PM 06411
JUNE 1964

THE BUNKER-RAMO CORPORATION

340

2185

PROGRAMMING MANUAL

THE BUNKER-RAMO CORPORATION

340

PROGRAMMING MANUAL

PM 06411
JUNE 1964

FOR ADDITIONAL INFORMATION, OR FOR THE ADDRESS OF YOUR NEAREST BUNKER-RAMO OFFICE, CONTACT:

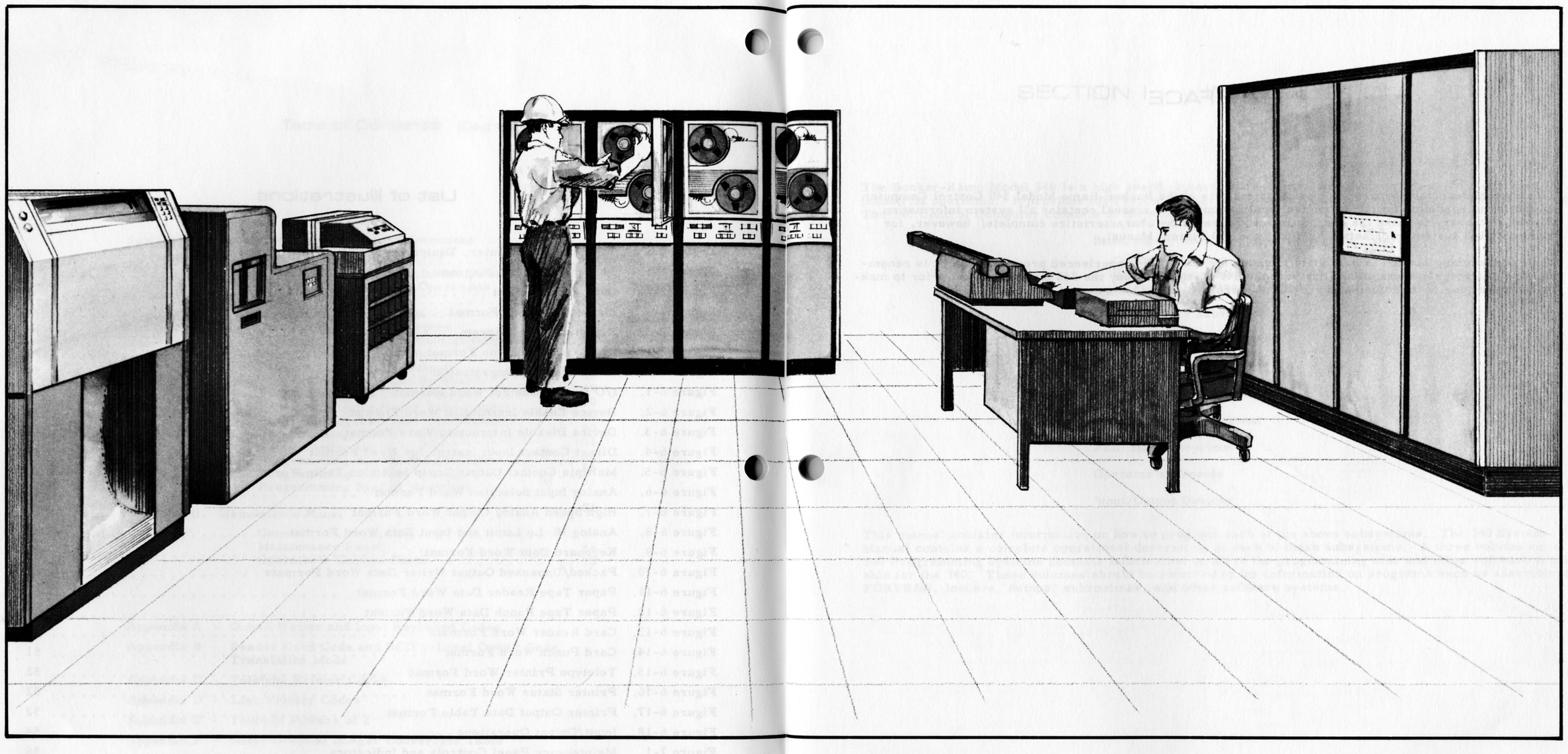
INFORMATION SERVICES
PROCESS CONTROL SYSTEMS
THE BUNKER-RAMO CORPORATION
8433 FALLBROOK AVENUE
CANOGA PARK, CALIFORNIA 91304
TELEPHONE 213-346-6000

The Bunker-Ramo Corporation reserves the right to modify this equipment in order to improve its performance and operational flexibility.

Table of Contents

Table of Contents (Cont'd)

List of Illustrations



Typical 340 Computer, Equipment and Programming Room

PREFACE

This manual describes programming characteristics of the Bunker-Ramo Model 340 Control Computer as applied to real time industrial control applications. The manual contains all system information deemed necessary to make the description of programming characteristics complete; however, for more detailed system information, refer to the 340 System Manual.

The Programming Manual is primarily intended for use by the experienced programmer. It is recommended that programmers unfamiliar with the 340 Computer study the 340 System Manual prior to making practical use of the information contained herein.

SECTION I INTRODUCTION

The Bunker-Ramo Model 340 is a high speed, binary, core-drum control computer system designed for industrial operation. It provides high performance, modular design, and availability of over 99 percent. The 340 Control Computer consists of the subsystems shown in Figure 1-1 and listed below.

Basic Model 340 Computer

- Central Processor
- Core Memory Unit
- Master I/O Control and Interrupt Unit

Drum Memory Subsystem

Contact Input Subsystem

Contact Output Subsystem

Analog Input Subsystem

Analog Output Subsystem

Fail-Safe Subsystem

Operator's Console

Input/Output Devices

This manual contains information on how to program each of the above subsystems. The 340 System Manual contains a complete operational description of each of these subsystems. A three volume set of 340 Programming Systems contains information on all of the programming aids and other software available for the 340. These volumes should be referred to for information on programs such as assemblers, FORTRAN, loaders, dumps, subroutines, and other software systems.

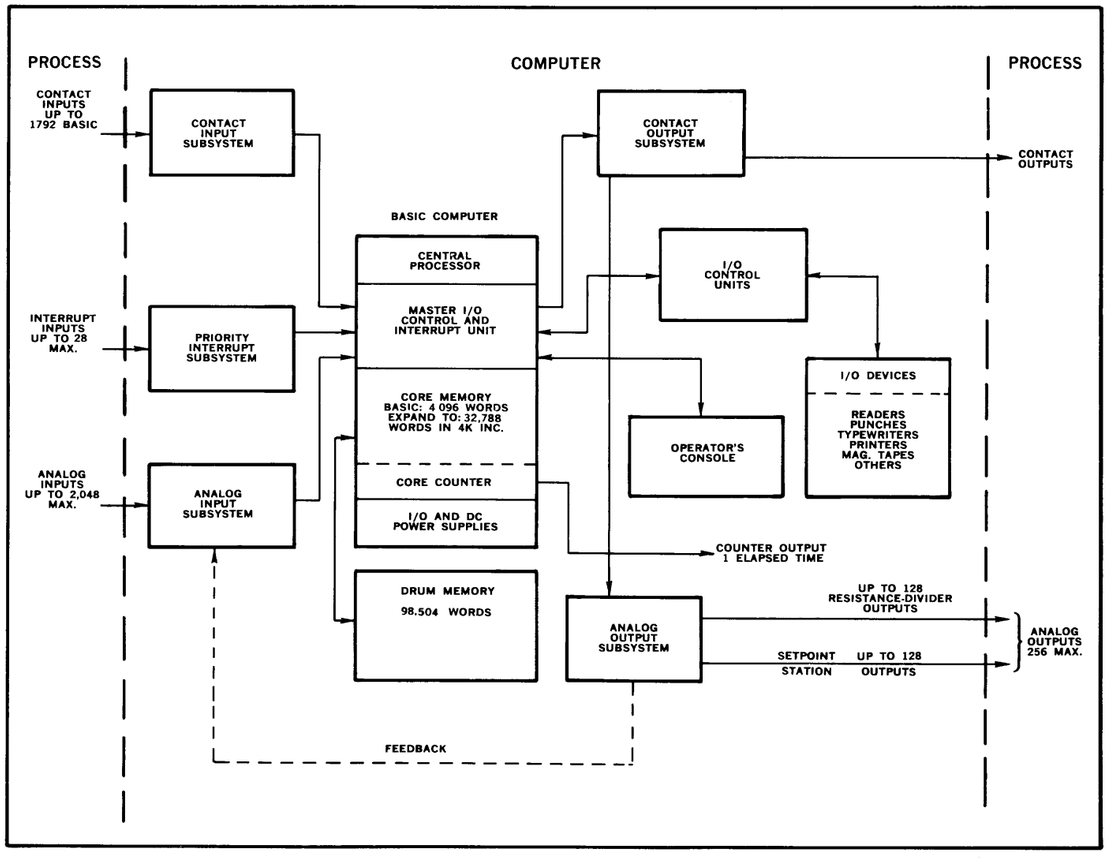


Figure 1-1. 340 System Block Diagram

SECTION II
GENERAL PROGRAMMING CONSIDERATIONS

WORD FORMATS

All the information used by a 340 program is represented as configurations of 28 binary digits (bits). The computer interprets each group of bits either as a 28-bit instruction word, or a 28-bit data word, depending on the use of the word in the program.

A. DATA WORD FORMAT (Figure 2-1)

In the Data Word, the first 27 bits specify the magnitude of the number. The 28th bit expresses the sign of the number where 0 = positive a number; and 1 = negative a number.

A 29th bit is included with each computer word to be used by the hardware to record and check parity. The 340 uses an odd-bit parity system. As information is stored in core memory, the circuitry checks the number of 1 bits in the word. If the word contains an even number of 1 bits, bit 29 is set = 1. If the number of 1 bits is odd the parity bit is set = 0. When either an instruction or data word is read from memory it is again checked for odd-bit parity. One of four parity indicators is set, if a parity failure is detected.

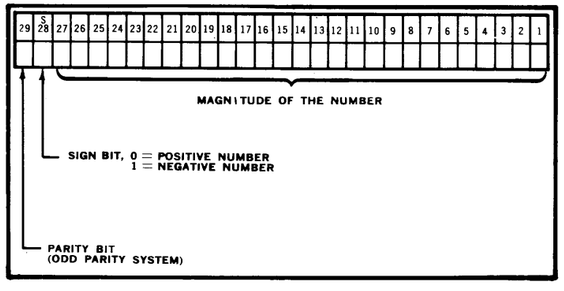


Figure 2-1. Data Word Format

B. INTERPRETING DATA WORDS

A 28-bit binary number can be difficult to interpret and convert to a more meaningful decimal equivalent. However, the relationship between the binary and octal number systems makes it convenient to express the binary number as an octal equivalent.

To convert a binary word to its 10-digit octal equivalent, do the following:

1. Group the bits in sets of 3, starting with the least significant bit.
2. Next, convert each set of 3 bits to its octal equivalent, using the following chart:

Binary	=	Octal
000		0
001		1
010		2
011		3
100		4
101		5
110		6
111		7

3. Example:

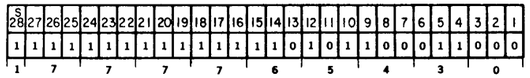
Binary	0	101	111	100	110	011	101	010	011	100
Octal	0	5	7	4	6	3	5	2	3	4

C. NEGATIVE NUMBERS

In the 340, negative numbers are expressed in 2's complement form. To convert any positive magnitude to the 2's complement notation (or any negative magnitude to its 2's complement form):

1. Change all 1 bits to 0 bits, and all 0 bits to 1 bits, including the sign bit. (This number is the 1's complement.)
2. Add 1 to the least significant bit position. The result is the 2's complement of the original number.

   For example, find the 2's complement of the octal number 0000012350.



The two numbers 0000012350 and 1777765430 have the same magnitude although their signs are different.

A simplified method for converting octal numbers to their 2's complement form involves:

1. Change the sign of the number.
2. Subtract the least significant non-zero octal digit from 8 and the remaining digits from 7.
3. For example, given the number 0015700346:

-0015700346	Base Number
1762077432	2's complement
0000000000	Check sum zero

If the complementing has been done correctly, the octal sum of the base number and its 2's complement is zero.

D. INSTRUCTION WORD FORMAT (Figure 2-2)

An instruction word is divided into two general fields: the Operation Code Field and the Operand Field. The figure below illustrates the instruction layout.

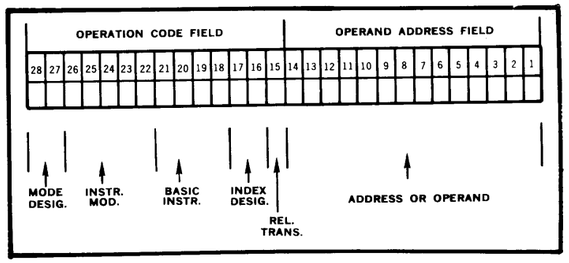


Figure 2-2. Instruction Word Format

1. Operand Field

The Operand Field uses bits 1 through 14 of the instruction word to express either an Operand Address or, in some instructions, the actual operand. In either use, the maximum Operand Field expressed (in octal) is 37777.

2. Operation Field

The Operation code, its mode indicators and modifiers, plus other designators are given in the 14-bit Operation Field (bits 15 through 28). Bit 15 is the Relative Transfer Modifier flag bit. Bits 16 and 17 designate one of the three index registers. Bits 18-21 specify the basic operation code (Load, Multiply, Store, etc.). Bits 22-26 indicate various basic operation code modifier options. The function of these bits varies with the basic operation involved and will be discussed in detail in the description of the operations in Section III. Bits 27 and 28 select the operating mode.

REGISTERS AND INDICATORS

The 340 has 10 standard program registers and 10 control registers. The information in the program registers is available either automatically under program control or manually from the indicators on the Maintenance Panel. Seven of the control registers are accessible from the indicators on the Maintenance Panel. Systems using Priority Interrupts will have two additional registers.

Register flow is shown in block form in Figure 2-3.

A. PROGRAM REGISTERS

A Register

The A Register or Accumulator, is the principal arithmetic register in the computer. It is a 28-bit signed register used with arithmetic, shift, logical, jump, extract, and merge operations. In addition, the contents of A may be stored into or loaded from core, and also may be exchanged with other registers. The overflow and carry indicators are set as a result of operations involving the A Register.

B Register

The B Register is the secondary arithmetic register, acting in some operations as the A Register extension. B is a 28-bit, signed register used in multiplication to hold the least significant bits of the product, and in divide to hold the remainder. B may be loaded from or stored into memory, exchanged with A, have its contents shifted left or right in conjunction with the contents of A, and hold the least significant half of the dividend on a special mode divide. The B register also provides convenient temporary storage for intermediate answers. In addition, the B Register may be used for masking operations.

C Register

The C Register is a 28-bit signed register used for temporary storage, and with multiply/divide instructions. During multiply the multiplicand is transferred into C, and during divide the divisor is held in C. The contents of A and C may be exchanged, and C may be loaded from or stored into core. The C Register is also used in some replace, search, compare, arithmetic and logical operations.

G Register

The G Register is the Relativization Base Address register used in core/drum and drum/core transfers to automatically modify the Operand Field of tagged instructions. G is a 14-bit unsigned register. A may be replaced by G or G by A.

I_i Register

There are three I or Index Registers available in the 340. These are 14-bit unsigned registers used to modify the Operand Field during the execution of any "indexed" instruction. Any of the I Registers may be loaded from and stored into core memory, exchanged with A, and decremented by any amount less than 16, 384.

X Register

The X Register is a 14-bit unsigned register used primarily as an instruction execution control counter. X is used in shift operations to control the number of places shifted, in the multiply and divide operations to control the number of multiplier bits used or the number of quotient bits generated, and in search operations to define the maximum number of items searched. X may be loaded from or stored into core and exchanged with A. In operation extension operations, X is used for operand address linkage.

M Register

The M Register is a 28-bit register that controls interrupt priority by masking the interrupt lines which will be allowed to cause program interrupts. The M Register may be loaded from or stored into core and exchanged with A.

Q Register

The Q Register is used to record the type-2 (momentary) interrupt signals. Each type-2 interrupt has one Q-Register bit position assigned. The basic minimum 10-bit Q may be extended to a 24-bit unsigned Q Register. The Q Register may be loaded from or stored into core, and exchanged with A. In addition to any Q Register bits used with the interrupt system, bits 4 through 7 are always reserved for the special core memory extension control.

B. CONTROL REGISTERS

N Register

The N Register is a 14-bit unsigned register used to designate the next instruction address. N is not available to the programmer except through the maintenance panel indicators. N is set and changed automatically by the computer.

R Register

The R Register is a 14-bit register used to hold the core address of information transferred to and from memory.

F Register

The F Register is a 28-bit intermediate storage register used to hold all information transferred to and from memory.

FC Register

The FC Register contains the upper 14 bits of the instruction just executed. The FC Register is displayed in the lower neons on the Maintenance Panel.

V Register

The V Register is a 14-bit register used to hold the starting core address for core/drum or drum/core transfers.

Z Register

The Z Register is a 14-bit register used to indicate the block length of the information being transferred between drum and core.

S Register

The S Register is a 7-bit register used to hold the current drum pickup or store sector address for drum-core transfers.

T Register

The T Register is a 10-bit register used to hold the current drum track address on drum/core transfers.

U Register

The U Register is a 28-bit register used for intermediate storage of the word being transferred in a drum/core transfer.

Y Register

The Y Register is a 14-bit register used by the scan and Compare Table operations for intermediate address storage.

J Register

This register holds each control word. (28 bits).

K Register

A 28 bit register used to hold data being output, group select table address, and results of comparisons.

L Register

This register is a device function code counter which stops on an enabled device code. (7 bits).

LS Register

This is a 7-bit register which holds I/O device select codes.

C. INDICATORS

Carry Indicator

The Carry Indicator is a flip-flop which is set on addition or subtraction whenever there is a Carry out of bit position 28 of the A Register. The Carry Indicator status may be interrogated and reset by the Branch Carry command, or by the Reset command.

Overflow Indicator

The Overflow Indicator is a flip-flop which is set whenever the magnitude of the result of an operation exceeds the capacity of the A Register. Overflow may occur on add, subtract, divide, and shift instructions. The status of the overflow indicator may be interrogated and reset by the Branch on Overflow command, or by the Reset command.

Core Parity

Two Core Parity indicators are provided to record parity errors detected during either the transfer of an instruction or the transfer of an operand. The status of the Core Parity indicators may be interrogated and reset by the Branch on Core Parity operation. Parity errors also generate a high priority interrupt.

Drum Parity

The Drum Parity Indicator is a flip-flop which is set when a parity error is detected in the word being transferred from drum to core. The Drum Parity indicator may be interrogated and reset using the Branch on Drum Parity or Direct Access Parity Error operation. A drum parity error also generates a high priority interrupt signal.

Direct Access Parity

The Direct Access Indicator (CS) is a flip-flop which is set when a parity error is detected during the direct access transfer of information from core to the enabled device. The indicator may be interrogated under program control using the Branch on Drum Parity or Direct Access Parity Error operation.

Inhibit Interrupt

The Inhibit Interrupt Indicator is a flip-flop which is used to control the occurrence of interrupts in the operating program. The flip-flop may be set or reset using SET and RST operations.

I/O Transfer-Complete Indicator

The I/O Transfer-Complete Indicator is a flip-flop used by the Master I/O Control Unit to indicate that a device has finished a data transfer. This flip-flop may be controlled with the RESET operations.

NOTE: All indicators may be set or reset with the SET or RESET instruction except the I/O Transfer-Complete indicator.

PROGRAM EXECUTION MODES

A. NORMAL ADDRESS MODE (N)

The Normal Mode is the conventional direct accessing method for executing instructions. In the Normal Mode, the Operand Field of the instruction word, bits 1-14, specifies the base address of the memory location from which the Operand is obtained (or stored). If the instruction is not indexed, this base address is the Operand Address. If indexing is specified, this base address will be modified by the contents of an index register to give the Effective Operand Address (EOA). The Normal Mode of most instructions is maskable.

B. INDIRECT ADDRESS MODE (I)

In the Indirect Address Mode the Operand Field of the instruction specifies the base memory location which will contain the actual Operand Address to be used by the instruction.

