End-of-Tape
Skip exit if end-of-tape indicator is not set

11210 Skip on Reader, Assembly Mode
Skip exit if the assembly mode of operation selected

11211 Skip on Reader, Character Mode
Skip exit if character mode of operation selected

PAPER TAPE PUNCH

21200 Skip on Punch Out-of-Tape
Skip exit if punch out-of-tape condition exists

21201 Skip on Punch, Not Out-of-Tape
Skip exit if punch out-of-tape condition does not exist

Magtape Autoload

0	74	0	04022	Select 48-bit mode
1	32	0	00007	
2	74	3	00000	
3	74	7	32000	Delay until ready to read
4	32	0	00003	
5	76	0	00000	Lower portion gets current address
6	00	0	10000	Terminal address for channel 3
7	74	0	32011	Select read tape 1 binary
10	74	0	32005	Rewind selected tape
11	74	7	32000	Sense ready to read
12	32	0	00011	Delay until ready to read
13	32	0	00002	
14	76	0	00000	
15	00	0	03332	
16	32	0	45001	
17	32	0	00017	

4

Operation

DESCRIPTION OF INDICATORS AND CONTROL SWITCHES

All main computer controls and indicators are on the console. Functional significance of console background lights is listed in table 4-1; computer controls are described in table 4-2.

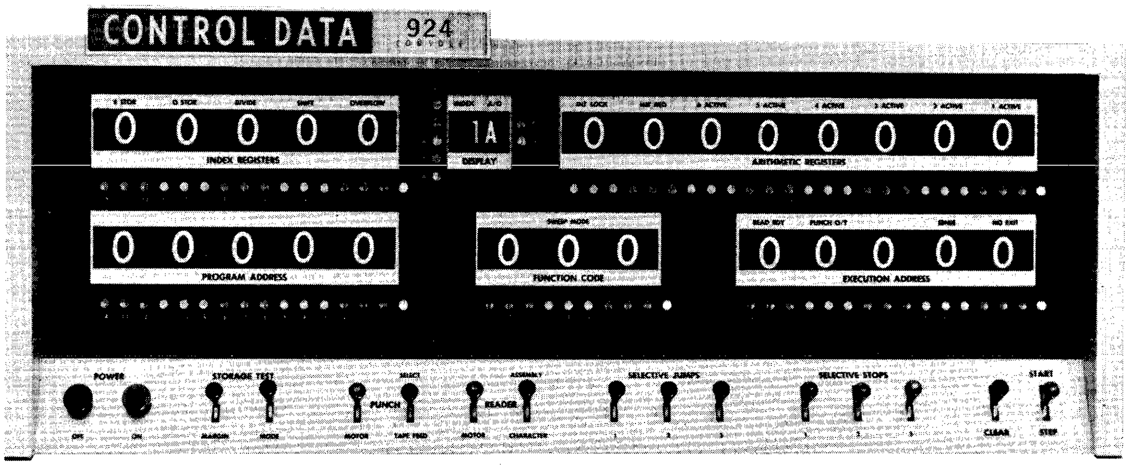


Figure 4-1. Center Panel of Console

The indicators are lamp modules, each of which displays a single octal digit. The lamps, in response to signals from the computer, display the contents of the operational registers in octal form only when the computer is stopped; the display is blank when the computer is running. Each indicator has three push buttons which are numbered in the powers of two, from right to left, starting with zero. Pressing a push button forces that particular stage of the register to the SET state. Each group of three buttons represents an octal digit. Different shades of blue are used in adjacent octal groups; within an octal group the three buttons are of the same shade.

At the right end of each register is a CLEAR push button (white). This button will clear the individual FFs within that register. SET and CLEAR push buttons should be used only when the computer is stopped; otherwise errors may result.

Conditions which stop the computer are listed below. When these conditions exist register contents may be altered by setting or clearing.

1) Illegal function codes 00, 77, and 52.3-52.6
2) Selective Stops (instruction 76)
3) Pressing START-STEP switch
4) Pressing CLEAR switch (internal master clear)

At some of the modules there are colored background lights which indicate certain internal conditions (figure 4-2, table 4-1). A light is identified by the register in which it is located and its position in the register. For example, A/Q-4 is fourth from the left in the A/Q register indicator panel. In general, red lights signify faults and blue lights signify special operating conditions. The background lights may be illuminated when the computer is running as well as when it is stopped.

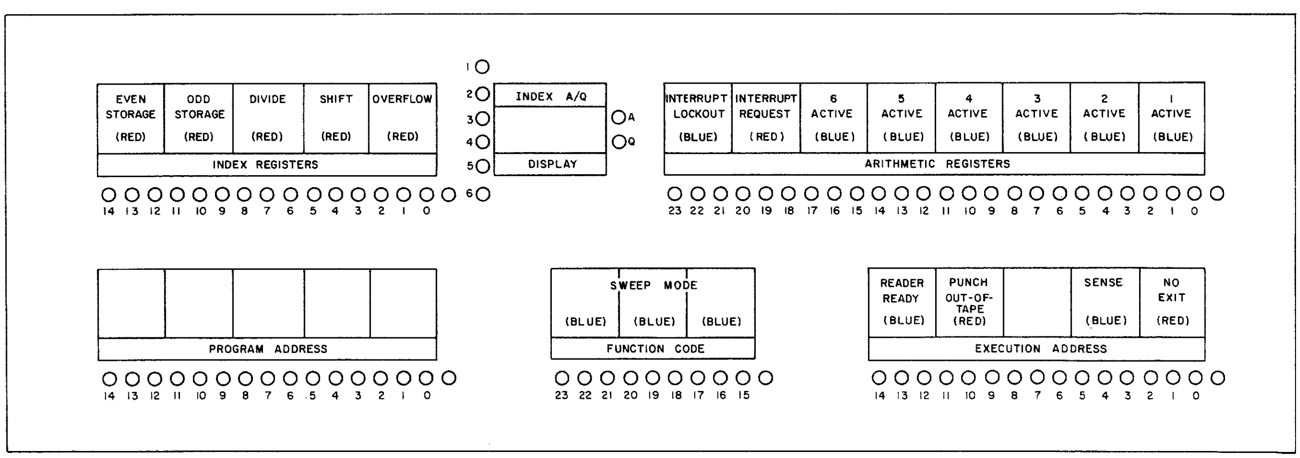


Figure 4-2. Console Display

TABLE 4-1. CONDITIONS INDICATED BY CONSOLE BACKGROUND LIGHTS

Light	Condition
A/Q-1 (blue)	Interrupt Lockout - Computer is in interrupt routine.
A/Q-2 (red)	Interrupt Request - Interrupt request signal is being received from interrupt circuit.
A/Q-3 (blue)	Channel 6 Active - Channel 6 is in use for output buffer.
A/Q-4 (blue)	Channel 5 Active - Channel 5 is in use for input buffer.
A/Q-5 (blue)	Channel 4 Active - Channel 4 is in use for output buffer.
A/Q-6 (blue)	Channel 3 Active - Channel 3 is in use for input buffer.
A/Q-7 (blue)	Channel 2 Active - Channel 2 is in use for output buffer.
A/Q-8 (blue)	Channel 1 Active - Channel 1 is in use for input buffer.
A/Q Select "A" "Q"	A Register Selected; Q Register Selected — Indicates selected register whose contents are being displayed in the appropriate indicator panel. Quantities can now be manually inserted into selected registers.
Index Register Select "1.2.3. etc."	Index Register Selected — Indicates selected register whose contents are being displayed in the appropriate indicator panel. Quantities can now be manually inserted into selected registers.
EA-1 (blue)	Reader Ready - (1) Paper tape is at load point, ready for an input buffer; or (2) paper tape input buffer is in progress.
EA-2 (red)	Punch Out of Tape - Punch tape reel is nearly empty.
IR-2 (red)	Odd Storage Fault - Fault in sequence chain of odd storage unit is inoperative until master cleared.
IR-1 (red)	Even Storage Fault - Fault in sequence chain of even storage unit; storage unit is inoperative until master cleared.
IR-3 (red)	Divide Fault - Improper divide instruction executed.
IR-4 (red)	Shift Fault - Shift instruction executed with shift count greater than 63 (decimal).
IR-5 (red)	Overflow Fault - Required sum or difference exceeds capacity of A register
EA-4 (blue)	Sense - Computer is sensing for various internal or external conditions. Lit when function code = 747 xxxx
EA-5 (red)	No Exit - Restart operation. If unable to proceed, master clear and restart program. If condition persists, notify maintenance. Stays lit when an instruction is executed (i.e., MEQ, THS, etc) takes longer than 10 microseconds.
FUNCTION CODE (blue) (3 lights)	Computer in sweep mode of operation.

TABLE 4-2. MAIN COMPUTER CONTROLS

Control		Function
POWER push button	ON - blue	Applies d-c and a-c power to computer by energizing contactor in primary power lines of motor-generator.
	OFF - red	Removes d-c and a-c power from computer by de-energizing contactor in primary power lines of motor-generator.
STORAGE TEST Lever switches lock in up, down and neutral positions	MARGIN	Varies the bias applied to storage sense amplifiers. Used for maintenance purposes only; should be in neutral position at all other times.
	MODE	UP: an instruction is executed repeatedly in either the Step or Start mode. DOWN: contents of consecutive storage locations may be manually examined by depressing Step. Consecutive words are displayed in function code and execution address but are not executed.
SELECTIVE JUMPS Three lever switches lock in upper positions, momentary in down positions.		Provide manual conditions for instruction 75, normal ~~jumps~~, b = 1, 2 ~~or~~ 3, return jumps, b = 5, 6 or 7 (Same function in all positions)
SELECTIVE STOPS Three lever switches lock in upper position, momentary in down positions.		Provide manual conditions for stopping the computer on instruction 76, b = 1, 2, 3, 5, 6 or 7 (Same function in all positions)
CLEAR Lever switch, momentary in up and down positions.		*DOWN master clears the computer, clears all operational registers and most control FFs. UP master clears external equipment, causing most of the registers and control FFs of the external equipment to be cleared. Also clears input and output channels. Selects paper tape reader.

TABLE 4-2. (CONT'D.)

Control	Function
START-STEP Lever switch, momentary in up and down positions.	START (up) selects high-speed mode in which a program of instructions and auxiliary operations proceeds until completed or stopped. STEP (down) selects Step mode. Each time switch is pressed a single instruction is executed and computer stops (all buffer requests are completed before operation stops). Step selection overrides any previous selection of Start. Buffer scanner stops on interrupt line.
SELECT A SELECT Q SELECT INDEX REGISTER	Selection of the designated register. Register contents are displayed on console and quantities may be inserted manually into the registers.
*SET Push Buttons Numbered in the powers of 2, beginning with zero. Each group of 3 is an octal digit.	Allow for manual entry of a quantity into a given register. Forces that particular stage of the register to the Set state. The register to be loaded (A, Q, or B^b) must be selected.
*CLEAR Push Button	Clears the individual FFs within the selected register.

