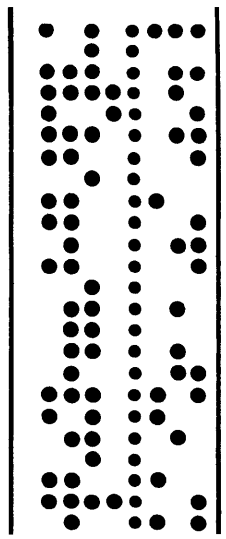
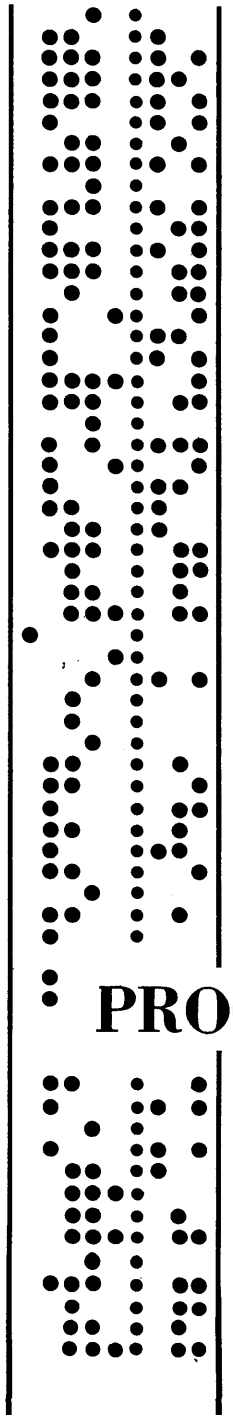


TP 1134

RCA 110 COMPUTER



PROGRAMMERS' REFERENCE MANUAL

AUGUST 1962

FOREWORD

The purpose of this manual is to delineate the programming features of the RCA 110 Computer and to provide a convenient guide to those features. The extreme versatility of the RCA 110 Computer will become apparent to the programmer as he becomes familiar with its programming characteristics.

The RCA 110 Computer is a general-purpose computer with powerful input/output capabilities for real-time process control. Thus, this manual discusses general-purpose programming with special emphasis on input/output methods. The manual has been organized into five sections; each section is a self-contained unit which pertains to various basic functions or characteristics of the computer.

- COMPUTER ORGANIZATION - Basic logical organization with emphasis on special functions such as priority interrupt and automatic trapping.

- INSTRUCTION DESCRIPTIONS - Detailed explanation of the functions of each instruction in the instruction repertoire of the RCA 110 Computer. Address-modification features are also discussed.

- INPUT/OUTPUT - RCA 110 Computer Input/Output functions, devices, and programming methods, with special emphasis on magnetic tape.

- COMPUTER OPERATION - Methods of computer operation of interest and importance to the programmer.

- APPENDIXES OF BASIC INFORMATION - Basic discussion of arithmetic, input/output assignments, standard core locations, and number conversion tables for the RCA 110 Computer.

This manual for the RCA 110 Computer provides a basic, thorough, and facile source of reference for the programmer. It is recognized that the computer state-of-the-art is constantly changing and that no reference manual can be all inclusive. Therefore, as new programming techniques are discovered, they will be made available to the persons concerned with programming and operating the RCA 110 Computer.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

LIST OF ILLUSTRATIONS

LIST OF TABLES

RCA 110 COMPUTER FEATURES

The RCA 110 Computer is a high-speed digital computer with solid-state circuitry and serial logic. It is organized with a fixed-word length of 24 bits and with all basic arithmetic performed in fixed-point binary. A brief synopsis of the RCA 110 Computer functional features are:

GENERAL

- serial logic
- solid state
- 936 kc internal clock rate
- 24-bit word size
- fixed-point arithmetic
- single-address instructions
- repertoire - 72 instructions
- 7 index registers (stored in memory)
- left and right accumulators (L and R)
- 4 levels of priority interrupt (2 programs per level)

BASIC TIMING

- word time - 28.85 microseconds
- add/subtract - 57.7 microseconds
- multiply - 779 microseconds
- divide - 865 microseconds

DATA AND INSTRUCTION STORAGE

- high-speed coincident-current core memory (HSM)
  - 3.5-microsecond memory-access time
  - 10.25-microsecond memory-cycle time
  - 512 to 4,096 words of storage
  - word size - 24 bits plus parity bit
- bulk storage - magnetic drum
  - average access time - 8.5 milliseconds
  - maximum access time - 17.0 milliseconds
  - drum speed - 3600 rpm
  - word size - 24 bits plus parity bit
  - 4,096 to 32,768 words of main storage capacity (32 to 256 tracks of 128 words each, with additional storage capacity available)
  - up to 16 buffer tracks

INPUT/OUTPUT CAPABILITIES

- magnetic-tape stations (15,000 characters/second) (1 to 10 stations)
- paper-tape reader (60 characters/second)
- paper-tape punch (60 characters/second)
- monitor typewriter (10 characters/second)
- input/output buffer registers (1 to 8)
- I/O sense lines - 24 lines/set (1 to 8 sets)
- I/O address lines - 24 lines/set (1 to 8 sets)
- programmed simultaneity

SECTION 1

COMPUTER ORGANIZATION

1.1 GENERAL

The RCA 110 Computer is a general-purpose, fast, and highly accurate digital computer capable of automatic monitoring and control. It has many industrial and military applications.

For some purposes, the computer may consist of the basic mainframe; for other uses it may be combined with a variety of peripheral equipment to form a complex computer system. Regardless of the ultimate use, the basic computer concept is the same in all applications. In any digital computer, there are five basic operational functions. These basic functions are shown in Figure 1-1.

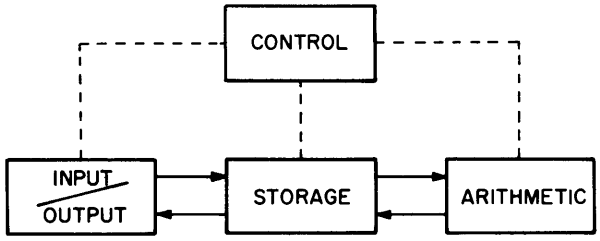


Figure 1-1. Typical Digital Computer, Block Diagram

The same operational sections as applied to an RCA 110 Computer system are:

- INPUT/OUTPUT — Information is transferred into, and out of, the computer by means of input/output devices. Data to be processed, or programs to be performed, are "read" into the machine by paper tape, or by magnetic-tape readers, or by other peripheral equipment. Information is returned from the computer by a paper-tape punch, a magnetic-tape recorder, typewriter, or other type of visual display. Within the computer are several registers that sense, select, and control the information to, and from, the input/output equipment.

- CONTROL — The control section is the command unit. It governs all operations in the machine such as information transfers, arithmetic performance, and the sequence of instructions. The control section may be a complete unit consisting of several registers, such as the program counter, the instruction register, and the timer.

- ARITHMETIC — This section of a computer performs mathematical operations: addition, subtraction, multiplication, and division. It also performs "logical" operations. The arithmetic section will contain such units as the left and right accumulators, the adder, and the counter.

- STORAGE — The storage unit is where information is placed (in machine language) until it is required for use during program execution. "Memory" is usually referred to as the storage within the computer. Information is retained in units such as a coincident core or a magnetic drum. Storage outside the computer is generally on paper or magnetic tape.

1.2 STORAGE

An important factor affecting the speed of the computer is the memory system. The RCA 110 Computer memory consists of a high-speed coincident-current core and magnetic drum.

The core normally has a storage capacity of 4096 words, with an access time of 3.5 microseconds and a cycle time of 10.25 microseconds.

The drum has a maximum of 256 tracks with 128 words per track, and can store as many as 32,768 words. The average access time for the drum is 8.5 milliseconds with a maximum of 17 milliseconds.

1.3 REGISTERS

The RCA 110 Computer contains several arithmetic and control registers which are of interest to the programmer. These registers perform the basic arithmetic operations and control the sequential operation of the computer.

1.3.1 Left Accumulator (L)

The Left Accumulator (L) is the principal arithmetic register, where most operations are performed. Containing 24 bits, it is used to hold an operand and most of the results of all arithmetic operations. Most conditional control transfers are also dependent on the contents of L.

1.3.2 Right Accumulator (R)

This register contains 24 bits and is primarily used as an extension of L. It holds the least significant half of double-length operands and results. R is also used to hold the remainder in division or the multiplier in multiplication.

1.3.3 Index Registers (XR)

The Index Registers (XR) are actually seven consecutive HSM locations (0001 to 0007) which are individually addressable. They are treated as 12-bit registers when an instruction is being interpreted. The primary function of the index registers is to provide for address modification; however, they may also be used for counting and looping. The status of an XR may be tested by the use of the instructions TXI and TXD.

1.3.4 Program Counter (PC)

The program counter is a 12-bit register which controls the sequential operation of the computer. At the conclusion of each instruction, the PC is updated, and specifies to the computer the HSM location from which the next instruction is to be taken.

1.3.5 Priority Registers

A system of automatic program interrupt is available with the RCA 110 Computer. This system allows a program with a higher priority to interrupt a current program on the computer. A full description of priority interrupt appears in Section 1.5.2.

Priority Request Register (JR) — This is an 8-bit register which holds incoming priority interrupt requests from eight request lines until the computer can process them.

Priority Status Register (JS) — This is an 8-bit register which holds the priority level and program number of the program on which the computer is currently working.

1.4 MACHINE ORGANIZATION

1.4.1 Instruction Word Format

The RCA 110 Computer has a fixed-word length, using two basic types of 24-bit computer words — an instruction word and a data word.

1.4.2 Data Word

The two types of data words are numeric and alphanumeric. Numeric data words consist of data stored in binary form as 23 magnitude bits plus a sign bit. The binary point is fixed and is placed to the left of the magnitude bits. Each bit occupies a particular bit position which is numbered from right to left, bit position 0-23. Each bit position corresponds to a power of 2. (See Appendix A for a discussion of binary numbers). A zero in bit 23 (the sign position) is used to signify positive number, and a one signifies a negative number. Negative numbers are stored as sign plus two's complement. A numeric data word as used in the RCA 110 Computer is illustrated in Figure 1-2.

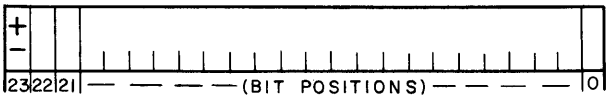


Figure 1-2. Numeric Data Word

An alphanumeric data word consists of a special configuration of bits which can uniquely represent an entire

character set. In the RCA 110, six bits are required for each alphanumeric character; therefore, four alphanumeric characters can be stored in one computer word. An alphanumeric data word as used in the RCA 110 Computer is illustrated in Figure 1-3.

Figure 1-3. Alphanumeric Data Word

1.4.3 Instruction Word

An instruction word consists of three basic groups of bits which are used to provide information for performance of various operations on the RCA 110 Computer. The RCA 110 Computer instruction word is illustrated in Figure 1-4.

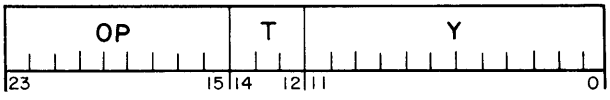


Figure 1-4. Instruction Word

The three basic parts of the instruction word are the operation code, the indexer, and the operand address.

OPERATION CODE — The operation code, or OP portion of the instruction word, contains 9 bits (bits 15-23) which specify the operation to be performed. On instructions TXI and TXD which modify the index registers, bits 15-17 are used to specify K, the amount the index register is to be modified ($0 \leq K \leq 7$). With the input/output instructions SUN, STN, LDN, and SDN bits 15-17 are used to specify N, the I/O register, sense lines, or address lines, affected by the instruction ($0 \leq N \leq 7$).

INDEXER — The indexer, or T portion (bits 12-14) of the instruction word, contains 3 bits which specify an index register to be used ($1 \leq T \leq 7$) for address modification.

OPERAND ADDRESS — The operand address, or Y portion (bits 0-11) of the instruction word, contains 12 bits which specify a memory address in HSM for the operand, on most instructions. However, on the transfer instruction, bits 0-11 specify the address to which the control is to be transferred.

1.5 SPECIAL CONDITIONS

1.5.1 Trapping

Trapping is the operation whereby the computer automatically transfers control to standard locations when special conditions occur. There are three types of trapping operations that can occur on the RCA 110 Computer: (1) Overflow traps, (2) Alarm traps, and (3) Power failure. The ALARM switch on the Maintenance and Control panel must be in the JUMP position for the computer to store information and transfer control to a standard location (except for Power failure). Trapping operations require three word times in addition to the quoted instruction timing.

OVERFLOW TRAPS — Overflow is the condition which exists when the result of an arithmetic operation exceeds the length of the arithmetic register. Overflow can be caused by any of the arithmetic instructions: Add, Subtract, and Divide. However, trapping will not occur on an overflow unless the special instructions provided for trapping are used. (Refer to Section 2. 2.)

If an overflow trap is initiated, the contents of the program counter (PC) will automatically be stored in HSM location 0023_8 (bits 0-11), and control will be transferred to HSM location 0013_8. HSM location 0023 will then contain the HSM location of the instruction being executed when the overflow occurred.

STD Location	Data Transferred	Alarm
0023	Contents of PC	OVERFLOW: BINARY DIVIDE
0013	Next instruction	

ALARM TRAPS — There are four conditions during computer operation that can cause alarm trapping:

- Parity error detected in reading from, or writing to, the HSM.
- Parity error detected in reading from, or writing to, the drum.
- Parity error detected in reading from, or writing to, an input/output device.
- Detection of an attempt to execute an undefined instruction.

STD Location	Data Transferred	Alarm
0027	Location of current instruction when alarm occurs	PARITY: HSM DRUM I/O NO INSTR
0017	Next instruction	

POWER FAILURE — In case of a power failure, the contents of memory are not destroyed. The computer will continue to process the instruction active at the time of failure, then automatically store program information as shown:

STD Location	Data Transferred	Alarm
0040	Contents of PC and JS	POWER FAILURE
0041	Contents of L	
0042	Contents of R	

In the event of a power failure during a drum transfer or an input/output instruction (LDN or STN), the program will be interrupted and the incrementing of PC will be inhibited. The contents of PC, JS, L, and R will be stored in standard locations. When operation is resumed following a power failure, the Return after Interrupt (RAI) instruction addressed to location 0040_8 must be used. This restores conditions present when the power failed.

1.5.2 Priority Interrupt

Automatic priority interrupt is available on the RCA 110 Computer. Priority interrupt is the function that permits an external signal to interrupt the current program and transfer the computer operation to a routine of higher priority. Relative priority is determined by the wiring of the priority request input lines.

Upon the occurrence of a priority interrupt, the contents of the Program Counter (PC), the Priority Status register (JS), the Left Accumulator (L), and the Right accumulator (R) are automatically stored in standard memory locations. When the interrupting routine has been completed, the conditions of the interrupted program are restored through the use of the RAI instruction.

The computer has five levels of operation, of which four levels are priority, with two programs possible on each level. Each interrupt signal is stored in the Priority Request register (JR) until it is executed. The priority of the current program is stored in the Priority Status register (JS). The highest priority corresponds to Interrupt Level 4. The program which has no priority assigned will assume the lowest level.