The Indirect Access Mode may be indexed. If it is, the base address is modified before the Memory Access is made. Because the computer must bring the true (or direct) address from Memory before executing the instruction, six microseconds are added to the Normal Mode execution time of Indirect Mode instructions. The Indirect Access Mode of Arithmetic, Logical, Load, and Store instructions is maskable.

C. IMMEDIATE ADDRESS LOWER HALF MODE (L)

The Immediate Address Lower Half Mode is a memory and time saving feature on the 340. Arithmetic, Logical and Load instructions use the 14-bit operand field as the effective operand when this mode of operation is used. Thus the data is contained in the instruction itself. This mode requires one less memory access, 6 μseconds, than the normal mode. In executing these instructions in Lower Mode, the 14 bits of the operand field operate on the lower 14 bits of the specified register or memory location. The upper 14 bits of the operand are interpreted as zeros.

The Lower Mode may also be used on store instructions, however, the operand field is a true operand address. Lower Mode Store instructions store bits 1-14 in the specified memory location, leaving bits 15-28 unchanged.

D. IMMEDIATE ADDRESS UPPER HALF MODE (U)

This mode operates like the Lower Half Mode, except in Immediate Address Upper Half Mode the 14 bits of the Operand Field operate on the upper 14 bits of the specified register or memory location. The lower 14 bits of the register or memory location are unchanged. The 14th bit of this Operand Field affects the 28th or sign bit of register or memory location involved. The Upper Mode may also be used on Store Instructions, however, the operand field is a true operand address. Upper Mode Store instructions store bits 15-28 in the specified memory location, leaving bits 1-14 unchanged.

E. B-REGISTER MODE (B)

In the B-Register Mode the operand is taken from or stored into the B Register. The B-Register Mode is specified by the special operand address 377778. This mode requires 6 microseconds less to execute than the normal mode, because no memory access is required for the operand. This mode may be used with arithmetic, logical, load, and store instructions. It is not indexable.

F. X-REGISTER MODE (X)

The X-Register Mode is used with Branch and Shift instructions. For branch instructions the X-Mode indicates the branch address should be taken from the X Register rather than the operand field. For shift instructions, this mode indicates that the number of places to shift should be taken from the X Register rather than the operand field of the instructions.

G. H-MODE (H)

The H-Mode is available only on Divide instructions to indicate that the contents of the combined A and B Registers are divided by the Effective Operand.

H. BRANCH MODE (Br)

The Branch Mode is available on the Replace and Exchange instructions. In the Branch Mode, the next instruction address is taken from the Operand Field.

I. MASK MODIFIER (M)

A Mask Modifier bit is provided on operations such as Load, Store, Replace, Exchange, Add, Merge, Extract, Exclusive OR, Compare, Search, and Compare Tables instructions. The Modifier bit causes the Effective Operand or the designated register transfer to be masked by the contents of the B Register. Masking is equivalent to a logical "and" operation. A "one" bit will be placed in the A Register only when the corresponding bits in both the B Register and EOA are "one" bits.

J. ZERO TEST (Z)

The Zero Test Modifier operates on the following 11 instructions: Add, Half Word Add, Add to Memory, Subtract, Subtract from Memory, Merge, Extract, Exclusive OR, and Decrement Index 1, 2, and 3.

This mode extends the capability of the above instructions by providing a conditional branch to be executed based on the results of these operations.

When the result of the operation is zero, the Next Instruction Address (NIA) is the current Instruction Address (CIA) plus 2.

When the condition is not met, the NIA is the CIA plus 1.

K. INDEXING (I1, I2, I3)

Three 14 bit program controlled index registers are provided for automatic address modification. When an instruction is tagged to use one of the I Registers, the effective operand address (EOA) is the base operand address contained in the operand field less the contents of the designated I Register. This automatic address modification mode requires no additional time since the modification takes place as the instruction is accessed from core memory.

All arithmetic, logical, load, store, branch, replace, exchange, and operation extension instructions are indexable in the normal and indirect mode of operation. On the indirect address mode, indexing occurs before the indirect address is accessed.

L. RELATIVE TRANSFER

The Relative Transfer mode of operation is part of the drum-core transfer subsystem. It is explained in Section V.

CORE MEMORY ORGANIZATION

A. GENERAL

The basic 340 main memory consists of 4,096 28-bit words of magnetic core storage. Up to 7 supplementary Core Memory units of 4,096 words each may be optionally added to expand the Core Memory capacity to a maximum of 32,768 words. Four of these 4,096-word blocks of core can be accessible at any one time; special core extension control bits allow the programmer to enable additional blocks of 4,096 words. Access time for any word is 6 microseconds.

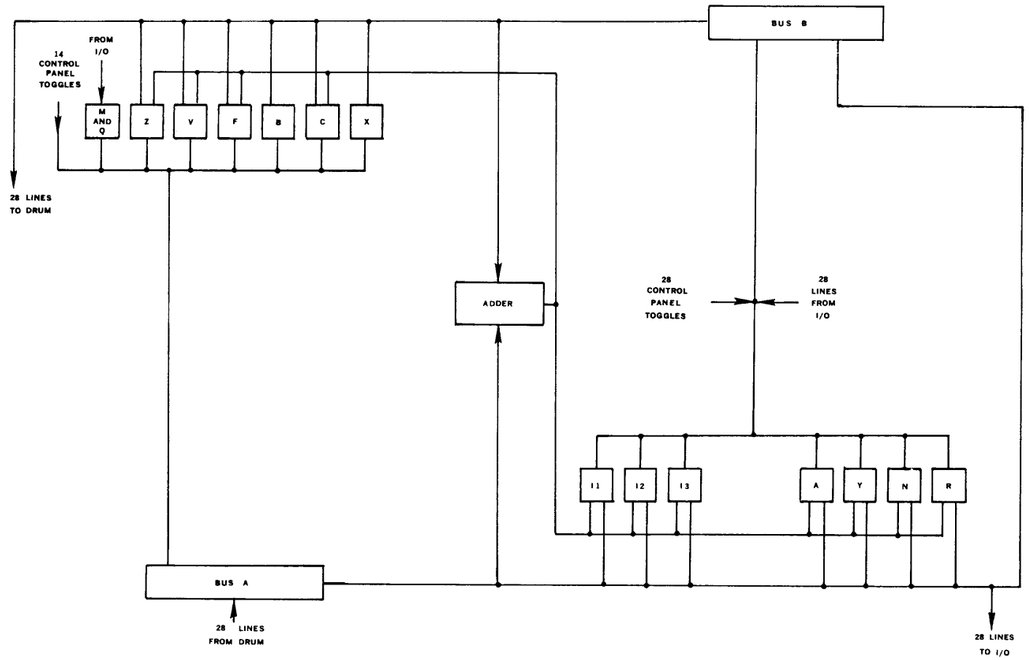


Figure 2-3. Register Flow Diagram

SECTION III CENTRAL PROCESSOR INSTRUCTIONS

GENERAL

The 340 utilizes parallel, random access, core memory with auxiliary high capacity drum bulk storage. The 340 operates under program control using 140 single address instructions which are accessed sequentially. Program operations are grouped and described below. All instructions are summarized in Appendix H using definitions given below.

OPERATION CLASSES

Machine operations are divided into the following 10 classes for ease in description of their functions.

Load/Store Operations
Arithmetic Operations
Logical Operations
Replace/Exchange Operations
Branch Operations
Shift Operations
Program Control Operations
Input/Output Operations
Drum/Core Operations
Operation Extension

Operations are summarized in Appendix I.

DEFINITIONS

The following symbols are used throughout the 340 operation descriptions as defined:

Symbol	Definition
CIA	Current Instruction Address
NIA	Next Instruction Address
OPF or OPA	Operand field — the number in the least significant 14 bits of the instruction word
EO	Effective operand
ER	Effective register
EOA	Effective operand address — the contents of the operand field or the contents as adjusted by Index and indirect modes defined by the operation mode
OEA	Operation extension address
n	Specified length of a multiply, divide, or shift operations. Specified block length of search or compare tables operations
c	Conditional number of places shifted in normalize shifts or number of words searched or scanned
⟷	Exchange
( )	The contents of the register or memory location indicated
(( ))	Contents of the contents of the register indicated
→	Transfer to, becomes, or replaces
μs	Microseconds

LOAD/STORE OPERATIONS

The B mode is specified by a normal mode code with OPF = 37777_8.

Lower Mode Loading: $(OPF)_{1-14} \rightarrow (ER)_{1-14}$ and $(ER)_{15-28} = 0$. When masking, $(OPF)_{1-14}$ masked by $(B)_{1-14} \rightarrow (ER)_{1-14}$.

Upper Mode Loading: $(OPF)_{1-14} \rightarrow (ER)_{15-28}$ and $(ER)_{1-14}$ remain unchanged. If masking, $(OPF)_{1-14}$ masked by $(B)_{15-28} \rightarrow (ER)_{15-28}$.

Lower Mode Storing: $(ER)_{1-14} \rightarrow (EOA)_{1-14}$ and $(EOA)_{15-28}$ remain unchanged. $(ER)_{1-14}$ masked by $(B)_{1-14} \rightarrow EOA_{1-14}$ when masking.

Upper Mode Storing: $(ER)_{15-28} \rightarrow (EOA)_{15-28}$ and $(EOA)_{1-14}$ remain unchanged. $(ER)_{15-28}$ masked by $(B)_{15-28} \rightarrow (EOA)_{15-28}$ when masking.

LOD A

LOAD A REGISTER

(EOA) or (OPF)→(A).
(EOA) or (OPF) are unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	000440	X	X
I	18	100440	X	X
L	6	040440	-	X
U	6	140440	-	X
B	6	0004437777	-	-

LOD B

LOAD B REGISTER

(EOA) or (OPF)→(B).
(EOA) or (OPF) are unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	016440	X	X
I	18	116440	X	X
L	6	056440	-	X
U	6	156440	-	X
Uc	6	157440	-	X

Note

Uc means load B_{15-28} and clear bits 1-14.

LOD C

LOAD C REGISTER

(EOA) or (OPF)→(C).
(EOA) or (OPF) are unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	002440	X	X
I	18	102440	X	X
L	6	042440	-	X
U	6	142440	-	X
B	6	0024437777	-	-

LOD X

LOAD X REGISTER

Bits 1-14 of (EOA) or (OPF)→bits 1-14 of (X).
(EOA) or (OPF) remain unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	006440	X	X
I	18	106440	X	X
L	6	046440	-	X
B	6	0064437777	-	-

LOD I1

LOAD INDEX REGISTER 1

Bits 1-14 of (EOA) or (OPF)→bits 1-14 of (I1).
(EOA) or (OPF) remain unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	003440	X	X
I	18	103440	X	X
L	6	043440	-	X
B	6	0034437777	-	-

LOD I2

LOAD INDEX REGISTER 2

Same as LOD I1, except bits 1-14 of (I2) are loaded

Mode	µs	Numerical Code	Indexable	Maskable
N	12	004440	X	X
I	18	104440	X	X
L	6	044440	-	X
B	6	0044437777	-	-

LOD I3

LOAD INDEX REGISTER 3

Same as LOD I1, except bits 1-14 of (I3) are loaded.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	005440	X	X
I	18	105440	X	X
L	6	045440	-	X
B	6	0054437777	-	-

LOD M

LOAD INTERRUPT MASK REGISTER M

(EOA) or (OPF)→effective bits of (M).
(EOA) or (OPF) remain unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	007440	X	X
I	18	107440	X	X
L	6	047440	-	X
U	6	147440	-	X
B	6	0074437777	-	-

LOD Q

LOAD INTERRUPT REGISTER Q

(EOA) or (OPF)→effective bits of (Q).
(EOA) or (OPF) remain unchanged.
NIA = CIA + 1.

Note

For Q Register less than 28 bits, the unused bits appear as one bits.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	013440	X	X
I	18	113440	X	X
L	6	053440	-	X
U	6	153440	-	X
B	6	0534437777	-	-

STR A

STORE A REGISTER

(A)→(EOA) and remain unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	000400	X	X
I	18	100400	X	X
L	12	040400	X	X
U	12	140400	X	X
B	6	0004037777	-	-

STR B

STORE B REGISTER

(B)→EOA and remain unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	016400	X	-
I	18	116400	X	-
L	12	056400	X	-
U	12	156400	X	-

STR C

STORE C REGISTER

(C)→(EOA) and are unchanged.
NIA = CIA + 1.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	002400	X	X
I	18	102400	X	X
L	12	042400	X	X
U	12	142400	X	X
B	6	0024037777	-	-

STR X

(X)→bits 1-14 of (EOA) and remain unchanged; Bits 15-28 of (EOA) = 0 in N, I, and B modes and remain unchanged in L mode.
NIA = CIA + 1.

STORE X REGISTER

Mode	µs	Numerical Code	Indexable	Maskable
N	12	006400	X	X
I	18	106400	X	X
L	12	046400	X	X
B	6	0064037777	-	-

STR I1

(I1)→(EOA) bits 1-14 and remain unchanged.
(EOA) bits 15-28 = 0 in N, I, and B modes and are unchanged in L mode.
NIA = CIA + 1.

STORE INDEX REGISTER 1

Mode	µs	Numerical Code	Indexable	Maskable
N	12	003400	X	X
I	18	103400	X	X
L	12	043400	X	X
B	6	0034047777	-	-

STR I2

(I2)→(EOA) bits 1-14 and remain unchanged.
(EOA) bits 15-28 = 0 in N, I, and B modes and are unchanged in L mode.
NIA = CIA + 1.

STORE INDEX REGISTER 2

Mode	µs	Numerical Code	Indexable	Maskable
N	12	004400	X	X
I	18	103400	X	X
L	12	044400	X	X
B	6	0044037777	-	-

STR I3

(I3)→(EOA) bits 1-14 and remain unchanged.
(EOA) bits 15-28 = 0 in N, I, and B modes and are unchanged in L mode.
NIA = CIA + 1.

STORE INDEX REGISTER 3

Mode	µs	Numerical Code	Indexable	Maskable
N	12	005400	X	X
I	18	105400	X	X
L	12	045400	X	X
B	6	0054037777	-	-

STR M

(M)→(EOA) effective bits and remain unchanged.
NIA = CIA + 1.

STORE INTERRUPT MASK REGISTER M

Mode	µs	Numerical Code	Indexable	Maskable
N	12	007400	X	X
I	18	107400	X	X
L	12	047400	X	X
U	12	147400	X	X
B	6	0074037777	-	-

STR Q

(Q)→(EOA) effective bits and are unchanged.
Q minimum size is 14 bits, expandable in increments of 7 bits, with unused bits stored as one bits.
NIA = CIA + 1.

STORE INTERRUPT REGISTER Q1

Mode	µs	Numerical Code	Indexable	Maskable
N	12	013400	X	X
I	18	113400	X	X
L	12	053400	X	X
U	12	153400	X	X
B	6	0134037777	-	-

STR T2

The bit configuration of the 28 data toggles on the computer control panel→(EOA) and remain unchanged.
NIA = CIA + 1.

STORE DATA TOGGLES

Mode	µs	Numerical Code	Indexable	Maskable
N	12	015400	X	X
I	18	115400	X	X
L	12	055400	X	X
U	12	155400	X	X
B	6	0154037777	-	-

ARITHMETIC OPERATIONS

MULTIPLY

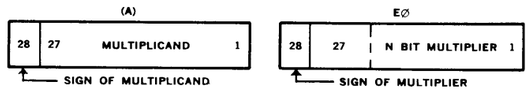
MPY n

(A) are multiplied by the n least significant bits of (EOA) or (OPF). The most significant part of the product is located in (A) and the least significant in (B). The algebraic sign of the product is in bit 28 of both (A) and (B). The length, n, of the multiplier is specified in bits 26-22 of the instruction and n may = 0 to 27.

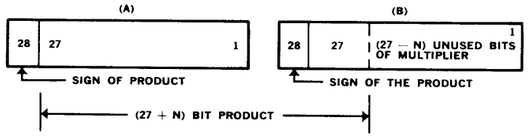
Mode	µs	Numerical Code	Indexable	Maskable
N	14+2n	000740*	X	-
I	20+2n	100740*	X	-
L	14+2n	040740*	-	-
U	14+2n	140740	-	-
B	14+2n	0007437777	-	-

*To include execution code in the op code, add n_8 at 2^{-6} to mode op code.

Before multiplication



After multiplication



Note: C Register = (A) initially and (X) = 0

Hence: $[(A)_{\text{bits } 27\text{-}1} \cdot EO_{\text{bits } n\text{-}1}]$
$\rightarrow [(A)_{27\text{-}1}, (B)_{27 \text{ through } (27\text{-}n)}]$

Where the binary scaling = (A at 2^X) (EO at 2^Y) →(A) (B) at $2^{X+Y+(27-n)}$ and no overflow or carry occurs.

If n = 0, the register status at completion of multiplication is (A) = 0, (B) = EO, (C) = (A) initial and (X) = 0.

If n = 1, a maximum of 27 bits of product (excluding sign) are developed such that the most significant digit is in bit 26 of (A) and the least significant in bit 27 of (B) — Bit 27 of (A) = sign = Bit 28 of (A). A zero product results from a multiplication where the multiplier and/or the multiplicand are zero. If either (A) or EO are full scale negative, $(1000000000)_8$, the product is a function of the positive or negative number used for the other multiplying element.
NIA = CIA + 1.

Example: A normal mode multiplication with register results shown in octal as:

MPY 7 EO Numeric op code = 007740
Execution time = 28µs

Register	before multiplication	after multiplication
A	0000001234	0000000143
EO	0000000023	unchanged
B		0120000000
C	any number	0000001234
X		0000000000

MPY 27 EO

Register	before multiplication	after multiplication
A	1777777755	1777777777
EO	0002000000	unchanged
B		1732000000
C	any numbers	1777777755
X		0000000000

DIV n

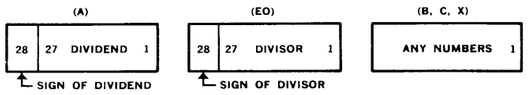
DIVIDE

Mode	μs	Numerical Code	Indexable	Maskable
N	26+2n	000700*	X	-
I	32+2n	100700*	X	-
L	26+2n	040700*	-	-
H	26+2n	140700*	X	-
B	26+2n	0007037777	-	-

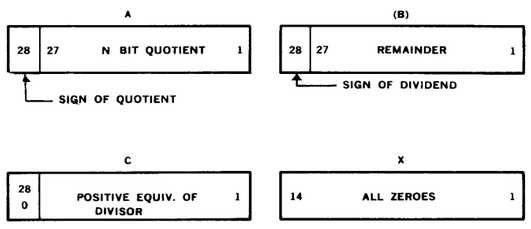
*To include execution code in the op code, add n_8 at 2^{-6} to mode op code.

(A) are divided by (EOA) or (OPF) developing an n bit quotient in the n least significant bit position of (A). The algebraic sign of the quotient is in bit 28 of (A). The remainder sign (sign of the dividend) and remainder are in (B) 28-1. The length, n, of the quotient is specified in bits 26-22 of the instruction word where n may equal 0 to 27.

Before division



After division



DIV 27 EO

Register	before division	after division
A	0000000133	0443146314
EO	0000000240	unchanged
B		0000000200
C	any numbers	Divisor
X		0000000000

DIV 27 EO

Register	before division	after division
A	0000000144	1660000000
EO	1777776600	unchanged
B		0000000000
C	any numbers	Divisor
X		0000000000

The binary scaling of the divide is (A) at 2^x/(B) $2^y \rightarrow$ (A) at $2^{x-y-(27-n)}$ and the remainder is in the B Register at 2^{x+n}. Overflow occurs if the quotient is $\geq$ (1 at 2^o). If an overflow has occurred such that only one most significant digit is lost, the actual quotient may be formed by adding ± 1 at 2^{-o} to the indicated quotient.

A zero divisor yields a quotient and remainder of zero and turns the overflow indicator on. A full scale negative divisor, $(1000000000)_8$, also sets A and B to zero and sets overflow on. Zero divided by any number other than the above two, yields a zero quotient but does not set overflow.