* Should be used only when the computer is stopped

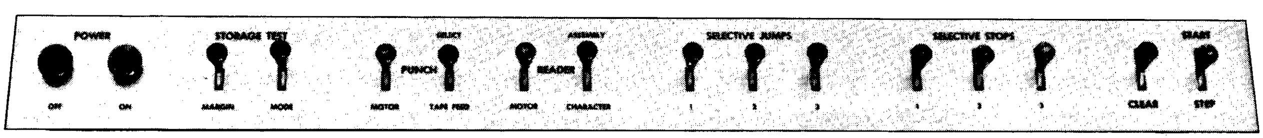


Figure 4-3. Manual Controls

READER AND PUNCH CONTROLS

TABLE 4-3. READER AND PUNCH CONTROLS

Switch	Function
PUNCH MOTOR Up: On Down: Off	Turns punch motor on or off. (Motor may also be turned on under program control.)
SELECT Up: Locks	Enables use of punch.
TAPE FEED Down: Momentary	Causes leader to be punched out.
READER MOTOR Momentary: Up: On Down: Off	Turns reader motor on or off. (Motor cannot be turned on by any other means.)
CHARACTER Momentary	Selects character mode. The reader sends each 7-bit character to computer separately.
ASSEMBLY Momentary	Selects Assembly mode. Reader sends four consecutive 6-bit characters assembled into a word to computer.

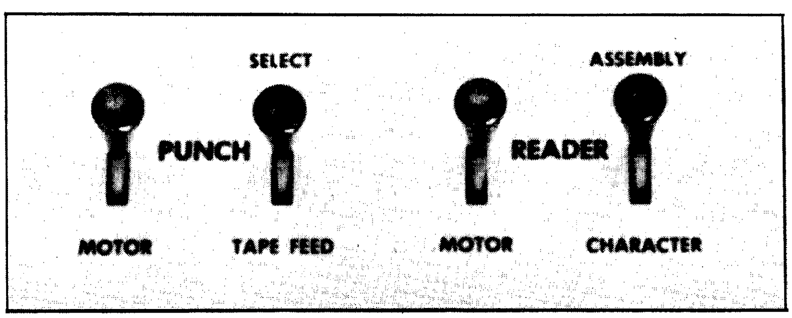


Figure 4-4. Reader and Punch Controls

1607 CONTROLS AND INDICATORS

Each tape unit is provided with push buttons for manual operation. These controls are mounted on a panel above the front door.

TABLE 4-4. 1607 CONTROLS AND INDICATORS

Control		Function
REWIND	S*	Controls manual rewind to load point.
	I**	Indicates rewind in progress.
CHANGE TAPE	S	Drops any manual selection and places tape unit in automatic or program control mode.
	I	Indicates tape rewound under program control and interlocked at metal leader; operation prevented until Stop Manual switch operated and moved off metal leader.
WRITE LOCKOUT	S	Drops power from unit and removes program designation.
	I	When lighted, indicates that tape unit is loaded with a reel which does not contain a file protection ring. The tape cannot be written as long as the light is on, but may be read.
1, 2, 3 or 4	S	Designates program selection of unit and applies power to unit. Each new unit designation cancels an existing designation
	I	Indicates unit selection and power-on condition.
REVERSE	S	Initiates reverse tape motion during manual operation.
	I	Indicates reverse tape motion.
STOP MANUAL	S	Drops unit from program control or drops Forward or Reverse selection and places unit in manual mode.
	I	Indicates manual mode.
FORWARD	S	Indicates forward tape motion during manual mode.
	I	Indicates forward tape motion.

* switch
** indicator

OPERATION

The main computer and external equipment are placed in operation by procedures which include loading and unloading data and programs, making necessary manual selections, and starting a program. Steps are listed in the recommended order of performance, beginning with the computer and external equipment in a shut down condition.

STARTING OPERATION WITH PRE-STORED LOAD PROGRAM

When a general loading program which provides for loading other programs is held in storage, the starting procedure is as follows:

1) Turn on power (Power On).
2) If punch is to be used, set Punch switch to Select and check for sufficient paper in reel. (If supply of tape is low see paragraph on additional procedures.)
3) Make required manual selections:
   - Selective Jumps
   - Selective Stops
4) Prepare paper tape baskets and empty chad box.
5) If paper tape is to be read, load into reader.
   a. Turn tape release lever clockwise to raise tape guide plate.
   b. Insert tape as shown in figure 4-5.
   c. Turn on reader (Reader switch).
   d. If tape is bi-octal with 7th level control holes, select Assembly mode.
   e. If tape is in flex or other code, select Character mode.
6) If magnetic tape is to be used: (figure 4-6)
   a. Open door to handler.
   b. Check that file reel to be loaded has been file protected as necessary.
   c. Mount the reel on the file reel hub and tighten the hub knob. To insure proper reel alignment push the reel firmly against the reel hub stop before tightening the knob. If the file protection ring has been removed from the reel, check that the Write Lockout lamp turns on when the reel is loaded. If the lamp does not turn on call maintenance.

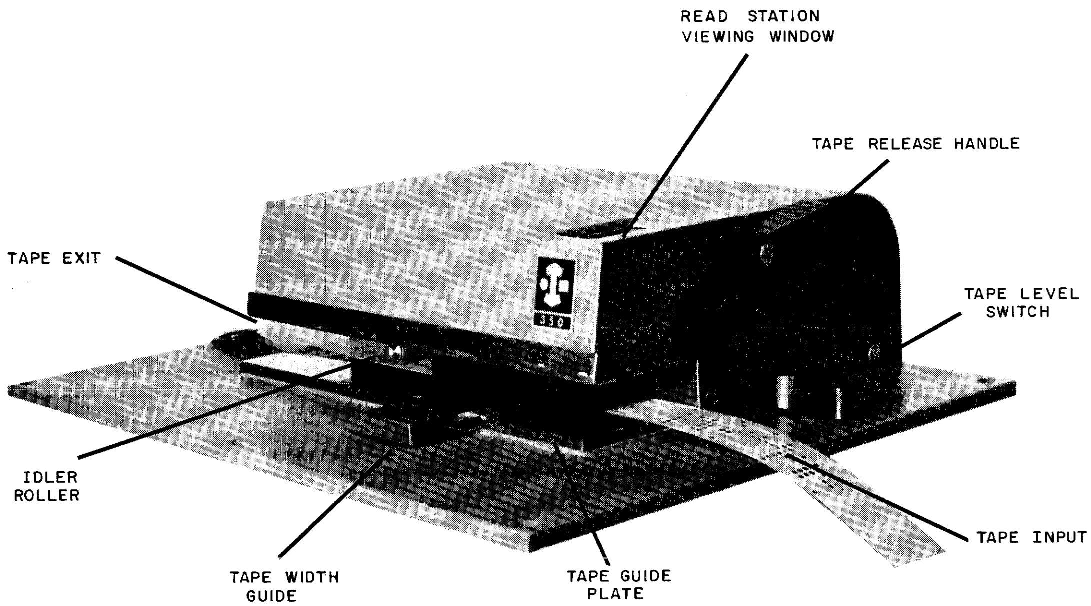


Figure 4-5. Paper Tape Reader

d. Press upper Reel Brake pushbutton to release mechanical brake and check that pulling tape from reel causes it to rotate clockwise. Pull sufficient tape from reel to reach end of permanent machine leader held by leader clamp.

e. Connect file tab to permanent machine leader.

f. Take up slack by turning file reel while pressing upper Reel Brake pushbutton.

g. Lift leader clamp and close door.

h. Press one of the unit selection switches (1, 2, 3, 4) to apply power to the unit and assign the unit a logical program selection number. Wait 2 minutes. The Stop Manual lamp should turn on, if not, call maintenance.

i. Press Forward button; wait 10 seconds.

j. Press Stop Manual.

k. Press Rewind button. Unit is ready when Rewind lamp turns off. If Stop Manual lamp remains on unit is not ready; call maintenance.

7) Master Clear both internal and external (press Clear then raise it).

8) Set Program Address register to address of first instruction of program.

9) Begin computer operation (set Start switch).

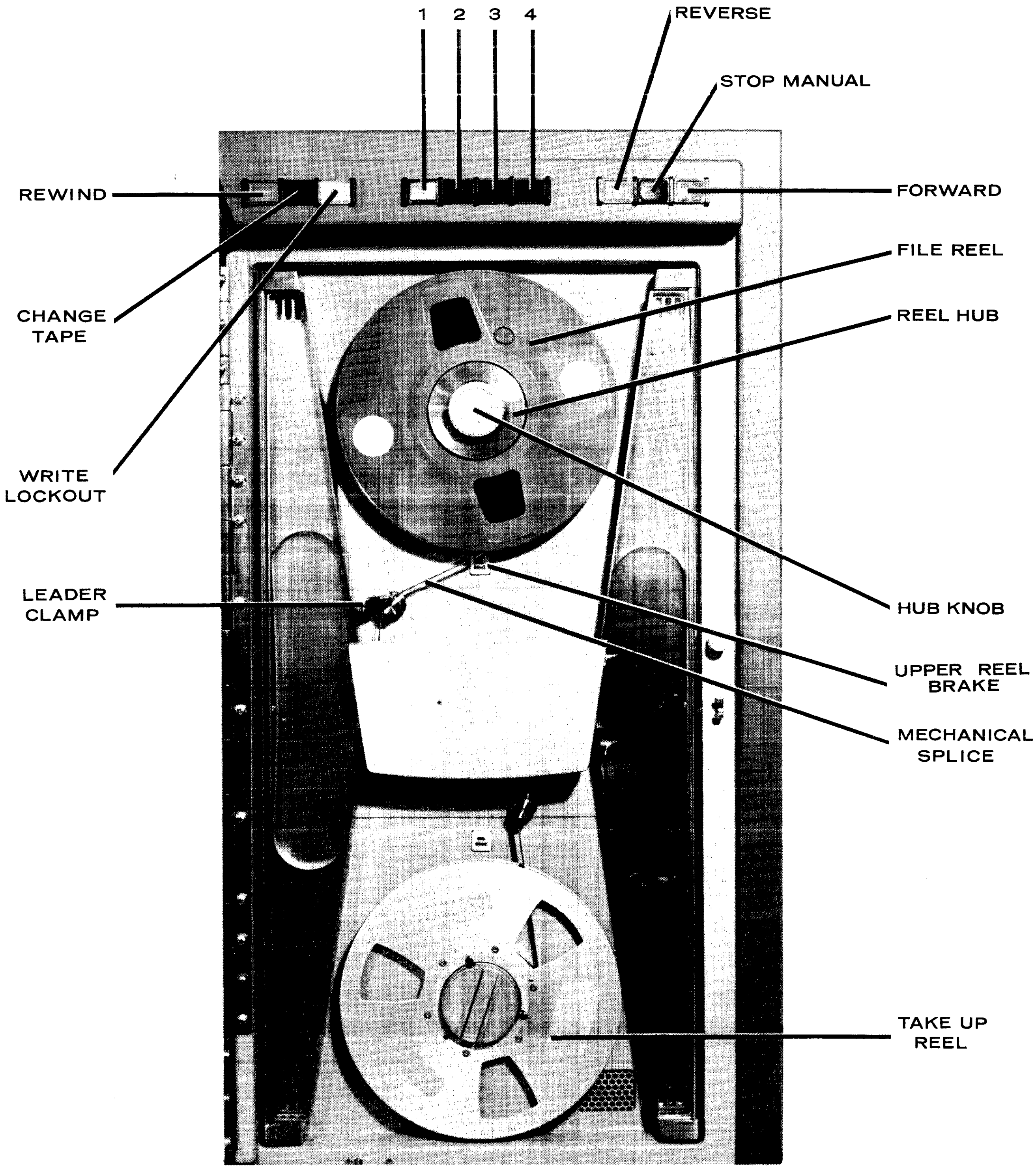


Figure 4-6. Tape Unit

STARTING OPERATION WITHOUT PRE-STORED LOAD PROGRAM

A load program to be entered in storage is usually on bi-octal paper tape. The following procedure enters the load program:

1) Turn on power.
2) Master clear both internal and external.
3) Press Start-Step switch once.
4) Clear function code and set to 200. (STA instruction)
5) Clear execution address and set to 00002. (channel 1 terminal address location)
6) Set terminal address of buffer in lowest five octal digits of A register
7) Press Start-Step switch once.
8) Load tape into reader.
9) Turn on reader motor (wait 10 seconds).
10) Raise reader mode switch to Assembly position.
11) Clear function code and set to 741. (Activate channel 1 code)
12) Clear execution address and set to initial address of buffer.
13) Press Start-Step switch once. Wait until tape loads (console lights come on).
14) Press Clear switch.
15) Steps 2 through 9 of operation with stored program may be performed.