The information flow, which takes place when an interrupt occurs, is as follows:

At the end of each instruction, the JS and JR registers are automatically compared. If a higher priority request is sensed (JR > JS), the contents of the PC are stored automatically in one of four standard locations corresponding to the priority level of the interrupting program.

On the READ or WRITE Drum instructions, the contents of the PC are the HSM location of the read or write instructions. On all other instructions, the contents of the PC are the HSM location of the next instruction to be executed when control is transferred back to the interrupted program. APPENDIX F lists the standard location.

The contents of L and R are also stored, and program control is transferred to one of the eight standard locations specified by the interrupt. The entire storage procedure requires five word times from the start of an interrupt to the beginning of the first instruction in the interrupting program.

The last instruction performed in a priority program must be a Return after Interrupt (RAI) instruction in order to return to the next instruction of the program which was interrupted. If the interrupted instruction was a READ or WRITE Drum instruction, an RAI instruction restores the contents of L and JS and places the contents of the stored PC in the Right accumulator. Program control is transferred to HSM location 0000_8. This location contains the starting address of the first instruction in a subroutine, which determines, from the contents of R and L, the proper starting address, the number of words that remain to be transferred, the starting drum track, and sector address.

Priority interrupt may be inhibited in two ways: by the INTERRUPT CONTROL switch or by programming. These methods are discussed in Section 4. 1. 6 and 2. 6.

SECTION 2

INSTRUCTION DESCRIPTION

This section provides detailed descriptions of each RCA 110 Computer instruction as well as a brief discussion of address modification.

The RCA 110 Computer instruction descriptions are divided into the following categories:

- Load and Store Instructions
- Arithmetic Instructions
- Logical Instructions
- Shift Instructions
- Transfer Instructions
- Control Instructions
- Input/Output Instructions

For a list of the instructions and their mnemonic codes in alphabetical order and octal codes in numerical order refer to Appendixes J and K.

The format for the instruction is illustrated in Figure 2-1. Preceding each figure is the alphabetic code which identifies the instruction. The timing for each instruction is given in "x" word times, where one word time is 28.85 microseconds. An additional word time must be added if an index register (XR) is used for address modification. All instructions may have modification of address, except those where bits 0-14 are not interpreted and also on TXI and TXD instructions.

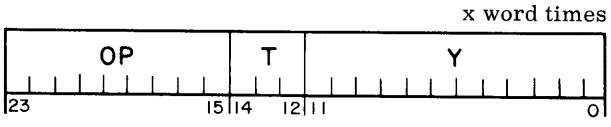
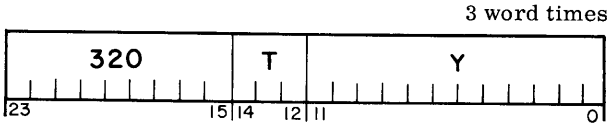


Figure 2-1. Sample Format of Instruction

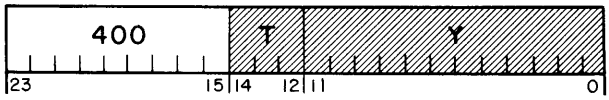
2.1 LOAD AND STORE INSTRUCTIONS

STP — Store PC Complemented



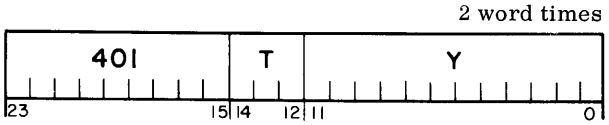
The contents of the Program Counter (PC) are two's complemented, and the result replaces the contents of Y, bits 0-11. At the time of execution, the PC contains the HSM location of the STP instruction. The contents of Y, bits 12-23, are unchanged. Y may be modified by T.

LDZ — Load Zero into Left Accumulator



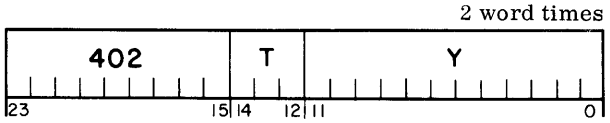
The contents of L, bits 0-23, are set to zero. Bits 0-14 of the LDZ instruction are not interpreted.

LDL — Load Left Accumulator



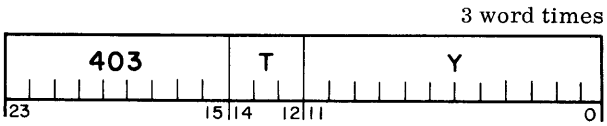
The contents of Y replace the contents of L. The contents of Y are unchanged. Y may be modified by T.

LDR — Load Right Accumulator



The contents of Y replace the contents of R. The contents of Y are unchanged. Y may be modified by T.

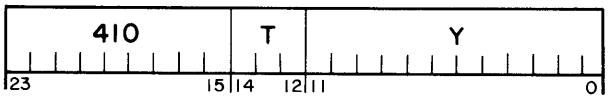
LDB — Load Both



The contents of Y replace the contents of R; the contents of Y + 1 replace the contents of L. The contents of Y and Y + 1 are unchanged. Y may be modified by T.

LDA — Load Address

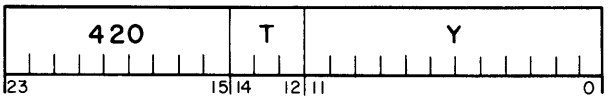
2 word times



The contents of Y, bits 0-11, replace the contents of L, bits 0-11. The contents of L, bits 12-23, and the contents of Y are unchanged. Y may be modified by T.

STZ — Store Zero into Y

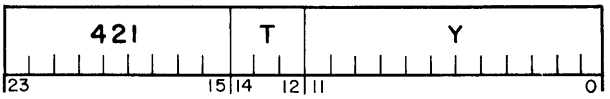
3 word times



The contents of Y, bits 0-23, are set to zero. Y may be modified by T.

STL — Store Left Accumulator

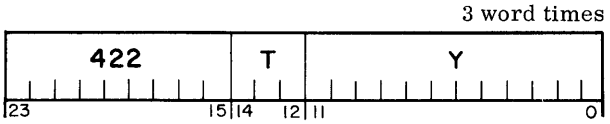
3 word times



The contents of L replace the contents of Y. The contents of L are unchanged. Y may be modified by T.

STR — Store Right Accumulator

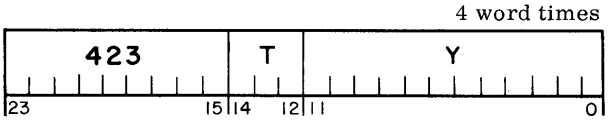
3 word times



The contents of R replace the contents of Y. The contents of R are unchanged. Y may be modified by T.

STB — Store Both

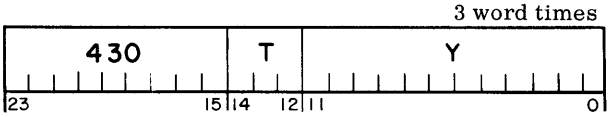
4 word times



The contents of R replace the contents of Y; the contents of L replace the contents of Y + 1. The contents of R and L are unchanged. Y may be modified by T.

STA — Store Address

3 word times

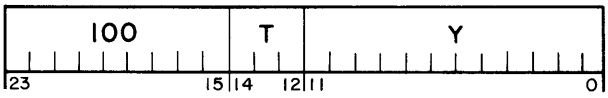


The contents of L, bits 0-11, replace the contents of Y, bits 0-11. The contents of Y, bits 12-23, and the contents of L are unchanged. Y may be modified by T.

2.2 ARITHMETIC INSTRUCTIONS

ADD — Add

2 word times

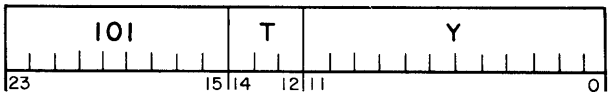


The contents of Y are algebraically added to the contents of L, and the resultant sum replaces the contents of L. If the sum is negative, the contents of L are in two's complement. The contents of Y are unchanged. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

ADR — Add and Replace

3 word times

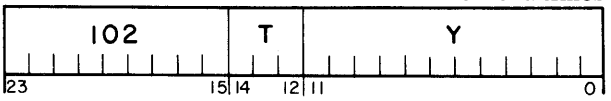


The contents of Y are algebraically added to the contents of L, and the resultant sum replaces the contents of both L and Y. If the sum is negative, the contents of both L and Y are in two's complement. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

ADT — Add and Trap on Overflow

3 word times

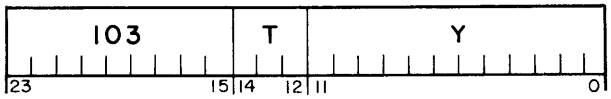


The contents of Y are algebraically added to the contents of L, and the resultant sum replaces the contents of L. If the sum is negative, the contents of L are in two's complement. The contents of Y are unchanged. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator comes ON, and a trapping operation is initiated. The HSM location of the ADT instruction replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

ART — Add, Replace, and Trap on Overflow

3 word times

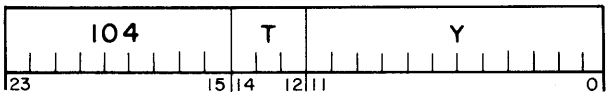


The contents of Y are algebraically added to the contents of L, and the resultant sum replaces the contents of both L and Y. If the sum is negative, the contents of both L and Y are in two's complement. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator comes ON, and a trapping operation is initiated. The HSM location of the ART instruction replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

ADL — Add Long

3 word times

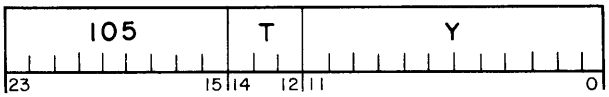


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The contents of Y are algebraically added to the contents of R, and the contents of Y + 1 are algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The contents of Y and Y + 1 are unchanged. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the ADL instruction, the sign position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

ALR — Add Long and Replace

5 word times

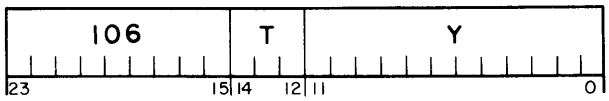


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The contents of Y are algebraically added to the contents of R, and the contents of Y + 1 are algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The sum in R replaces the contents of both R and Y; the sum in L replaces the contents of both L and Y + 1. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the ALR instruction, the sign position of both R and Y, bit 23, is made to agree with the sign position of both L and Y + 1. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

ALT — Add Long and Trap on Overflow

4 word times



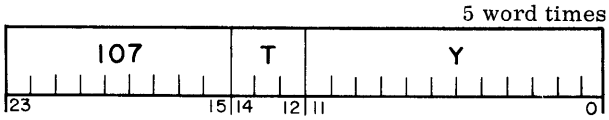
The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The contents of Y are algebraically added to the contents of R, and the contents of Y + 1 are algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The contents of Y and Y + 1 are unchanged. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the ALT instruction, the sign position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is

turned ON, and a trapping operation is initiated. The location of the ALT instruction replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

ALW — Add Long, Replace, and Trap on Overflow

5 word times

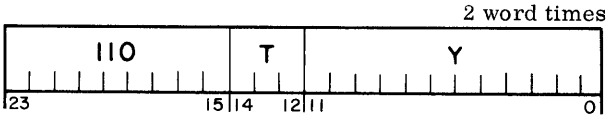


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The contents of Y are algebraically added to the contents of R, and the contents of Y + 1 are algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The resultant sum in R replaces the contents of both R and Y; the sum in L replaces the contents of both L and Y + 1. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the ALW instruction, the sign position of both R and Y, bit 23, is made to agree with the sign position of both L and Y + 1. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is turned ON, and a trapping operation is initiated. The location of the ALW instruction replaces the contents of standard location 0023_8, positions 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

SUB — Subtract

2 word times

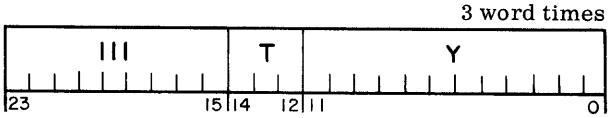


The two's complement of the contents of Y is algebraically added to the contents of L, and the sum replaces the contents of L. If the sum is negative, the contents of L are in two's complement. The contents of Y are unchanged. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

SUR — Subtract and Replace

3 word times

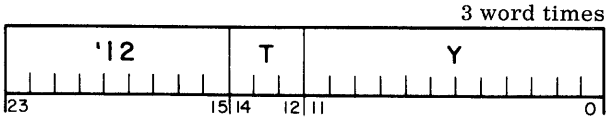


The two's complement of the contents of Y is algebraically added to the contents of L, and the sum replaces the contents of both L and Y. If the sum is negative, the contents of both L and Y are in two's complement. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

SUT — Subtract and Trap on Overflow

3 word times

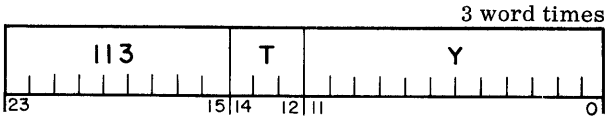


The two's complement of the contents of Y is algebraically added to the contents of L, and the sum replaces the contents of L. If the sum is negative, the contents of L are in two's complement. The contents of Y are unchanged. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is turned ON, and a trapping operation is initiated. The HSM location of the SUT instruction replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

SRT — Subtract, Replace, and Trap on Overflow

3 word times



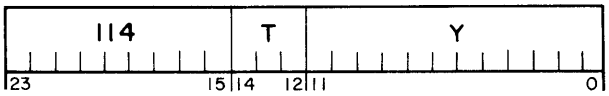
The two's complement of the contents of Y is algebraically added to the contents of L, and the sum replaces the contents of both L and Y. If the sum is negative, the contents of both L and Y are in two's complement. Y may be modified by T.

Note: If an overflow occurs, the BINARY OVERFLOW indicator is turned ON, and a trapping operation is initiated. The HSM location of the SRT instruction

replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

SUL — Subtract Long

3 word times

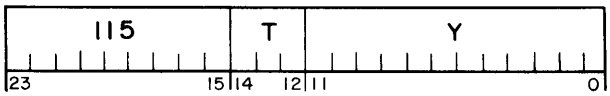


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The two's complement of the contents of Y is algebraically added to the contents of R, and the two's complement of the contents of Y + 1 is algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The contents of Y and Y + 1 are unchanged. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the SUL instruction, the sign position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

SLR — Subtract Long and Replace

5 word times

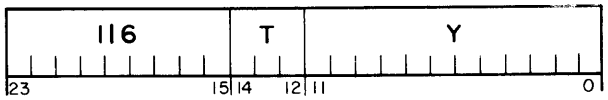


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The two's complement of the contents of Y is algebraically added to the contents of R, and the two's complement of the contents of Y + 1 is algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The sum in R replaces the contents of both R and Y; the sum in L replaces the contents of both L and Y + 1. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the SLR instruction, the sign position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is not turned ON, and a trapping operation is not initiated.