Special Modes:

(1) The divide operations with n = 0 produces an absolutizing function such that, in 14 μs, the (A) = | EO| and (C) = | A| initial with (B) unchanged and (X) set to zero.

(2) The H mode of the divide uses both (A) and (B) as a 54 bit plus sign dividend. This mode produces a full 27-bit plus-sign quotient without truncation and is designed specifically for the n = 27 case. If the 54-bit dividend is negative, (A) should contain the negative most significant digits and (B) should contain the positive equivalent of the least significant bits.
NIA = CIA + 1.

ADD

ADD

Mode	μs	Numerical Code	Indexable	Maskable
N	12	000640	X	X
I	18	100640	X	X
L	6	040640	-	X
U	6	140640	-	X
B	6	0006437777	-	-

Note: EO masked by (B) prior to addition. Zero test on all modes.

The effective operand is added to (A) and the algebraic sum placed in (A). Overflow will occur if the result $> (2^o - 2^{-27})$. A carry bit is generated by a carry out of bit position 28.

If masking, (EOA) or (OPF) is masked by (B) and the result added to (A). On a lower mode addition a carry bit may be generated in bit position 15 which is included in the sum.
NIA = CIA + 1.

Note

Add Lower - This operation is effective for positive operands only. Assuming that the OPA of instruction contains a negative number, the results yielded will be inconsistent with the two's complement arithmetic.

Add Upper - Bit 14 of the OPA is interpreted as the sign, therefore both positive and negative numbers will yield consistent results.

SUB

SUBTRACT

Mode	μs	Numerical Code	Indexable	Maskable
N	12	002640	X	X
I	18	102640	X	X
L	6	042640	-	X
U	6	142640	-	X
B	6	0026437777	-	-

Note: 1's complement taken of (EO) then masked by (B) prior to subtraction. Zero test on all modes.

The effective operand is subtracted from (A) and the algebraic result $\rightarrow$(A). Overflow Indicator, carry bit generation, and masking are as defined in the ADD operation. In effect, the subtraction is executed by taking the 1's complement of EO and performing the addition EO' + 1 + (A).
NIA = CIA + 1.

Note

Subtract Lower - This operation is effective for positive operands only. Assuming that the OPA of instruction contains a negative number, the results yielded will be inconsistent with the two's complement arithmetic.

Subtract Upper - Bit 14 of the OPA is interpreted as the sign, therefore both positive and negative numbers will yield consistent results.

HWA

HALF WORD ADD

Mode	μs	Numerical Code	Indexable	Maskable
N	12	001640	X	X
I	18	101640	X	X
L	6	041640	-	X
B	6	0016437777	-	-

Note: Zero test on all modes.

The algebraic sum of (A) plus (EOA) or (OPF) $\rightarrow$ (A). Addition is as described in ADD operation except no carry bit is generated between bits 14 and 15-28 (only if a carry is generated at n = 28, will the carry indicator be set). An overflow occurs only when the absolute sum of $A + EO_{\text{bits } 15\text{-}28} > (2^o - 2^{-27})$.
NIA = CIA + 1.

Note

Overflow is set only when the absolute value of the result in A exceeds $(2^o - 2^{-27})$.

ADM

ADD TO MEMORY

Mode	μs	Numerical Code	Indexable	Maskable
N	18	006640	X	X
I	24	106640	X	X

Notes: Zero test on both modes.
In masked mode EO is masked by (B) prior to addition.

The (EOA) are added algebraically to (A) and the sum is stored in (EOA) and (C). The overflow and carry bit indicators are set as for the ADD operation. (A) remain unchanged. Full scale negative added to a number ≤ 0, changes the bit in position 28 of EO and sets overflow indicator.
NIA = CIA + 1.

SBM

SUBTRACT FROM MEMORY

(A) are subtracted from (EOA) and the result stored in (EOA) and (C). (A) are unchanged. Rules for overflow and carry set are as for SUB. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	18	007640	X	X
I	24	107640	X	X

Notes: Zero test on both modes.
In masked mode EO is masked by (B) prior to subtraction.

DIX I1

DECREMENT INDEX REGISTER 1

(I1) is reduced by (EOA) or (OPF) and the result stored in (I1). The subtraction is made modulo 2^{14} assuming $(I)_i$ and (EO) are unsigned 14-bit numbers. No indicators are set. If $(I)_i = [(EO)-1]$ the next decrement will refill $(I)_i$, modulo 2^{14}. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	003640	X	X
I	18	103640	X	X
L	6	043640	-	X
B	6	0036437777	-	-

Notes: All modes accept Zero test modifier.
EO masked by (B) prior to subtraction.

DIX I2

DECREMENT INDEX REGISTER 2

(I2) is reduced by (EOA) or (OPF) and the result stored in (I2). The subtraction is made modulo 2^{14} assuming $(I)_i$ and (EO) are unsigned 14-bit numbers. No indicators are set. If $(I)_i = [(EO)-1]$ the next decrement will refill $(I)_i$, modulo 2^{14}. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	004640	X	X
I	18	104640	X	X
L	6	044640	-	X
B	6	0046437777	-	-

Notes: All modes accept Zero test modifier.
EO masked by (B) prior to subtraction.

DIX I3

DECREMENT INDEX REGISTER 3

(I3) is reduced by (EOA) or (OPF) and the result stored in (I3). The subtraction is made modulo 2^{14} assuming $(I)_i$ and (EO) are unsigned 14-bit numbers. No indicators are set. If $(I)_i = [(EO)-1]$ the next decrement will refill $(I)_i$, modulo 2^{14}. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	005640	X	X
I	18	105640	X	X
L	6	045640	-	X
B	6	0056437777	-	-

Notes: All modes accept Zero test modifier.
EO masked by (B) prior to subtraction.

LOGICAL OPERATIONS

EXT

EXTRACT

(A) and (EOA) or (OPF) are COMPARED, bit by bit. (A) retains a binary one where corresponding bits in both (A) and (EO) are ones. This is a logical "AND" operation.

For L mode:
(OPF) AND $(A)_{14-1} \rightarrow (A)_{14-1}$; $0 \rightarrow (A)_{28-15}$

For U mode:
(OPF) AND $(A)_{28-15} \rightarrow (A)_{28-15}$; $(A)_{14-1}$ unchanged

NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	000340	X	X
I	18	100340	X	X
L	6	040340	-	X
U	6	140340	-	X

Note: All modes are maskable by (EO) masked by (B) after Extract. Zero test modifier on all modes.

EXM

EXTRACT TO MEMORY

(A) and (EOA) are compared. (EOA) and (C) retain a binary one where corresponding bits of (EOA) and (A) are ones. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	18	005340	X	X
I	24	105340	X	X

Note: Both modes accept Zero test modifiers.

MRG

MERGE

(A) and (EOA) or (OPF) are compared. (A) retains a binary one where either (A) or (EO) contains a one bit. This is a logical "OR" operation.

For L mode:
(OPF) OR $(A)_{14-1} \rightarrow (A)_{14-1}$; $0 \rightarrow (A)_{28-15}$

For U mode:
(OPF) OR $(A)_{28-15} \rightarrow (A)_{28-15}$; A_{14-1} unchanged

NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	001340	X	X
I	18	101340	X	X
L	6	041340	-	X
U	6	141340	-	X
B	6	0013437777	-	-

Note: N, I, L, and U are maskable — (EOA) and (A) are merged and the result masked by (B) = effective result. N, I, L and U modes accept zero test modifier.

MGM

MERGE TO MEMORY

(A) and (EOA) are compared. (EOA) and (C) retain a binary one wherever corresponding bits of either (A) or (EOA) are ones. NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	18	004340	X	X
I	24	104340	X	X

Note: Both modes accept Zero test modifiers.

XOR

EXCLUSIVE OR

(A) and (EOA) or (OPF) are compared. (A) retain a binary one wherever the corresponding bits of (A) or (EO) are different. That is, (A) · EO' + (A)' · EO → (A).

For L mode:
(OPF) compared $(A)_{14-1} \rightarrow A_{14-1}$; $0 \rightarrow A_{28-15}$

NIA = CIA + 1.

Mode	μs	Numerical Code	Indexable	Maskable
N	12	002340	X	X
I	18	102340	X	X
L	6	042340	-	X

Note: All modes are maskable — (EO) is masked by (B) prior to comparison with (A). Zero test modifier on all modes.

COM EQ

COMPARE EQUAL

(A) are compared algebraically with (EOA). If (A) = (EOA), NIA = CIA + 2. If coincidence is not found, NIA = CIA + 1. No changes are made to registers. $(1000000000)_8$ is interpreted as the most negative number.

Overflow and carry indicators are unchanged.

Mode	µs	Numerical Code	Indexable	Maskable
N	12	043340	X	X

Note: On masking, both (EOA) and (A) are masked prior to comparison.

COM NE

COMPARE NOT EQUAL

Same as COM EQ except NIA = CIA + 2 when (A) # (EOA).

Mode	µs	Numerical Code	Indexable	Maskable
N	12	143340	X	X

Note: On masking, both (EOA) and (A) are masked prior to comparison.

COM GR

COMPARE GREATER THAN

Same as COM EQ except NIA = CIA + 2 when (A) > (EOA).

Mode	µs	Numerical Code	Indexable	Maskable
N	12	003340	X	X

Note: On masking, both (EOA) and (A) are masked prior to comparison.

COM LS

COMPARE LESS THAN

Same as COM EQ except NIA = CIA + 2 when (A) < (EOA).

Mode	µs	Numerical Code	Indexable	Maskable
N	12	103340	X	X

Note: On masking, both (EOA) and (A) are masked prior to comparison.

SCN EQ

SCAN EQUAL

(A) are compared algebraically with each word in a specified table beginning with the effective operand address and continuing for n words, where n is specified in the X register. If (A) are found equal to a word in the table, coincidence is found; NIA = CIA + 2, and the address of coincidence is placed in the X register. If coincidence is not found, NIA = CIA + 1, and (X) contain the last address of the table. No other registers are changed. If (X) is initially zero, the scan stops in 6 µs with (X) = EOA_1.

Overflow and carry indicators remain unchanged.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+6c	046340	X	X

where c = number of words actually scanned.

SCN NE

SCAN NOT EQUAL

Same as SCN EQ, except coincidence is found when (A) are not equal to a word in the table specified by EOA.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+6c	146340	X	X

SCN GR

SCAN GREATER THAN

Same as SCN EQ, except coincidence is found when (A) is greater than a word in the table specified by EOA.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+6c	006340	X	X

SCN LS

SCAN LESS THAN

Same as SCN EQ, except coincidence is found when (A) is less than a word in the table specified by EOA.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+6c	106340	X	X

CMT EQ

COMPARE TABLES EQUAL

The contents of a reference table are compared algebraically, word by word, with the contents of an unknown table. The comparison begins with the reference table and unknown table addresses specified in EOA and (C) respectively, and continues for the number of words specified in (X), where the maximum number in X = 37777_8. Coincidence occurs when the contents of a word in the reference tables equal the contents of the corresponding word in the unknown tables and NIA = CIA + 2. When no coincidence occurs, NIA = CIA + 1. Register contents prior to and after execution of CMT for both coincidence and no coincidence conditions are shown below:

Mode	µs	Numerical Code	Indexable	Maskable
N	6+12c	047340	X	X

where c = number of words compared.

REGISTER	INITIAL CONDITIONS	FINAL CONDITIONS	
		COINCIDENCE	NO COINCIDENCE
A		Contents of Reference table word at coincidence	Contents of last word in Reference table
B	Mask if specified	Unchanged	Unchanged
C	First address of Unknown table	Address of Unknown table word at coincidence	Address of Last word of Unknown table
X	Number of words to be compared	Same as $(C)_{14-1}$	Same as $(C)_{14-1}$
EOA	First address of Reference table	Unchanged	Unchanged

Overflow and carry indicators remain unchanged.

If (X) = 0 initially, the compare stops after 6 µs with final register conditions equal to no coincidence conditions, except for the A Register, which is unchanged.

CMT NE

COMPARE TABLES NOT EQUAL

Same as CMT EQ except coincidence is found when a word in the reference table is not equal to the corresponding word in the unknown table.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+12c	147340	X	X

CMT GR

COMPARE TABLES GREATER THAN

Same as CMT EQ except coincidence is found when a word in the reference table is greater than the corresponding unknown table word.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+12c	007340	X	X

CMT LS

COMPARE TABLES LESS THAN

Same as CMT EQ except coincidence is found when a word in the reference table is less than the corresponding unknown table word.

Mode	µs	Numerical Code	Indexable	Maskable
N	6+12c	107340	X	X

REPLACE AND EXCHANGE OPERATIONS

All replace and exchange operations cause a transfer of register contents but do not affect memory. Both normal and branch mode are available. If the branch mode is specified, NIA = OPA; hence a transfer and branch are accomplished in one instruction.

In the normal mode, NIA = CIA + 1.

RAW C

REPLACE A WITH C

(C) are transferred to (A); (C) are unchanged. If mask is specified, (C) are masked by (B) prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	002500	X
Br	6	102500	X

RAW I1

REPLACE A WITH I1

(I1) bits 1-14 are transferred to bits 1-14 of (A). Bits 15-28 of (A) are set to zero and (I1) are unchanged. If mask is specified, (I1) is masked by (B) bits 1-14 prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	003500	X
Br	6	103500	X

RAW I2

REPLACE A WITH I2

Same as RAWI1, except the (I2) are transferred.

Mode	µs	Numerical Code	Maskable
N	6	004500	X
Br	6	104500	X

RAW I3

REPLACE A WITH I3

Same as RAWI1, except (I3) are transferred.

Mode	µs	Numerical Code	Maskable
N	6	005500	X
Br	6	105500	X

RAW X

REPLACE A WITH X, HOLD UPPER A BITS

The (X) bits 1-14 are transferred to (A) bits 1-14. (A) 15-28 and (X) remain unchanged. If masking, (X) is masked by $(B)_{1-14}$ prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	001500	X
Br	6	101500	X

RAW XC

REPLACE A WITH X, CLEAR UPPER A BITS

Same as RAWXH, except $(A)_{15-28}$ are set to zeros.

Mode	µs	Numerical Code	Maskable
N	6	006500	X
Br	6	106500	X

RAW M

REPLACE A WITH M

(M) are transferred to (A) and (M) remain unchanged. (M) are masked by (B) prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	007500	X
Br	6	107500	X

RAW Q

REPLACE A WITH Q

Same as RAWM, except (Q) are transferred.

Mode	µs	Numerical Code	Maskable
N	6	013500	X
Br	6	113500	X

RAW G

REPLACE A WITH G (RELATIVE ADDRESS REGISTER)

$(G)_{1-14}$ are transferred to $(A)_{1-14}$. Zero → $(A)_{15-28}$ and (G) are unchanged. Masking is as for RAWI1.

Mode	µs	Numerical Code	Maskable
N	6	016500	X
Br	6	116500	X

RAW T1

REPLACE A WITH PANEL ADDRESS TOGGLES

(Console address toggles)$_{1-14}$ → (A) and 0 → (A_{15-28}). Masking is as in RAWI1.

Mode	µs	Numerical Code	Maskable
N	6	017500	X
Br	6	117500	X

RAW T2

REPLACE A WITH PANEL DATA TOGGLES

Same as RAWT1, except (Data Toggles)$_{1-28}$ → $(A)_{1-28}$. Masking is on all 28 bits.

Mode	µs	Numerical Code	Maskable
N	6	015500	X
Br	6	115500	X

RWA C

REPLACE C WITH A

(A) → (C) and remain unchanged. When masking, (A) is masked by (B) prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	042500	X
Br	6	142500	X

RWA T1

REPLACE I1 WITH A

$(A)_{1-14}$ → $(I1)_{1-14}$ and (A) are unchanged. When masking, $(A)_{1-14}$ are masked by $(B)_{1-14}$ prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	043500	X
Br	6	143500	X

RWA I2

REPLACE I2 WITH A

Same as RWAI1, except (A) → (I2).

Mode	µs	Numerical Code	Maskable
N	6	044500	X
Br	6	144500	X

RWA I3

REPLACE I3 WITH A

Same as RWAI1, except (A) → (I3).

Mode	µs	Numerical Code	Maskable
N	6	045500	X
Br	6	145500	X

RWA X

REPLACE X WITH A

Same as RWAI1, except $(A)_{1-14}$ → (X).

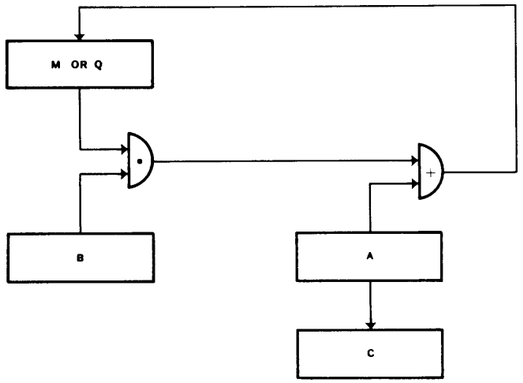
Mode	µs	Numerical Code	Maskable
N	6	046500	X
Br	6	146500	X

RWA M

REPLACE M WITH A

The (A) → (M) and are unchanged. (M) is masked by (B) and are merged with (A). The result stored in (M). The (C) are set to (A) initial.

Mode	µs	Numerical Code	Maskable
N	12	047500	-
Br	12	147500	-



RWA Q

REPLACE Q WITH A

Same as RWAM, except (Q) are replaced.

Mode	µs	Numerical Code	Maskable
N	12	053500	-
Br	12	153500	-

RWA G

REPLACE G WITH A

Same as RWAI1, except (A) → (G).

Mode	µs	Numerical Code	Maskable
N	6	056500	X
Br	6	156500	X

EAW B

EXCHANGE A WITH B

(A) are interchanged with (B).

Mode	µs	Numerical Code	Maskable
N	6	016600	-
Br	6	116600	-

EAW C

EXCHANGE A WITH C

(A) ⟷ (C). When masked, (A) → (C) but (C) are masked by the (B) prior to transfer to (A).

Mode	µs	Numerical Code	Maskable
N	6	005600	X
Br	6	105600	X

EAW XH

EXCHANGE A WITH X, HOLD UPPER A BITS

$(A)_{1-14}$ and (X) are exchanged. $(A)_{15-28}$ remain unchanged. When masked, only (X) are masked by (B) prior to transfer.

Mode	µs	Numerical Code	Maskable
N	6	001600	X
Br	6	101600	X

EAW XC

EXCHANGE A WITH X, CLEAR UPPER A BITS

Same as EAWXH, except $(A)_{15-28}$ are set to zero.

Mode	µs	Numerical Code	Maskable
N	6	006600	X
Br	6	106600	X

EAW I1

EXCHANGE A WITH I1

$(A)_{1-14}$ are exchanged with (I1). $0 \rightarrow (A)_{15-28}$. When specified, (I1) are masked by (B) prior to transfer.

Mode	µs	Numerical Code	Maskable
N	12	003600	X
Br	12	103600	X

EAW I2

EXCHANGE A WITH I2

Same as EAWI1, except (I2) are exchanged with (A).

Mode	µs	Numerical Code	Maskable
N	12	004600	X
Br	12	104600	X

EAW I3

EXCHANGE A WITH I3

Same as EAWI1, except (I3) are exchanged with (A).

Mode	µs	Numerical Code	Maskable
N	12	005600	X
Br	12	105600	X

EAW M

EXCHANGE A WITH M

(A) ⟷ (M). When specified, (M) is masked by (B) prior to exchange.

Mode	µs	Numerical Code	Maskable
N	12	007600	X
Br	12	107600	X

EAW Q

EXCHANGE A WITH Q

Same as EAWM, except exchange is with (Q).

Mode	µs	Numerical Code	Maskable
N	12	013600	X
Br	12	113600	X

BRANCH OPERATIONS

In all branch operations, the next instruction address is CIA + 1 if the condition of the branch is not met. If the condition is met, NIA = EOA, where the effective operand address may be address in the operand field of the instruction modified by indexing or indirect addressing or, as in the X mode, it may be (X).

The execution time is 6 µs if the condition is not met. If the condition is met, the I mode requires 12 µs for execution.

BUN

BRANCH UNCONDITIONALLY

The condition is always met. Hence NIA = EOA.