SHUTTING DOWN EQUIPMENT

After operation has stopped, shut down the equipment.

1) Remove paper tape from reader and baskets; rewind tapes.
2) Turn off reader motor.
3) If punch was used, generate a foot of leader by pressing Tape Feed; remove tape and wind it up.
4) To unload magnetic tape:
    a) Press Stop Manual button to select manual mode.
    b) Press Reverse button to move tape backwards to change tape position.
    c) Open front door of tape unit.
    d) To secure tape, lower leader clamp.
    e) Press the upper Reel Brake button to release the mechanical brake and pull tape from file reel to provide slack.
    f) Unfasten mechanical splice which connects the file tab to the permanent machine leader.

g) Loosen file reel hub knob and remove the file reel.

h) Check if reel needs to be file protected and also if it is labelled adequately prior to storage.

5) Press Power Off button, which disconnects power from all equipments.

ADDITIONAL PROCEDURES

REPLACING TAPE ROLL AT PUNCH

The paper tape punch (figure 4-7) is mounted on a sliding rack on the left side of the console. To replace a roll of tape:

1) Pull out punch drawer.
2) Remove the tape reel from cradle at side of punch.
3) Unscrew tape hold-down assembly, remove old roll, and place new roll on reel. Replace hold-down assembly and mount reel in cradle.
4) Thread tape as shown in figure 4-7. Bring tape around lower roller and into guides leading to punch block.
5) Turn on punch motor and advance tape through the punch block by pressing the tape feed-out lever (at top).
6) Bring leader out through slot in compartment.
7) Slide punch back into compartment.

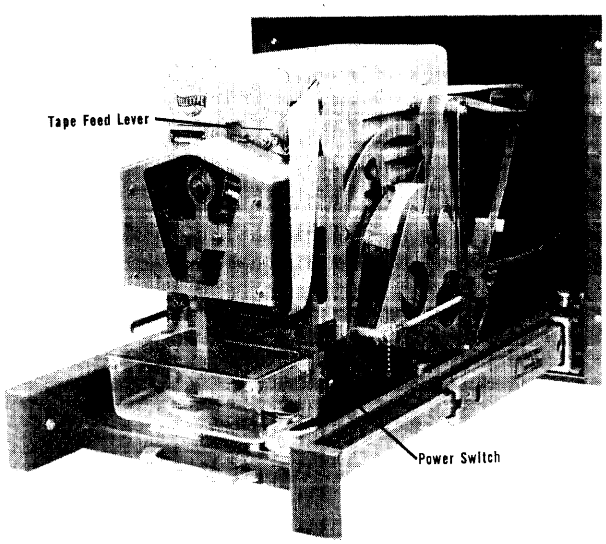


Figure 4-7. Paper Tape Punch

FILE PROTECTION RING

The back of the 1607 file reel has a slot near the hub which accepts a plastic file protection ring (figure 4-8). Writing on a tape is possible only when the reel contains a file protection ring. When the ring is in place the Write Lockout indicator goes out immediately after the reel is loaded onto the tape unit. The ring should be removed from the reel after writing is completed to avoid accidental rewriting. Tape may be read either with the ring in place or without it.

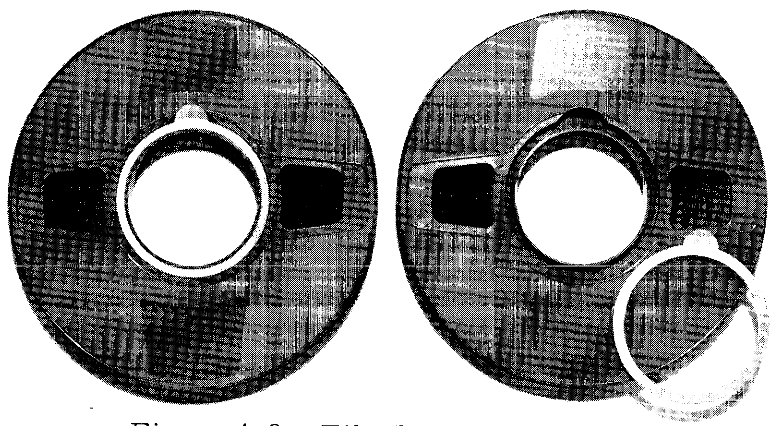


Figure 4-8. File Protection Ring

EMERGENCY PROCEDURES

A fault indication, or a warning signal from the buzzer, may call for special procedures on the part of the operator.

TABLE 4-5. EMERGENCY PROCEDURES

Punch out of tape	Load new roll of tape in punch at end of current operation.
Odd Storage Fault	Master clear. Restart program.
Even Storage Fault	Master clear. Restart program.
No Exit	Restart operation. If unable to proceed, master clear and restart program. If condition persists, notify maintenance.
Sweep	Place Mode switch in neutral position.
Buzzer Signal	Notify maintenance engineer immediately.

Faults for which the program provides corrective action are: Divide, Shift and Overflow Faults. (Refer to appendix.)

APPENDIX SECTION

APPENDIX I

COMMON NOTATIONS AND POWERS

COMMON PURE NOTATIONS

Decimal	Binary	Octal
00	00000	00
01	00001	01
02	00010	02
03	00011	03
04	00100	04
05	00101	05
06	00110	06
07	00111	07
08	01000	10
09	01001	11
10	01010	12
11	01011	13
12	01100	14
13	01101	15
14	01110	16
15	01111	17
16	10000	20
17	10001	21

POWERS OF COMMON NUMBER SYSTEMS

2^0 = 1	8^0 = 1	10^0 = 1
2^1 = 2	8^1 = 8	10^1 = 10
2^2 = 4	8^2 = 64	10^2 = 100
2^3 = 8	8^3 = 512	10^3 = 1,000
2^4 = 16	8^4 = 4,096	10^4 = 10,000
2^5 = 32	8^5 = 32,768	10^5 = 100,000
2^6 = 64	8^6 = 262,144	10^6 = 1,000,000
2^7 = 128	8^7 = 2,097,152	
2^8 = 256	8^8 = 16,777,216	
2^9 = 512		
2^{10} = 1,024		

APPENDIX II

FAULTS

Certain fault conditions may occur in the execution of a computer program which may be sensed by EXF instructions. The occurrence of the fault does not stop operation but sets an indicator that can be sensed. A fault is visually indicated on the console. The FFs set by fault conditions may be cleared by a computer Master Clear.

Shift Fault

Any attempt to shift a register more than 63_{10} (77_8) places right or left results in a shift fault. If the fault exists, the indicator is set prior to execution of the shift instruction and the shift fault background light on the console display panel is lighted. The shifts will be performed regardless of the status of the fault indicator. If an interrupt has been selected, the main program will be interrupted after executing the shift instruction. The shift fault may be sensed by 74 7 00120, 1.

Divide Fault

A divide fault occurs in fixed point divide instructions (25) when the divisor is zero or the required quotient exceeds the 23-bit capacity of the quotient register, A. The sign bit of A is examined at the end of the division phase. If it is equal to "1", a divide fault has occurred. If an interrupt has been selected, the main program will be interrupted after the divide instruction is completed. A divide fault is sensed by a 74 7 00110, 1.

Overflow Fault

An overflow fault is produced when the capacity of the A register ($>\pm 2^{23}-1$) is exceeded. The fault is sensed after the arithmetic operation causing the overflow is completed. An overflow may be sensed by a 74 7 00130, 1.

If an interrupt has been selected, the main program may or may not be halted before the next instruction can be executed because the scanner may not immediately recognize the interrupt.

A pass or sense overflow instruction could be programmed after the arithmetic instruction which causes the overflow. The interrupt would, in this case, be recognized before executing succeeding instructions which might alter the contents of the A register.

Even and Odd Storage Faults

These faults indicate a failure in computer storage and turn on background lights on the console display. The indicators may be cleared by an internal master clear. If a storage fault is produced, maintenance should be notified.

APPENDIX III

TABLE OF POWERS OF 2

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

APPENDIX IV

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000	0000
to	to
0777	0511
(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
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	0512
to	to
1777	1023
(Octal)	(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 to 6777 (Octal) | 3072 to 3583 (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	4022	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 to 7777 (Octal) | 3584 to 4095 (Decimal)

APPENDIX V

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	.00109[illegible]	.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 VI
EXF AND CHARACTER CODES

SELECT

74 0 000C0	Interrupt on Channel C inactive; C=1-6
000C1	Remove Selection above on Channel C
00070	Clear Arithmetic faults
00100	Interrupt on Arithmetic faults
00101	Remove Selection above
01000	Select Interrupt from real-time clock
01001	Remove Selection above

SENSE

74 7 000C0	Skip exit on Channel C active; C=1-6
000C1	Skip exit on Channel C inactive; C=1-6
00110	Skip exit on Divide fault
00111	Skip exit on No Divide fault
00120	Skip exit on Shift fault
00121	Skip exit on No Shift fault
00130	Skip exit on Overflow fault
00131	Skip exit on No Overflow fault

CONSOLE I/O EQUIPMENT

SELECT

74 0 11200	Reader and no interrupt on end-of-tape
11210	Reader and set end-of-tape indicator
11220	Reader and interrupt on end-of-tape
21200	Punch Assembly mode
21210	Punch Character mode
21240	Turn Punch Motor off

SENSE

74 7 11200	Skip exit on Reader, end-of-tape
11201	Skip exit on Reader, no end-of-tape
11210	Skip exit on Reader, assembly mode
11211	Skip exit on Reader, character mode
21200	Skip exit on Punch out-of-tape
21201	Skip exit on Punch not out-of-tape
00200	Skip exit on Real Time Clock Interrupt
00201	Skip exit on no Real Time Clock Interrupt

1607 CODES

SELECT

74 0	320n1	Select read tape n, binary
	320n2	Select read tape n, coded
	32001	Read selected read tape, binary
	32002	Read selected read tape, coded
	32004	Interrupt when selected read tape ready
	32005	Rewind selected read tape
	32006	Backspace selected read tape
	32007	Rewind selected read tape, interlock
	420n1	Select write tape n, binary
	420n2	Select write tape n, coded
	42001	Write selected write tape, binary
	42002	Write selected write tape, coded
	42003	Write end-of-file mark
	42004	Interrupt when selected write tape ready
	42005	Rewind selected write tape
	42006	Backspace selected write tape
	42007	Rewind selected write tape, interlock
	---	Status Request

SENSE

74 7	32000-1	Ready to read
	32002-3	Read parity error
	32004-5	Read length error
	32006-7	End-of-file mark
	42000-1	Ready to write
	42002-3	Write reply parity error
	42004-5	Write reply length error
	42006-7	End of tape marker

1608 CODES

SELECT

74 05 77n1	Select tape N to read binary
77n2	Select tape N to read coded
7001	Prepare selected tape to read binary
7002	Prepare selected tape to read coded
7004	Interrupt when selected tape ready
7005	Rewind selected tape
7006	Backspace selected tape
7007	Rewind and unload selected tape
7101	Turn off "Tape Indicator" on read unit
7102	Set low density on read unit
7103	Set high density on read unit
7104	Search file mark forward on read unit
7105	Search file mark backward on read unit
7106	Remove interrupt selection on read unit
74 06 77n1	Select tape N to write binary
77n2	Select tape N to write coded
7001	Prepare selected tape to write binary
7002	Prepare selected tape to write coded
7003	Write end-of-file (tape mark) on selected unit
7004	Interrupt when selected tape ready
7005	Rewind selected tape
7006	Backspace selected tape
7007	Rewind and unload selected tape
7101	Turn off "Tape Indicator" on write unit
7102	Set low density on write unit
7103	Set high density on write unit
7104	Skip bad spot on selected write unit
7106	Remove interrupt selection on write unit

Note: All codes EXCEPT those specifying a unit number "N" refer to previously selected unit.