SLT — Subtract Long and Trap on Overflow

4 word times

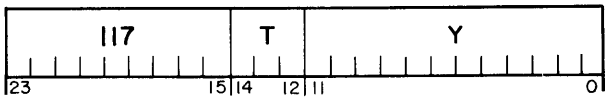


The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The two's complement of the contents of Y is algebraically added to the contents of R, and the two's complement of Y + 1 is algebraically added to the contents of L. Any carry generated by the addition to R will be added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The contents of Y and Y + 1 are unchanged. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the SLT instruction, the sign position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is turned ON, and a trapping operation is initiated. The location of the SLT instruction replaces the contents of standard location 0023_8, positions 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

SLW — Subtract Long, Replace, and Trap on Overflow

5 word times



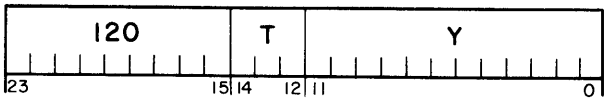
The contents of Y and Y + 1 are treated as a single double-precision word; the contents of Y are the least significant portion of the word. The two's complement of the contents of Y is algebraically added to the contents of R, and the two's complement of Y + 1 is algebraically added to the contents of L. Any carry generated by the addition to R is added into bit 0 of L. If the sum is negative, the contents of L and R (as a single word) are in two's complement. The sum in R replaces the contents of both R and Y; the sum in L replaces the contents of both L and Y + 1. The initial sign of Y must agree with the sign of Y + 1. If the signs do not agree, the sum is erroneous. At the conclusion of the SLW instruction, the sign

position of R, bit 23, is made to agree with the sign position of L. Y may be modified by T.

Note: If an overflow occurs (generated by a carry-out of bit 22 in L), the BINARY OVERFLOW indicator is turned ON, and a trapping operation is initiated. The location of the SLW instruction replaces the contents of standard location 0023_8, positions 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

MPY — Multiply

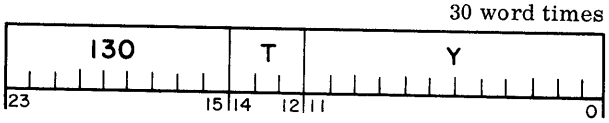
27 word times



The contents of L are multiplied by the contents of Y. The 23 most significant bits of the 46-bit product replace the contents of L, bits 0-22; the least significant bits replace the contents of R, bits 0-22. The sign positions of L and R, bits 23, are set to the algebraic sign of the product. If the product is negative, the contents of both L and R (as a single word) are in two's complement. The contents of Y are unchanged. Y may be modified by T.

DVD — Divide

30 word times

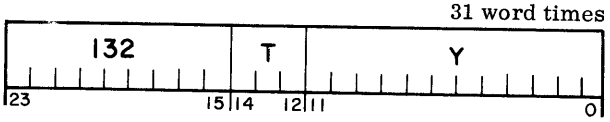


The contents of Y (the divisor) are divided into the contents of both L and R (the double-length dividend). The 23-bit quotient replaces the contents of L, bits 0-22; the 23-bit remainder (absolute value) replaces the contents of R, bits 0-22. The sign positions of L and R, bits 23, are set to the algebraic sign of the quotient. If the quotient is negative, the remainder is not affected, but the contents of L are in two's complement. The contents of Y are unchanged.

Note: If the initial absolute value in L is equal to, or greater than, the absolute value in Y ($|L| \geq |Y|$), the quotient is erroneous. The DIVIDE OVERFLOW indicator is not turned ON, and a trapping operation is not initiated. Y may be modified by T.

DVT — Divide and Trap on Overflow

31 word times



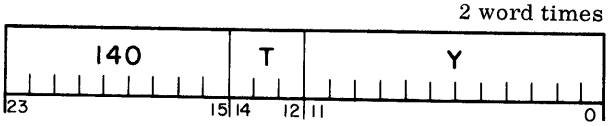
The contents of Y (the divisor) are divided into the contents of both L and R (the double-length dividend). The 23-bit quotient replaces the contents of L, bits 0-22; the 23-bit remainder (absolute value) replaces the contents of R, bits 0-22. The sign positions of L and R, bits 23, are set to the algebraic sign of the quotient. If the quotient is negative, the remainder is not affected, but the contents of L are in two's complement. The contents of Y are unchanged. Y may be modified by T.

Note: If the initial absolute value in L is equal to, or greater than, the absolute value in Y ($|L| \geq |Y|$), the quotient is erroneous. The DIVIDE OVERFLOW indicator is turned ON, and a trapping operation is initiated. The HSM location of the DVT instruction replaces the contents of standard location 0023_8, bits 0-11. The computer then receives its next instruction from standard location 0013_8 and continues from that location.

2.3 LOGICAL INSTRUCTIONS

LAN — Logical AND

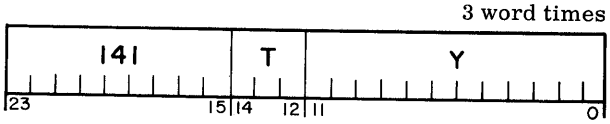
2 word times



Each position in the contents of Y, bits 0-23, is matched with the corresponding position in L, bits 0-23. If the contents of corresponding positions of both Y and L are one's, a one replaces the contents of that position in L. If the corresponding positions of either Y or L, or both, are zero's, a zero replaces the contents of the position in L. The contents of Y are unchanged. Y may be modified by T.

LAR — Logical AND and Replace

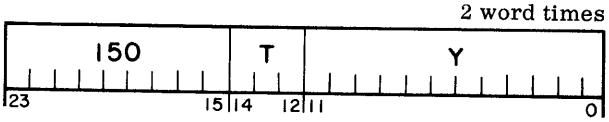
3 word times



The LAR instruction is the same as the LAN instruction, except that the result replaces the contents of both L and Y. Y may be modified by T.

LIO — Logical Inclusive OR

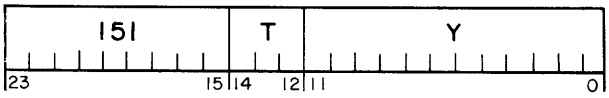
2 word times



Each position in the contents of Y, bits 0-23, is matched with the corresponding position in L, bits 0-23. If the corresponding positions of both Y and L are zero's, a zero replaces the contents of that position in L. If the corresponding position of either Y or L, or both, is a one, a one replaces the contents of that position in L. The contents of Y are unchanged. Y may be modified by T.

LIR — Logical Inclusive OR and Replace

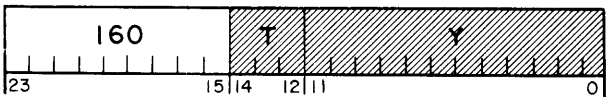
3 word times



The LIR instruction is the same as the LIO instruction, except that the result replaces the contents of both L and Y. Y may be modified by T.

CML — Complement L

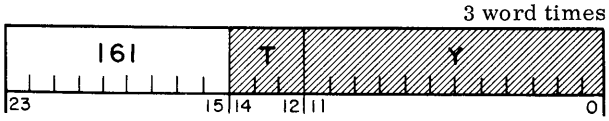
2 word times



The two's complement of the contents of L is formed, and the result replaces the contents of L. Bits 0-14 of the CML instruction are not interpreted.

CMB — Complement Both

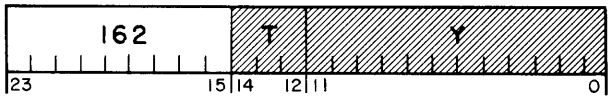
3 word times



Bits 0-22 of both L and R are treated as a single 46-bit number. The two's complement of this number is formed, and the result replaces the contents of L and R, bits 0-22. The initial sign of R, bit 23, is ignored, and the sign of R is made to agree with the sign of L. Bits 0-14 of the CMB instruction are not interpreted.

CLM — Complement L on Minus

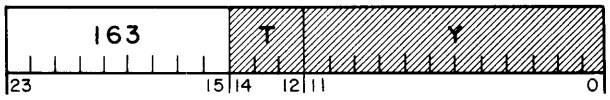
2 word times



If the sign position of L, bit 23, is a one (minus), the two's complement of the contents of L, bits 0-22, is formed, and the result replaces the contents of L. If the sign position of L is a zero (plus), no operation is performed, and the computer takes the next sequential instruction. In either case, the sign position of L is unchanged. Bits 0-14 of the CLM instruction are not interpreted.

CBM — Complement Both on Minus

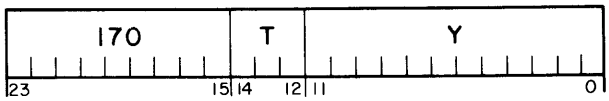
3 word times



Bits 0-22 of both L and R are treated as a single 46-bit number. If the sign position of L, bit 23, is a one (minus), the two's complement of the double-length number is formed, and the result replaces the contents of L and R, bits 0-22. If the sign position of L is a zero (plus), no operation is performed, and the computer takes the next sequential instruction. In either case, the sign position of L is unchanged, and the sign of R is made to agree with the sign of L. Bits 0-14 of the CBM instruction are not interpreted.

LEO — Logical Exclusive OR

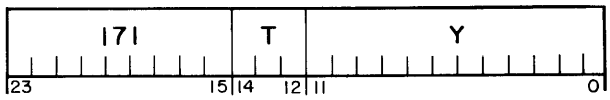
2 word times



Each position in the contents of Y, bits 0-23, is matched with the corresponding position in L, bits 0-23. If the corresponding positions of Y and L are the same, a zero replaces the contents of that position in L. If the corresponding positions of Y and L are not the same, a one replaces the contents of that position in L. The contents of Y are unchanged. Y may be modified by T.

LER — Logical Exclusive OR and Replace

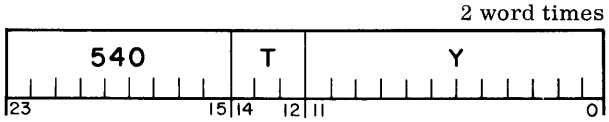
3 word times



The LER instruction is the same as the LEO instruction, except that the result replaces the contents of both L and Y. Y may be modified by T.

GTB — Gray to Binary

2 word times



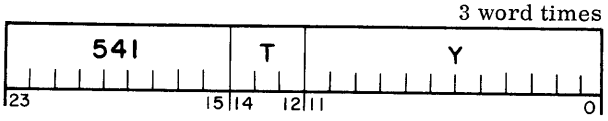
The contents of Y are treated as a single Gray-coded number and converted to the equivalent binary number, replacing the contents of L with the result. If the contents of Y are negative, the resulting binary number is in one's complement. Y may be modified by T.

Example:

$$0101110_{Gray} = 0110100_{Binary}$$

GBR — Gray to Binary and Replace

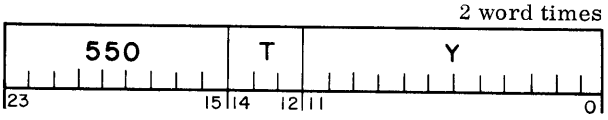
3 word times



The GBR instruction is the same as the GTB instruction, except that the result replaces the contents of both L and Y. Y may be modified by T.

BTG — Binary to Gray

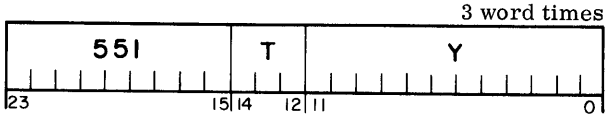
2 word times



The contents of Y are treated as a single binary number and are converted to the equivalent Gray-coded number, replacing the contents of L with the result. If the contents of Y are negative, the resulting Gray-coded number will be one greater than the two's complement. Y may be modified by T.

BGR — Binary to Gray and Replace

3 word times

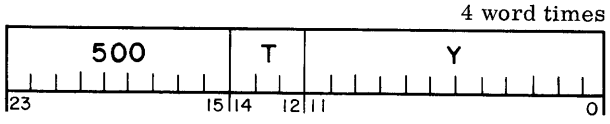


The BGR instruction is the same as the BTG instruction, except that the result replaces the contents of both L and Y. Y may be modified by T.

2.4 SHIFT INSTRUCTIONS

RBA — Rotate Both Algebraic

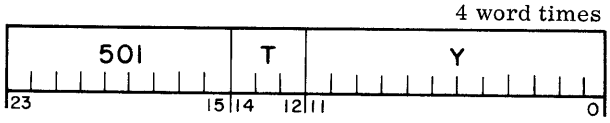
4 word times



Bits 0-22 of both L and R are treated as a single 46-bit register. The sign position, bit 23, in both L and R is ignored and retained. The contents of L and R are shifted right the number of positions specified by Y (after address modification). Bits shifted out of position 0 in L are shifted into position 22 in R; bits shifted out of position 0 in R are shifted into position 22 in L, giving a circular effect. The length of the shift (after address modification) is interpreted modulo 46_{10}.

SBA — Shift Both Algebraic

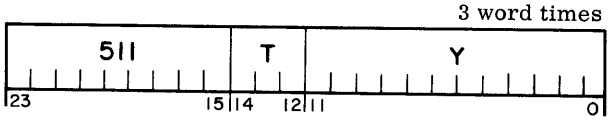
4 word times



Bits 0-22 of both L and R are treated as a single 46-bit register. The sign position, bit 23, in both L and R is ignored and retained. The contents of L and R are shifted right the number of positions specified by Y (after address modification). Bits shifted out of position 0 in L are shifted into position 22 in R; bits shifted out of position 0 in R are lost. On positive numbers, vacated positions in L, starting with bit 22, are filled with zero's. On negative numbers, vacated positions in L, starting with bit 22, are filled with one's. The length of the shift (after address modification) is interpreted modulo 46_{10}.

SLL — Shift L Logical

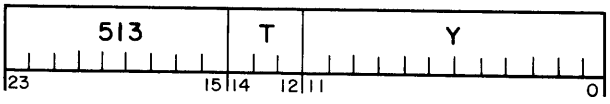
3 word times



The contents of L, bits 0-23, are shifted right the number of positions specified by Y (after address modification). Bits shifted out of bit position 0 in L are lost, vacated positions in L, starting with bit 23, are filled with zero's. The length of the shift (after address modification) is interpreted modulo 24_{10}.

SBL — Shift Both Logical

4 word times

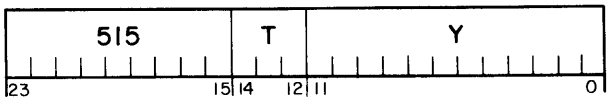


The contents of L and R are treated as a single 48-bit register. The contents of L and R are shifted right the number of positions specified by Y (after address modification). Bits shifted out of position 0 in L are

shifted into position 23 in R; bits shifted out of position 0 in R are lost. Vacated positions in L, starting with bit 23, are filled with zero's. The length of the shift (after address modification) is interpreted modulo 48_{10}.

RLL — Rotate L Logical

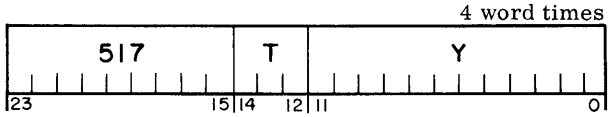
3 word times



The contents of L, bits 0-23, are shifted right the number of positions specified by Y (after address modification). Bits shifted out of position 0 in L are moved into position 23 in L, giving a circular effect. The length of the shift (after address modification) is interpreted modulo 24_{10}.

RBL — Rotate Both Logical

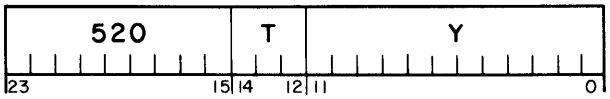
4 word times



The contents of L and R are treated as a single 48-bit register. The contents of L and R are shifted right the number of positions specified by Y (after address modification). Bits shifted out of position 0 in L are shifted into position 23 in R; bits shifted out of position 0 in R are shifted into position 23 in L. The length of the shift (after address modification) is interpreted modulo 48_{10}.