Mode	µs	Numerical Code	Indexable
N	6	000000	X
I	12	100000	X
X	6	040000	-

BZE

BRANCH IF (A) = 0

The condition is met when $(A)_{28-1} = 0$.

Mode	µs	Numerical Code	Indexable
N	6	016000	X
I	6,12	116000	X
X	6	056000	-

BZM

BRANCH IF A MASKED = 0

The condition is met if $(A)_{28-1}$, when masked by $(B)_{28-1}$, equal zero.

Mode	µs	Numerical Code	Indexable
N	6	001000	X
I	6,12	101000	X
X	6	041000	-

BNZ

BRANCH IF A NOT ZERO

The condition is met when $(A)_{28-1}$ contain one bit.

Mode	µs	Numerical Code	Indexable
N	6	012000	X
I	6,12	112000	X
X	6	052000	-

BNM

BRANCH IF A MASKED NOT ZERO

The condition is met if $(A)_{28-1}$, as masked by $(B)_{28-1}$, contain a one bit.

Mode	µs	Numerical Code	Indexable
N	6	002000	X
I	6,12	102000	X
X	6	042000	-

BPO

BRANCH IF A IS POSITIVE

The condition is met if $(A)_{28}$ is zero; that is, if (A) is a positive number.

Mode	µs	Numerical Code	Indexable
N	6	013000	X
I	6,12	113000	X
X	6	053000	-

BNG

BRANCH IF A IS NEGATIVE

The condition is met if $(A)_{28}$ is a one; that is, if (A) is a negative number.

Mode	µs	Numerical Code	Indexable
N	6	003000	X
I	6,12	103000	X
X	6	043000	-

BLB

BRANCH IF A CONTAINS A LOW BIT

The condition is met if $(A)_1$ is a one bit.

Mode	µs	Numerical Code	Indexable
N	6	004000	X
I	6,12	104000	X
X	6	044000	-

BNX 11

BRANCH IF INDEX 1 NOT ZERO

The condition is met if (I1) is not zero.

Mode	µs	Numerical Code	Indexable
N	6	007000	X
I	6,12	107000	X
X	6	047000	-

BNX 12

BRANCH IF INDEX 2 NOT ZERO

The condition is met if (I2) is not zero.

Mode	µs	Numerical Code	Indexable
N	6	010000	X
I	6,12	110000	X
X	6	050000	-

BNX 13

BRANCH IF INDEX 3 NOT ZERO

The condition is met if (I3) is not zero.

Mode	µs	Numerical Code	Indexable
N	6	011000	X
I	6,12	111000	X
X	6	051000	-

BOF

BRANCH IF OVERFLOW INDICATOR ON

The condition is met if the Overflow Indicator has been set on since the last test or overflow set/reset operation.
The Overflow Indicator is turned off by this command.

Mode	µs	Numerical Code	Indexable
N	6	005000	X
I	6,12	105000	X
X	6	045000	-

BCY

BRANCH IF CARRY INDICATOR ON

The condition is met if the Carry Indicator has been turned on since the last test or set/reset operation.
The indicator is turned off by this command.

Mode	µs	Numerical Code	Indexable
N	6	015000	X
I	6,12	115000	X
X	6	055000	-

BCP

BRANCH IF CORE PARITY ERROR

The condition is met if the Core Parity Instruction Indicator and/or Core Parity Operand Indicator have been turned on since the last test or set/reset operation. Both indicators are turned off by this command.

Mode	μs	Numerical Code	Indexable
N	6	006000	X
I	6,12	106000	X
X	6	046000	-

BDP

BRANCH IF DRUM OR DIRECT ACCESS PARITY ERROR

The condition is met if the Drum Parity and/or the Direct Access Parity Indicators have been turned on. These indicators are turned off by this instruction.

Mode	μs	Numerical Code	Indexable
N	6	014000	X
I	6,12	114000	X
X	6	054000	-

SHIFT AND NORMALIZE OPERATIONS

For all shift instructions, the execution time is equal to (8 + 2n) μs where n is the number of bit positions shifted and may be equal to from 0 to (2^{14} - 1). The number of places to shift may be indicated in OPF or X Register.

In the Normal mode (bit 27 of instruction = 1), n is specified in OPF; and in the X mode (bit 27 = 0), n is specified by (X). The N-mode code is formed by adding 040000_8 to the X-mode code shown for each operation. (X) is always zero at completion of any shift and NIA = CIA + 1. If overflow occurs, the shift operation is not stopped.

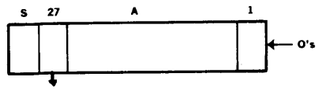
Normalize operations all require (8 + 2c) μs execution time, where c equals the number of bit positions shifted. Both the normal and X modes are available on normalize as on shifts. Overflow will not occur. Normalize will stop or will not begin if (X) = 0. If (A) = 0, the shifting operation will continue until (X) = 0. NIA = CIA + 1.

ALA

SHIFT A LEFT ARITHMETIC

Bits 27-1 of (A) are shifted left n bit positions, zeros are placed in the n least significant bits of (A), and the sign of (A) is unchanged. An overflow occurs if a bit is shifted out of $(A)_{27}$ when $(A)_{27} \neq$ (A) sign.

Mode	Numerical Code
N	040540
X	000540

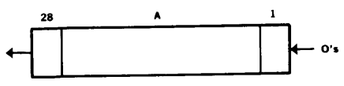


ALO

SHIFT A LEFT LOGICAL OPEN

All 28 bits of (A) are shifted left n bit positions and zeros are placed in the n least significant bits of (A). No overflow occurs.

Mode	Numerical Code
N	041540
X	001540

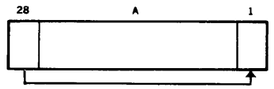


ALC

SHIFT A LEFT LOGICAL CLOSED

All 28 bits of (A) are shifted left n bit positions and bit 28 of (A) is placed in bit 1 of (A) at each shift. There is no overflow.

Mode	Numerical Code
N	042540
X	002540



ARA

SHIFT A RIGHT ARITHMETIC

Bits 27-1 of (A) are shifted right n bit positions. The sign (bit 28) of (A) remains unchanged, but at each shift $(A)_{28} \rightarrow (A)_{27}$; that is, the sign is propagated to the right.

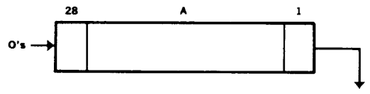
Mode	Numerical Code
N	043540
X	003540

ARO

SHIFT A RIGHT LOGICAL OPEN

The 28 bits of (A) are shifted right n bit positions. Zeros are entered in the n most significant bits of (A).

Mode	Numerical Code
N	044540
X	004540



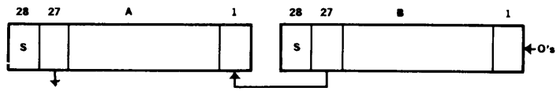
BLA

SHIFT A, B LEFT ARITHMETIC

Bits 27-1 of (A) and (B) are shifted left n bit positions such that bit 27 of (B) is shifted into bit 1 of (A) and zero is shifted into bit 1 of (B). The sign bits (28) of (A) and (B) remain unchanged.

Overflow will occur if a bit shifted out of $(A)_{27}$ is not equal to (A) sign.

Mode	Numerical Code
N	045540
X	005540



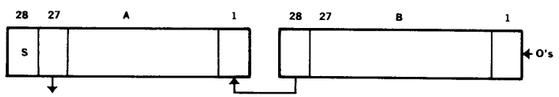
BLL

SHIFT A LEFT ARITHMETIC, B LEFT LOGICAL OPEN

Bits 27-1 of (A) and the 28 bits of (B) are shifted left n bit positions such that on each shift: (A) sign → (A) sign, $(B)_{28} \rightarrow (A)_1$, $(B)_{27} \rightarrow (B)_{28}$, and $0 \rightarrow (B)_1$.

Mode	Numerical Code
N	046540
X	006540

Overflow occurs if a bit shifted out of $(A)_{27} \neq$ (A) sign.

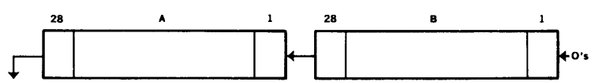


BLO

SHIFT A, B LEFT, A LOGICAL, B LOGICAL, OPEN

All 28 bits of (A) and (B) are shifted left n bit positions such that on each shift: $(B)_{28} \rightarrow (A)_1$, $0 \rightarrow (B)_1$, $(A)_{27} \rightarrow (A)_{28}$ and $(A)_{28}$ is lost. No overflow occurs.

Mode	Numerical Code
N	047540
X	007540

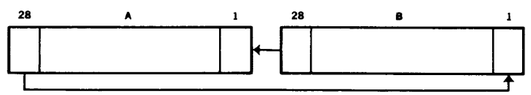


BLC

SHIFT A, B LEFT, A LOGICAL, B LOGICAL, CLOSED

All 28 bits of (A) and (B) are shifted left n bit positions such that on each shift: $(B)_{28} \rightarrow (A)_1$ and $(A)_{28} \rightarrow (B)_1$. No overflow occurs.

Mode	Numerical Code
N	050540
X	010540

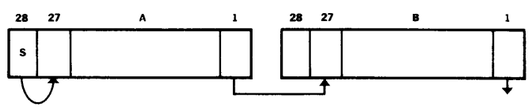


BRA

SHIFT A, B RIGHT ARITHMETIC

Bits 27-1 of (A) and (B) are shifted right n bit positions such that on each shift: $(A)_{28} \rightarrow (A)_{27}$, $(A)_1 \rightarrow (B)_{27}$, and (A) and (B) signs are unchanged.

Mode	Numerical Code
N	052540
X	012540

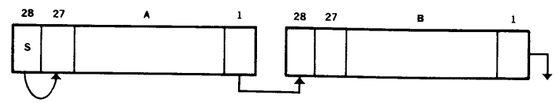


BRL

SHIFT A, B RIGHT - A ARITHMETIC, B LOGICAL OPEN

Bits 27-1 of (A) and the 28 bits of (B) are shifted right n bit positions such that on each shift: $(A)_{28} \rightarrow (A)_{27}$, $(A)_1 \rightarrow (B)_{28}$, $(B)_{28} \rightarrow (B)_{27}$, and $(A)_{28}$ is unchanged.

Mode	Numerical Code
N	051540
X	011540

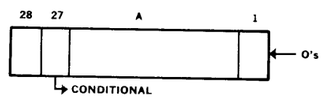


NAA

NORMALIZE A ARITHMETIC

$(A)_{27-1}$ is shifted left until $A_{27} \neq A_{28}$, X = 0, or until (A) = 1400000000_8. On each shift, $0 \rightarrow A_1$, (X) - 1 → (X), and (A) sign is unchanged.

Mode	Numerical Code
N	053540
X	013540

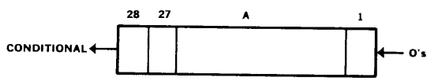


NAO

NORMALIZE A LOGICAL OPEN

All 28 bits of (A) are shifted left until $(A)_{28} = 1$. No shift occurs if $(A)_{28}$ is initially a one. On each shift, $0 \rightarrow A_1$ and (X) - 1 → (X).

Mode	Numerical Code
N	054540
X	014540



NAC

NORMALIZE A LOGICAL CONTINUED

The 28 bits of (A) are unconditionally shifted left one position. Shifting will then continue until $A_{28} = 1$ or X = 0 (or stop if condition $A_{28} = 1$ is met upon initial shift). Shifting will not start if X is initially = 0. This mode provides a means of easily continuing a normalizing operation by nullifying the previous stop condition. $0 \rightarrow (A)_1$ and (X) - 1 → (X) on each shift.

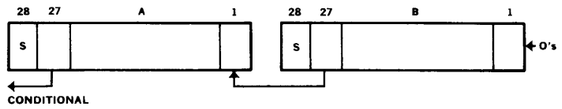
Mode	Numerical Code
N	055540
X	015540

NBA

NORMALIZE A AND B ARITHMETIC

(A) and (B) are shifted left until $(A)_{27} \neq (A)_{28}$ or until (A) = 1400000000_8.
$0 \rightarrow (B)_1$, $(B)_{27} \rightarrow (A)_1$, (X) - 1 → (X), and $(A)_{28}$, $(B)_{28}$ are unchanged on each shift.

Mode	Numerical Code
N	056540
X	016540



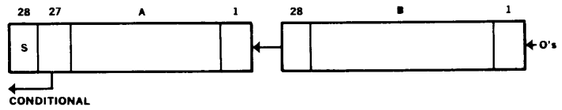
NBL

NORMALIZE A, B, A ARITHMETIC, B LOGICAL, OPEN

Bits 27-1 of (A) and all 28 bits of (B) are shifted left until $(A)_{27} \neq (A)_{28}$ or (A) = 1400000000_8.

On each shift, $0 \rightarrow (B)_1$ and (X) - 1 → (X). (A) sign remains unchanged and (B) sign is shifted into $(A)_1$.

Mode	Numerical Code
N	057540
X	017540



PROGRAM CONTROL

NOP

NO OPERATION

Skip to CIA + 1 for the NIA with no change of register status.

Mode	µs	Numerical Code	Indexable
N	6	017000	-

STP

STOP AND BRANCH

The program stops at this instruction with (N) = EOA. When the computer console resume button is depressed, NIA = EOA.

Mode	µs	Numerical Code	Indexable
N	6	046600	-
I	12	146600	-

SRB

SET RETURN AND BRANCH

This operation branches to a specified routine after saving the return address. The return address, CIA + 1, is stored in (EOA) and the program branches to EOA + 1 to begin the subroutine. $(EOA)_{28-15}$ are unchanged.

Mode	µs	Numerical Code	Indexable
N	12	045600	X
I	18	145600	X

SET

SET

The indicators or functions specified by the bits in the operand field of the instruction are set, singly or in combination. A 1 bit in the OPF turns on or activates.

Mode	µs	Numerical Code
N	6	064600

NIA = CIA + 1.

The following table lists the OPF bit designations.

OPF bit	Turns on or affects
1	Drum parity indicator
2	Direct access parity indicator
3	Core instruction parity indicator
4	Core operand parity indicator
5	I/O transfer-complete indicator (under RESET only)
6	Interrupt inhibit indicator
7	Carry indicator
8	Overflow indicator
9	Not assigned
10	Not assigned
11	*1's complement of (B) → (B)
12	*1's complement of (A) → (A)
13	*2's complement of (A) → (A)
14	Set (A) and (B) to zero.

*Note: 1. In the 2's complement mode, overflow is set if 1000000000_8 is complemented and the accumulator remains unchanged. The carry indicator is set if 0000000000_8 is complemented and the accumulator is unchanged.

2. Bits 11 through 14 (A and B Register effects) cause the same action in either the set or reset instruction.

RST

RESET

The indicator or function as specified by the bits in OPF are reset, singly or in combination. The reset operation is controlled like the set operation using the same OPF bit assignments. For bits 11 through 14 (A and B Register effects), the same action occurs in either the set or reset operation.

Mode	µs	Numerical Code
N	6	044600

NIA = CIA + 1.

STA

SET CORE ADDRESS

A 4,096-word block of additional core memory above a 16,384 word memory size is specified. This additional 4K block is substituted for the third quarter of regular memory. Bits 1 through 4 of the OPF specify which 4K block addition is to be addressed as follows:

Mode	µs	Numerical Code
N	6	047600

OPF bits	Addresses Core Block
0000	The regular 4K block in the third quarter (8K-12K)
0001	16K-20K
0010	20K-24K
0011	24K-28K
0100	28K-32K

Addressing within the 4K block uses bits 1-12 of the EOA for all operations. Bits 4-7 of Q retain control of the additional block used. The SCA instruction modifies $(Q)_{4-7}$ to equal $(OPF)_{1-4}$ but does not change other Q bits. The maximum expansion of core memory above 16,384 is four 4,096-word blocks, for a total memory of 32,768 words.

NIA = CIA + 1.

OPERATION EXTENSION

OEX

Mode	µs	Numerical Code	Indexable
N	18	Specified by group number basic code	X
I	24	plus 100000_8	X

The Operation Extension provides automatic entry into up to 320 subroutine-type programs stored in core memory. These instructions are written in programs as though they were real instructions. Their operand field may be modified by an index register and may be an indirect address. The following actions take place when an operation extension is executed:

A. Interrupts are inhibited
B. (EOA) → (B)
C. EOA → (X)
D. CIA + 1 → (OEA)
E. NIA = OEA + 1

The initial (A) and (C) remain unchanged. OEA + 1 normally contains and branch unconditional instruction to the location of the subroutine. If all 320 operation extensions are used, a block of 640 locations in core memory must be reserved for linkage to these routines. The operation extension format is shown in Figure 3-1.

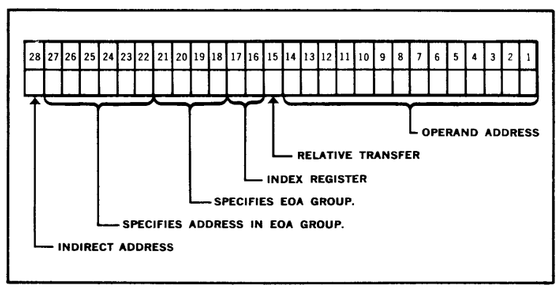


Figure 3-1. Operation Extension Format

OEA is the specified operation extension address. OEA is always even-numbered and is specified as follows in bits 18 to 21 of instructions:

bits 21	20	19	18	OEA Group
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5

Bits 22-27 specify the addresses $(0 \text{ to } 76)_8$ for each group as follows:

Group	$Addresses_8$
1	0200 to 0376
2	0400 to 0576
3	0600 to 0776
4	1000 to 1176
5	1200 to 1376

SECTION IV PROGRAMMING INTERRUPTS

GENERAL

A basic 340 program can be interrupted from up to 28 sources. The priority for each source is assigned and changed under program control; however, scanning for interrupt conditions is done automatically by the Interrupt Subsystem independent of the program being executed. Thus, a minimum of computer time and memory is involved in handling interrupts.

At the time an interrupt signal is recognized, the 340 automatically stores the next instruction address of the program in progress in a program re-entry table, and goes to the interrupt response address to pick up the first instruction of the response program. The response program takes the programmed interrupt response actions. When the response is finished, the interrupted program is resumed at the place the interrupt occurred. Therefore, no program has to be repeated due to an interrupt. This flexibility in handling interrupts comes through a versatile set of program instructions that allow the programmer to override interupt requests, alter priority, and cancel interrupts. These instructions are described below.

INTERRUPT PRIORITY

The 340 Interrupt Subsystem is able to determine the priority of the program-in-progress and to remember up to 28 individual interrupts, reacting to each in the order of relative priority.

Priority is assigned both by the hardware and by the programmer. Hardware priority assignment is permanent, involving the physical arrangement of the interrupt input lines made before installation. The programmer priority assignment is variable and involves the bit configuration in special interrupt priority masking register, the M Register. There is one mask bit position for each interrupt line. An interrupt signal is accepted or rejected on the basis of a one or zero in the mask position associated with the interrupt line. For example, if the 4th highest priority interrupt occurred, the mask can be changed to block all lower priority interrupts from being recognized, yet retain the 3 higher interrupts. If a 1, 2, or 3 priority interrupt occurs before the 4th priority interrupt response is completed, the lower response is interrupted, the higher priority response executed, and then the lower priority response re-entered and completed.

Inhibiting interrupts is not restricted to blocking lower priority interrupts while higher priority responses are in progress. In addition, it is possible to do any of the following:

A. Temporarily inhibit all interrupts from interfering with a critical program calculation.

B. Block higher priority interrupts, while responding or checking out lower priority responses.

C. Enable lower priority responses from higher priority response routines.

D. Re-establish the higher-lower priority relationship between interrupts.

For example, when the interrupt system has all lines enabled and is interrupted from priority 20 line, the priority mask may be changed to keep lines 1, 2, 5, and 27 enabled. Since enabled interrupts can override the program in progress and thus have a higher priority, thus, line 27 now has higher priority than line 20 and can override the line 20 response.

To provide the priority assignment function, two special registers are used with the interrupt system. These are the M Register (priority masking register) and the Q Register (the type-2 interrupt holding register).