1608 FUNCTION CODES

SENSE

74 75	7000	Skip exit on ready to read
	7001	Skip exit on not ready to read
	7002	Skip exit on read parity error
	7003	Skip exit on no read parity error
	7004	Skip exit on read length error
	7005	Skip exit on no read length error
	7006	Skip exit on end-of-file mark
	7007	Skip exit on no end-of-file mark
	7106	Skip exit when read unit is rewinding or is at Load Point
	7107	Skip exit when read unit is not rewinding or is at Load Point
	7102	Skip exit on even bus ready
	7103	Skip exit on not even bus ready
	7104	Skip exit on odd bus ready
	7105	Skip exit on not odd bus ready
74 76	7000	Skip exit on ready to write
	7001	Skip exit on not ready to write
	7002	Skip exit on write reply parity error
	7003	Skip exit on no write reply parity error
	7004	Skip exit on write reply length error
	7005	Skip exit on no write reply length error
	7006	Skip exit on end-of-tape marker
	7007	Skip exit on no end-of-tape marker
	7106	Skip exit when write unit is rewinding
	7107	Skip exit when write unit is not rewinding
	7102	Skip exit on even bus ready
	7103	Skip exit on not even bus ready
	7104	Skip exit on odd bus ready
	7105	Skip exit on not odd bus ready

1609 ADAPTER CODES

SELECT

74 0	54001	Read Station A
	54002	Read station B
	54003	Read station A and B
	54004	Punch
	54005	Punch and Read A
	54006	Punch and Read B
	54007	Punch, Read A and B

SENSE

74 5	54002	Skip exit on 1604 Selected
	54003	Skip exit on 1604 Not Selected
	64004	Skip exit on unit ready
	64005	Skip exit on unit not ready

SELECT

74 0	54011	Read station A with Interrupt
	54012	Read station B with Interrupt
	54013	Read stations A and B with Interrupt
	54014	Punch with Interrupt
	54015	Punch and Read A with Interrupt
	54016	Punch and Read B with Interrupt
	54017	Punch, Read A and B, with Interrupt

161 TYPEWRITER CODES

SELECT

74 0	C4210	Select Typewriter Output
	C4220	Select Typewriter Input

C = Channel number

1610 INSTRUCTIONS

SELECT

74 05 4001	Primary sequence read
4002	Secondary sequence read
4003	Primary and secondary sequence read
4005	Primary sequence read with interrupt
4006	Secondary sequence read with interrupt
4007	Primary and secondary sequence read with interrupt
74 06 4001	Print
4002	Punch
4005	Print with Interrupt
4006	Punch with Interrupt

SENSE

74 76 4002	Skip exit in read ready
4003	Skip exit on read not ready
4002	Skip exit on print ready
4003	Skip exit on print not ready
4004	Skip exit on punch ready
4005	Skip exit on punch not ready

1612 EXF CODES

SELECT

74 0	* y6000	Select Printer
	y6001	Advance Paper One Line
	y6002	Advance Paper Two Lines
	y6003	Skip on Channel 7 (space to selected area of form)
	y6004	Skip on Channel 8 (space to top of form)
	y6006	Suppress Paper Advance
	y6007	Interrupt on Printer Ready
	y6010	Clear Monitor Channels 1-6
	y6011	Select Monitor Channel 1
	y6012	Select Monitor Channel 2
	y6013	Select Monitor Channel 3
	y6014	Select Monitor Channel 4
	y6015	Select Monitor Channel 5
	y6016	Select Monitor Channel 6

SENSE

74 7	y6000	Skip exit on Printer Ready
	y6001	Skip exit on Printer not Ready

* y = Channel designator

1615 EXF CODES

1604 EXTERNAL FUNCTION CODES

$(N = 1_8 \rightarrow 10_8)$

OUTPUT

74 0 C	20N1	Select Tape N to Write Binary
	20N2	Select Tape N to Write Coded
	2001	Prepare Selected Tape to Write Binary
	2002	Prepare Selected Tape to Write Coded
	2003	Write End-of-File Mark on Selected Tape
	2004	Select Interrupt when Write Tape Next Ready

OUTPUT

74 0 C	2005	Rewind Selected Write Tape
	2006	Backspace Selected Write Tape
	2007	Rewind-Unload Selected Write Tape
	2400	Clear Interrupt Selections on Write Tape
	2401	Set Low Density on Selected Write Tape
	2402	Set High Density on Selected Write Tape
	2403	Skip Bad Spot on Selected Write Tape
	2404	Select Interrupt on Next Error

SENSE

74 7 C	2000	Exit on Ready to Write
	2001	Exit on Not Ready to Write
	2002	Exit on Write Reply Parity Error
	2003	Exit on No Write Reply Parity Error
	2004	Exit on Write Reply Length Error
	2005	Exit on No Write Reply Length Error
	2006	Exit on End of Tape Marker
	2007	Exit on Not End of Tape Marker
	2400	Exit on Ready to Select
	2401	Exit on Not Ready to Select
	2402	Exit on Load Point
	2403	Exit on Not Load Point
	2404	Exit on Interrupt on Write Tape
	2405	Exit on No Interrupt on Write Tape
	2406	Exit on Write Program Error
	2407	Exit on No Write Program Error

INPUT

74 0 C	20N1	Select Tape N to Read Binary One Record
	20N2	Select Tape N to Read Coded One Record
	22N1	Select Tape N to Read Binary One File
	22N2	Select Tape N to Read Coded One File
	2001	Prepare Selected Tape to Read Binary One Record
	2002	Prepare Selected Tape to Read Coded One Record
	2201	Prepare Selected Tape to Read Binary One File
	2202	Prepare Selected Tape to Read Coded One File
	2003	Move Selected Read Tape Forward One Record

INPUT

74 0 C	2203	Search File Mark Forward
	2004	Select Interrupt when Read Tape Next Ready
	2005	Rewind Selected Read Tape
	2006	Backspace Selected Read Tape
	2206	Search File Mark Backward
	2007	Rewind-Unload Selected Read Tape
	2400	Clear Interrupt Selections on Read Tape
	2401	Set Low Density on Selected Read Tape
	2402	Set High Density on Selected Read Tape
	2404	Select Interrupt on Next Error

SENSE

74 7 C	2000	Exit on Ready to Read
	2001	Exit on Not Ready to Read
	2002	Exit on Read Parity Error
	2003	Exit on No Read Parity Error
	2004	Exit on Read Length Error
	2005	Exit on No Read Length Error
	2006	Exit on End of Tape Marker
	2007	Exit on Not End of Tape Marker
	2400	Exit on Ready to Select
	2401	Exit on Not Ready to Select
	2402	Exit on Load Point
	2403	Exit on Not Load Point
	2404	Exit on Interrupt on Read Tape
	2405	Exit on No Interrupt on Read Tape
	2406	Exit on Read Program Error
	2407	Exit on No Read Program Error

160 EXTERNAL FUNCTION CODES

($N = 1_8 \rightarrow 7_8$)

WRITE OPERATIONS

60N1	Select Tape N to Write Binary
60N2	Select Tape N to Write Coded
6001	Prepare Selected Tape to Write Binary
6002	Prepare Selected Tape to Write Coded
6003	Write End-of-File on Selected Tape
6005	Rewind Selected Write Tape
6006	Backspace Selected Write Tape
6007	Rewind-Unload Selected Write Tape
6010	Set Low Density on Selected Write Tape
6020	Set High Density on Selected Write Tape
6030	Skip Bad Spot on Selected Write Tape
6053	Request Status

READ OPERATIONS

50N1	Select Tape N to Read Binary One Record
50N2	Select Tape N to Read Coded One Record
52N1	Select Tape N to Read Binary One File
52N2	Select Tape N to Read Coded One File
5001	Prepare Selected Tape to Read Binary One Record
5002	Prepare Selected Tape to Read Coded One Record
5201	Prepare Selected Tape to Read Binary One File
5202	Prepare Selected Tape to Read Coded One File
5003	Move Selected Read Tape Forward One Record
5203	Search File Mark Forward
5005	Rewind Selected Read Tape
5006	Backspace Selected Read Tape
5206	Search File Mark Backward
5007	Rewind-Unload Selected Read Tape
5010	Set Low Density on Selected Read Tape
5020	Set High Density on Selected Read Tape

STATUS RESPONSE

X2XX	Ready to Read
X1XX	Ready to Write
X4XX	Read Parity Error
XX2X	Write Reply Parity Error
XX1X	End-of-File Mark
XX4X XXX4	End of Tape Marker

SATELLITE EXTERNAL FUNCTION CODES

1604 EXTERNAL FUNCTION CODES

OUTPUT SELECT

74 0 C	2500	Release Direct Selections
	2501	Select Write Control for 160
	2502	Release Write Control to 1604
	2503	Select Direct 1604 to 160
	2504	Select Action Request
	2520	Clear Communication Flag 2
	2540	Set Communication Flag 1
	2560	Clear Communication Flag 1

OUTPUT SENSE

74 7 C	2500	Exit on Write Control Available
	2501	Exit on Write Control not Available
	2520	Exit on Communications Flag 2 Set
	2521	Exit on Communications Flag 2 not Set
	2560	Exit on Communications Flag 1 Set
	2561	Exit on Communications Flag 1 not Set

INPUT SELECT

74 0 C	2501	Select Read Control for 160
	2502	Release Read Control to 1604
	2503	Select Direct 160 to 1604
	2505	Release Interrupt

INPUT SENSE

74 7 C	2500	Exit on Read Control Available
	2501	Exit on Read Control not Available
	2504	Exit on 160 Interrupt
	2505	Exit on No. 160 Interrupt

160 EXTERNAL FUNCTION CODES

WRITE SELECT

6050	Release Action Request
6051	Set Communications Flag 2
6052	Release Write Control to 1604
6055	Clear Communications Flag 1
6056	Clear Communications Flag 2

READ SELECT

5051	Set Communications Flag 1
5052	Release Read Control to 1604
5053	Select Interrupt

STATUS RESPONSE

4XXX	Read Control Available
2XXX	Write Control Available
1XXX	Direct 160 to 1604
X4XX	Direct 1604 to 160
XXX2	160 Action Request
XXX1	Communications Flag 1 Set