NRM — Normalize

4 word times

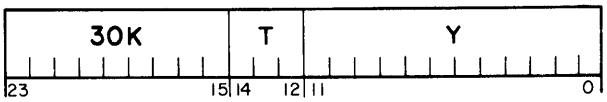


Bits 0-22 in both L and R are treated as a single 46-bit register. The contents of L and R are shifted left until the content of position 22 in L is the complement of position 23 in L, or until the contents of R, bits 0-22, replace the contents of L, bits 0-22. Bits shifted out of position 22 in R are shifted into position 0 in L. Vacated positions in R, starting with bit 0, are filled with zero's. The number of shifts required for normalization replaces the contents of Y, bits 0-11. The sign bit of both L and R is retained. Y may be modified by T.

2.5 TRANSFER INSTRUCTIONS

TXI — Transfer on XR NOT ZERO and Increment

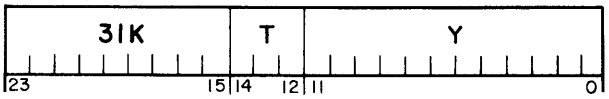
3 word times



If the initial contents of the index register specified by T are not zero nor greater than 7777_8 (the overflow occurred on the last TXI instruction), the contents of the specified index register are incremented by K ($0 \leq K \leq 7$). Then the computer receives its next instruction from Y and proceeds from that location. If the initial contents of the specified index register are zero or greater than 7777_8, the index register is not incremented, and the computer then processes the next sequential instruction. Y may not be modified by T.

TXD — Transfer on XR NOT ZERO and Decrement

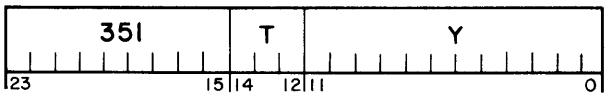
3 word times



If the initial contents of the index register specified by T are greater than zero, the contents of the specified index register are decremented by K, ($0 \leq K \leq 7$). The computer then receives its next instruction from Y and proceeds from that location. If the initial contents of the specified index register are zero or less, the index register is not decremented, and the computer processes the next sequential instruction. Y may not be modified by T.

TRN — Transfer on Negative

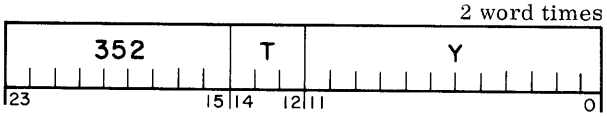
2 word times



If the sign position in L, bit 23, contains a one, the computer receives its next instruction from Y and proceeds from that location. If the sign position in L contains a zero, the computer processes the next sequential instruction. The contents of Y and L are unchanged. Y may be modified by T.

TRZ — Transfer on Zero

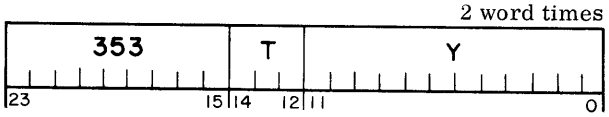
2 word times



If all positions in L, bits 0-23, contain zero's, the computer receives its next instruction from Y and proceeds from that location. If any position in L (bits 0-23) contains a one, the computer processes the next sequential instruction. The contents of Y and L are unchanged. Y may be modified by T.

TNZ — Transfer on Negative or Zero

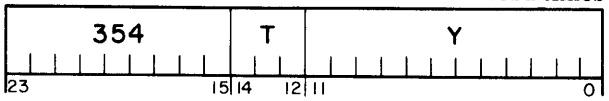
2 word times



If all bits in L equal 0, or if the sign bit of L is a one (bit 23), the computer takes its next instruction from Y. If the sign position is 0 and any of the bits 0-22 contains a zero, the computer processes the next instruction. The contents of Y and L are unchanged.

TRP — Transfer on Positive

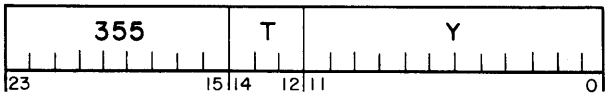
2 word times



If the sign position in L, bit 23, contains a zero and any position in L (bits 0-22) contains a one, the computer receives its next instruction from Y and proceeds from that location. If the sign position in L contains a one, or if every position in L (bits 0-23) contains a zero, the computer processes the next sequential instruction. The contents of Y and L are unchanged. Y may be modified by T.

TPN — Transfer on Positive or Negative

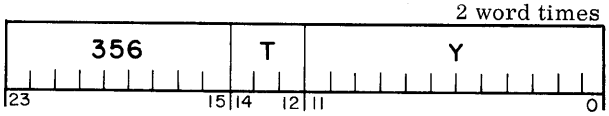
2 word times



If any position in L (bits 0-23) contains a one, the computer receives its next instruction from Y and proceeds from that location. If every position in L, bits 0-23, contains a zero, the computer processes the next sequential instruction. The contents of Y and L are unchanged. Y may be modified by T.

TPZ — Transfer on Positive or Zero

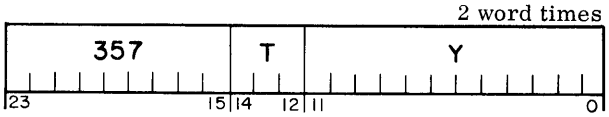
2 word times



If the sign position in L, bit 23, contains a zero, the computer receives its next instruction from Y and proceeds from that location. If the sign position in L contains a one, the computer processes the next sequential instruction. The contents of Y and L are unchanged. Y may be modified by T.

TRA — Transfer Unconditionally

2 word times

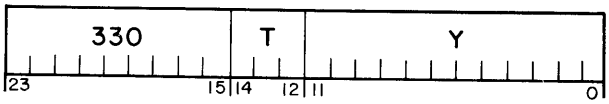


The computer receives its next instruction from Y and proceeds from that location. The contents of Y are unchanged. Y may be modified by T.

2.6 CONTROL INSTRUCTIONS

RAI — Return after Interrupt

4 word times



This instruction is intended to permit automatic restoration of an interrupted program. It must be the last instruction executed by the interrupting program. The contents of L, R, PC, and JS are restored to the status held at the time of the interrupt. Specified by Y is the HSM location of the storage of the set of standard locations used for saving the status of the interrupted program. Y is a function of the priority of the program executing the RAI instruction and may have the following values:

0010_8 - Priority 1
0014_8 - Priority 2
0020_8 - Priority 3
0024_8 - Priority 4
0040_8 - Power failure

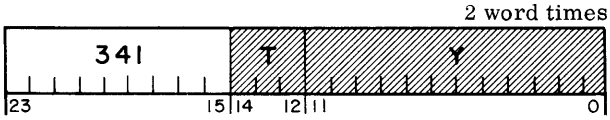
Bits 12-14 of the RAI instruction are interpreted.

Note: An RAI instruction is used following a power failure during computer operation. It is addressed

to standard location 0040_8, and restores those conditions which were present when the power-failure interrupt occurred. Y may be modified by T; therefore, T should be zero for normal applications of the RAI instruction.

INI — Inhibit Interrupt

2 word times

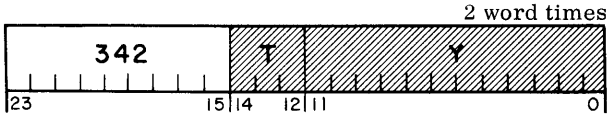


The INI instruction places the computer in the INHIBIT mode where priority interrupt requests are not serviced by the computer. The computer will remain in the INHIBIT mode until an ACI instruction is executed. Any interrupt requests that occur while the computer is in the INHIBIT mode are automatically saved in the JR register, and are executed in order of priority when the computer returns to the NORMAL mode. Bits 0-14 of the INI instruction are not interpreted.

Note: The INI instruction is treated as an NOP instruction unless the INTERRUPT CONTROL switch on the Maintenance and Control panel is in the NORMAL position. (See Section 4.1.6.)

ACI — Activate Interrupt

2 word times

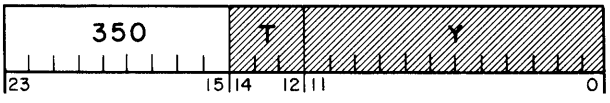


The ACI instruction places the computer in the ENABLE mode where priority interrupts can be serviced. The computer will remain in the ENABLE mode until an INI instruction is executed. Bits 0-14 of the ACI instruction are not interpreted.

Note: The ACI instruction is treated as an NOP instruction unless the INTERRUPT CONTROL switch on the Maintenance and Control panel is in the NORMAL position. (See Section 4.1.6.)

NOP — No Operation

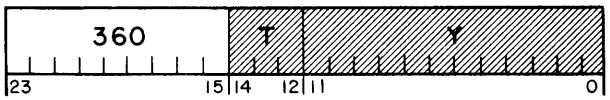
2 word times



No operation is performed, and the computer receives the next sequential instruction. Bits 0-14 of the NOP instruction are not interpreted.

HLT — Halt

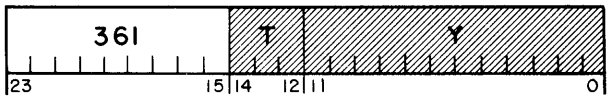
1 word time



The computer unconditionally halts. The PC contains the HSM location of the HLT instruction. Restart of the computer is initiated by depressing the START button on the Maintenance and Control panel. Bits 0-14 of the HLT instruction are not interpreted.

HTI — Halt Interruptable

1 word time

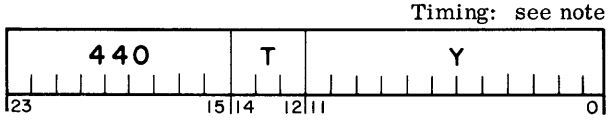


The computer halts and can be restarted either by depressing the START button on the Maintenance and Control panel or by an interrupt request. If the interrupt request is used, the computer returns to the HTI instruction after servicing the interrupt. If the computer is restarted by means of the START button, operation is returned to the NORMAL mode. Bits 0-14 of the HTI instruction are not interpreted.

2.7 INPUT/OUTPUT INSTRUCTIONS

WDR — Write Drum

Timing: see note



The WDR instruction requires two additional words of information in order to transfer information from HSM to the drum. The memory location of the first of these two words is specified by Y, and the computer automatically assumes the second word to be at Y + 1.

Location Y contains the following information:

- Bits 0-11, the memory location of the first word to be written on the drum.
- Bits 12-23, the number of consecutive words to be taken from memory and written onto the drum (including the first word).

Location Y + 1 contains the following information:

- Bits 0-6, the starting sector address on the drum where the information is to be written.
- Bits 7-13, the starting track address.
- Bit 14, specifying whether track is a buffer track (indicated by a one) or a main track (indicated by a zero).
- Bit 16, specifying the upper half of drum (indicated by a one) or the lower half of drum (indicated by a zero).

Note: The half of drum specified will be employed until otherwise programmed.

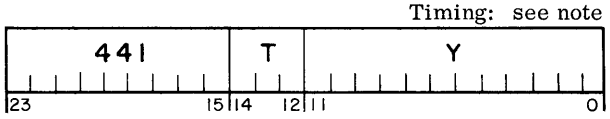
CAUTION

If bit 15 is a one, the computer will start writing on the drum 128 microseconds after the WDR instruction is given. The information will be written on the specified track, but the sector address will be random.

Timing: Transfer time (average) = 8.32 + 0.128n in milliseconds, where n is the number of words to be written. Y may be modified by T.

RDR — Read Drum

Timing: see note



The RDR instruction requires two additional words of information in order to transfer information from the drum to HSM. The memory location of the first of these two words is specified by Y, and the computer automatically assumes the second word to be at Y + 1.

Contained in Y is the following information:

- Bits 0-11, the memory location for storing the first word that is read from the drum.
- Bits 12-23, the number of consecutive words to be read from the drum and stored into HSM (inclusing the first word).

Contained in Y + 1 is the following information:

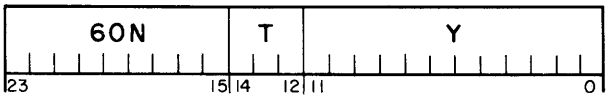
- Bits 0-6, the starting sector address on the drum where the information is to be read.
- Bits 7-13, the track address.
- Bit 14, specifying whether track is a buffer track (indicated by a one), or a main track (indicated by a zero).
- Bit 16, specifying the upper half of drum (indicated by a one) or the lower half of drum (indicated by a zero).

Note: The half of drum specified will be employed until otherwise programmed.

Timing: Transfer time (average) = 8.32 + 0.128n in milliseconds, where n is the number of words to be read. Y may be modified by T.

SUN — Set Up I/O Device

2 word times

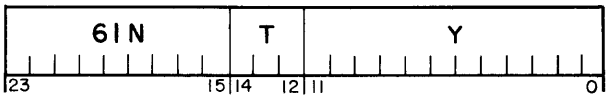


The SUN instruction is used to set up an I/O device in a particular operating mode through the eight groups of 24 IOA lines. The N portion of the OP code, where $0 \leq N \leq 7$, specifies which of the eight groups is to be used. The SUN instruction requires one additional word of information in order to set up the I/O device for an operation. The memory location of this word is specified by Y. The contents of Y, bits 0-23, are used to specify which IOA lines of a particular group are to be activated. The IOA assignments are listed in Appendix H. See Section 3 for a complete discussion of Input/Output. Y may be modified by T.

Note: PRIORITY INTERRUPT is automatically inhibited when Read Tape or Write Tape instruction is specified.

STN — Store I/O Register N

3 word times



The contents of the I/O register specified by N, where $0 \leq N \leq 7$, replace the contents of Y. If the specified I/O register has not been filled by an input device at the time the STN instruction is encountered, the computer delays until the input device has completed its transmission of the word.

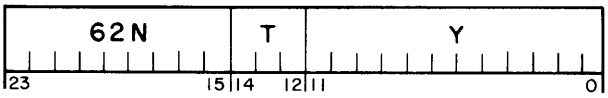
Note: If N = 0, the STN instruction must be preceded by a SUN instruction, or the computer delays

indefinitely. See Section 3 for a complete discussion of Input/Output. I/O Register assignments are listed in Appendix I.

Y may be modified by T unless the SUN instruction which preceded the STN instruction specified the Read Tape or the Write Tape instruction.

LDN — Load I/O Register N

2 word times



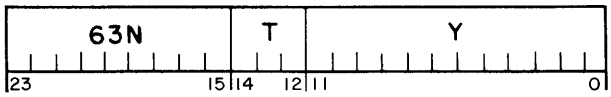
The contents of Y replace the contents of the I/O register specified by N, where $0 \leq N \leq 7$. If the specified I/O register has not been emptied by an output device at the time the LDN instruction is encountered, the computer delays until the output device has taken the word before loading the new word.

Note: If N = 0, the LDN instruction must be followed by an SUN instruction so that the I/O register will be unloaded. In the case of a magnetic-tape instruction, an SUN instruction must precede an LDN. See Section 3 for a complete discussion of Input/Output. I/O Register assignments are listed in Appendix I.