Note that a type-1 interrupt is a continuous interrupt signal generated when a normally false line goes true. The line remains true until some action resets it false. A type-2 interrupt is a short duration interrupt signal generated when a line momentarily goes true and then false. True signals must remain true for at least 12 microseconds.

M REGISTER

The M Register controls priority by masking the interrupt lines, allowing those unmasked to cause program interrupts. Each type-1 and -2 interrupt line requires a corresponding M Register bit. Program control over interrupts is established by loading 1's and 0's into the M Register. A 1 bit enables the associated interrupt lines to cause a program interrupt. A 0 bit inhibits the interrupt signal from causing an interrupt. Thus, interrupt

priority may be controlled by changing the M Register bit configuration. The M Register is set and reset under program control using the special instructions outlined below. The basic M Register is 28 bits in length.

Q REGISTER

The 340 Interrupt Subsystem scans the type-1 interrupt lines directly because these interrupt signals are continuous. However, type-2 interrupt signals are momentary, and must be "held" in order to be scanned. A special buffer register, the Q Register, is used to record, or remember, type-2 interrupt signals. Each type-2 interrupt has one Q Register bit position which is automatically set to 1 when an interrupt signal occurs on its line. During the scanning, the Q Register bits are examined, in their proper priority sequence, along with the type-1 lines. The basic minimum 10-bit Q Register may be extended to a 28-bit Q Register.

When a type-2 interrupt signal sets a bit in the Q Register and its respective response routine is entered, that bit in the Q Register must be reset to 0 by the response routine. This may be accomplished with one of the instructions described below.

Because bits 4-7 of Q are always reserved for special memory extension control, the maximum number of type-2 interrupts allowed is 24.

The Q Register is discussed in terms of groups of bits; however, a flexible system of patch-panel wiring permits the Q bits to be discontinuously intermixed with the type-1 interrupt lines in any desired combination. The exact arrangement is determined by the particular installation.

PROGRAM RE-ENTRY ADDRESS

A continuous block of 56 core words is reserved for the interrupt system. 28 of these words are used for recording the next program instruction address (re-entry address) and the other 28 words are used for the first word of the interrupt response routine (the response address). These two groups of core words are interlaced such that each interrupt line has an associated "pair" of core addresses, starting with core address 37400. Core address 37400 holds the program next instruction address after the computer has been interrupted by interrupt line No. 1, and core address 37401 contains the first instruction of the associated interrupt response routine for line No. 1. Refer to Table 4-1.

After completing an interrupt routine, the computer must return to the program that was interrupted. The next instruction address of the program that was being executed when the interrupt occurred was automatically merged with a JUMP operation code, then stored in the first of the "pair" of addresses associated with the interrupt line. Thus, to exit from an interrupt routine, the response program goes to this re-entry address and program control jumps back to the interrupted program.

Example: To re-enter the program that was in progress when interrupt line number 25 went true the programmer uses a jump to the instruction in core address 37450.

INTERRUPT RESPONSE ADDRESS

Each interrupt line also has a unique response address assigned by the 340 Interrupt Subsystem. When an interrupt occurs, the computer automatically stores the next program instruction address in the assigned re-entry address, then transfers to the first address of the associated interrupt response routine which is stored in the second core location of the pair.

Example: When interrupt line No. 25 goes true, the computer stores the program next instruction address in core address 37450 and automatically takes its next instruction from core address 37451. Thus the programmer places a transfer instruction in memory location 37451 to begin the interrupt response routine that is appropriate for an interrupt on Line No. 25. Refer again to Table 4-1.

TIMING

The timing considerations for the Interrupt Subsystem may be divided into two parts; the time required to scan the interrupt lines, and the time required to begin the response to interrupt once it is recognized. Scanning is done in 18 μseconds for the interrupt system. The time to respond to the interrupt signal depends on how long it takes to complete the instruction in progress.

INTERRUPT SYSTEM CONFIGURATION

The basic 340 Interrupt System has 28 interrupt levels. A unique method of patch-panel wiring permits the type-1 and -2 interrupts to be intermixed in any convenient combination.

In all interrupt configurations, five of the basic interrupt lines are reserved for program control and system monitoring functions shown in Table 4-2. Table 4-3 lists other internal signals that may be used for interrupt sources. Each system uses these internal interrupts at the discretion of the system programmer.

Table 4-1. Interrupt Re-Entry and Response Addresses

Interrupt Line No.	Re-Entry Address	Response Address
1	37400	37401
2	02	03
3	04	05
4	06	07
5	10	11
6	12	13
7	14	15
10	16	17
11	20	21
12	22	23
13	24	25
14	26	27
15	30	31
16	32	33
17	34	35
20	36	37
21	40	41
22	42	43
23	44	45
24	46	47
25	50	51
26	52	53
27	54	55
30	56	57
31	60	61
32	62	63
33	64	65
34	37466	37467

Table 4-2. Reserved Interrupts

Interrupt Source	Type	Function
Elapsed Time	2	Signals run-down of elapsed time counter
End Drum-Core Transfer	2	Signals end of information transfer between drum and core
I/O Transfer Complete	1	Computer/device information transfer completed
Drum Parity Error	1	Signals odd-bit parity error in a drum word
Instruction Parity Error	1	Signals odd-bit parity error in a core word

Table 4-3. Other Interrupts

Interrupt Source	Type	Function
Operand Parity Error	1	Core parity error detected when accessing operand
I/O Access Parity Error	1	Core parity error detected when transferring data to Master I/O Controller.
Analog Input Out-of-Limits	2	Generated when analog input exceeds limits. This interrupt is required by the standard analog scan programs.
Contact Input Non-Comparison	2	Generated when contact input does not compare. This interrupt is not required by the standard contact scan programs.
Operator's Console Request	1	
Operator's Console Cancel	1	
Operator's Console Unassigned	1	For optional use in system program.
Operator's Console Unassigned	1	
Operator's Console Connect Outputs	1	
Operator's Console Freeze Outputs	1	
Card Punch Alarm	1	Equipment malfunction.
Card Reader Alarm	1	Equipment malfunction.
Paper Tape Punch, Low Paper	1	Signals low paper level.
Paper Tape Reader, Parity Error	1	Signals parity error on tape.
Once-A-Second Pulse (64/60)	2	Not used in standard programming system.

SECTION V
PROGRAMMING DRUM-CORE TRANSFER OPERATIONS

GENERAL

In addition to the high-speed core main memory, auxiliary magnetic drum memory for bulk storage may be added to the basic 340 Computer. Maximum drum storage capacity is 98,304 28-bit words.

DRUM MEMORY ADDRESSING

The drum is divided into basic units called tracks. Each track is in turn divided into 128 or 256 sectors. By assigning each track a unique identification number, and numbering the sectors within a track, any word in the memory may be referenced by giving its track and sector number address. In referring to memory addresses it is customary to list the track first followed by the sector. Thus the memory address 57-132 identifies track 57, sector 132. Drum memory addressing is illustrated in Figure 5-1.

DRUM PROTECTION

The Memory Subsystem includes a memory protect feature that assures that a program on the drum is not accidentally overwritten by either a programmer error or hardware malfunction. This protect feature provides a set of up to 16 manually controlled toggle switches for selecting writeable and non-writeable areas of memory. Although the protect feature disables the write circuits of the "guarded" (non-writeable) areas of memory, the respective read circuits are not affected and information may still be read from these sections of memory. The number of toggle switches active in controlling memory write circuits varies with the size of the memory involved. Each switch controls 16 tracks.

DRUM-CORE INFORMATION TRANSFER OPERATIONS

Information transfers between core and drum memory occurs in parallel with program execution. Program intervention is required only to set up and initiate the next information transfer. Access to the core memory is shared between the drum memory subsystem, the Master I/O Control and Interrupt Unit, and the Central Processor on a priority basis. Approximately 4 percent of core access time is used by the drum memory subsystem during drum-core transfers.

Drum-core transfers operate in various modes under program control providing a great deal of flexibility and versatility. The Modes are:

A. Variable - Provides the transfer of a variable length block of words, either to or from the drum. The block may be of any length from one word to 16,384 words and transfer may start at any drum or core location. An average of 8.5 ms drum access time (max. 17 ms) can be expected at the initiation of the transfer.

B. Immediate - This transfer begins immediately and does not wait for drum latency time. The block must consist of one or more complete tracks (drum) of words. The block specified must start at sector zero of a drum track and the starting core address must contain zeros in the least significant seven bits. There is a one-to-one correspondence between the last seven bits of the address of core and of drum as each word is transferred.

C. Relative - This transfer provides an automatic operand address modification during transfer. The contents of the G Register is used to increment the operand address of those words which contain a relative modifier bit in position 15 during transfer from drum to core and core to drum, respectively. This mode is normally used for transferring instructions which were originally written to operate in one section of core such that they can operate, with correct operand addressing, in another. Data is normally transferred in the non-relative mode.

D. Parity Error Stop - The parity error stop mode allows the program to specify whether to stop the transfer when a parity error is detected or to finish the transfer. In either case, the Parity Error Indicator is turned on.

DRUM-CORE TRANSFER COMPLETE SIGNAL

A short 12 microsecond pulse is generated in drum memory circuits upon completion of a drum transfer. This signal is normally used to generate a type 2 program interrupt and is connected to a position of the Q Register. The response routine will normally reset the bit position in Q and initiate the next drum-core transfer; and then return to the interrupted program.

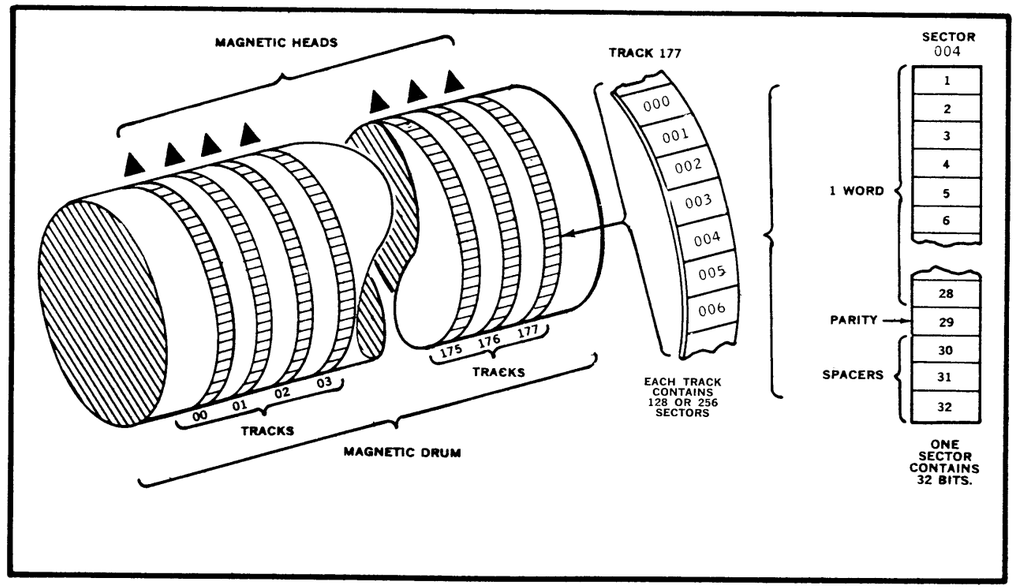


Figure 5-1. Drum Memory Organization

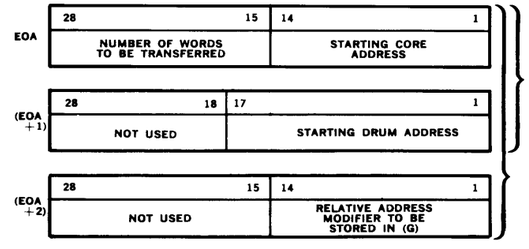
DRUM-CORE TRANSFER INSTRUCTIONS

START DRUM-TO-CORE TRANSFER

SDC

Information is transferred from drum to core as specified by bits 23-26 of the instruction. The OPF of the instruction specifies the address of the first of two or three consecutive core cells which contain the following information.

Mode	μs	Numerical Code
N	18-24	000300
I	24-30	100300



N	18	
I	24	
N	24	
I	30	

Those bits marked as "not used" may be used by the programmer to store other information. (EOA + 2) relative transfer information need to be set up only when the relative type of transfer is specified.

Bits 23-26 contain the following information:

Bit	Use if a one bit
23	RELATIVIZED, use (G)
24	IMMEDIATE
25	STOP ON PARITY
26	LOAD G with (EOA + 2)

NIA = CIA + 1.

SCD

START CORE-TO-DRUM TRANSFER.

Mode	μs	Numerical Mode
N	18-24	040300
I	24-30	140300

This operation is the same as SDC, with the exception that information is being transferred from core to drum. All modes and modifier bits are as previously described.

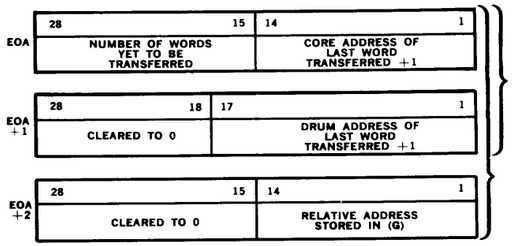
NIA = CIA + 1.

HDT

HALT DRUM TRANSFER

Mode	μs	Numerical Code	Indexable
N	24	001300	X
I	30	101300	X

The drum-to-core or core-to-drum transfer is stopped by this operation. Bits 23 through 26 of the instruction define the type of transfer operation being stopped as previously described. The OPF contains the address of the first of 2 or 3 consecutive core locations which will contain the following information after execution of HDT:



N	18
I	24
N	24
I	30

These words are stored, in proper format for restarting. A drum-core transfer-complete signal is generated by this instruction.

The Halt Mode transfer operation can be used to interpose a higher priority drum/core transfer while a transfer is in operation. At the end of the higher priority transfer, the computer is interrupted. A single instruction then initiates the resumption of the previous transfer by referring to the location where the data required to continue the transfer has been stored.

NIA = CIA + 1.

SECTION VI
PROGRAMMING INPUT-OUTPUT OPERATIONS

PRINCIPLE OF OPERATION

The 340 input/output design allows blocks of information to be transferred automatically between core memory and the input/output device. The Master I/O Controller buffers all peripheral devices, thus driving them all at maximum speed. Core memory accesses are shared by the Master I/O Controller and the rest of the computer. Thus programs run concurrently with input/output operations. The program need only initiate the transfer and the input/output system completes the transfer automatically. Direct contact input is also provided allowing programs to react quickly to emergency conditions.

MASTER I/O CONTROL AND INTERRUPT UNIT

The Master I/O Control and Interrupt Unit contains circuits for the processing of input, output, and priority interrupt operations. How it functions in the Interrupt Subsystem is described in Section IV.

The Master I/O Control and Interrupt Unit can control up to 128 input/output devices through their respective control units. The program activates a specific I/O function or device with a single ENABLE instruction and a setup of the proper boundary conditions. Beyond this, the Master I/O takes over and completes the input or output operation automatically at the maximum speed of the function or device. The Master I/O drives many input/output functions or devices at their maximum rates by interleaving their operations. (See Figure 6-18.)

The lower 7 bits of each ENABLE instruction define the I/O function or device to be enabled. These 7 bits also define a dedicated location in core memory assigned to the function or device. In this location a control word of two 14-bit fields is stored. The upper field contains n, the number of words of data to be transferred, and the lower field contains the first address of a sequential block of addresses which contain the data to be transferred.

The speed of operation of the Master I/O is governed by a fast counter with a 478 kc/s stepping rate. This counter steps through all possible states of all input/output functions and devices at 2 μseconds per step to determine if (1) the program has enabled a function or device, (2) the table specified for the function or device has not been exhausted, and (3) the function or device control unit is ready to accept more information. When these three conditions are obtained, a specific device is known to be "ready." The Central Processor is then signalled, and within 6 μseconds the contents of the Central Processor registers and the action of the fast counter are "frozen" while three direct memory accesses take place:

A. The control word in the dedicated location for this device is transferred to the J Register in the Master I/O controller.

B. Using the address in the lower half of the J Register, one word of data is transferred between the buffer register in the device control unit and the table in core memory.

C. The contents of the J Register are transferred back to the dedicated location. During the transfer the data block is decremented by one and the data word address is incremented by one. This operation updates the control word for the next operation.

The fast counter then continues testing each successive I/O function or device for a ready signal. As the counter reaches successive ready signals, it performs the three direct accesses described above for that device. The counter continues testing all I/O devices sequentially for ready signals. The counter may pass any one enabled device several times before it is again in a ready state. This is due to the relatively slow speed of I/O devices compared to the fast scanning speed of the counter.

The Master I/O Controller continues this scanning operation on any enabled device until the last data word is transferred. The device is automatically disabled and direct memory accessing stops.

I/O END-OF-TABLE OPERATIONS

An I/O transfer-complete flip-flop is set when an enabled device completes its operation. This flip-flop causes an automatic program interrupt. At the same time the enable code for the device is automatically stored in dedicated core memory location 37472_8. These enable codes are shown

in Table 6-1. The interrupt response program for this condition typically performs at least these functions:

(a) Examine location 37472_8 to see which device is finished.

(b) Reset the I/O end-of-table flip-flop using a RESET instruction with bit 5 a one, and reset inhibit-interrupt flip-flops with a one in bit 6.

(c) Initiate the transfer of the next block of data for this device if required.

Other devices which are finished remain enabled until the computer has processed the previous end-of-table interrupt, at which time the next finished device causes an interrupt. This procedure continues automatically until all finished devices are recognized and processed.

Table 6-1. Typical Function and Device Codes

Code (Octal)	Function or Device
000	Contact Input, Compare and Store Contents of Input Lines
001	Contact Input, Compare Only
002	Contact Input, Store Contents of Input Lines Only
003	Not Available
004	Multiple Contact Output
005	Not Available
006	Not Available
007	Analog, Limit Scan and Store Data
010	Analog, Limit Scan
011	Analog, Store Data
012	Not Available
013	Not Available
014	Output Typewriter No. 1
015	Output Typewriter No. 2
016	Keyboard No. 1
017	Card Reader
020	Paper Tape Reader
021	Unassigned
022	Line Printer (Status)
023	Unassigned
024	Teletype Input
025	Unassigned
026	Teletype Printer
027	Card Punch
030	Paper Tape Punch
031	Line Printer (Output)
032	Unassigned
033	Unassigned
034	Unassigned
035	Unassigned
036	Unassigned
037	Unassigned
040	Unassigned
041	Unassigned
042	Not Available
043	Unassigned
044 through 177	Unassigned

CONTROL WORD FORMAT

The control word format for dedicated core memory location for input/output devices is shown in Figure 6-1.

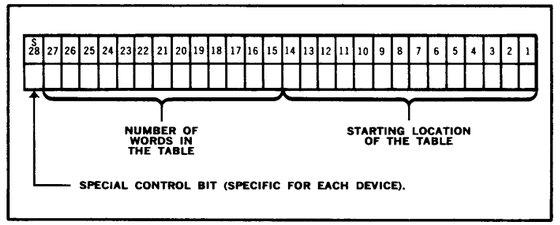


Figure 6-1. I/O Device Control Word Format

The control word indicates to the Master I/O Controller where the data is located in core memory and how many words are contained in the table to be transferred. Table 6-2 shows the assigned dedicated locations for standard I/O equipment.

Table 6-2.

Dedicated Control Word Location	Input/Output Device
37200	Contact Input Group Selection Table
37201	Contact Input Compare Table
37202	Contact Input Data Storage Table
37203	Not Available
37204	Multiple Contact Output Group Selection Table
37205	Multiple Contact Output Data Table
37206	Direct Contact Output Set or Reset
37207	Analog Input Selection Table
37210	Analog High-Low Limit Table
37211	Analog Input Data Storage Table
37212	Time of Day Counter
37213	Elapsed Time Counter
37214	Output Typewriter No. 1
37215	Output Typewriter No. 2
37216	Keyboard No. 1
37217	Card Reader
37220	Paper Tape Reader
37221	Unassigned
37222	Line Printer (Status)
37223	Unassigned
37224	Teletype Input
37225	Unassigned
37226	Teletype Printer
37227	Card Punch
37230	Paper Tape Punch
37231	Line Printer (Output)
37232	Unassigned
37233	Unassigned
37234	Unassigned
37235	Unassigned
37236	Unassigned
37237	Unassigned
37240	Unassigned
37241	Unassigned
37242	Not Available
37243	Unassigned
37244 through 37377	Unassigned

INPUT-OUTPUT INSTRUCTIONS

All input/output operations except Direct Contact Input are controlled by the ENABLE and DISABLE instructions. The enable codes for all input/output devices are shown in Table 6-1.