163 MAGNETIC TAPE CODE

EXTERNAL FUNCTION

111X	Write Tape X (6-bit) if OUT Command Given
111X	Write End of File Mark (if NO OUT Command Given)
211X	Write Tape X (12-bit) if OUT Command Given
112X	Backspace Tape X One Record (if INA Command Given)
112X	Backspace Tape X to End of File Mark (if NO INA Command Given)
113X	Read Forward Tape X (6-bit) (if an INPUT Command Given)
213X	Read Forward Tape X (12-bit) (if an INPUT Command Given)
113X	Search Tape X for End of File Mark (if NO INPUT Command Given)
114X	Request Tape X Status
115X	Rewind Unload Tape X
116X	Rewind Load Tape X

EXTERNAL FUNCTION

1171	Set Tapes to Odd Parity
1172	Set Tapes to Even Parity (BCD)

STATUS RESPONSE

0000	Odd Parity Selected - No Errors
0001	Even Parity Selected - No Errors
0002	Tape X Not Ready
0004	Horizontal and/or Vertical Parity Error
0015	Illegal BCD Detected on Write
0020	End of File Mark Read
0040	End of Tape or Load Point Sensed
7777	163 or 164 Off

166 LINE PRINTER CODES

160 SELECT CODES

0700	Asynchronous Print
0710	Synchronous Print
072X 0727	Paper Advance
0740	Status Request

STATUS RESPONSE

0000	166 Ready
0001	Buffer Busy
0002	Out of Paper
0004	Paper Moving
0010	Drum Stationary
0020	Off Line

167 CARD READER CODES

160 SELECT CODES	
4500	External Function Clear
4501	Free Run Read (online only)
4502	Single Cycle Read
4540	Check Status (online only)

STATUS RESPONSE	
0000	167 Ready
0001	Hopper Empty
0002	Stacker Full
0004	Feed Failure
0010	Program Error
0020	Amplifier Failure
0040	Motor Power Off

1617 EXTERNAL FUNCTION CODES

SELECT

160	1604	Function
4500	74 0 4000	Unit Clear
4501	4001	Free Run Read
4502	4002	Single Cycle Read
4504	4004	Translate H→BCD
4540	-	Check Status

160 STATUS RESPONSE	
0000	Read Ready
0001	Hopper Empty
0002	Stacker Full
0004	Feed Failure
0010	Program Error
0020	Amplifier Failure
0040	Motor Power Off

1604 SENSE CODES

74 7 X 4002	Read Ready
4003	Not Read Ready
4004	Hopper Empty
4005	Hopper Not Empty
4010	Stacker Full
4011	Stacker Not Full
4020	Amplifier Failure
4021	No Amplifier Failure
4040	Feed Failure
4041	No Feed Failure
4100	Reader Active
4101	Reader Not Active

SPECIAL ADDRESSES FOR BUFFER CONTROL WORDS

Special Address	
00000	Initial start
00002	Channel 1 control (terminal address)
00003	Channel 1 control (current address)
00004	Channel 2 control (terminal address)
00005	Channel 2 control (current address)
00006	Channel 3 control (terminal address)
00007	Channel 3 control (current address)
00010	Channel 4 control (terminal address)
00011	Channel 4 control (current address)
00012	Channel 5 control (terminal address)
00013	Channel 5 control (current address)
00014	Channel 6 control (terminal address)
00015	Channel 6 control (current address)

BUFFER MODE INSTRUCTIONS

74 0 04010 - Select 24-bit mode for channel 1 and 2

04011 - Select 160 (12-bit) mode for channel 1 and 2

04012 - Select 1604 (48-bit) mode for channel 1 and 2

04020 - Select 24-bit mode for channel 3 and 4

04021 - Select 160 (12-bit) mode for channel 3 and 4

04022 - Select 1604 (48-bit) mode for channel 3 and 4

04030 - Select 24-bit mode for channel 1 and 2, 3 and 4

04031 - Select 160 (12-bit) mode for channel 1 and 2, 3 and 4

04032 - Select 1604 (48-bit) mode for channel 1 and 2, 3 and 4

04040 - Select 24-bit mode for channel 5 and 6

04041 - Select 160 (12-bit) mode for channel 5 and 6

04042 - Select 1604 (48-bit) mode for channel 5 and 6

04050 - Select 24-bit mode for channel 1 and 2, 5 and 6

04051 - Select 160 (12-bit) mode for channel 1 and 2, 5 and 6

04052 - Select 1604 (48-bit) mode for channel 1 and 2, 5 and 6

04060 - Select 24-bit mode for channel 3 and 4, 5 and 6

04061 - Select 160 (12-bit) mode for channel 3 and 4, 5 and 6

04062 - Select 1604 (48-bit) mode for channel 3 and 4, 5 and 6

04070 - Select 24-bit mode for channel 1 and 2, 3 and 4, 5 and 6

04071 - Select 160 bit mode for channel 1 and 2, 3 and 4, 5 and 6

04072 - Select 1604 bit mode for channel 1 and 2, 3 and 4, 5 and 6

04012 - Clear channel 1 and 2 160 mode select

04011 - Clear channel 1 and 2 1604 mode select

04022 - Clear channel 3 and 4 160 mode select

04021 - Clear channel 3 and 4 1604 mode select

04042 - Clear channel 5 and 6 160 mode select

04041 - Clear channel 5 and 6 1604 mode select

Input - Output Typewriter Codes

CHARACTERS UC	LC	CODE	CHARACTERS UC	LC	CODE
A	a	30	X	x	27
B	b	23	Y	y	25
C	c	16	Z	z	21
D	d	22	)	0	56
E	e	20	*	1	74
F	f	26	@	2	70
G	g	13	#	3	64
H	h	05	$	4	62
I	i	14	%	5	66
J	j	32	¢	6	72
K	k	36	&	7	60
L	l	11	½	8	33
M	m	07	(	9	37
N	n	06	_	-	52
O	o	03	?	/	44
P	p	15	"	'	54
Q	q	35	°	+	46
R	r	12	.	.	42
S	s	24	:	;	50
T	t	01	,	,	40
U	u	34	÷	=	02
V	v	17	tab	tab	51
W	w	31	space		04
Backspace		61	Carriage Return		45
Lower Case		57	Upper case		47

Magnetic Tape BCD Codes

Character	Code (Octal)	Character	Code (Octal)
A	61	2	02
B	62	3	03
C	63	4	04
D	64	5	05
E	65	6	06
F	66	7	07
G	67	8	10
H	70	9	11
I	71	~~&~~ +	60
J	41	-	40
K	42	(blank)	20
L	43	/	21
M	44	. (period)	73
N	45	$	53
O	46	*	54
P	47	, (comma)	33
Q	50	~~%~~ (	34
R	51	~~#~~ =	13
S	22	~~@~~ -	14
T	23	~~¤~~)	74
U	24		
V	25	record mark	32
W	26		
X	27		
Y	30	group mark	77
Z	31	tape mark	17
0	12	For [illegible]	56
1	01		16

Flexowriter Codes

UC	LC	CODE	UC	LC	CODE
A	a	30	Y	y	25
B	b	23	Z	a	21
C	c	16	0	0	56
D	d	22	1	1	74
E	e	20	2	2	70
F	f	26	3	3	64
G	g	13	4	4	62
H	h	05	5	5	66
I	i	14	6	6	72
J	j	32	7	7	60
K	k	36	8	8	33
L	l	11	9	9	37
M	m	07	-	-	52
N	n	06	'	/	44
O	o	03	(	)	54
P	p	15	+	,	46
Q	q	35	=	.	42
R	r	12	:	;	50
S	s	24	CR		45
			Upper Case (UC)		47
T	t	01	Lower Case (LC)		57
			Back Space (BS)		61
U	u	34	Color Shift (CS)		02
			Tabulate (TAB)		51
V	v	17	Stop		43
			Space		04
W	w	31	Tape Feed		00
			Delete		77
X	x	27			

Notes:
1. Leader - Blank Tape, Delete - Deleted Character
   Stop - Stop Flexowriter reader,
2. 10, 40, 41, 53, 55, 63, 65, 67, 71, 73, 75, and 76 - illegal

Punched Card Codes

Char	Card	BCD	Char	Card	BCD	Char	Card	BCD	Char	Card	BCD
			+	12	60	---	11	40			20
1	1	01	A	12 1	61	J	11 1	41	/	0 1	21
2	2	02	B	12 2	62	K	11 2	42	S	0 2	22
3	3	03	C	12 3	63	L	11 3	43	T	0 3	23
4	4	04	D	12 4	64	M	11 4	44	U	0 4	24
5	5	05	E	12 5	65	N	11 5	45	V	0 5	25
6	6	06	F	12 6	66	O	11 6	46	W	0 6	26
7	7	07	G	12 7	67	P	11 7	47	X	0 7	27
8	8	10	H	12 8	70	Q	11 8	50	Y	0 8	30
9	9	11	I	12 9	71	R	11 9	51	Z	0 9	31
0	0	12									
=	8,3	13	.	12 8,3	73	$	11 8,3	53	,	0 8,3	33
-	8,4	14	)	12 8,4	74	*	11 8,4	54	(	0 8,4	34

1612 Printer Codes

CHAR	CODE	CHAR	CODE	CHAR	CODE	CHAR	CODE
Blank	20	F	66	V	25	≤	15
0	12	G	67	W	26	|	16
1	01	H	70	X	27	[	17
2	02	I	71	Y	30	]	32
3	03	J	41	Z	31	→	35
4	04	K	42	.	73	≡	36
5	05	L	43	-	40	~, ∧	37
6	06	M	44	+	60	% or ∨	52
7	07	N	45	=	13	$ or ¬	53
8	10	O	46	(	34	↑	55
9	11	P	47	)	74	↓	56
A	61	Q	50	/	21	>	57
B	62	R	51	*	54	<	72
C	63	S	22	,	33	≥	75
D	64	T	23	:	00	?	76
E	65	U	24	≠	14	;	77
						(	34
						)	74

In last column, codes ~ % $ appear if business application, ∧ ∨ ¬ for scientific application.