Y may be modified by T unless the SUN instruction which preceded the LDN instruction specified the Read Tape or the Write Tape instruction.

SDN — Sense Device N

2 word times



The SDN instruction provides for the sensing the I/O Status (IOS) lines. The bit pattern is generated by the set of IOS lines specified by N, where $0 \leq N \leq 7$; the bits are AND'd with the contents of Y, and the result replaces the contents of L.

The status of an input/output device will be indicated by a signal on the corresponding IOS line. A zero indicates that the input/output device is not active; a one indicates an active condition.

2.8 ADDRESS MODIFICATION

Address modification can be performed automatically on the RCA 110 Computer by specifying or "tagging" one of the seven index registers (XR) in an instruction word. All except 10 of the 72 RCA 110 Computer instructions can have address modifications.

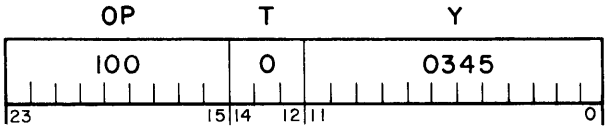
Address modification is performed during the execution of a program by computer operation on an instruction-operand address (Y portion). The computer is instructed to perform this modification by analyzing the T portion of each instruction (bits 14-12). If the T portion contains a number from one (1) to seven (7), address modification is performed.

The number contained in the T portion of the instruction refers to a particular XR (locations 0001 through 0007). If an XR is specified, the contents of the register are then subtracted from the Y portion of the instruction word (bits 11-0).

2.8.1 Reducing the Operand Address Y

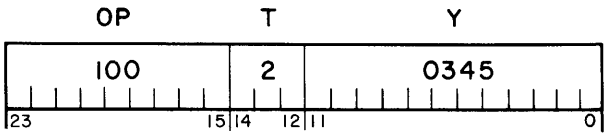
For example, consider the following ADD instructions:

In the first example, the instruction would add the contents of location 0345 to the contents of the Left accumulator (L).



Since the T portion of the instruction = 0, no index register is specified, and the instruction is performed using the address specified in Y.

For the second example, assume it is necessary to perform the same instruction using the operand stored in location 0340, or 5 locations below 0345. Also, suppose that index register 2 (XR 2) contains the value 0005_8. The instruction is performed as before, except for specifying XR 2 by placing a two (2) in the T portion of the instruction.



The computer would then modify the base address in Y (0345) by subtracting the contents of XR 2 = (0005_8), thus resulting in an effective address of 0340.

Example:

```
 0345  Y (base address)
-0005  contents of index register 2
 0340  effective address
```

This instruction now adds the contents of 0340 to the contents of L.

Note: The effective address does not replace the Y portion of the instruction. Y still remains as the base address. Thus, the actual memory location used to perform an instruction is different from that given in the instruction.

2.8.2 Increasing the Operand Address Y

Since the contents of the XR is subtracted from the base address specified in Y, the base address can be increased by placing a minus number (two's complement) in the XR. The programmer should consider bit 11 (of the number being placed in XR) as a sign bit.

If bit 11 is a one, then bits 0-11 form the two's complement of the number by which the operand is to be increased. Thus, any octal number of 3777 or less decreases the base address, and 4000 or more will increase the base address.

For example, suppose the base address is to be increased by one. Since address modification is performed by subtracting the contents of an XR, it is necessary to place a negative one (-1 = 7777 in two's complement) in the specified register. Thus when the negative one is subtracted from the base address, the net effect is to increase the base address by one.

Example:

Computer Operation	Equivalent Operation	
0345	0345	Y (base address)
-7777	+0001	XR (two's complement of -1) to be subtracted from Y
0346	0346	Effective address

SECTION 3

INPUT/OUTPUT

3.1 GENERAL

The RCA 110 Computer has flexible and powerful input/output capabilities. It can be used to monitor and control continuous processes, operate displays, and communicate with a wide variety of peripheral devices.

In order to effectively utilize the full input/output capabilities of the RCA 110 Computer, the programmer should become extremely familiar with the input/output programming methods and operations.

RCA 110 Computer input/output operations are explained in the following sections:

- Communication Channels - Section 3.2
- Input/Output Instructions - Section 3.3
- RCA-110 Input/Output Equipment Programming - Section 3.4 and 3.5

3.2 COMMUNICATION CHANNELS

The RCA 110 Computer communicates with the outside world through five different channels:

- Input/Output Sense Lines (IOS Lines)
- Input/Output Address Lines (IOA Lines)
- Input/Output Buffer Registers (IOR's)
- Priority Request Lines
- Drum Buffer Tracks

The relationship of these channels to the RCA 110 Computer is illustrated in Figure 3-1.

3.2.1 Input/Output Sense Lines

The Input/Output Sense lines (IOS lines) are used by the computer to detect the status of peripheral devices, console sense switches, analog-to-digital converters, relays, and on-line input/output devices.

The RCA 110 Computer can have up to 192 IOS lines. These lines operate in from one to eight different levels (groups) of 24 lines per level.

The programmer can individually select each of the IOS lines by the use of the Sense Device N (SDN) instruction, which is explained in Section 2.7.

The IOS line assignments are shown in Appendix G.

3.2.2 Input/Output Address Lines

The Input/Output Address lines (IOA lines) are used by the computer to instruct a particular device to "get ready" for data transmission. An IOA signal will activate the device into a send or receive condition. The IOA lines are also used to operate displays, set sequence-control relays, drive digital-to-analog converters, initiate warning alarms, and select analog or digital signal lines.

The RCA 110 Computer can have up to 192 IOA lines. These lines operate in from one to eight different levels (groups) of 24 lines per level. The programmer can activate any of the IOA lines by the use of the Set Up I/O Device (SUN) instructions which is explained in Section 2.7.

The IOA line assignments are shown in Appendix H.

3.2.3 Input/Output Buffer Registers

The Input/Output Buffer registers (IOR's) are general purpose buffer registers used to receive and send information to the peripheral devices of the RCA 110 Computer. The RCA 110 Computer can have from one to eight I/O buffer registers. Each IOR is individually addressable and contains 24 bits.

IOR's are numbered from 0-7. The IOR zero is used to communicate with the basic RCA 110 Computer input/output devices, such as the magnetic-tape station,

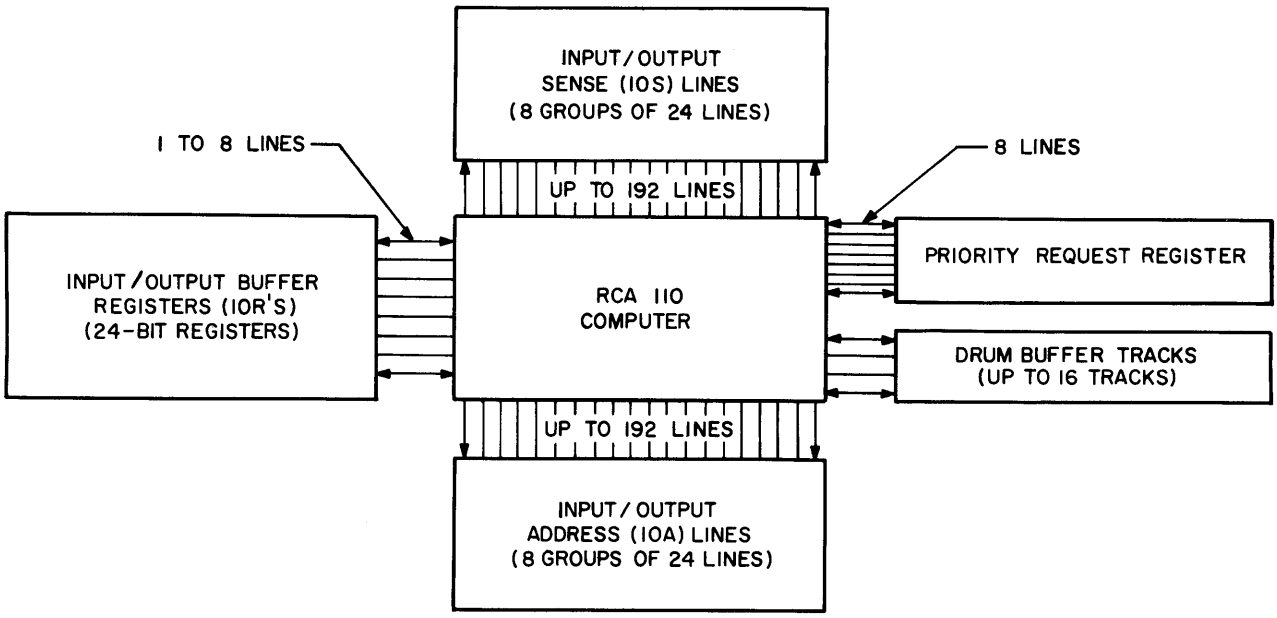


Figure 3-1. RCA 110 Computer Communication Channels

paper-tape reader, paper-tape punch, and monitor typewriter. The other IOR's are used to communicate with special peripheral devices, such as telemetry equipment, guidance computers, payloads, transmission lines, and other miscellaneous pulse sources.

The programmer can transmit data through the IOR's to and from input/output devices by the use of the Load I/O Register N (LDN) and Store I/O Register N (STN) instructions. These instructions are explained in Section 2.7.

The IOR assignments are shown in Appendix I.

3.2.4 Priority Request Lines

The eight Priority Request lines can be connected to outside devices to notify the RCA 110 Computer of an important event or alarm condition. Each of the priority request lines can be assigned a priority value from 1-8, depending on the relative importance of the signal carried by the line. Each line is connected to the Priority Request register (JR), which informs the computer of the relative importance of incoming signals.

3.2.5 Drum Buffer Tracks

The drum memory of the RCA 110 Computer can be implemented for "data buffering". This is the operation which allows the computer to read or write to an external device while performing other internal operations during the data transfer. The drum accomplishes this data buffering by providing intermediate storage for information between the RCA 110 Computer and the input/output devices. Intermediate or drum buffer storage is provided by equipping the drum with special read and write heads, logic, and storage tracks, in addition to the main storage capabilities.

There are two types of drum buffer tracks - READ tracks and WRITE tracks. Refer to Figure 3-2.

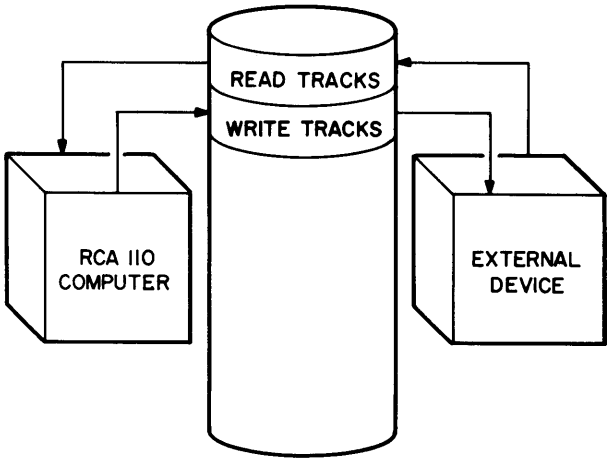


Figure 3-2. Drum Buffer Tracks

READ tracks are loaded from an external device and read by the RCA 110 Computer with a Read Drum instruction. WRITE tracks are loaded from the computer and then unloaded to an external device by a Write Drum instruction.

Each READ or WRITE track may be used independently of the other tracks. Thus, the RCA 110 Computer may write or read any buffer track while the other tracks are sending or receiving information from external devices. Likewise, the RCA 110 Computer may also perform internal operations while the buffer tracks are being used to communicate with external devices.

A complete explanation of drum buffer programming is given in Section 3.3.5.

3.3 INPUT/OUTPUT INSTRUCTIONS

The basic functions of these instructions are to:

- Determine if an input/output device is ready to send or receive information.
- Prepare the device for sending or receiving.
- Instruct the computer to send or receive information to and from the device.
- Transfer information to or from the drum.

The six input/output instructions of the RCA 110 Computer are:

- SDN - Sense Device N
- SUN - Set Up I/O Device
- STN - Store I/O Register N
- LDN - Load I/O Register N
- WDR - Write Drum
- RDR - Read Drum

These instructions use an "extended addressing" technique. By "extended addressing" is meant that the address portion (Y) of the input/output instruction refers to an address that contains the pertinent information required to begin the input/output operation.

Note: The information in A + 3 is used to test status of a particular I/O device.

Example:

HSM LOCATION	MNEMONIC OP CODE	OPERAND ADDRESS (Y)
A	SDN	A + 3
A + 1	Next instructions	
A + 2	Or data	
A + 3	Information (mask) required to test status of I/O device.	

3.3.1 SDN — Sense Device N – 63(N) T Y

The SDN instruction is used to sense or detect the operating status of an input/output device through the eight groups of IOS lines. SDN may also be used to detect the settings of sense switches on the Maintenance and Control panel.

The SDN instruction consists of three basic parts:

- Numeric OP Code - 63(N) - Where N may be any number 0-7 which refers to a particular group of IOS lines.

  Example: 630 refers to IOS group 0
  635 refers to IOS group 5

- T or Tag - 63(N) T - The T portion is used when modification of the address contained in Y is desired. (Refer to Address modification for further information.)

- Y or Operand Address - 63(N) T Y - The Y portion specifies an HSM address which contains a "mask" used to test the status of various I/O devices.

The status is indicated by the presence of a one on a a particular IOS line. When an SDN instruction is used, the contents of the location specified by Y are masked with the data received on the IOS lines. The result of the masking operation replaces the contents of L. Thus, after an SDN instruction, L may be tested to determine the result of the masking operation.

3.3.2 SUN — Set Up I/O Device – 60(N) T Y

The SUN instruction is used to select and signal an I/O device to "get ready" for data transmission. The

SUN instruction results in a signal being sent to an I/O device via the eight groups of IOA lines.

The SUN instruction consists of three basic parts:

- Numeric OP Code - 60(N) - Where N may be any number 0-8 which refers to a particular group of IOA lines.

  Example: 600 refers to IOA group 0
  606 refers to IOA group 6

- T or Tag - 60(N) T - The T portion is used when modification of the address contained in Y is desired.

- Y or Operand Address-60(N) T Y - The Y portion of the SUN instruction specifies an HSM address which contains the mask to select the operating mode of a particular I/O device.

Explanation of the SUN instruction is provided in Section 2.7.

3.3.3 LDN — Load I/O Register N – 62(N) T Y

The LDN instruction is used to transfer the contents of a selected memory location to a particular IOR. If the IOR addressed by the LDN is in the process of being unloaded, the computer will wait until the IOR is empty. When the IOR is empty, the data referenced by the LDN instruction is transferred from memory to the IOR. The LDN instruction must then be followed by an SUN instruction, except when using magnetic tape. (For magnetic tape see Section 3.5.) The SUN instruction instructs the computer to transfer the contents of IOR 0 to a peripheral device.

The LDN instruction consists of three basic parts:

- Numeric OP Code - 62(N) - Where N may be any number 0-7 which refers to a particular IOR.

  Example: 620 will load the contents of a memory location into IOR 0.
  623 will load the contents of a memory location into IOR 3.

- T or Tag-62(N) T - The T portion is used when modification of the address contained in Y is desired, except on Write Tape instruction.