ENA

ENABLE

Mode	µs	Numerical Code
N	12	040600

The device specified in the operand field of this instruction is enabled. OPF bits 1 through 7 define the device, bit 9 controls the mode of operation of those devices having a dual mode capability. Bit 10 controls the on-off status of the motor in the device, with a one indicating that the motor is to be turned on, and a zero indicating that the motor is to be left in its previous state.

Each device must be enabled before any communication to or from core memory can occur. The control word in the dedicated location in core memory and any tables associated with the data transfer must be established before enabling any device. Any or all devices may be enabled at any given time.

NIA = CIA + 1.

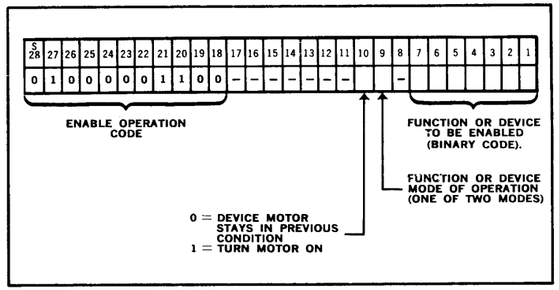


Figure 6-2. Device Enable Instruction Word Format

DIS

DISABLE

Mode	µs	Numerical Code
N	12	060600

The device or function specified by bits 1-7 of OPF is disabled. If bit 10 of OPF is a one, the device motor is turned off.

Any device may be Disabled at any time except the analog input device, which cannot be disabled. It is not necessary to use this instruction each time a device has finished transferring data, since the Master I/O Controller automatically disables the device at that time. If the device has a motor which must be turned off, however, the program must execute a disable instruction with bit 10 set to a one.

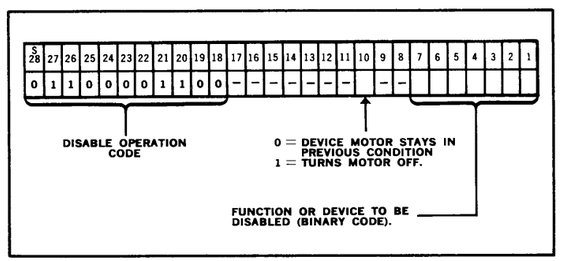


Figure 6-3. Device Disable Instruction Word Format

CIN

DIRECT CONTACT INPUT

The group of contact closure input lines, as specified in (OPF) bits 1-7, is loaded into the A Register.

By use of this instruction, the contents of a specified group of 28 contact input lines is placed directly into the A Register. This instruction enables the programmer to gain immediate access to the data in a group of contact inputs in the A Register without having to rely on securing the data from core memory via the ENABLING operation. Emergency situations may require this treatment. The Contact Input Control Unit need not be ENABLED prior to this instruction. The contacts of the input lines may be masked by the B Register while being transferred into the A Register.

NIA = CIA = 1.

Mode	µs	Numerical Code	Maskable
N	18	041600	X

Note: If specified, the input lines are masked by (B) prior to storage in (A).

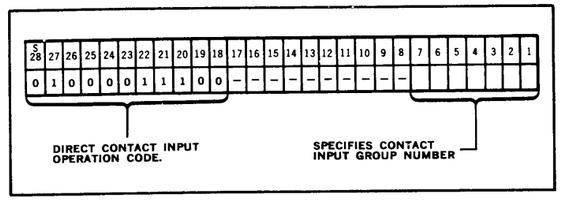


Figure 6-4. Direct Contact Input Instruction Word Format

PROGRAMMING CONTACT INPUTS

A. DESCRIPTION

The basic 340 Contact Input Subsystem is capable of accepting 1,792 contact inputs. The system enables the program to initiate contact input comparisons and/or data storing. Once the proper parameters have been established and the desired mode of operation Enabled, the Master I/O unit automatically processes the contact inputs. The comparison and/or storing operation continues until either a noncomparison or end-of-table condition occurs. The noncomparison interrupt condition occurs when the selected input group fails to compare with the specified comparison word in core memory. At that time all modes of operation in the Contact Input Control Unit are disabled. If the input storing mode is active, the group that did not compare will not be stored. The result of the attempted comparison is stored in location 37470_8 with the group number stored in location 37471_8.

B. MODES OF OPERATION

The 340 Contact Input Subsystem has three modes of operation:

1. Enable Code 000 initiates the compare and store mode of operation.
2. Enable Code 001 initiates the compare only mode.
3. Enable Code 002 initiates the data storing mode.

Three dedicated control word locations, 37200_8, 37201_8, and 37202_8, contain the location of the Group Selection Table, the Comparison Table, and the Contact Input Data Table respectively. If bit 28 in control word location 37200_8 is a one, the first contact group will be read n times. The words in the Group Selection Table contain the group address in bits 1-7. The words in the Comparison Table contain the 28 bits to be compared with the contact input data. The words in the Contact Input Data Table contain the status of the contacts after the completion of the Contact Input Operation. These dedicated locations and tables must be set up properly before the program enables the contact input operation.

C. TIMING

The computer time required to process each group of contacts is:

1. Compare and Store Mode : 66 microseconds
2. Compare only Mode : 42 microseconds
3. Store only Mode : 42 microseconds

The elapsed time between processing each group depends on how many other input/output devices are being controlled by the Master I/O Control Unit. For example, assume that there are a total of 10 other devices and all are disabled at this time. The elapsed time would then be 20 microseconds between groups. The percentage of computer time required during Contact Input Operation would be: 76 percent for Compare and Store Mode, 44 percent for Compare Only Mode, and 44 percent for Store Only Mode. The large percentage is due to the fact that contact inputs operate at computer speeds with solid state circuitry employed for the selection. Contact input operations are not continuous, but rather happen in spurts at the beginning of some programmed time interval. For instance, a typical system may scan a group of 112 contact inputs at the beginning of every 1 second period and therefore use effectively 0.03 percent of the computer's time.

PROGRAMMING MULTIPLE CONTACT OUTPUTS

A. DESCRIPTION

The 340 Multiple Contact Output Subsystem has the capability of 1,792 multiple contact output lines, in groups of 28 contacts per group. These output contacts can be used for many purposes, one of which is for driving resistance-divider circuits used to generate analog voltage outputs. The location of the Output Group Selection Table is specified in dedicated location 37204_8 and the location of the Output Data Table is specified in dedicated location 37205_8. Once these parameters have been specified, the Multiple Contact Output Subsystem is enabled with code 004_8. The Master I/O Control Unit then processes the outputs in sequence from these tables. The rate of output is governed by the individual length of the pulse specified for each group output. The word format for the Output Group Selection table is shown in Figure 6-5.

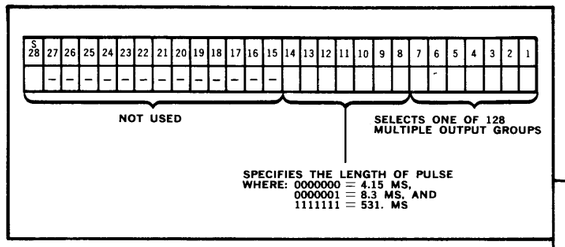


Figure 6-5. Multiple Contact Output Group Selection Table Word

The 28 bits of each word in the Output Data Table specify the desired status of the 28 contacts in the group. A one bit corresponds to a closed contact and a zero corresponds to an open contact. In the case of latching relays, 1's set the relay. In the case of momentary relays, 1's pulse the relay and 0's have no effect.

B. TIMING

The computer time required for each output operation is 36 microseconds and the elapsed time between operations is dependent on the individual pulse durations. Typically 0.90 percent of the computer's time is required for 100 VA relays, and 0.30 percent for 250 VA relays.

PROGRAMMING ANALOG INPUTS

A. DESCRIPTION

The Analog Control Unit controls the actions of high and low speed analog input multiplexers and an Analog-to-Digital Converter. This control unit also can control optional high speed analog output channels. Normal analog outputs are implemented by means of the multiple contact outputs.

The Analog Control Unit allows the programmer to direct the inputting, high speed outputting, and processing of analog data in three different ways, each way defined by separate input/output function codes. These three analog functions are defined by function codes 007_8, and 010_8, and 011_8. By Enabling the first function, the programmer can perform input selection, limit scanning, and data storage. The second function performs input selection and limit scanning only, and the third function performs input selection and data storage only. Only one function should be enabled at a time. If two or more functions are enabled, the control unit will perform the operation defined by the last function address given.

Once a particular analog function is Enabled, the computer loads in the assigned function response address for one or two control words, one of which gives the starting address of a block of analog input selection words, the number of words in the block (or the number of samples to be taken of a given input), and an indication of whether repeated sampling of a single input is required. The format of analog input selection words enables the programmer to specify; (1) mode selection, (2) input/output addressing, and (3) input signal conditioning. This format is shown in Figure 6-6.

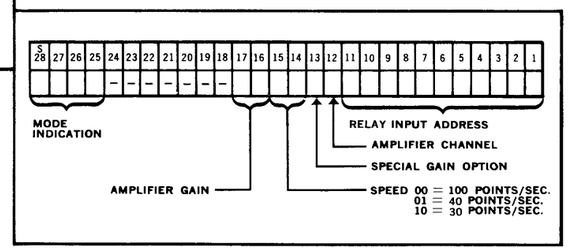


Figure 6-6. Analog Input Selection Word Format

1. Mode Selection

The Analog Control Unit provides the interface between the Master I/O Control Unit and the following analog devices:

(a) A normal speed ($\leqslant$ 100 points/second) input multiplexer (mercury relays) and digitizer which accepts high or low level input signals.

(b) Optional high speed ($\leqslant$ 8,000 points/second) input multiplexer (solid state) and digitizer which accepts high or low level signals.

(c) Optional high speed ($\leqslant$ 8,000 points/second) analog output channels.

Mode selection is controlled by the mode indicator bits in the input/output selection word. An input mode change is effective for the word in which the information appears. An input/output change takes a time period equal to the current operating mode.

Indicator bits specify:

Bit 28 — 0 = Inputs, 1 = Outputs

Bit 27 — 0 = No Priority, 1 = Priority

Bit 26 — 0 = Lo Speed Input, 1 = Hi Speed Input

Bit 25 — 0 = No Mode Change, 1 = Mode Change

Bit 26 selects the high speed or low speed multiplexer.

If bit 27 is a ONE, the analog control unit inhibits inputting or outputting from all other peripheral devices. This mode ensures that data transfer between the Master I/O and analog unit can occur at the maximum rate (8,000 points/second). If the high speed mode without priority is selected, maximum data rate is limited by the time interlace of any other I/O device communicating with the Master I/O Control Unit.

If Bit 28 is a ONE the control unit can communicate with optional high speed digital to analog output channels. Again maximum data rate can only be realized on a priority basis.

Bit 25 is a "strobe." Information in bits 26-28 will only take effect when bit 25 is a ONE. This bit allows the programmer to make mode selection once and then ignore mode selection on following analog input operations until a mode change is desired, at which time bit 25 must be a ONE.

2. Input/Output Addressing

The Analog Control Unit can provide random access to:

(a) 2,048 analog inputs sampled at speeds up to 100 points/second.

(b) 2,048 analog inputs sampled at speeds up to 8,000 points/second.

(c) 2,048 analog outputs which can be updated at a rate of 8,000 points/second.

If a specific system has both high and low speed input multiplexing capability, the analog control unit can provide random access to a total of 4,096 analog inputs, 2,048 at high speed and 2,048 at low speed. The maximum sampling or updating rate is 8,000 points/second. If high speed inputs and outputs are interlaced, maximum inputting and outputting rates are 4,000 points/second each.

The selection of analog inputs and outputs can be effected by the Analog Control Unit in blocks of up to 2,048 addresses.

For those systems utilizing the optional high speed analog output capability, the format of the high speed output selection/data word is shown in Figure 6-7.

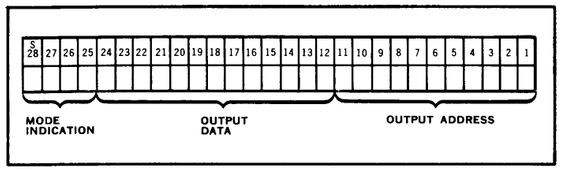


Figure 6-7. High Speed Analog Output Word Format

3. Input Signal Conditioning

Bits 12 through 17 of the analog input selection word control input signal conditioning. Bit 12 controls the selection of a second (redundant) amplifier which can be provided as an option. If bit 12 is a ZERO, the normal amplifier is used. If bit 12 is a ONE, the redundant amplifier is used.

Bits 14 and 15 control the conditioning of input signals with respect to noise rejection. At the normal inputting speed of 100 points/second, noise rejection is provided by amplifier band pass and input filtering. This level of conditioning is indicated by ZEROS in bits 14 and 15 (period code 00). Two other period codes are available, 01 and 10, to specify the use of time integral noise rejection techniques in the control unit. The use of period code 01 results in an inputting speed of 50 points/second; period code 10 results in an inputting speed of 35 points/second.

Bits 16 and 17 control the gain of the amplifier as follows:

GAIN CODE (bits 17 and 16)	AMPLIFIER GAIN
00	50
01	200
10	500
11	1000

If bit 13 is a ZERO, the normal amplifier is used. If bit 13 is a ONE, an optional buffer amplifier is used in addition to the normal amplifier. The latter configuration produces an overall gain of less than 50. The gain can be specified to satisfy the system requirements.

Depending on the analog functions desired, the programmer must specify two or three of the following control words before Enabling the function: (1) input selection control word, (2) high-low limit table control word, (3) data storage table control word. The formats (Figure 6-8) of these control words and their addresses are specified below.

Also before Enabling an analog input function, the programmer must specify and/or fill two or three of the following tables in core memory: (1) analog input selection table, (2) analog high-low limit table, (3) analog input data storage table. The format of the words in the analog input selection table is illustrated above. The analog high-low limit table contains a table of high and low limits against which each input is compared. Each word of this table has a high limit in bit positions 16 through 28, and a low limit in bit positions 2 through 14. Bit 1 and 15 are indicator bits through which the program can control the meaning of the analog high/low limit interrupt. If bit 1is a ZERO, an interrupt occurs if the input is not within the low limit, but if bit 1 is a ONE, the interrupt will occur if the input is within the low limit. Bit 15 exercises the same control for the high limit condition.

The analog input data storage table contains a block of unused words into which analog inputs will be stored, if storage is desired. Data is stored twice in the same word, in bits 2 through 14 and 16 through 28. The sign is in bits 14 and 28. Bits 1 and 15 are ZERO.

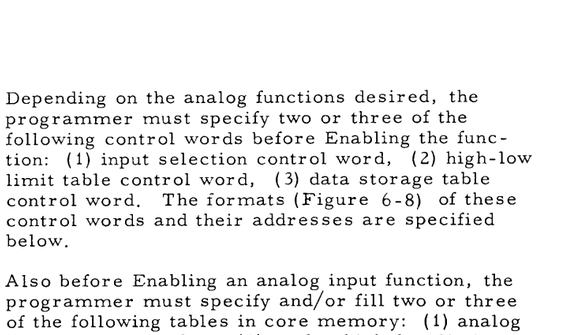


Figure 6-8. Analog Hi-Lo Limit and Input Data Word Format

An analog input which makes a limit transition either out of or back into limits initiates the following sequence of events:

(a) If the limit check shows the input to be out of limits, the Analog Limit Interrupt is set.

(b) The value of the input is stored in the analog input data storage table.

(c) All Enable Flip-flops are reset, inhibiting any further inputting of data in the analog subsystem.

It is not possible to directly disable the analog input functions by means of a DISABLE operation. In an emergency situation, however, the programmer can load a number into the analog input function control word such that n in the control word is equal to ZERO.

A second enable following a 007_8 enable has no effect. However, a second enable which requests data storage following a 010_8 enable will cause the data storage function to be performed. Similarly, a second enable which requests limit scan following a 011_8 enable will cause the limit scan function to be performed.

B. MODES OF OPERATION

The 340 Analog Input Subsystem has three modes of operation: (1) Enable code 007 initiates a limit scan and store data mode, (2) Enable code 010 initiates a limit scan mode only, (3) and Enable code 011 initiates a store data mode only. Three dedicated control word locations, 37207_8, 37210_8, and 37211_8, specify the location of the analog input selection table, the analog high-low limit table, and the analog input data table respectively. If bit 28 in location 37207_8 is a one, the first analog input will be sampled n times. If the bit 28 in location 37211_8 is a one, the analog input data readings will be stored on top of each other in the first word in the analog input data table. The formats for the words in the analog input selection table, analog high-low limit table, and analog input data storage table are shown above. These dedicated locations and tables must be set up properly before one of the three analog input modes is enabled.

C. TIMING

The computer time required for limit scan and store data mode is 66 microseconds, for limit scan mode is 42 microseconds, and for store data mode is 42 microseconds per analog input. The elapsed time between inputs is 10 milliseconds at 100 points per second scan rate. The percentage of computer time required for analog input operations are: 0.6 percent for limit scan and store data mode, 0.4 percent for limit scan only, and 0.4 percent for store data only.

PROGRAMMING ANALOG OUTPUTS

Analog outputs for 340 systems are implemented using the Multiple Contact Output Subsystem. Both resistance-divider circuits and set point stations are controlled using contact outputs from the computer. See paragraphs above for the details of programming Multiple Contact Outputs.

PROGRAMMING THE INPUT KEYBOARD

A. DESCRIPTION

The Model 34-417 Input Keyboard is "locked" as long as its control unit is Disabled. When the program Enables the Keyboard Control Unit the Keyboard becomes "unlocked." The operator can then enter data. A typical sequence follows:

1. Program will set up the Keyboard control word with n = 1.
2. Program Enables Keyboard Control Unit.
3. Operator presses a key and the Keyboard is locked at this point.
4. The Master I/O Control Unit stores the six-bit character into the first location of the input-data table.
5. The Master I/O Control Unit now interrupts the computer with the I/O End-of-Table interrupt, and disables the Keyboard Control Unit.
6. The program recognizes the character code as being the start of a data-input sequence.
7. The program sets up the Keyboard control word with n = 6.
8. The program Enables the Keyboard Control Unit.
9. The operator can now enter the data in blocks of six characters. The programmer may wish to print out each character or blocks of characters as they are entered.

Keyboard codes are shown in Appendix A.

B. OPERATION

The dedicated locations for the control words associated with Input Keyboard No. 1 and 2 are locations 372168 and 372218. These locations must be set up prior to enabling either keyboard. The Enable codes are 016 and 021 for keyboard 1 and 2 respectively. A table of n locations must be reserved for keyboard input data and has the format shown in Figure 6-9.

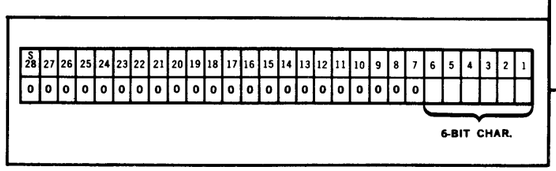


Figure 6-9. Keyboard Data Word Format

PROGRAMMING THE OUTPUT WRITERS

A. DESCRIPTION

Each Output Writer Control Unit controls two Output Writers. Output Writer 1 has priority over 2. This means that if both 1 and 2 are enabled, 1 will finish printing and be automatically disabled before 2 can print.

B. OPERATION

The dedicated control word location must be set up prior to enabling an Output Writer. The control word locations and enable codes for the standard Output Writers are shown below:

Output Writer	Control Word Location	Enable Code
1	37214	014
2	37215	015

Once these parameters are specified, the Master I/O Control Unit transfers the contents of each word in the output data table to the appropriate Output Writer Control Unit. Typing takes place at the maximum rate of 10 characters per second. The two possible formats for the output data table are shown in Figure 6-10.