APPENDIX VII
924 REPERTOIRE

FULL-WORD TRANSMISSION

LDA	LOAD A	$(M) \rightarrow A$
LAC	LOAD A, COMPLEMENT	$(M)^1 \rightarrow A$
LDQ	LOAD Q	$(M) \rightarrow Q$
LQC	LOAD Q COMPLEMENT	$(M)^1 \rightarrow Q$
STA	STORE A	$(A) \rightarrow M$
STQ	STORE Q	$(Q) \rightarrow M$
XAQ	INTERCHANGE A AND Q	$(A) \rightarrow Q$; $(Q) \rightarrow A$

ADDRESS TRANSMISSION

LIL	LOAD INDEX	$(M) \rightarrow B^b$
SIL	STORE INDEX	$(B^b) \rightarrow m$
SAL	SUBSTITUTE ADDRESS	$(A_{14}-A_{00}) \rightarrow M$
ENA	ENTER A	$Y \rightarrow A$, Extend sign Y
ENQ	ENTER Q	$Y \rightarrow Q$, Extend sign Y
ATI	A TO INDEX	$(A) \rightarrow B^b$ (m not used)
QTI	Q TO INDEX	$(Q) \rightarrow B^b$ (m not used)
ENI	ENTER INDEX	$y \rightarrow B^b$; $b = 0$: pass

FULL-WORD ARITHMETIC

ADD	ADD	$[(A) + (M)] \rightarrow A$
SUB	SUBTRACT	$[(A) - (M)] \rightarrow A$
MUI	MULTIPLY	$(M)\ (A) \rightarrow QA$
DVI	DIVIDE	$(QA)/(M) \rightarrow A$; Remainder $= Q_f$
TAL	TALLY	Count "1's" in A (m not used)

ADDRESS ARITHMETIC

INA	INCREASE A	$[Y + (A)] \rightarrow A$, Extend sign Y
INI	INCREASE INDEX	$y + (B^b) \rightarrow B^b$
INQ	INCREASE Q	$[Y + (Q)] \rightarrow Q$, Extend sign Y

LOGICAL

SST	SELECTIVE SET	Set (A_n) for $(M_n) = 1$
SCM	SELECTIVE COMPLEMENT	Complement (A_n) for $(M_n) = 1$
SCL	SELECTIVE CLEAR	Clear (A_n) for $(M_n) = 1$
SSU	SELECTIVE SUBSTITUTE	$(M_n) \rightarrow (A_n)$ for $(Q_n) = 1$
LDL	LOAD LOGICAL	$L(Q)(M) \rightarrow A$
ADL	ADD LOGICAL	$[(A) + L(Q)(M)] \rightarrow A$
SBL	SUBTRACT LOGICAL	$[(A) - L(Q)(M)] \rightarrow A$
STL	STORE LOGICAL	$L(Q)(A) \rightarrow M$
CMA	COMPLEMENT A	$(A)^1 \rightarrow A$ (m not used)
CMQ	COMPLEMENT Q	$(Q)^1 \rightarrow Q$ (m not used)

SHIFTING

ARS	A RIGHT SHIFT	Shift (A) right by K
QRS	Q RIGHT SHIFT	Shift (Q) right by K
LRS	AQ RIGHT SHIFT	Shift (AQ) right by K
LLS	AQ LEFT SHIFT	Shift (AQ) left by K
QLS	Q LEFT SHIFT	Shift (Q) left by K
ALS	A LEFT SHIFT	Shift (A) left by K
SCA	SCALE A	Shift (A) left until $\lvert(A)\rvert \geq .5$ or $k = 0$; (k - No. of shifts) $\rightarrow B^b$
SCQ	SCALE AQ	Shift (AQ) left until $\lvert(AQ)\rvert \geq .5$ or $k = 0$; (k - No. of shifts) $\rightarrow B^b$

REPLACE

RAD	REPLACE ADD	$[(M) + (A)] \rightarrow$ M and A
RSB	REPLACE SUBTRACT	$[(M) - (A)] \rightarrow$ M and A
RAO	REPLACE ADD ONE	$[(M) + 1] \rightarrow$ M and A
RSO	REPLACE SUBTRACT ONE	$[(M) - 1] \rightarrow$ M and A

STORAGE SEARCH

EQS	EQUALITY SEARCH	Search (B^b) words, if (M-1), (M-2), etc. = (A) : Skip Exit
THS	THRESHOLD SEARCH	Search (B^b) words, if (M-1), (M-2), etc. > (A) : Skip Exit

MEQ	MASKED EQUALITY	Search (B^b) words, if L(Q)(M-1), (M-2), etc. = (A) : Skip Exit
MTH	MASKED THRESHOLD	Search (B^b) words, if L(Q)(M-1), (M-2), etc. > (A) : Skip Exit
PTS	PATTERN SEARCH	Search (B^b) words, if $[(A) \forall (M-1), (M-2)]$, etc. = (A) : Skip Exit

STORAGE TEST

SSK	STORAGE SKIP	(M_{23}) Neg: Skip Exit; (M_{23}) Pos: Exit
SSH	STORAGE SHIFT	(M_{23}) Neg: Skip Exit, left 1; (M_{23}) Pos: Exit, left 1

SKIP

ISK	INDEX SKIP	$(B^b) = y$: Skip NI; $(B^b) \neq y$: $(B^b) + 1 \rightarrow B^b$ and exit
SKH	SKIP HIGH	$(B^b) \geq y$: Exit, $(B^b) < y$: Skip Exit
SKL	SKIP LOW	$(B^b) < y$: Exit, $(B^b) \geq y$: Skip Exit

JUMPS AND STOPS

AJP	A JUMP	Jump to m on condition j
QJP	Q JUMP	Jump to m on condition j
SLJ	SELECTIVE JUMP	Jump to m on condition j
SLS	SELECTIVE STOP	Stop on j, and jump to m
UJP	UNCONDITIONAL JUMP	Jump to M; b = 0, 7, jump to m
IJP	INDEX JUMP	$(B^b) \neq 0 : B^b - 1 \rightarrow B^b$, jump to m; $(B^b) = 0$, NI
XEC	EXECUTE	Execute instruction at M
URJ	UNCONDITIONAL RETURN JUMP	Return jump to M

EXTERNAL FUNCTION

EXF	EXTERNAL FUNCTION	j = 1-6 Activate Channel; j = 0 Select conditional y; j = 7 on condition y, exit or skip exit

APPENDIX VIII
NUMBER SYSTEMS

Any number system may be defined by two characteristics, the radix or base and the modulus. The radix or base is the number of unique symbols used in the system. The decimal system has ten symbols, 0 through 9. Modulus is the number of unique quantities or magnitudes a given system can distinguish. For example, an adding machine with ten digits, or counting wheels, would have a modulus of $10^{10}-1$. The decimal system has no modulus because an infinite number of digits can be written, but the adding machine has a modulus because the highest number which can be expressed is 9, 999, 999, 999.

Most number systems are positional, that is, the relative position of a symbol determines its magnitude. In the decimal system, a 5 in the units column represents a different quantity than a 5 in the tens column. Quantities equal to or greater than 1 may be represented by using the 10 symbols as coefficients of ascending powers of the base 10. The number 984_{10} is:

$$
\begin{array}{rcl}
9 \times 10^2 = 9 \times 100 & = & 900 \\
+8 \times 10^1 = 8 \times 10 & = & 80 \\
+4 \times 10^0 = 4 \times 1 & = & \underline{4} \\
 & & 984_{10}
\end{array}
$$

Quantities less than 1 may be represented by using the 10 symbols as coefficients of ascending negative powers of the base 10. The number 0.593_{10} may be represented as:

$$
\begin{array}{rcl}
5 \times 10^{-1} = 5 \times .1 & = & .5 \\
+9 \times 10^{-2} = 9 \times .01 & = & .09 \\
+3 \times 10^{-3} = 3 \times .001 & = & \underline{.003} \\
 & & 0.593_{10}
\end{array}
$$

BINARY NUMBER SYSTEM

Computers operate faster and more efficiently by using the binary number system. There are only two symbols, 0 and 1; the base = 2. The following shows the positional value.

. . .	2^5	2^4	2^3	2^2	2^1	2^0	
	=32	=16	=8	=4	=2	=1	Binary point

The binary number 0 1 1 0 1 0 represents:

$$
\begin{aligned}
0 \times 2^5 &= 0 \times 32 = 0 \\
+1 \times 2^4 &= 1 \times 16 = 16 \\
+1 \times 2^3 &= 1 \times 8 = 8 \\
+0 \times 2^2 &= 0 \times 4 = 0 \\
+1 \times 2^1 &= 1 \times 2 = 2 \\
+0 \times 2^0 &= 0 \times 1 = 0 \\
\hline
&\quad 26_{10}
\end{aligned}
$$

Fractional binary numbers may be represented by using the symbols as coefficients of ascending negative powers of the base.

		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5} . . .
Binary Point	.	=1/2	=1/4	=1/8	=1/16	=1/32

The binary number 0.10 110 may be represented as:

$$
\begin{aligned}
1 \times 2^{-1} &= 1 \times 1/2 = 1/2 = 8/16 \\
+0 \times 2^{-2} &= 0 \times 1/4 = 0 = 0 \\
+1 \times 2^{-3} &= 1 \times 1/8 = 1/8 = 2/16 \\
+1 \times 2^{-4} &= 1 \times 1/16 = 1/16 = 1/16 \\
\hline
&\quad 11/16_{10}
\end{aligned}
$$

OCTAL NUMBER SYSTEM

The octal number system uses eight discrete symbols, 0 through 7. With the base eight the positional value is:

. . .	8^5	8^4	8^3	8^2	8^1	8^0
	32,768	4,096	512	64	8	1

The octal number 513_8 represents:

$$
\begin{aligned}
5 \times 8^2 &= 5 \times 64 = 320 \\
+1 \times 8^1 &= 1 \times 8 = 8 \\
+3 \times 8^0 &= 3 \times 1 = 3 \\
\hline
&\quad 331_{10}
\end{aligned}
$$

Fractional binary numbers may be represented by using the symbols as coefficients of ascending negative powers of the base.

8^{-1}	8^{-2}	8^{-3}	8^{-4}	. . .
1/8	1/64	1/512	1/4096	

The octal number 0.4520 represents:

$$
\begin{aligned}
4 \times 8^{-1} &= 4 \times 1/8 = 4/8 = 256/512 \\
+5 \times 8^{-2} &= 5 \times 1/64 = 5/64 = 40/512 \\
+2 \times 8^{-3} &= 2 \times 1/512 = 2/512 = \underline{2/512} \\
& \qquad 298/512 = 149/256_{10}
\end{aligned}
$$

ARITHMETIC

ADDITION AND SUBTRACTION

Binary numbers are added according to the following rules:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 with a carry of 1

The addition of two binary numbers proceeds as follows (the decimal equivalents verify the result):

Augend	0111	(7)
Addend	+0100	+(4)
Partial Sum	0011	
Carry	1	
Sum	1011	(11)

Subtraction may be performed as an addition:

8 (minuend)		8 (minuend)
-6 (subtrahend)	or	+4 (10's complement or subtrahend)
2 (difference)		2 (difference - omit carry)

The second method shows subtraction performed by the "adding the complement" method. The omission of the carry in the illustration has the effect of reducing the result by 10.

One's Complement

The 924 performs all arithmetic operations in the binary one's complement mode. In this system, positive numbers are represented by the binary equivalent and negative numbers in one's complement notation.

The one's complement representation of a number is found by subtracting each bit of the number from 1. For example:

1111	
-1001	9
0110	(one's complement of 9)

This representation of a negative binary quantity may also be obtained by substituting "1's" for "0's" and "0's" for "1's".

The value zero can be represented in one's complement notation in two ways:

$0000 \rightarrow 00_2$	Positive (+) Zero
$1111 \rightarrow 11_2$	Negative (-) Zero

The rules regarding the use of these two forms for computation are:

1) Both positive and negative zero are acceptable as arithmetic operands.
2) If the result of an arithmetic operation is zero, it will be expressed as positive zero. There are two exceptions to this rule: (1) When negative zero is added to negative zero, the result is negative zero. (2) If a positive zero is subtracted from negative zero, the result will be expressed as a negative zero. This is because the 924 complements the minuend and adds.

One's complement notation applies not only to arithmetic operations performed in A, but also to the modification of execution address. During address modification, the modified address will equal 77777_8 only if the unmodified execution address equals 77777_8 and b = 0 or $(B^b) = 77777_8$.

Two's Complement

The counters in the computer use two's complement arithmetic. A counter is a register with provisions for increasing its contents by one if it is additive (P register) or decreasing its contents by one if it is subtractive (U register). A two's complement counter is open-ended; there is no end-around carry or borrow.

Positive numbers have the same representation in both systems while negative values differ by one count.