- Y or Operand Address - 62(N) T Y - The Y portion of the LDN instruction specifies the HSM address which contains the information to be transferred to the IOR.

3.3.4 STN — Store I/O Register N – 61(N) T Y

The STN instruction is used to transfer the contents of an IOR to a selected memory location. If the specified IOR is in the process of being filled by an input device when an STN instruction is encountered, the computer will wait until the IOR is completely loaded before executing the STN instruction.

The STN instruction must be preceded by an SUN instruction or the computer will delay indefinitely.

The STN instruction consists of three basic parts:

- Numeric OP Code - 61(N) - Where N may be any number 0-7 which refers to a particular IOR.

  Example: 610 will store the contents of IOR 0 into a selected memory location.
  615 will store the contents of IOR 5 into a selected memory location.

- T or Tag - 61(N) T - The T portion is used when modification of the address contained in Y is desired, except on Read Tape instruction.

- Y or Operand Address - 61(N) T Y - The Y portion of the STN instruction contains the HSM location where the contents of the specified IOR are to be stored.

3.3.5 RDR – Read Drum; WDR – Write Drum

The drum in the RCA 110 Computer may be used as an input/output buffer as well as for bulk storage. The programmer can transmit data to the drum by the RDR or WDR instructions. With minor differences, both of these instructions can be used to transfer information to and from the drum buffer tracks as well as main storage. Refer to Figure 3-3.

The RCA 110 Drum Memory is composed of three basic functional sections:

- MAIN STORAGE TRACKS - Tracks which are used for the bulk storage of programs and data. Communicates with the High Speed Memory section of the computer and consists of up to 256 tracks with 128 words, or sectors, per track.

- TIMING TRACKS - Four prerecorded tracks which provide basic timing reference pulses for the transfer of information to and from the drum.

- BUFFER TRACKS - A portion of the drum which is used for the intermediate storage of information being transferred to or from an input/output device to the computer central processor. The drum can have up to sixteen tracks with 128 words, or sectors, per track.

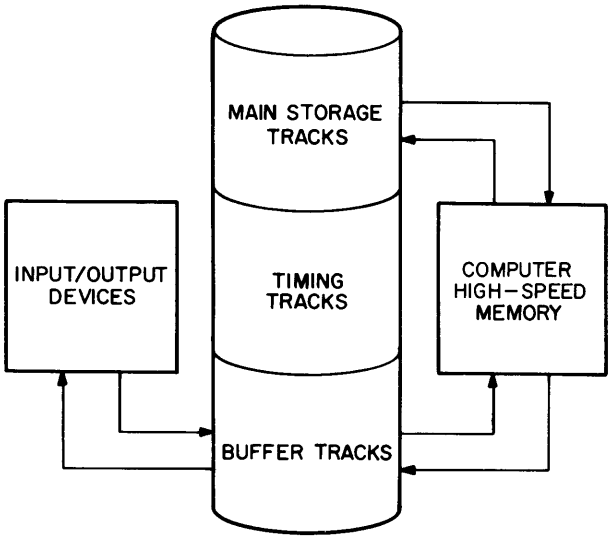


Figure 3-3. RCA 110 Computer Drum Memory

Each track and sector in either the buffer or main storage sections can be addressed by the programmer.

The instructions used to read and write to the drum are as follows:

Write Drum - 440 T Y
Read Drum - 441 T Y

Both instructions operate in exactly the same manner. The operation and programming of these instruction is as follows:

- Numeric OP Code 440 or 441 - The numeric OP code 440 results in information being transferred from the core to the drum, while OP code 441 transfers information from the drum to the core.

- T or Tag 440 T or 441 T - The T portion is used when address modification is desired.

- Y or Operand Address 440 T Y or 441 T Y - The Y portion of the instruction specifies the first of two consecutive HSM addresses which contain further information required by the computer in the transfer.

The first of the two consecutive addresses (address in Y) specifies the number of words to be transferred (bits 12-23) and the HSM address of the first word to be transferred or received (bits 0-11).

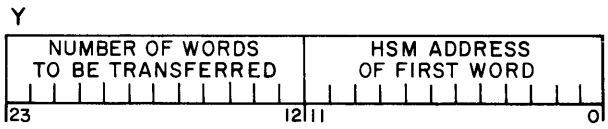
1) Main Storage Tracks - Bit Positions 7-13 and 16 are used to specify the track address. These bit positions are divided into three groups which are used to designate the track number.

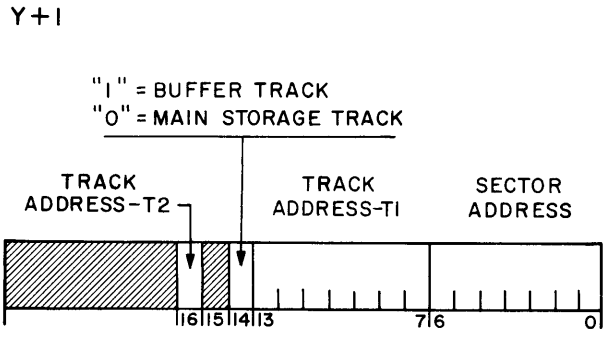
   Bits 13 and 16, 1st group, most significant track number
   Bits 10 to 12, 2nd group, next significant track number
   Bits 7 to 9, 3rd group, least significant track number

2) Buffer Tracks - A buffer track may be specified by a one in bit position 14. The buffer track number is then specified in the same manner as a main storage track.

3) Sector Address - Bit positions 0-6 are used to specify the sector address. These bit positions are divided into 3 groups which are used to designate the sector address.

   Bit 6, 1st group
   Bits 3-5, 2nd group
   Bits 0-2, 3rd group





CAUTION

If bit 15 is a one, the computer will start writing on the drum 128 microseconds after the WDR instruction is given. The information will be written on the specified track, but the sector address will be random.

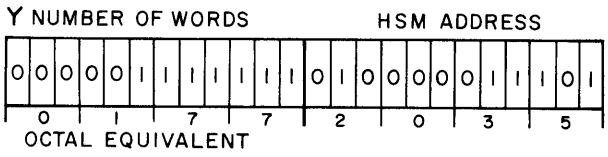
Note: Bits 7-13, and 16 are required to fully specify tracks numbered from 000 to 377.

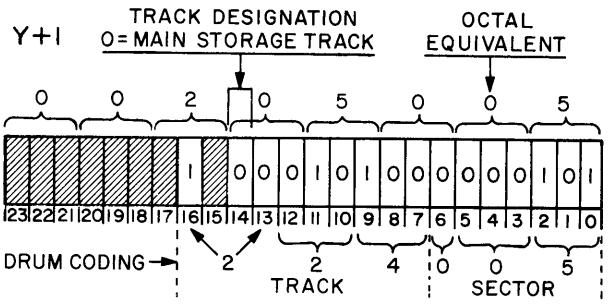
Bit 14 is used to specify a buffer track. Bit 15 is unused.

Example 1:

127_{10} words (177_8) are to be transferred from HSM location 3025 to the drum, starting with main storage track 224_8, sector 005_8.

Y and Y + 1 would appear as follows:





Therefore, the contents of Y and Y + 1 would be:

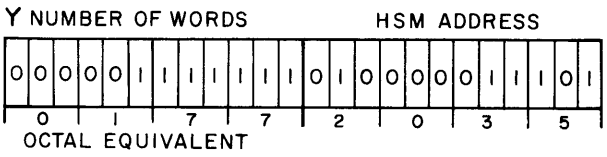
Y = 01772035 or 127_{10} words, HSM location 2035
Y + 1 = 00205005 or main storage track 224, Sector 005

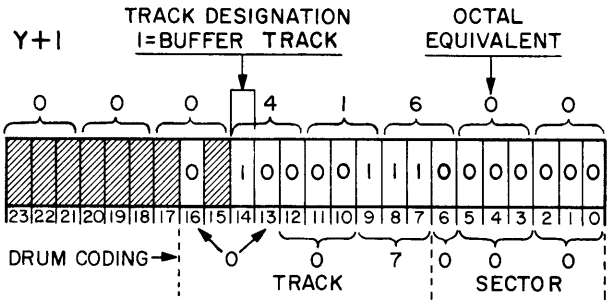
Note that the track address 224 is represented by ones in bit positions 16, 11, and 9, and sector address 005 is represented by one's in bit positions 2 and 0.

Example 2:

127_{10} words (177_8) are to be transferred from HSM location 2035 to buffer track 07_8, Sector 000.

Y and Y + 1 would appear as follows:





Therefore, the contents of Y and Y + 1 would be:

Y = 01772035 or 127_{10} words, HSM location 2035
Y + 1 = 0041600 or Buffer track 07, Sector 000

SUMMARY

- Y Bits - 0-11 contain HSM address
  12-23 number of words
- Y + 1 Bits - 0-6 sector address $(000\text{-}177)_8$
  7-13 and 16 track address $(000\text{-}377)_8$
  14 buffer track indicator

3.4 RCA 110 INPUT/OUTPUT EQUIPMENT PROGRAMMING

The RCA 110 basic computer system includes several standard input/output devices which all have common programming requirements.

The basic RCA 110 Computer input/output devices include:

- Standard Equipment

  Paper-Tape Reader
  Paper-Tape Punch
  Monitor Typewriter

- Optional Equipment

  Magnetic Tape Station

3.4.1 Paper-Tape Reader

The paper-tape reader reads an eight-column punched paper tape at a rate of up to 60 characters per second. The reader always stops on the next character to be read. The reader operates in four basic modes:

1) Mode I - Octal Mode - Eight octal characters are read into IOR 0 for each SUN instruction. The most significant three bits are filled first,

followed by the next significant three bits until IOR 0 is completely filled. An STN instruction then transfers the contents of IOR 0 to memory as one computer word.

Special Conditions - During this operating mode, the reader checks for the special conditions:

Condition	Computer Action
• Parity Error	Reader stops on character causing error. I/O Alarm Indicator is lighted on Maintenance and Control panel, and automatic trapping occurs.
• Non-Octal Character (Includes carriage return (CR) and delete characters.)	Passed over by reader and not read into IOR. Does not constitute part of a computer word.
• Forward STOP character $(13)_8$	Causes the computer to skip after the next STN instruction. Does not enter IOR 0.

2) Mode II - Alphanumeric - One 6-Bit Character - One 6-bit character is read into IOR 0 for each SUN instruction. The 6-bit character will occupy bit positions 0-5, with the remaining positions 6-23 filled with zeros.

Special Conditions - During this mode, the reader checks for the special conditions:

Condition	Computer Action
• Incorrect Parity	Reader stops on character causing error, and automatic trapping occurs.
• "Delete" or "Blank character	Passed over and not recognized by reader.
• Forward STOP character $(13)_8$	Causes the computer to skip after the next STN instruction. Does not enter IOR 0.
• Carriage Return character	Converted to octal 56.

3) Mode III - Alphanumeric - Four 6-Bit Characters - Four 6-bit characters are read into IOR 0 for each SUN instruction. The most significant six bits are filled first, followed by the next most significant six bits until IOR 0 is completely filled.

Special Conditions - During this mode, the reader checks for the special conditions:

Condition	Computer Action
• Incorrect Parity	Reader stops on character causing error.
• Carriage Return character	Converted to octal 56.

4) Rewind - During this mode, the paper tape reader is rewinding and will stop upon a REVERSE STOP character $(14)_8$. IOR 0 is free during the rewind operation.

3.4.2 Paper-Tape Punch

The paper-tape punch punches an eight column paper tape at rates up to 60 characters per second. The punch operates in three different modes.

1) Mode I - Alphanumeric - One 6-Bit Character - One 6-bit character is punched for each SUN instruction. This character is transferred to IOR 0, bits 0-5, by an LDN instruction. The remainder of IOR 0, bits 6-23, contains zeros and is not transferred to the punch.

2) Mode II - Alphanumeric - Four 6-Bit Characters - Four 6-bit frames are punched for each SUN instruction. Each 6-bit frame corresponds to a pair of octal digits. Four 6-bit frames are loaded into IOR 0 and are punched onto the tape, the most significant 6 bits first, followed by the next significant 6 bits, until the IOR is completely unloaded. An SUN instruction is used to place the punch into Mode II and is followed by an LDN instruction, which loads the data to be punched into IOR 0.

SIMULTANEOUS OPERATION - The paper-tape punch and typewriter may be operated simultaneously in Mode II. This is accomplished by giving an SUN instruction with the operand $(00000140)_8$, which sets up both devices in Mode II.

3) BUZZ Mode - The BUZZ mode is initiated by depressing the TAPE FORWARD switch on the

front of the paper-tape punch console. This causes the punch to feed the tape forward while punching sprocket holes, which enables the operator to manually produce a short leader on any tape.

3.4.3 Monitor Typewriter

The RCA 110 monitor typewriter can print at a rate of 10 characters per second. The typewriter has two basic modes of operation:

1) Mode I - Alphanumeric - One 6-Bit Character - This mode is identical to punch Mode I, in which a single character is typed for each SUN instruction. Each character is represented by the least significant 6 bits of IOR 0, bits 0-5, and the remaining bits (6-23) are zeros. Typewriter control-function codes, such as carriage return $(56)_8$ and tab $(36)_8$, are transferred to the typewriter in the same manner as alphanumeric information. An SUN instruction, with an operand as specified in Appendix H, sets up the typewriter in Mode 1. The SUN instruction must precede an LDN instruction, which results in the transfer of information from HSM through IOR 0 to the typewriter.

2) Mode II - Alphanumeric - Four 6-Bit Characters - This mode corresponds to punch Mode II in which four 6-bit frames are transferred through IOR 0 for each SUN instruction. The most significant frame of IOR 0 is typed first, followed by the next significant frames.

   SIMULTANEOUS OPERATION - The monitor typewriter and paper-tape punch may be operated simultaneously in Mode II by using an SUN instruction with the operand $(00000140)_8$.

3.4.4 Paper-Tape Code

The punched-paper tape has eight columns across the tape. Refer to Figure 3-4 and Table 3-1 for the paper-tape code format. A punched hole is a binary one, except for the column of small sprocket holes. A two-digit binary coded number or carriage return may be punched in each row with the corresponding parity bit. Row 4 may be used as 2^3 for the bits to the right, or as 2^0 for those to the left.

3.5 MAGNETIC TAPE

The RCA 110 Computer has provisions for magnetic-tape operations. These operations are on-line, under program control, and compatible with many off-line tape format requirements.

The RCA 110 Computer can control from one to ten magnetic-tape stations through one central tape-station buffer. Each of the units is individually addressable under program control.

In the discussion of the magnetic-tape-stations features, and programming methods, refer to Figures 3-5 and 3-6.