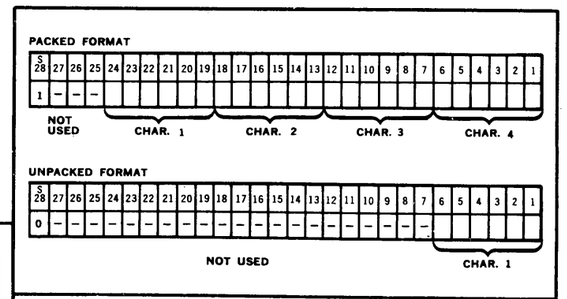


Figure 6-10. Packed/Unpacked Output Writer Data Word Formats

C. TIMING

The computer time required per operation is 18 microseconds and the elapsed time between outputs is 100 milliseconds or 400 milliseconds depending on whether the data is packed or unpacked. The percentage of computer time required is 0.0045 percent for packed data and 0.018 percent for unpacked data.

PROGRAMMING THE PAPER TAPE READER

A. DESCRIPTION

The Model 34-410 Paper Tape Reader reads at a maximum rate of 300 characters per second. When the tape is placed in the reader, the tape automatically runs up to and stops on the first character. No allowance must therefore be made for leader on the tape. Parity is checked on each character and an indication bit is stored in the data table with the character.

B. OPERATION

Dedicated control word location 37220_8 must be set up before the enable code of 020 is given by the program. Once these parameters have been specified, the Master I/O Control Unit processes characters from the reader at the maximum rate until n becomes zero. The tape then stops, and an I/O end-of-table interrupt is generated. A table of n locations must be set aside for input data and in the format shown in Figure 6-11.

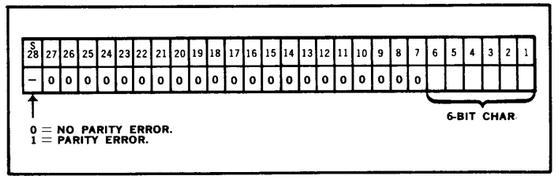


Figure 6-11. Paper Tape Reader Data Word Format

C. TIMING

The computer time required for each input is 18 microseconds with 3 milliseconds between each input. This means 0.6 percent of the computer's time is required while reading tape.

PROGRAMMING THE PAPER TAPE PUNCH

A. DESCRIPTION

The Model 34-412 Paper Tape Punch punches at a maximum rate of 110 characters per second. An odd parity bit is generated and punched with each character in position number 7 on the tape. A low-paper alarm line is available at the interface for use as an interrupt or contact input to the system program.

B. OPERATION

The dedicated control word location 37230_8 must be set up prior to the program giving an enable code of 030. Once these are specified, the Master I/O controller transfers data to the punch controller at the maximum rate. The format of the output data table is always unpacked and is shown in Figure 6-12.

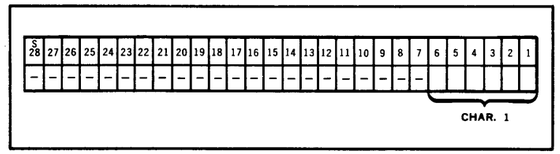


Figure 6-12. Paper Tape Punch Data Word Format

C. TIMING

The computer time required for each output is 18 microseconds every 9.5 milliseconds. Thus the punch requires 0.19 percent of the computer's time during punching operations.

PROGRAMMING THE CARD READER

A. DESCRIPTION

The Model 34-402 Card Reader reads at a maximum rate of 200 cards per minute and will read under a binary or character format. A line is available at the interface for indicating such things as: out of cards, card jam, and stacker full. Translation codes are shown in Appendix B.

B. OPERATION

The dedicated control word location 37217_8 must be set up prior to giving the enable code of 017. Once the Card Reader Control Unit is enabled, the Card Reader begins to feed cards at a rate of 200 cards per minute. The first column of data is available to the Master I/O Control Unit approximately 84 milliseconds after the card begins to feed and subsequent data arrives 2.5 milliseconds apart. After the 80th column, 104 milliseconds elapses before the first column on the next card is available. This timing information is not important to the programmer since once the Card Reader Control Unit is enabled, the Master I/O Control Unit automatically transfers the data into the input data table until n becomes zero. At which time, an end-of-table interrupt is generated. The number n, which specifies the number of columns to be read, may be any number less than

81. However, the moving card cannot be stopped and if n becomes zero before column 80 is processed, the remaining columns are ignored. Bit 10 of the ENABLE instruction specifies the mode of reading and causes the formats shown in Figure 6-13.

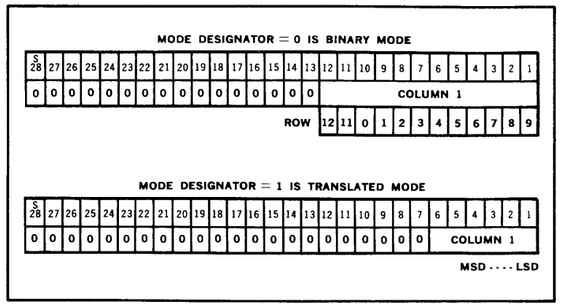


Figure 6-13. Card Reader Word Formats

C. TIMING

The computer time required for each column is 18 microseconds, thus 1.44 milliseconds for 80 columns. The time between columns is 2.5 milliseconds and, between cards is 104 milliseconds. 300 milliseconds is required for each card. The percentage of computer time required during reading is 0.48 percent.

PROGRAMMING THE CARD PUNCH

A. DESCRIPTION

The Model 34-404 Card Punch punches at a rate of 100 cards per minute. Once the Card Punch Control Unit is enabled, the Master I/O Control Unit scans the output data table twelve times. Each time, the Card Punch Control Unit converts the column data to row data and causes one row to be punched on the card, a maximum of 80 bits. After the twelfth row has been punched, the computer receives a transfer-complete interrupt. The program must initiate the control word and output data table for the next card within 20 milliseconds to attain the rate of 100 cards per minute. It should be noted that it is not necessary to set up a complete card image (80 columns) when less than that are to be punched.

B. OPERATION

The dedicated control word location 37227_8 must be set up before enable code 027 is given by the program. The value for n is given from 1 to 40 as indicated by the format of the output data table shown in Figure 6-14.

C. TIMING

The computer time required for each card is 5.8 milliseconds; for each row, 486 microseconds; and the time between rows is 42 milliseconds. Each card requires 600 milliseconds with 20 milliseconds between cards. The percentage of computer time required during punching is 0.98 percent.

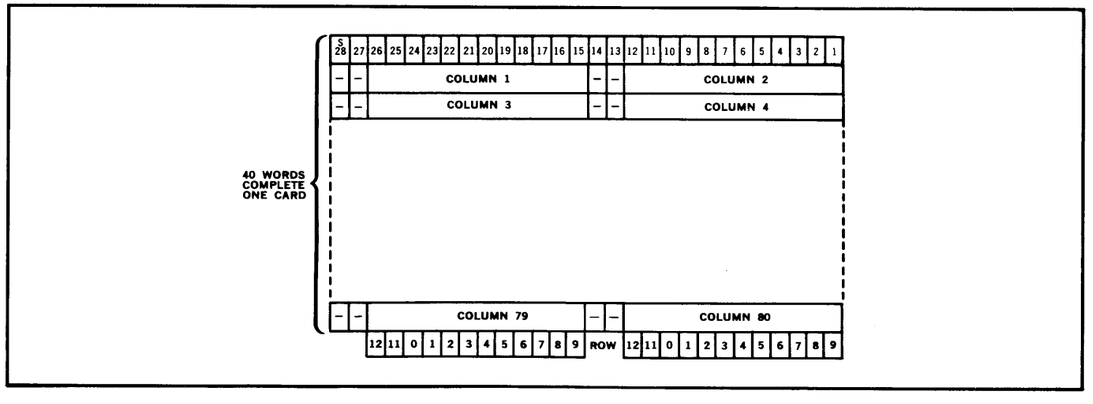


Figure 6-14. Card Punch Word Format

PROGRAMMING THE TELETYPE PRINTER

A. DESCRIPTION

Four Model 34-408 Teletype Printers may be driven in series by one Printer Control Unit. The printers operate at 10 characters per second. The character codes and selection format are shown in Appendix C.

B. OPERATION

The dedicated control word location 37226_8 must be set up prior to giving an enable code of 026. Once these parameters and the output data table have been specified, the Master I/O Control Unit transfers the contents of the output data table sequentially to the Printer Control Unit. The motor control bit is not used for this device. The format of the output data table is shown in Figure 6-15.

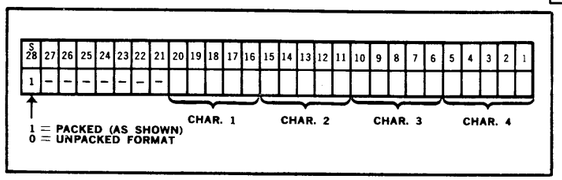


Figure 6-15. Teletype Printer Word Format

C. TIMING

The computer time required for each output operation is 18 microseconds every 400 milliseconds for packed data, and 100 milliseconds for unpacked data. Thus 0.0045 percent of computer time is required during output of packed data, and 0.018 percent for unpacked.

PROGRAMMING THE LINE PRINTER

A. DESCRIPTION

The Model 34-425 High-Speed Line Printer prints up to 120 characters per line at 300 lines per minute. Vertical line spacing is six per inch. Vertical formatting may be done using a punched paper tape loop or under program control.

The 64-character code set for the line printer is shown in Appendix D.

B. OPERATION

Two dedicated locations are required for line printer operation. 37222_8 is the Printer Status Control Word, and 37231_8 is the Line Printer Control Word.

1. Printer Status

Before enabling the printer, the program must determine the operating status of the printer. An Enable code of 022 stores the Printer Status Word in the address indicated by the Control Word 37222. The format of the Printer Status Word is shown in Figure 6-16. Note that a new print operation cannot be enabled unless bits 1-5 of the status word are zeros.

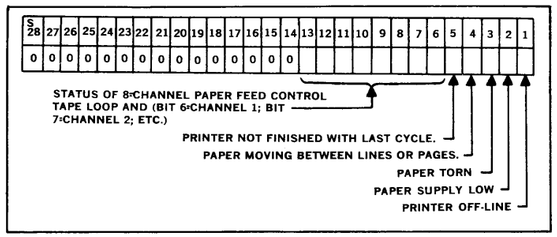


Figure 6-16. Printer Status Word Format

2. Printer Control

When it has been determined that the printer is operative (bits 1-5 are zeros), the program may enable the printer. The length and location of the output message (Printer Output Data Table) must be set up in dedicated location 37231 before the printer is enabled by a code of 031.

The first word of the Printer Output Data Table is a print control word; the subsequent words in the table contain characters to be printed, as shown in the table format in Figure 6-17. Note that the length specified in the Line Printer Control Word must be one more than the actual number of character words, to accommodate the print control word.

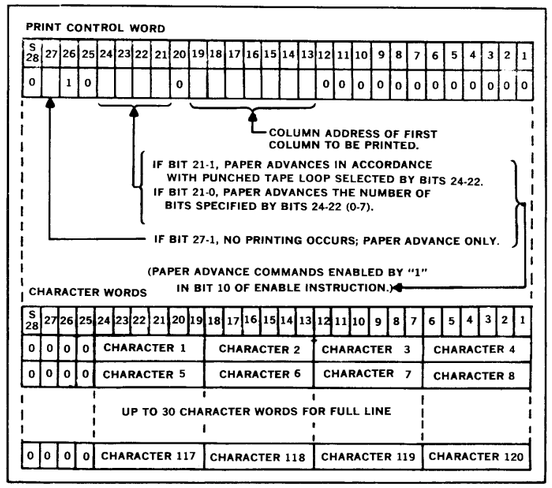


Figure 6-17. Printer Output Data Table Format

When bit 10 in the enable instruction is a ONE, the paper advance command in the print control word is interpreted; if bit 10 is a ZERO, the paper advance instructions are skipped.

Channel One on the punched paper tape format control loop must be punched for "top of form" only. This accommodates the "top of form" button on the printer.

C. TIMING

The computer time for each word transferred is 18 microseconds. The Line Printer Control Unit accepts one word each 60 microseconds; thus, for full line printing, 30 character words are transferred in 1800 microseconds to do the printing. Each full line of printing thus takes 201.8 milliseconds. Overall computer time required during continuous printing is 0.27 percent.

PROGRAMMING SPECIAL FUNCTIONS

A. TIME-OF-DAY COUNTER

1. Description

The Time-of-Day Counter control word location is used as a counter and is incremented each 16.67 milliseconds. The "ready" signal that causes the increase is synchronized with the 60 cycle AC line frequency. This core location can be treated by the programmer as any other core location. All 28 bits are used and the overflow indicator will not be SET when the last addition of one causes the count to initialize.

2. Specifications

Control-word address: 37212_8.

3. Timing

Computer time required for each operation: 12 μseconds.

Time between operations: 16.67 milliseconds.

Percentage of computer time required for each count: 0.072 percent.

B. CORE TIMER

1. Description

This Timer uses one word of core memory. The core timer control word is decremented by one immediately after the Time of Day Counter is incremented. It uses the same 60 cycle "ready" signal. All 28 bits are used and when the subtraction occurs that causes the number in these bits to become ZERO, a type 2 interrupt is generated by the Master I/O Control Unit. Again, this core location can be treated by the programmer the same as any other core location.

2. Specifications

Control word address: 37213_8.

3. Timing

Computer time required for each operation: 12 μseconds.

Time between operations: 16.67 milliseconds.

Percentage of computer time required for each count: 0.072 percent.

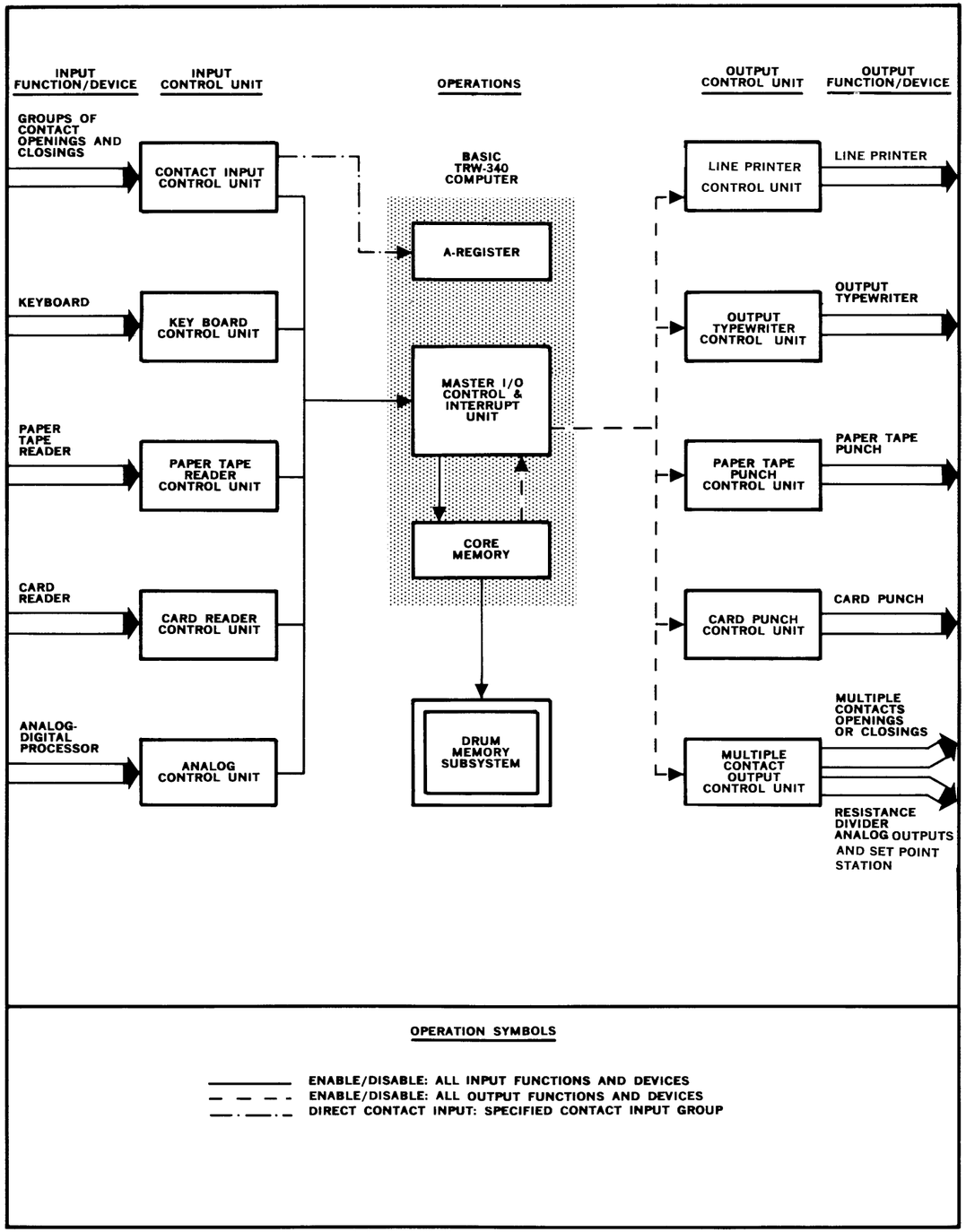


Figure 6-18. 340 Input-Output Operations

SECTION VII MAINTENANCE PANEL

MAINTENANCE PANEL

The Maintenance Panel (Figure 7-1) is located on the Central Processor cabinet. The panel contains the controls and indicators used by programming and maintenance personnel for off-line program checking and equipment maintenance. Functions that can be performed through the use of this panel include:

A. Entering data into the computer.

B. Off-line debugging of computer programs.

C. Circuit troubleshooting and maintenance.

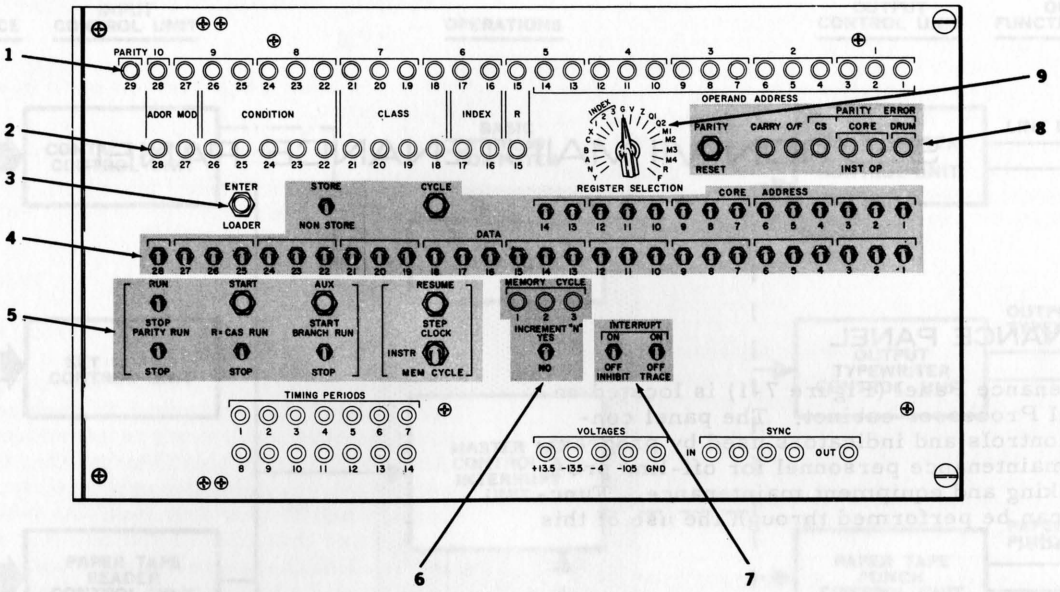
D. Changing portions of the program.

E. Stepping through programs by instructions, memory cycles, or clock time.

F. Loading bootstrap program into core memory.

G. Displaying the contents of all major registers during program execution in the step mode.

A Programming Panel, a system option, may be mounted on the Operator's Console to provide the same types of program-checking functions currently available with the Maintenance panel.



1 Neon indicators to display the contents of registers selected by REGISTER SELECTION switch 9

2 Neon indicators to display the operation code.

3 The ENTER LOADER button provides for transferring the first 32 words in drum to the first 32 core cells. These cells contain a bootstrap program using the panel switches to load additional information into core.