Count	2's comp. rep.	1's comp. rep.
+2	00010	00010
+1	00001	00001
0	00000	00000
-1	11111	11110
-2	11110	11101

The difference in the representation of negative values in these two systems is due to the skipping of the "all one's" count in one's complement notation. In the one's complement system the end-around-carry feature of the register automatically changes a count of all one's to all zeros. (Note exception under one's complement.)

As an example, if the content of a subtractive counter is positive seven (0111) and is to be reduced by one, add the two's complement expression of negative one, (1111), to 0111 as shown below. The result is six.

$$\begin{array}{r} 0111 \\ +1111 \\ \hline 0110 \end{array}$$

Note that the two's complement expression for a negative number may also be formed by adding one to the one's complement representation of the number.

MULTIPLICATION

Binary multiplication proceeds according to the following rules:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

Multiplication is always performed on a bit-by-bit basis. Carries do not result from multiplication, since the product of any two bits is always a single bit.

Decimal example:

multiplicand	14	
multiplier	12	
partial products	28	
	14	(shifted one place left)
product	168_{10}	

The shift of the second partial product is a shorthand method for writing the true value 140.

Binary example:

multiplicand	(14)	1110	
multiplier	(12)	1100	
partial products		0000	
		0000	shift to place digits in proper columns
		1110	
		1110	
product	(168_{10})	10101000_2	

The computer determines the running subtotal of the partial products. Rather than shifting the partial product to the left to position it correctly, the computer right shifts the summation of the partial products one place before the next addition is made. When the multiplier bit is "1", the multiplicand is added to the running total and the results are shifted to the right one place. When the multiplier bit is "0", the partial product subtotal is shifted to the right (in effect, the quantity has been multiplied by 10_2).

DIVISION

The following example shows the familiar method of decimal division:

		14	quotient
divisor	13	185	dividend
		13	
		55	partial dividend
		52	
		3	remainder

The computer performs division in a similar manner (using binary equivalents):

```
                          1110      quotient  (14)
divisor             1101 |10111001  dividend
                          1101
                          10100
                           1101
                            1110    partial dividends
                            1101
                              11    remainder  (3)
```

However, instead of shifting the divisor right to position it for subtraction from the partial dividend (shown above), the computer shifts the partial dividend left, accomplishing the same purpose and permitting the arithmetic to be performed in the A register. The computer counts the number of shifts, which is the number of quotient digits to be obtained; after the correct number of counts, the routine is terminated.

CONVERSIONS

The procedures that may be used when converting from one number system to another are power addition, double dabble, and substitution.

Recommended Conversion Procedures (Integer and Fractional)

Conversion	Recommended Method
Binary to Decimal	Power Addition
Octal to Decimal	Power Addition
Decimal to Binary	Double Dabble
Decimal to Octal	Double Dabble
Binary to Octal	Substitution
Octal to Binary	Substitution

GENERAL RULES

$r_i > r_f$: use Double Dabble, Substitution

$r_i < r_f$: use Power Addition, Substitution

r_i = Radix of initial system

r_f = Radix of final system

POWER ADDITION

To convert a number from r_i to r_f ($r_i < r_f$) write the number in its expanded r_i polynominal form and simplify using r_f arithmetic.

EXAMPLE 1 Binary to Decimal (Integer)

$$
\begin{aligned}
010\ 111_2 &= 1\,(2^4) + 0\,(2^3) + 1\,(2^2) + 1\,(2^1) + 1\,(2^0) \\
&= 1\,(16) + 0\,(8) + 1\,(4) + 1\,(2) + 1\,(1) \\
&= 16 + 0 + 4 + 2 + 1 \\
&= 23_{10}
\end{aligned}
$$

EXAMPLE 2 Binary to Decimal (Fractional)

$$
\begin{aligned}
.0101_2 &= 0\,(2^{-1}) + 1\,(2^{-2}) + 0\,(2^{-3}) + 1\,(2^{-4}) \\
&= 0 + 1/4 + 0 + 1/16 \\
&= 5/16_{10}
\end{aligned}
$$

EXAMPLE 3 Octal to Decimal (Integer)

$$
\begin{aligned}
324_8 &= 3\,(8^2) + 2\,(8^1) + 4\,(8^0) \\
&= 3\,(64) + 2\,(8) + 4\,(1) \\
&= 192 + 16 + 4 \\
&= 212_{10}
\end{aligned}
$$

EXAMPLE 4 Octal to Decimal (Fractional)

$$
\begin{aligned}
.44_8 &= 4\,(8^{-1}) + 4\,(8^{-2}) \\
&= 4/8 + 4/64 \\
&= 36/64_{10}
\end{aligned}
$$

DOUBLE DABBLE

To convert a whole number from r_i to r_f ($r_i > r_f$):

1) Divide r_i by r_f using r_i arithmetic
2) The remainder is the lowest order bit in the new expression
3) Divide the integral part from the previous operation by r_f
4) The remainder is the next higher order bit in the new expression
5) The process continues until the division produces only a remainder which will be the highest order bit in the r_f expression.

To convert a fractional number from r_i to r_f:

1) Multiply r_i by r_f using r_i arithmetic
2) The integral part is the highest order bit in the new expression
3) Multiply the fractional part from the previous operation by r_f
4) The integral part is the next lower order bit in the new expression
5) The process continues until sufficient precision is achieved or the process terminates.

EXAMPLE 1 Decimal to Binary (Integer)

45 ÷ 2 = 22 remainder 1; record	1
22 ÷ 2 = 11 remainder 0; record	0
11 ÷ 2 = 5 remainder 1; record	1
5 ÷ 2 = 2 remainder 1; record	1
2 ÷ 2 = 1 remainder 0; record	0
1 ÷ 2 = 0 remainder 1; record	1
	101101

Thus: $45_{10} = 101101_2$

EXAMPLE 2 Decimal to Binary (Fractional)

.25 x 2 = 0.5; record	0
.5 x 2 = 1.0; record	1
.0 x 2 = 0.0; record	0
	.010

Thus: $.25_{10} = .010_2$

EXAMPLE 3 Decimal to Octal (Integer)

273 ÷ 8 = 34 remainder 1; record	1
34 ÷ 8 = 4 remainder 2; record	2
4 ÷ 8 = 0 remainder 4; record	4
	421

Thus: $273_{10} = 421_8$

EXAMPLE 4 Decimal to Octal (Fractional)

.55 x 8 = 4.4; record	4
.4 x 8 = 3.2; record	3
.2 x 8 = 1.6; record	1
-- --	-
-- --	-
	.431...

Thus: $.55_{10} = .431\ldots_8$

SUBSTITUTION

This method permits easy conversion between octal and binary representations of a number. If a number in binary notation is partitioned into triplets to the right and left of the binary point, each triplet may be converted into an octal digit. Similarly each octal digit may be converted into a triplet of binary digits.

EXAMPLE 1 Binary to Octal

Binary = 110 000 . 001 010
Octal = 6 0 . 1 2

EXAMPLE 2 Octal to Binary

Octal = 6 5 0 . 2 2 7
Binary = 110 101 000 . 010 010 111

COMMON PURE NOTATIONS

Decimal	Binary	Octal
00	00000	00
01	00001	01
02	00010	02
03	00011	03
04	00100	04
05	00101	05
06	00110	06
07	00111	07
08	01000	10
09	01001	11
10	01010	12
11	01011	13
12	01100	14
13	01101	15
14	01110	16
15	01111	17
16	10000	20
17	10001	21

POWERS OF COMMON NUMBER SYSTEMS

$2^0 = 1$	$8^0 = 1$	$10^0 = 1$
$2^1 = 2$	$8^1 = 8$	$10^1 = 10$
$2^2 = 4$	$8^2 = 64$	$10^2 = 100$
$2^3 = 8$	$8^3 = 512$	$10^3 = 1,000$
$2^4 = 16$	$8^4 = 4,096$	$10^4 = 10,000$
$2^5 = 32$	$8^5 = 32,768$	$10^5 = 100,000$
$2^6 = 64$	$8^6 = 262,144$	$10^6 = 1,000,000$
$2^7 = 128$	$8^7 = 2,097,152$	
$2^8 = 256$	$8^8 = 16,777,216$	
$2^9 = 512$		
$2^{10} = 1,024$		

GLOSSARY

ABSOLUTE ADDRESS	A specific storage location; contrast with relative address.
ACCESS TIME	The time from request to delivery of data from storage (1.8 usec.).
ACCUMULATOR	A register with provisions for the addition of another quantity to its content. It is also the name of the A register.
ADDER	A device capable of forming the sum of two or more quantities.
ADDRESS	A 15-bit quantity which identifies a particular storage location.
ALPHABETIC CODING	A system of abbreviation used in preparing information for input into a computer, e.g., Q Right Shift would be QRS.
AND FUNCTION	A logical function in Boolean algebra that is satisfied (has the value "1") only when all of its terms are "1's". For any other combination of values it is not satisfied and its value is "0".
A REGISTER	Principal arithmetic register; operates as a 24-bit subtractive accumulator (modulus: $2^{24}-1$).
BASE	A quantity which defines some system of representing numbers by positional notation; radix.
BIT	Binary digit, either "1" or "0".
BLOCK	A group of words transported in and out of storage as a unit.
BOOTSTRAP	The coded instructions at the beginning of an input tape, together with the manually entered instructions.
BORROW	In a subtractive counter or accumulator, a signal indicating that in stage n, a "1" was subtracted from a "0". The signal is sent to stage n+1 which it complements.
BRANCH	A conditional jump.

B^1-B^6 REGISTERS	Index registers used primarily for modification of execution address.
BUFFER	A device in which data is stored temporarily in the course of transmission from one point to another. To store data temporarily. The operation in which either a word from storage is sent to an external equipment via an output channel (output buffer), or a word is sent from an external equipment to storage via an input channel (input buffer).
CAPACITY	The upper and lower limits of the numbers which may be processed in a register or the quantity of information which may be stored in a storage unit. If the capacity of a register is exceeded, an overflow is generated.
CARRY	In an additive counter or accumulator, a signal indicating that in stage n, a "1" was added to a "1". The signal is sent to stage n+1, which it complements.
CHANNEL	A transmission path that connects the computer to an external equipment.
CHARACTER	Two types of information handled by the computer: 1) A group of 6 bits which represents a digit, letter or symbol. In the assembly mode, four 6-bit characters make up a computer word. 2) A group of 7 bits which represents an item of information. In the character mode, this item is one 7-bit character and "0's" in the remaining (upper) 17 bits.
CLEAR	A command that removes a quantity from a register by placing every stage of the register in the "0" state.

COMMAND	A signal that performs a unit operation, such as transmitting the content of one register to another, shifting a register one place to the left or setting a FF.
COMPILER	A routine which automatically produces a specific program for a particular problem. The routine determines the meaning of information expressed in a psuedo-code, selects or generates the required subroutine, transforms the subroutine into specific coding, assigns storage registers, and enters the information as an element of the problem program.
COMPLEMENT	Noun: see One's Complement or Two's Complement. Verb: a command which produces the one's complement of a given quantity.
CONTENT	The quantity or word held in a register or storage location.
CORE	A ferromagnetic toroid used as the bistable device for storing a bit in a memory plane.
COUNTER	A register with provisions for increasing or decreasing its content by 1.
CYCLE TIME	The time required for a complete storage reference (6.4 usec.).
EVEN STORAGE	The storage unit which contains the 4096 even addresses.
EXCLUSIVE OR	A logical function in Boolean algebra that is satisfied when any but not all of its terms are "1". It is not satisfied when all terms are "0".
EXECUTION ADDRESS	The lower 15 bits of a 24-bit instruction. Most often used to specify the storage address of an operand. Sometimes used as the operand.
EXIT	Execute next instruction.