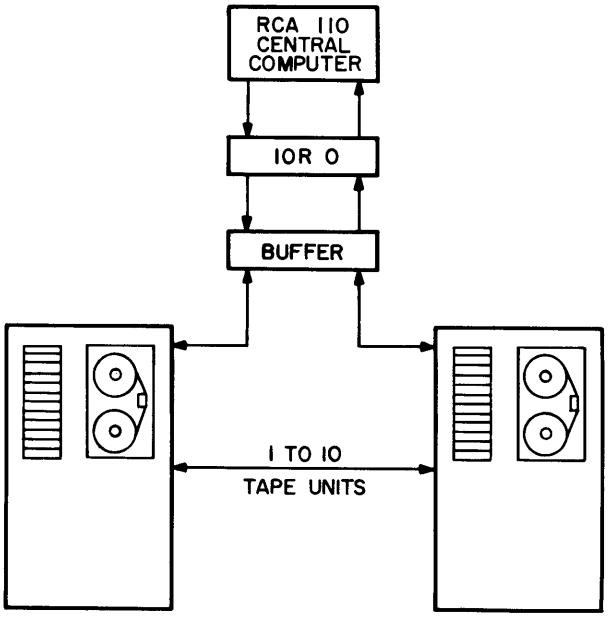


Figure 3-5. Magnetic-Tape Communication Channels

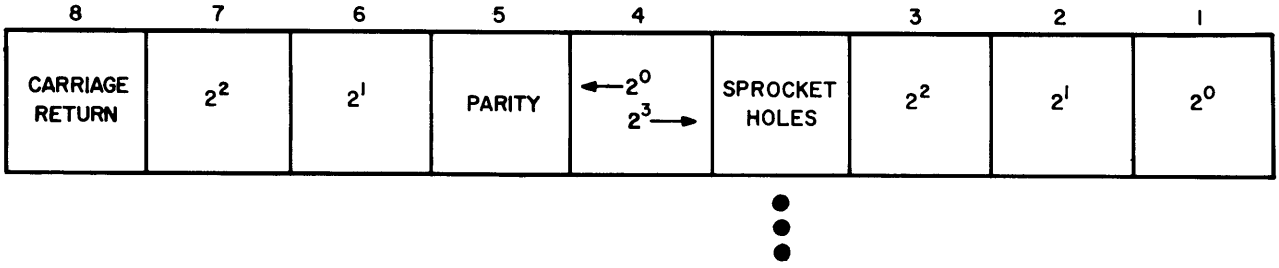


Figure 3-4. Paper-Tape Column Format

TABLE 3-1. INPUT/OUTPUT CODES-CHARACTER OR FUNCTION

UPPER CASE	LOWER CASE	PAPER-TAPE CODE								COMPUTER OCTAL CODES
		CR	2^2	2^1	PAR 2^0	2^3	SPR	2^2	2^1	2^0	
$	0			•			•				20
'	1						•			•	01
@	2						•		•		02
#	3				•		•		•	•	03
=	4						•	•			04
&	5				•		•	•		•	05
¢	6				•		•	•	•		06
?	7						•	•	•	•	07
÷	8					•	•				10
Δ	9				•	•	•			•	11
★	*		•			•	•		•		52
a	A		•	•			•			•	61
b	B		•	•			•		•		62
c	C		•	•	•		•		•	•	63
d	D		•	•			•	•			64
e	E		•	•	•		•	•		•	65
f	F		•	•	•		•	•	•		66
g	G		•	•			•	•	•	•	67
h	H		•	•		•	•				70
i	I		•	•	•	•	•			•	71
j	J		•		•		•			•	41
k	K		•		•		•		•		42
l	L		•				•		•	•	43
m	M		•		•		•	•			44
n	N		•				•	•		•	45
o	O		•				•	•	•		46
p	P		•		•		•	•	•	•	47
q	Q		•		•	•	•				50
r	R		•			•	•			•	51
s	S			•	•		•		•		22
t	T			•			•		•	•	23
u	U			•	•		•	•			24
v	V			•			•	•		•	25
w	W			•			•	•	•		26
x	X			•	•		•	•	•	•	27
y	Y			•	•	•	•				30
z	Z			•		•	•			•	31
%	(		•	•	•		•				60
+	)		•		•	•	•		•	•	53
,	, (comma)			•	•	•	•		•	•	33
.	. (period)		•	•		•	•		•	•	73
:	/			•	•		•			•	21
" (quote)	_ (underline)		•				•				40
	Upper case		•	•	•	•	•	•			74
	Lower case		•	•	•	•	•		•		72
	Tab			•	•	•	•	•	•		36
	Carriage Return	•					•				56
	Space				•		•				00
	Forward Stop					•	•		•	•	13
	Delete		•	•	•	•	•	•	•	•	77
	Reverse Stop				•	•		•			14

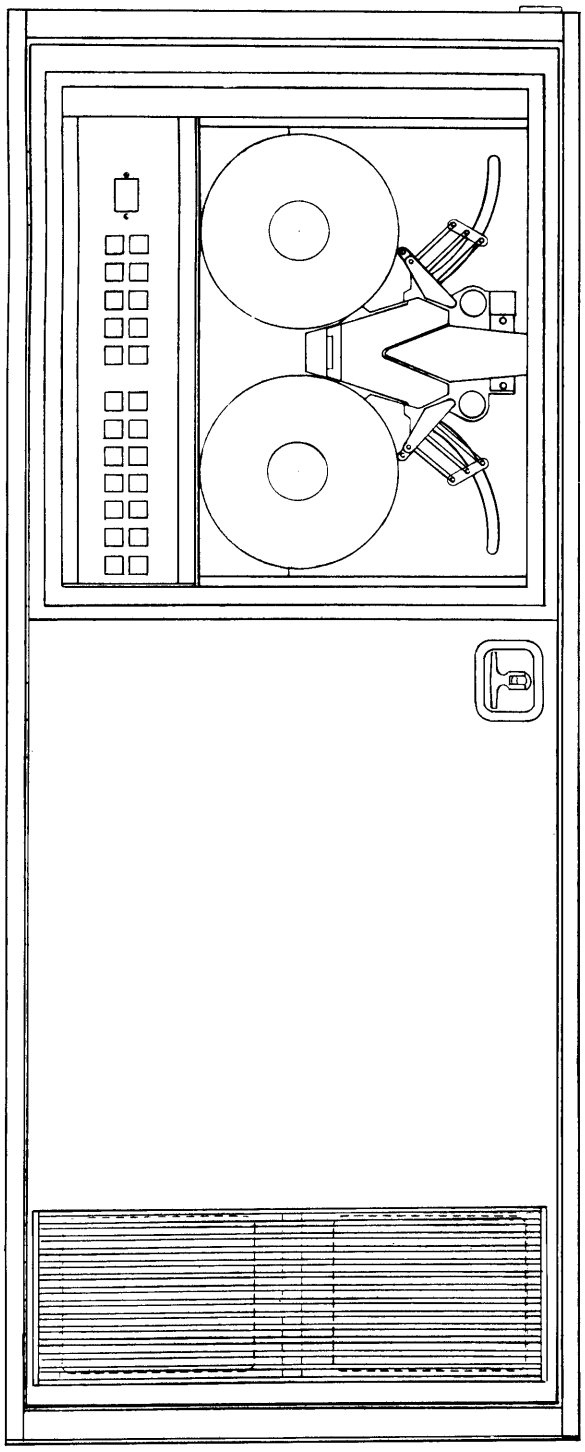


Figure 3-6. Magnetic-Tape Station

3.5.1 Tape-Station Buffer

All tape stations are linked to the RCA 110 Computer through a special magnetic-tape-station buffer, which communicates with the computer via Input/Output Buffer register (IOR 0).

The buffer unit is used to provide status and timing information and to control the interface between the tape units and the RCA 110 Computer. The buffer can be connected to from one to ten tape units. Each tape unit has the provision for altering the address to which it responds. The buffer may only select and operate one tape unit at a time. All other tape units cease operations, except rewind.

3.5.2 Error Checking

All data being transferred through the buffer is checked for errors in three different ways:

- VERTICAL REDUNDANCY CHECK - Each character is examined for either odd or even parity count. Normally, the character is considered to be binary and checked for an odd count, with a BCD signal from the computer resulting in an even count. Whenever the parity count is incorrect, a parity error indicator is illuminated and an error signal is sent to the computer.

- ECHO CHECK - During a WRITE operation, the buffer reads the tape as it is being written and compares the information being written with that being read. If the comparison does not agree, tape motion ceases and a TAPE UNIT INOPERABLE signal (TDNF) is sent to the computer.

- LONGITUDINAL CHECK - Each bit track is checked for an odd or even count. The count of bits in a WRITE operation should always be even. If it is not, the computer is signaled that there is a parity error.

3.5.3 Tape-Station Characteristics

Each tape unit communicates with the RCA 110 Computer through the magnetic-tape-station buffer, the IOA, and IOS lines. The buffer is used for data transfers, the IOA lines to provide control signals, and the IOS lines to sense the tape station operating status.

Each tape unit can write or read at a rate of 15,000 characters per second. Information is recorded on

the tape in IBM-729-II low density format, 200 characters to the inch, which is compatible with IBM 704, 709, and 7090 tape units.

3.5.4 Magnetic-Tape Characters

Each character consists of six magnitude bits plus one bit for parity. Characters may be either in binary or BCD formats, with odd parity for binary and even parity for BCD characters. The chart below indicates the BCD code to be used on the magnetic tape. All RCA 110 Computer characters are equivalent to IBM characters with the exception of SPACE and ZERO. These characters are recorded in IBM format on tape and converted to the equivalent RCA 110 Computer characters by the tape station electronics. Refer to Table 3-2.

	RCA 110 Character			IBM Character
SPACE	(000000)	=	BLANK	(010000)
ZERO	(010000)	=	ZERO	(001010)

The RCA 110 Computer has provisions for utilizing the AMPEX TM-4 Tape Station with the following recording characteristics:

1) Packing Density - Low density 200 characters per inch, 0.005 inch per character

2) Recording Modes - BCD or Binary - IBM - 729-II Format

3) Speed - 75 inches per second

4) Start Time - 75 IPS ± 10% = 3.3 milliseconds
   75 IPS ± 5% = 6.3 milliseconds

5) Starting Distance - 0.203 inch (maximum)
   0.183 inch (nominal)
   0.162 inch (minimum)

6) Stop Time - 1.8 millisecond (maximum)

7) Stop Distance - 0.100 inch (maximum)
   0.065 inch (nominal)
   0.030 inch (minimum)

8) Inter-Record Gap - 0.875 inch (maximum)
   0.750 inch (nominal)
   0.685 inch (minimum)

9) Write - EOF Mark - 45 milliseconds

10) Rewind Time - 3 minutes for 2400 foot-reel

TABLE 3-2. BCD TAPE CODES

CHARACTER	IN STORAGE	ON TAPE
0	01 0000	00 1010
1	00 0001	00 0001
2	00 0010	00 0010
3	00 0011	00 0011
4	00 0100	00 0100
5	00 0101	00 0101
6	00 0110	00 0110
7	00 0111	00 0111
8	00 1000	00 1000
9	00 1001	00 1001
#	00 1011	00 1011
@	00 1100	00 1100
&	01 0000	11 0000
A	01 0001	11 0001
B	01 0010	11 0010
C	01 0011	11 0011
D	01 0100	11 0100
E	01 0101	11 0101
F	01 0110	11 0110
G	01 0111	11 0111
H	01 1000	11 1000
I	01 1001	11 1001
+ } 0 }	01 1010	11 1010
.	01 1011	11 1011
¤	01 1100	11 1100
-	10 0000	10 0000
J	10 0001	10 0001
K	10 0010	10 0010
L	10 0011	10 0011
M	10 0100	10 0100
N	10 0101	10 0101
O	10 0110	10 0110
P	10 0111	10 0111
Q	10 1000	10 1000
R	10 1001	10 1001
0̄	10 1010	10 1010
$	10 1011	10 1011
*	10 1100	10 1100
	00 0000	01 0000
/	11 0001	01 0001
S	11 0010	01 0010
T	11 0011	01 0011
U	11 0100	01 0100
V	11 0101	01 0101
W	11 0110	01 0110
X	11 0111	01 0111
Y	11 1000	01 1000
Z	11 1001	01 1001
‡	11 1010	01 1010
,	11 1011	01 1011
%	11 1100	01 1100

3.5.5 Data Organization

WORD - Information is normally recorded on the tape in groups of six BCD characters which constitute a 36-bit tape word. Each WORD consists of four data characters followed by two BLANK characters.

Since the RCA 110 Computer word consists of 24 bits, only the first four data characters are interpreted, with the two BLANK characters being ignored. Thus, the programmer does not have to consider the packing and unpacking of words. Refer to Figure 3-7.

RECORD - A group of one or more tape WORDs constitutes a RECORD. Each RECORD on tape is followed by an END OF RECORD gap (EOR) of 0.75 inch. If a WORD contains less than six characters, the group of characters is sent to the computer followed by an EOR gap defined by the last character.

FILE - A group of one or more RECORDs constitutes a FILE. A FILE is followed by an END OF FILE (EOF) mark (00 1111), preceded by a ROW CHECK SUM which is used to maintain tape recording parity.

3.6 MAGNETIC-TAPE PROGRAMMING

The RCA 110 Computer communicates with the magnetic-tape station through several different channels which perform the following functions:

- DATA TRANSFER - All data transmitted between the magnetic tape stations and central computer passes through IOR 0.
- TAPE-STATION COMMANDS - The IOA lines are used to carry control signals from the computer to the tape stations.
- STATUS DETERMINATION - The operating status of the tape stations may be determined through the IOS lines.

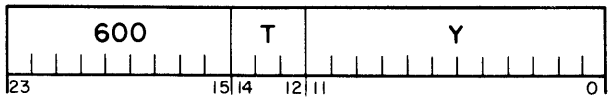
The programmer may instruct the RCA 110 Computer to perform each of these functions with normal RCA 110 input/output instructions.

3.6.1 Data Transfer

The RCA 110 Computer is instructed to perform all data transfers to and from the tape stations by two sets of instructions. The first set is used to READ TAPE; the second set to WRITE TAPE.

- READ TAPE - When the programmer wishes the RCA 110 Computer to read data from a magnetic tape station, two instructions are to be used:

SUN



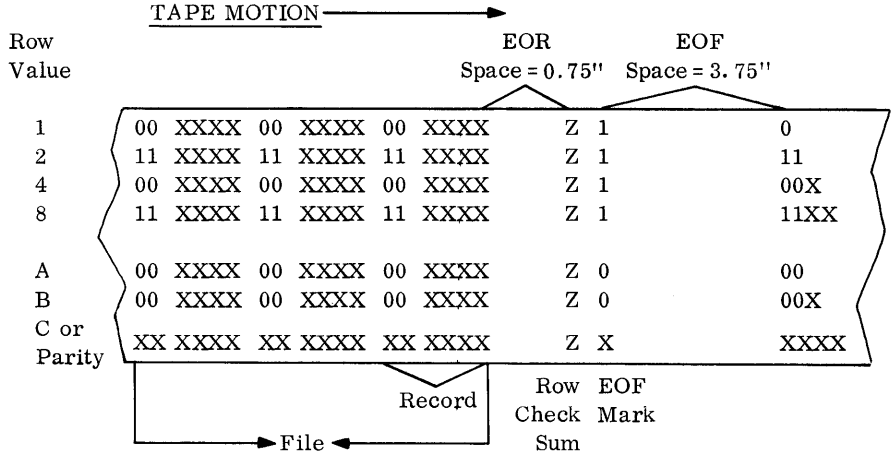


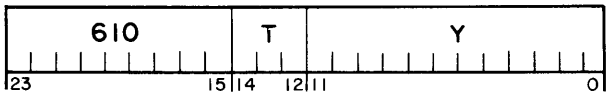
Figure 3-7. Tape Character Format

Set up tape station and send first word to IOR 0.