4 Switches to permit direct access to core memory in conjunction with the 14 CORE ADDRESS toggles and the 28 DATA toggles. The data word set up on the DATA toggles can be entered into the core address set up on the CORE ADDRESS toggles by setting the STORE-NON STORE switch to STORE, and pressing the CYCLE button. The contents of the core address specified by the CORE ADDRESS toggles is displayed on the neon indicators 1 if the STORE-NON STORE switch is in the NON-STORE position, the CYCLE button is pressed, and the REGISTER SELECTION switch is set to F.

5 Eight switches and buttons to control operation of the program. These switches provide for (a) stopping the computer program; (b) stepping through program by instruction, memory cycle, or clock time; (c) restarting computer at origin, at next instruction address, or auxiliary (emergency) origin; and (d) stepping through program from one Branch Unconditional instruction to the next. Provision is made to stop the computer when a core parity error is generated. The programmer can also set the address of an instruction at which he would like to stop on the CORE ADDRESS toggles. By operating the R=CAS switch, the program will run until the instruction address is the same as the setting of the CORE ADDRESS toggles, and then the program stops.

6 Three MEMORY CYCLE neon indicators and one switch are associated with the stepping mode described in 5 above. The MEMORY CYCLE indicators specify which memory cycle has just been completed. The INCREMENT "N" switch is a maintenance feature which allows repeated execution of the same instruction.

7 The two INTERRUPT switches permit continuous interruption of the computer, inhibition of all interrupts, or normal interrupt operation.

8 This section consists of four neon indicators to display the setting of PARITY ERROR detection flip-flops, two neon indicators which display the states of the O/F (Overflow) and CARRY flip-flops, and a PARITY RESET button which can set to zero all parity flip-flops.

9 The setting of the REGISTER SELECTION switch specifies the register to be displayed on neon indicators 1 .

The 24 test points at the bottom of the panel -- labelled TIMING PERIODS, VOLTAGES, and SYNC -- are used only during maintenance procedures for monitoring hardware operation.

Note: The Maintenance Panel does not contain a "run" light; however, a run condition is indicated by a flickering of the MEMORY CYCLE indicators in item (6).

Figure 7-1. Maintenance Panel Controls and Indicators

APPENDIX A
OUTPUT WRITER AND INPUT KEYBOARD CODES

Lower Case	Octal Code
A	26
B	27
C	30
D	31
E	32
F	33
G	34
H	35
I	36
J	37
K	40
L	41
M	42
N	43
Ø	44
P	45
Q	46
R	47
S	50
T	51
U	52
V	53
W	54
X	55
Y	56
Z	57

Upper Case	Lower Case	Octal Code
°(degree)	0	60
'	1	61
"	2	62
#	3	63
$	4	64
%	5	65
¢	6	66
&	7	67
?	8	70
:	9	71
_	,	24
^	*	72
.	.	25
=	+	22
θ	(	74
/	-	23
	SPACE	20
	C-RETURN	01
	TAB	02
	UPPER CASE	03
	LOWER CASE	04
	RED	05
	BLACK	06
$\frac{1}{4}$	$\frac{1}{2}$	21
X	)	73

APPENDIX B

READER CARD CODE AND BCD 6-LEVEL OUTPUT CODE TRANSLATION MODE

BCD 6-LEVEL	CARD CODE ZONE	CARD CODE NUM
010000	-	-
111011	12	8-3
111100	12	8-4
111101	12	8-5
111110	12	8-6
111111	12	8-7
110000	12	-
101011	11	8-3
101100	11	8-4
101101	11	8-5
101110	11	8-6
101111	11	8-7
100000	11	-
010001	0	1
011011	0	8-3
011100	0	8-4
011101	0	8-5
011110	0	8-6
011111	0	8-7
001011	-	8-3
001100	-	8-4
001101	-	8-5
001110	-	8-6
001111	-	8-7
111010	12	0
110001	12	1
110010	12	2
110011	12	3
110100	12	4
110101	12	5
110110	12	6
110111	12	7
111000	12	8

BCD 6-LEVEL	CARD CODE ZONE	CARD CODE NUM
111001	12	9
101010	11	0
100001	11	1
100010	11	2
100011	11	3
10100	11	4
100101	11	5
100110	11	6
100111	11	7
101000	11	8
101001	11	9
011010	0	8-2
010010	0	2
010011	0	3
010100	0	4
010101	0	5
010110	0	6
010111	0	7
011000	0	8
011001	0	9
001010	-	0
000001	-	1
000010	-	2
000011	-	3
000100	-	4
000101	-	5
000110	-	6
000111	-	7
001000	-	8
001001	-	9
000000	All other codes	

APPENDIX C TELETYPE PRINTER CODES

Code 8	Letters	Figures
03	A	-
31	B	?
16	C	:
11	D	$
01	E	3
15	F	!
32	G	&
24	H	#
06	I	8
13	J	'
17	K	(
22	L	)
34	M	.
14	N	,
30	O	9
26	P	0

Code 8	Letters	Figures
27	Q	1
12	R	4
05	S	BELL
20	T	5
07	U	7
36	V	;
23	W	2
35	X	/
25	Y	6
21	Z	"
00	BLANK	
37	LETTERS	
33	FIGURES	
04	SPACE	
10	CARRIAGE RETURN	
02	LINE FEED	

TELETYPE PRINTER SELECTION FORMATS

Turn all printers OFF: Figures, H

Turn all printers ON: Figures, H, Letters, U, Letters

Turn first printer ON: Figures, H, Letters, A, Letters

Turn second printer ON: Figures, H, Letters, B, Letters

Turn third printer ON: Figures, H, Letters, C, Letters

Turn fourth printer ON: Figures, H, Letters, D, Letters

APPENDIX D LINE PRINTER COPIES

Binary Code	Char	Binary Code	Char
000 000	0	100 000	- (dash)
000 001	1	100 001	J
000 010	2	100 010	K
000 011	3	100 011	L
000 100	4	100 100	M
000 101	5	100 101	N
000 110	6	100 110	O
000 111	7	100 111	P
001 000	8	101 000	Q
001 001	9	101 001	R
001 010	↑	101 010	←
001 011	=	101 011	$
001 100	' (apostrophe)	101 100	*
001 101	:	101 101	]
001 110	>	101 110	;
001 111	✓	101 111	@
010 000	+	110 000	(space)
010 001	A	110 001	/
010 010	B	110 010	S
010 011	C	110 011	T
010 100	D	110 100	U
010 101	E	110 101	V
010 110	F	110 110	W
010 111	G	110 111	X
011 000	H	111 000	Y
011 001	I	111 001	Z
011 010	?	111 010	%
011 011	. (period)	111 011	,
011 100	)	111 100	(
011 101	[	111 101	&
011 110	<	111 110	
011 111	#	111 111	"

APPENDIX E TABLE OF POWERS 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 F

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal)	0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal)	0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

	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)

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

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
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

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

	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

APPENDIX G
OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DECIMAL	OCTAL	DECIMAL	OCTAL	DECIMAL	OCTAL	DECIMAL
.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	OCTAL	DECIMAL	OCTAL	DECIMAL	OCTAL	DECIMAL
.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	OCTAL	DECIMAL	OCTAL	DECIMAL	OCTAL	DECIMAL
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX H INSTRUCTION FORMATS

MODE 28	MODE 27	OPERATION MODIFIERS 26	25	24	23	22	OPERATION CODE BASIC INST. FIELD 21	20	19	18	17	16	15	OPERAND ADDRESS FIELD 14 – 1
N = 0 / I = 1 / L = 0 / U = 1	0 / 0 / 1 / 1	NO MASK 0 / MASK 1	REGISTER				LOAD 1 0 0 1 / STORE 1 0 0 0				INDEX I_1 0 1 / I_2 1 0 / I_3 1 1		RELATIVE TRANSFER	
N = 0 / I = 1 / L = 0 / U = 1	0 / 0 / 1 / 1	NO. OF MULTIPLIER BITS					MULTIPLY 1 1 1 1							
		NO. OF QUOTIENT BITS					DIVIDE 1 1 0 1							
N = 0 / I = 1 / L = 0 / U = 1	0 / 0 / 1 / 1	MASK 1	NO Z = 0 / Z = 1	Add, Ch. Add, Subt. Dec. I_1, 2, 3 / Add, Subt. to Mem.			ARITHMETIC 1 1 0 1							
GR = 0 0 / LS = 0 1 / EQ = 1 0 / NEQ = 1 1		MASK 1	0	COMPARES			LOGICAL 0 1 1 1							
N = 0 / I = 1 / L = 0 / U = 1	0 / 0 / 1 / 1	MASK 1	$Z = 1$	EXTRACTS AND MERGES										
N = 0 / Br = 1	RAW = 0 / RWA = 1	MASK 1	REGISTER				REPLACE 1 0 1 0							
N = 0 / Br = 1	0	MASK 1	REGISTER				EXCHANGE 1 1 0 0							
N = 0 / I = 1	N = 0 / {X} = 1		15 BRANCHES / 1 NO OPERATION				BRANCH 0 0 0 0							
	N = 1 / {X} = 0		16 SHIFTS				SHIFT 1 0 1 1							
N = 0 / I = 1	1	SPECIFIC INSTRUCTION					I/O AND PROG. CONTROL 1 1 0 0							
N = 0 / I = 1	To Core = 0 / To Drum = 1	Not G = 0 / G = 1	No Parity 0 / Parity 1	No Imm = 0 / Imm = 1	No Rel = 0 / Rel = 1	Start = 0 / Stop = 1	DRUM/CORE TRANS. 0 1 1 0							
N = 0 / I = 1	0 / 0	64 ADDRESSES IN EACH GROUP; 5 GROUPS					OPER. EXT. 0 0 0 1 / 0 0 1 0 / . / . / 0 1 0 1							

APPENDIX I SUMMARY OF OPERATIONS

Class	Symbolic Code	Normal Numeric Code	Operation	Modes	Modes Indexable	Modes Maskable	Execution Time in μs		
							Normal	Indirect	L U or B
LOAD/STORE	LOD A	000440	Load A	NILUB	NI	NILU	12	18	6
	LOD B	016440	Load B	NILUUc	NI	NILUUc	12	18	6
	LOD C	002440	Load C	NILUB	NI	NILU	12	18	6
	LOD X	006440	Load X	NILB	NI	NIL	12	18	6
	LOD I1	003440	Load Index 1	NILB	NI	NIL	12	18	6
	LOD I2	004440	Load Index 2	NILB	NI	NIL	12	18	6
	LOD I3	005600	Load Index 3	NILB	NI	NIL	12	18	6
	LOD M	007440	Load Interrupt Mask M	NILUB	NI	NILU	12	18	6
	LOD Q	013440	Load Interrupt & Guard Cont. Q	NILUB	NI	NILU	12	18	6
	STR A	000400	Store A	NILUB	NILU	NILU	12	18	12, 6
	STR B	016400	Store B	NILU	NILU	----	12	18	12
	STR C	002400	Store C	NILUB	NILU	NILU	12	18	12, 6
	STR X	006400	Store X	NILB	NIL	NIL	12	18	12, 6
	STR I1	003400	Store Index 1	NILB	NIL	NIL	12	18	12, 6
	STR I2	004400	Store Index 2	NILB	NIL	NIL	12	18	12, 6
	STR 13	005400	Store Index 3	NILB	NIL	NIL	12	18	12, 6
	STR M	007400	Store Interrupt Mask M1	NILUB	NILU	NILU	12	18	12, 6
	STR Q	013400	Store Interrupt & Guard Cont. Q	NILUB	NILU	NILU	12	18	12, 6
	STR T2	015400	Store Data Toggles	NILUB	NILU	NILU	12	18	12, 6
ARITHMETIC	MPY n	000740	Multiply (n bits multiplier)	NILUB	NI	----	14+2n	20+2n	14+2n
	DIV n	000700	Divide (n bits quotient)	NILUBH	NIH	----	26+2n	32+2n	26+2n
	ADD	000640	Add	NILUBZ	NI	NILU	12	18	6
	SUB	002640	Subtract	NILUBZ	NI	NILU	12	18	6
	HWA	001640	Half Word Add	NILBZ	NI	NIL	12	18	6
	ADM	006640	Add to Memory	NIZ	NI	NI	18	24	-
	SBM	007640	Subtract from Memory	NIZ	NI	NI	18	24	-
	DIX I1	003640	Decrement Index I1	NILBZ	NI	NIL	12	18	6
	DIX I2	004640	Decrement Index I2	NILBZ	NI	NIL	12	18	6
	DIX I3	005640	Decrement Index I3	NILBZ	NI	NIL	12	18	6
LOGICAL	EXT	000340	Extract	NILUZ	NI	NILU	12	18	6
	EXM	005340	Extract to Memory	NIZ	NI	NI	18	24	-
	MRG	001340	Merge	NILUBZ	NI	NILU	12	18	6
	MGM	004340	Merge to Memory	NIZ	NI	NI	18	24	-
	XOR	002340	Exclusive or	NILZ	NI	NI	12	18	6
	COM EQ	043340	Compare Equal	N	N	N	12	-	-
	COM NE	143340	Compare Not Equal	N	N	N	12	-	-
	COM GR	003340	Compare Greater Than	N	N	N	12	-	-
	COM LS	103340	Compare Less Than	N	N	N	12	-	-
	SCN EQ	046340	Scan Equal	N	N	N			
	SCN NE	146340	Scan Not Equal	N	N	N	6+6c	where c=	
	SCN GR	006340	Scan Greater Than	N	N	N	words	scanned	
	SCN LS	106340	Scan Less Than	N	N	N			
	CMT EQ	047340	Compare Tables Equal	N	N	N			
	CMT NE	147340	Compare Tables Not Equal	N	N	N	6+12c	where	
	CMT GR	007340	Compare Tables Greater Than	N	N	N	c = words	compared.	
	CMT LS	107340	Compare Tables Less Than	N	N	N			
									(Br)
REPLACE/ EXCHANGE	RAW C	002500	Replace A with C	NBr	-	NBr	6	-	6
	RAW I1	003500	Replace A with Index 1	NBr	-	NBr	6	-	6
	RAW I2	004500	Replace A with Index 2	NBr	-	NBr	6	-	6
	RAW I3	005500	Replace A with Index 3	NBr	-	NBr	6	-	6
	RAW XH	001500	Replace A with x, Hold Upper	NBr	-	NBr	6	-	6
	RAW XC	006500	Replace A with x, Clear Upper	NBr	-	NBr	6	-	6
	RAW M	007500	Replace A with Int. Mask M	NBr	-	NBr	6	-	6
	RAW Q	013500	Replace A with Int. Cont. Q	NBr	-	NBr	6	-	6
	RAW G	016500	Replace A with G	NBr	-	NBr	6	-	6
	RAW T1	017500	Replace A with Address Toggles	NBr	-	NBr	6	-	6
	RAW T2	015500	Replace A with Data Toggles	NBr	-	NBr	6	-	6
	RWA C	042500	Replace C with A	NBr	-	NBr	6	-	6
	RWA I1	043500	Replace I1 with A	NBr	-	NBr	6	-	6
	RWA I2	044500	Replace I2 with A	NBr	-	NBr	6	-	6
	RWA I3	045500	Replace I3 with A	NBr	-	NBr	6	-	6
									(Br)
	RWA X	046500	Replace X with A	NBr		NBr	6	-	6
	RWA M	047500	Replace Int. Mask M with A	NBr	-		12	-	12
	RWA Q	053500	Replace Int. Cont. Q with A	NBr	-		12	-	12
	RWA G	056500	Replace G with A	NBr	-	NBr	6	-	6
	EAW B	016600	Exchange A with B	NBr	-	-	6	-	6
	EAW C	002600	Exchange A with C	NBr	-	NBr	6	-	6
	EAW XH	001600	Exchange A with X, Hold Upper	NBr	-	NBr	6	-	6
	EAW XC	006600	Exchange A with X, Clear Upper	NBr	-	NBr	6	-	6
	EAW I1	003600	Exchange A with Index 1	NBr	-	NBr	12	-	12
	EAW I2	004600	Exchange A with Index 2	NBr	-	NBr	12	-	12
	EAW I3	005600	Exchange A with Index 3	NBr	-	NBr	12	-	12
	EAW M	007600	Exchange A with Int. Mask M	NBr	-	NBr	12	-	12
	EAW Q	0013600	Exchange A with Int. Cont. Q	NBr	-	NBr	12	-	12
								(Condition Met)	

Class	Symbolic Code	Normal Numeric Code	Operation	Modes	Modes Indexable	Modes Maskable	Execution Time in μs		
							Normal	Indirect	L, U or B
BRANCH	BUN	000000	Branch Unconditionally	NIX	NI	-	6	12	6
	BZE	016000	Branch if A Zero	NIX	NI	-	6	12	6
	BZM	001000	Branch if A Masked Zero	NIX	NI	-	6	12	6
	BNZ	012000	Branch if A not Zero	NIX	NI	-	6	12	6
	BNM	002000	Branch if A Masked not Zero	NIX	NI	-	6	12	6
	BPO	013000	Branch if A Positive	NIX	NI	-	6	12	6
	BNG	003000	Branch if A Negative	NIX	NI	-	6	12	6
	BLB	004000	Branch if A Contains Low Bit	NIX	NI	-	6	12	6
	BNXI1	007000	Branch if Index 1 not Zero	NIX	NI	-	6	12	6
	BNXI2	010000	Branch if Index 2 not Zero	NIX	NI	-	6	12	6
	BNXI3	011000	Branch if Index 3 not Zero	NIX	NI	-	6	12	6
	BOF	005000	Branch if Overflow Indicator On	NIX	NI	-	6	12	6
	BCY	015000	Branch if Carry Indicator On	NIX	NI	-	6	12	6
	BCP	006000	Branch if Core Parity Error	NIX	NI	-	6	12	6
	BDP	014000	Branch if Drum or Direct Access Parity Error	NIX	NI	-	6	12	6
SHIFT/ NORMALIZE	ALA	040540	Shift A Left Arithmetic	NX	-	-			
	ALO	041540	Shift A Left Logical Open	NX	-	-			
	ALC	042540	Shift A Left Logical Closed	NX	-	-			
	ARA	043540	Shift A Right Arithmetic	NX	-	-			
	ARO	044540	Shift A Right Logical Open	NX	-	-			
	BLA	045540	Shift A, B Left Arithmetic	NX	-	-	8+2n where		
	BLL	046540	Shift A, B Left (A Arith, B Log) Open	NX	-	-	n = places shifted		
	BLO	047540	Shift A, B Left Logical Open	NX	-	-			
	BLC	050540	Shift A, B Left Logical Closed	NX	-	-			
	BRA	052540	Shift A, B Right Arithmetic	NX	-	-			
	BRL	051540	Shift A, B Right (A Arith, B Log Open	NX	-	-			
	NAA	053540	Normalize A Arithmetic	NX	-	-			
	NAO	054540	Normalize A Logical Open	NX	-	-	8+2c where		
	NAC	055540	Normalize A Logical Continued	NX	-	-	c = number of shift		
	NBA	056540	Normalize A, B Arithmetic	NX	-	-	to normalize		
	NBL	057540	Normalize A, B (A Arith, B Log) Open	NX	-	-			
PROGRAM CONTROL	NOP	017000	No operation	N	-	-	6	6	-
	STP	046600	Stop and Branch	NI	NI	-	6	12	-
	SRB	045600	Set Return and Branch	NI	NI	-	12	18	-
	SET	064600	Set	N	-	-	6	-	-
	RST	044600	Reset	N	-	-	6	-	-
	STA	047600	Set Core Address	N	-	-	6	-	-
INPUT/OUTPUT CONTROL	ENA	040600	Enable	N	-	-	12	-	-
	DIS	060600	Disable	N	-	-	12	-	-
	CIN	041600	Direct Contact Input	N	-	N	18	-	-
DRUM/CORE TRANSFER	SDC	000300	Start Drum-to-Core Transfer	NI	NI	-	18-24	24-30	-
	SCD	040300	Start Core-to-Drum Transfer	NI	NI	-	18-24	24-30	-
	HDT	001300	Halt Drum Transfer	NI	NI	-	24	30	-
OPERATION EXTENSION	OEX	variable	Operation Extension	NI	NI	-	12	18	-