EXTERNAL FUNCTION
: 1) External Function Select (74.0) sends a code to an external equipment to direct its operation.
2) External Function Sense (74.7) sends a code to an external equipment to sense its operating condition.

FAULT
: Operational difficulty which stops operation or sets an indicator.

FIXED POINT
: A notation or system of arithmetic in which all numerical quantities are expressed by a predetermined number of digits with the binary point implicitly located at some pre-determined position.

FLIP-FLOP (FF)
: A bistable storage device. A "1" input to the set side puts the FF in the "1" state; a "1" input to the clear side puts the FF in the "0" state. The FF remains in a state indicative of its last "1" input. A stage of a register consists of a FF.

FUNCTION CODE
: The upper 9 bits of a 24-bit instruction consisting of the operation and index codes.

INCLUSIVE OR
: A logical function in Boolean algebra that is satisfied when any of its terms are "1". It is <u>not</u> satisfied when all terms are "0".

INDEX CODE
: A 3-bit quantity, bits 15, 16, and 17 of an instruction; usually specifies an index register whose contents are added to the execution address; sometimes specifies the conditions for executing the instruction.

INSTRUCTION
: A 24-bit quantity consisting of a function code, execution address, and index designator.

INTERRUPT REQUEST
: A signal received from an external equipment that may cause a special sequence of instructions to be executed.

INVERTER
: A circuit which provides as an output a signal that is opposite to its input. An inverter output is "1" only if all the separate OR inputs are "0".

JUMP
: An instruction which alters the normal sequence control of the computer and, conditionally or unconditionally, specifies the location of the next instruction.

LOAD To place a quantity from storage in a register.

LOCATION A storage position holding one computer word, usually designated by a specific address.

LOGICAL PRODUCT In Boolean algebra, the AND function of several terms. The product is "1" only when all the terms are "1"; otherwise it is "0". Sometimes referred to as the result of "bit-by-bit" multiplication.

LOOP Repetition of a group of instructions in a routine.

MASK In the formation of the logical products of two quantities, one quantity may mask the other, i.e., determine what part of the other quantity is to be considered. If the mask is "0" that part of the other quantity is cleared; if the mask is a "1", the other quantity is left unaltered.

MASTER CLEAR (MC) A general command produced by placing the CLEAR switch up (external MC) or down (computer MC) which clears all the crucial registers and control FFs.

MNEMONIC CODE A three-letter code which represents the function or purpose of an instruction. Also called Alphabetic Code.

MODULUS An integer which describes certain arithmetic characteristics of registers, especially counters and accumulators, within a digital computer. The modulus of a device is defined by r^n for an open-ended device and r^n-1 for a closed (end-around) device, where r is the base of the number system used and n is the number of digit positions (stages) in the device. Generally, devices with modulus r^n use two's complement arithmetic; devices with modulus r^n-1 use one's complement.

NORMAL JUMP An instruction that jumps from one sequence of instructions to a second, and makes no preparation for returning to the first sequence.

NUMERIC CODING	A system of abbreviation in which all information is reduced to numerical quantities.
ODD STORAGE	The storage unit which contains the 4096 odd addresses.
ONE'S COMPLEMENT	With reference to a binary number, that number which results from subtracting each bit of the given number from "1". The one's complement of a number is formed by complementing each bit of it individually, that is, changing a "1" to "0" and a "0" to a "1". A negative number is expressed by the one's complement of the corresponding positive number.
ON-LINE OPERATION	A type of system application in which the input data to the system is fed directly from the external equipment to the computer.
OPERAND	Usually refers to the quantity specified by the execution address. This quantity is operated upon in the execution of the instruction.
OPERATIONAL REGISTERS	Registers which are displayed on the operator's console (B^1-B^6, A, Q, P, U).
OPERATION CODE	The upper 6 bits of a 24-bit instruction which identify the instruction. After the code is translated, it conditions the computer for execution of the specified instruction. This code, which is expressed by two octal digits, is designated by the letter f.
O^2, O^4, O^6 REGISTERS	Output registers used for output buffer operations.
OR FUNCTION	A logical function in Boolean algebra that is satisfied (has the value "1") when any of its terms are "1". It is not satisfied when all terms are "0". Often called the 'inclusive' OR function.
OVERFLOW	The capacity of a register is exceeded.
PARITY CHECK	A summation check in which the binary digits in a character are added and the sum checked against a previously computed parity digit; i.e., a check which tests whether the number of ones is odd or even.

PARTIAL ADD	An addition without carries. Accomplished by toggling each bit of the augend where the corresponding bit of the addend is a "1".
P REGISTER	The Program Address Counter is a two's complement additive register (Modulus 2^{15}) which generates in sequential order the storage addresses containing the individual program steps.
PROGRAM	A precise sequence of instructions that accomplishes a computer routine; a plan for the solution of a problem.
PROGRAM STEP	A 24-bit instruction contained in a 24-bit storage address; such an instruction is read from storage and executed.
Q REGISTER	Auxiliary arithmetic register which assists the A register in the more complicated arithmetic operations (Modulus: $2^{24}-1$).
RANDOM ACCESS	Access to storage under conditions in which the next position from which information is to be obtained is in no way dependent on the previous one.
READ	To remove a quantity from a storage location.
READY	1) The input-output control signal sent by the computer or an external equipment. The ready signal indicates that a word or character is available for transmission. 2) A status reponse indicating that the external device being addressed is ready for operation.
RELATIVE ADDRESS	Identifies a word in a subroutine or routine with respect to its position. Relative addresses are translated into absolute addresses by the addition of some specific reference address, usually that at which the first word of the routine is stored.
REPLACE	In the title of an instruction, the result of the execution of the instruction is stored in the location from which the initial operand was obtained.

RESUME	The input-output control signal sent by either the computer or an external equipment to indicate that it is prepared to receive another word (24 bits) or character (usually 6 bits). The resume signal is thus a request for data.
RETURN JUMP	An instruction that jumps from a sequence of instructions to initiate a second sequence and prepares for continuing the first sequence after the second is completed.
ROUTINE	The sequence of operations which the computer performs under the direction of a program.
S^1 REGISTER	Storage Address register (even storage). Selects the storage address specified by the contents of the P register.
S^2 REGISTER	Storage Address register (odd storage). Selects the storage address specified by the contents of the P register.
SCALE FACTOR	One or more coefficients by which quantities are multiplied or divided so that they lie in a given range of magnitude.
SCANNER	That portion of the computer which automatically samples the state of the buffer channels and interrupt line, and initiates action in accordance with the information obtained.
SECONDARY REGISTERS	Transient registers not displayed on the console (U^2, $S^{1,2}$, $Z^{1,2}$, X, $O^{2,4,6}$.
SHIFT	To move the bits of a quantity right or left.
SIGN BIT	In registers where a quantity is treated as signed by use of one's complement notation, the bit in tne highest-order stage of the register. If the bit is "1", the quantity is negative; if the bit is "0", the quantity is positive.
SIGN EXTENSION	The duplication of the sign bit in the higher-order stages of a register.

SKIP EXIT	Execute instruction at (P) + 2.
STAGE	The FFs and inverters associated with a bit position of a register.
STORE	To transmit information to a device from which the unaltered information can later be obtained.
TOGGLE	To complement each bit of a quantity as a result of an individual condition.
TRANSLATION	An indication of the content of a group of bit registers. A complete translation gives the exact content; a partial translation indicates only that the content is within certain limits.
TRANSMISSION, FORCED	A transfer of bits into a register which has not been cleared previously.
TWO'S COMPLEMENT	Number that results from subtracting each bit of a number from "0". The two's complement may be formed by complementing each bit of the given number and then adding one to the result, performing the required carries.
U REGISTER	Program Control register. A 24-bit register that holds a program step while the instruction contained in it is being executed. The lower 15 bits make up a subtractive accumulator (Modulus 2^{15}), used primarily as a counter in shift and search instructions.
WORD	A unit of information which has been coded for use in the computer as a series of bits. The normal word length is 24 bits.
WRITE	To enter a quantity into a storage location.

X REGISTER	Exchange register. Most internal transmissions between the arithmetic section and the rest of the computer are made through X. Output transmissions go through X.
Z^1 REGISTER	Storage Restoration register (even storage). Holds the word to be written into a given storage location.
Z^2 REGISTER	Storage Restoration register (odd storage). Holds the word to be written into a given storage location.

924 INSTRUCTIONS

			Page				Page
01	ARS	A Right Shift	2-26	41	SCL	Selective Clear	2-23
02	QRS	Q Right Shift	2-26	42	SCM	Selective Complement	2-22
03	LRS	AQ Right Shift	2-26	43	SSU	Selective Substitute	2-24
04	ENQ	Enter Q	2-15	44	LDL	Load Logical	2-24
05	ALS	A Left Shift	2-27	45	ADL	Add Logical	2-24
06	QLS	Q Left Shift	2-27	46	SBL	Subtract Logical	2-24
07	LLS	AQ Left Shift	2-26	47	STL	Store Logical	2-24
10	ENA	Enter A	2-15	50	ENI	Enter Index	2-16
11	INA	Increase A	2-21	51	INI	Increase Index	2-21
12	LDA	Load A	2-11	52 0	XAQ	Interchange A and Q	2-12
13	LAC	Load A Complement	2-11	52 1	CMA	Complement A	2-25
14	ADD	Add	2-17	52 2	CMQ	Complement Q	2-25
15	SUB	Subtract	2-17	53	LIL	Load Index	2-13
16	LDQ	Load Q	2-11	54	ISK	Index Skip	2-31
17	LQC	Load Q Complement	2-11	55	IJP	Index Jump	2-38
20	STA	Store A	2-12	56	INQ	Increase Q	2-21
21	STQ	Store Q	2-12	57	SIL	Store Index	2-13
22	AJP	A Jump	2-36, 40	60	XEC	Execute	2-41
23	QJP	Q Jump	2-37, 40	61	SAL	Substitute Address	2-13
24	MUI	Multiply	2-18	62	TAL	Tally	2-20
25	DVI	Divide	2-19	63	PTS	Pattern Search	2-34
26	ATI	A to Index	2-16	64	EQS	Equality Search	2-33
27	QTI	Q to Index	2-16	65	THS	Threshold Search	2-33
30	SKH	Skip High	2-31	66	MEQ	Masked Equality	2-34
31	SKL	Skip Low	2-31	67	MTH	Masked Threshold	2-34
32	UJP	Unconditional Jump	2-37	70	RAD	Replace Add	2-29
33	URJ	Unconditional Return Jump	2-41	71	RSB	Replace Subtract	2-29
34	SCA	Scale A	2-28	72	RAO	Replace Add One	2-29
35	SCQ	Scale AQ	2-28	73	RSO	Replace Subtract One	2-29
36	SSK	Storage Skip	2-30	74	EXF	External Function	3-3
37	SSH	Storage Shift	2-30	75	SLJ	Selective Jump	2-37, 40
40	SST	Selective Set	2-22	76	SLS	Selective Stop	2-37, 40

CONTROL DATA

CORPORATION

501 PARK AVENUE, MINNEAPOLIS 15, MINNESOTA • FEDERAL 9-0411