T portion refers to an index register which contains the amount by which the Y portion is to be modified. If T is 0, Y is not to be modified.

Y portion refers to an HSM location which contains a mask used to place the tape station in a particular operating mode.

STN



Store contents of IOR 0 in HSM, location Y.

T portion refers to an index register which contains the number of words to be read from tape.

The content of the index register is not changed by the execution of the READ TAPE instruction.

Y portion specifies the HSM location to receive the first word from tape; Y + 1 will receive second word, Y + 2 third word, etc.

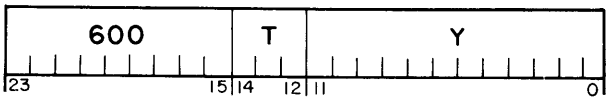
Notes:

(1) The SUN instruction must always precede the STN instruction on a READ TAPE operation.

(2) There are two modes of writing tape, binary and BCD. If the tape is written in one mode and read in another, the parity alarm will occur. If the programmer wishes to READ TAPE in the BCD mode, the SUN instruction mask should be (00000600).

- WRITE TAPE - When information is to be written on magnetic tape from the RCA 110 Computer, the following instructions are used:

SUN

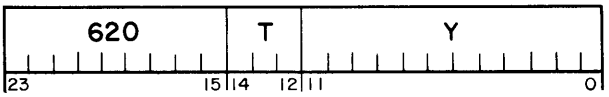


Set up tape station and reset IOR 0.

T portion refers to an index register which contains the amount by which the Y portion is to be modified. If T is 0, Y is not to be modified.

Y portion specifies a HSM location which contains the mask required to place the tape station in a particular recording mode.

LDN



Load contents of HSM location Y into IOR 0.

T portion refers an index register which contains the number of words to be written on tape.

The content of the index register is not changed by the execution of WRITE TAPE instruction.

Y portion specified the HSM location from which the first word is to be written on tape; Y + 1 will contain the second word; Y + 2 the third, etc.

When the WRITE TAPE operation is complete, the tape station automatically forms an EOR gap and writes a parity check character.

Notes:

(1) The SUN instruction must always precede the LDN instruction on a WRITE TAPE operation.

(2) If the last word being written on tape is also to be the last word in a file, the programmer should use an additional SUN instruction after the LDN to write an EOF mark.

(3) If the WRITE TAPE instruction is to be in the BCD mode, the SUN instruction mask should be 0000 1200.

Example:

LOCATION	MNEMONIC OP CODE	TAG	OPERAND ADDRESS
A	SUN	T	Y - Set up tape station for recording
A + 1	LDN	T	Y - Transfer data from HSM to tape
A + 2	SUN	T	Y - Write EOF

The HSM location specified in the Y portion of the second SUN instruction contains the mask (00040000),

which causes an EOF mark (00 1111) to be written on the tape.

3.6.2 Tape-Station Commands

The programmer will have the ability to issue ten different commands to each tape station. The commands are issued by the use of the SUN instruction, with the Y portion of the instruction specifying a particular HSM location which contains the command mask. Refer to IOA assignments in Appendix H.

The ten commands, their masks, and their functions are as follows:

1) SET UP BCD MODE - (00000200) - This command places the tape station in a BCD mode and can be used simultaneously with a TAPE READ (00000600) or a TAPE WRITE (00001200) instruction. If a TAPE WRITE or TAPE READ command is issued without specifying BCD, the data transfer is in binary coding.

2) READ TAPE - (00000400) - Commands a word to be read from a tape into IOR 0. Tape is read in binary unless BCD mode is specified (00000600). After completion of a READ TAPE instruction, the tape station remains inoperable for three milliseconds.

3) WRITE TO TAPE - (00001000) - Commands a word to be read from IOR 0 and written onto tape. Tape is written in binary unless BCD mode is also specified (00001200).

   An EOR gap is placed at the end of each group of words upon completion of the WRITE TAPE command.

   If a EOT marker is under the WRITE head as the WRITE TAPE command is given, forward tape motion is inhibited and the EOT indicator is lighted. If the EOT marker is detected during the execution of the command, the tape unit will only be able to receive a maximum of 800 additional words of data.

   After completion of a WRITE TAPE instruction, the tape station remains inoperable for seven milliseconds.

4) BACK SPACE ONE RECORD - (00002000) - Commands the tape station to rewind the tape to the previous EOR gap.

5) BACK SPACE ONE FILE MARKER - (00004000) - Commands the tape station to rewind the tape to the previous EOF mark.

6) SKIP NEXT RECORD (00010000) - Commands the tape station to move the tape forward to the next EOR gap.

7) SKIP FILE - (00020000) - Commands the tape station to move the tape forward to the next EOF mark.

8) WRITE EOF - (00040000) - Commands tape station to write an EOF mark (00 1111) and its associated check character on tape.

9) REWIND - (00100000) - Commands tape station to rewind tape at high speed to load point.

10) REWIND AND UNLOAD - (00200000) - Commands tape station to rewind tape at high speed to the conductive leader. If no conductive leader has been used, the tape will be pulled from the reel and the tape transport disabled.

3.6.3 Status Determination

The programmer can determine the operating status of a particular tape station by the use of the SDN instruction, with the Y portion referring to a particular HSM location which contains a status determination mask. See IOS assignments in Appendix G.

Upon completion of the SDN instruction, L will contain the results of the sensing operation. See SDN instruction description Section 2. 7.

The SDN instruction can be used to detect nine different operating states of a magnetic tape station.

The nine states and required SDN masks are as follows:

1) BUFFER BUSY? - (00000040) - Tape station buffer busy and cannot accept commands.

2) EOT MARKER? - (00000100) - End-of-tape photosensitive marker is sensed.

3) TAPE UNIT INOPERABLE (TDNF)? - (00000200) -
   Tape unit not ready.
   Tape unit does not respond to a command.
   Write echo pulses not received during a TAPE WRITE operation.

4) PARITY ? - (00000400) - Logitudinal or vertical parity error detected.

5) UNACCEPTABLE COMMAND ? - (00001000) - Tape unit busy
Tape unit inoperable
BACK SPACE or REWIND command when beginning of tape detected.
READ, WRITE, or SKIP command when EOT is detected.
READ or SKIP command following a WRITE command without intervening BACK SPACE or REWIND commands.
Command given when no tape station has been selected.

6) EOR GAP ? - (00002000) - Determines if EOR GAP is under read head of tape station.

7) EOF MARK ? - (00004000) - Determines if an EOF mark has been sensed during a read operation.

8) TAPE AT LOAD POINT ? - (00010000) - Determines if tape is at load point.

9) TAPE REWINDING ? - (00020000) - Determines if selected tape station is in a rewind mode.

TABLE 3-3. INPUT/OUTPUT ADDRESS LINES

SU(N) Set Up I/O Device N = 0	
	Mask
Paper-Tape Reader	
Read octal mode	00000002
Read alphanumeric (1) character mode	00000010
Read alphanumeric (4) character mode	40000000
Rewind	00000020
Paper-Tape Punch	
Punch alphanumeric (1) character mode	00000004
Punch alphanumeric (4) character mode	00000040
Typewriter	
Type alphanumeric (1) character mode	00000001
Type alphanumeric (4) character mode	00000100
Color black	10000000
Color red	20000000
Magnetic-Tape Station	
Set up BCD Mode	00000200
Read Tape	00000400
Write Tape	00001000
Back Space One Record	00002000
Back Space One File Marker	00004000
Skip Next Record	00010000
Move to Next File Mark	00020000
Write End-of-File (EOF) Marker	00040000
Rewind	00100000
Rewind to Load Point and Unload	00200000

TABLE 3-4. INPUT/OUTPUT SENSE LINES

SD(N) Sense Device N N = 0	
	Mask
Paper-Tape Reader	
Reader out of tape?	00000002
Reader rewinding?	00000020
Paper-Tape Punch	
Punch out of tape?	00000004
Typewriter	
Power off ?	00000001
Sense Switches	
No. 1 On ?	40000000
No. 2 On ?	20000000
No. 3 On ?	10000000
No. 4 On ?	04000000
No. 5 On ?	02000000
No. 6 On ?	01000000
IOR 0	
Busy?	00000010
Magnetic-Tape Station	
Buffer Busy?	00000040
End-of-Tape (EOT) Marker ?	00000100
Tape Unit Inoperable ?	00000200
Parity Error ?	00000400
Unacceptable Command?	00000100
End-of-File Mark (EOF)?	00004000
Tape at Load Point?	00010000
Tape Rewinding ?	00020000
Note: In the entire list a YES answer is indicated by a resultant one in the corresponding position of L after the SDN. A NO answer is indicated by a zero.	

SECTION 4

COMPUTER OPERATION

4.1 MAINTENANCE AND CONTROL PANEL

All computer controls and indicators are grouped on the Maintenance and Control panel. This panel displays the contents of the M REGISTER, P REGISTER, PRIORITY REQUEST, PRIORITY STATUS registers, PARITY and ALARM indicators, and Word Time (WT) status. This panel also provides a START control, manual input switches, and Program STOP (P STOP) switches. The positions of the controls and indicators may be identified in Figure 4-1.

Each indicator displays the contents of its associated circuit. Register indicators display each bit. Word Time indicators show the word-time level of the instructions when halted.

4.1.1 M REGISTER

All instruction and data exchanges in the computer involve the M REGISTER. The data is retained in M as 24-bit words. The contents of this register can be observed by means of the indicator lamp display (top row in Figure 4-1) on the Maintenance and Control panel. Toggle switches associated with each lamp provide means for manual settings of the register when data is to be inserted or altered. An illuminated lamp

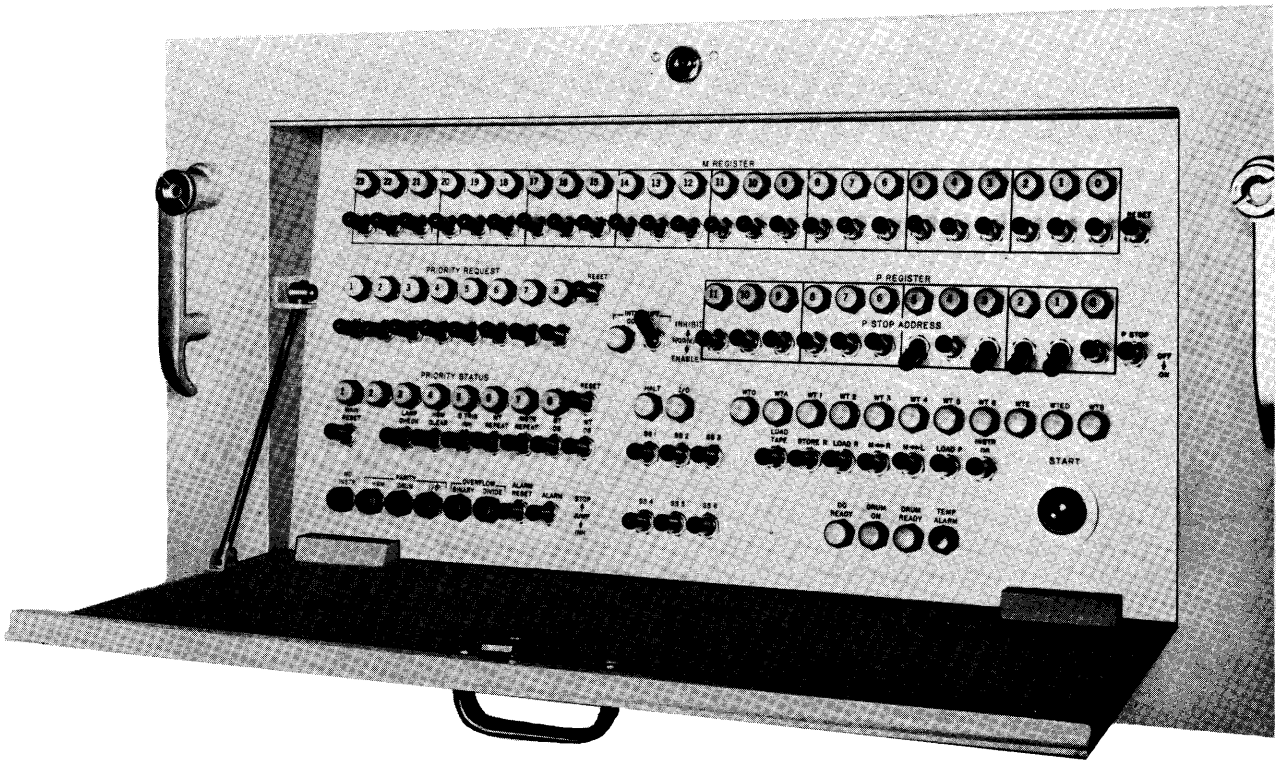


Figure 4-1. Maintenance and Control Panel

indicates that a binary one is present. If the lamp is not ON, a zero exists in that position of the word. The complete word (bits 0-23) is displayed on the panel. The register is cleared to zero with the RESET switch.

4.1.2 PRIORITY REQUEST

The eight PRIORITY REQUEST lamps indicate standby requests for the program to branch. These requests are in storage for the computer to execute under predesignated conditions. The associated request control switches enable the operator to request the program of a certain level. The register is cleared to zero with the RESET switch.

4.1.3 PRIORITY STATUS

The eight PRIORITY STATUS lamps under the request controls indicate the priority of the currently running program. No status is indicated if current program has no priority assignment. Only one status is displayed at any time. The register is cleared to zero with the RESET switch.

4.1.4 P REGISTER

The contents of the Program Counter (PC) are displayed by the indicator lamps. The contents are the address of the instruction currently being executed. If the computer is in a STOP condition, the contents may be the location of the next instruction, depending on the word time (WT) status.

The operator may set an address into the P REGISTER where a stop is requested. The P STOP ADDRESS 12-bit location is placed in the register by means of the P STOP ADDRESS toggle switches.

4.1.5 P STOP

If the P STOP switch is in the ON position, the computer stops when the contents of the P REGISTER equal that which has been set by the P STOP ADDRESS switches. The instruction of that HSM location is in the M REGISTER and ready to be executed. If the switch is in the OFF position, the P STOP address switches are disabled.

4.1.6 INTERRUPT CONTROL

When the INTERRUPT CONTROL lamp is ON, it indicates that the current program may be interrupted. The switch which sets this condition has three positions: INHIBIT, NORMAL, and ENABLE. When it is up (INHIBIT), no interrupts are possible. When the switch is down (ENABLE), no inhibits are possible and interrupts are permitted. In the center position (NORMAL), the interrupts and inhibits are under program control.

4.1.7 MAIN RESET Switch

The MAIN RESET switch clears all registers to zero and stops the internal clock.

4.1.8 LAMP CHECK Switch

The LAMP CHECK switch tests all lamps on the panel (except TEMP ALARM) without destroying the contents of the registers. If any lights fail to come on when the switch is manipulated, the bulbs should be replaced.

4.1.9 HSM CLEAR Switch

To clear the core memory locations to zero, the HSM CLEAR switch is held down while the START button is depressed for several seconds.

4.1.10 C TRIG INH Switch

To prevent checking the C counter when it is zero, the C TRIG INH switch is placed in the down position. The C counter is a six-bit counter which keeps track of the number of bit shifts in SHIFT and NORMALIZE instructions, and counts the iterations required in Divide and Multiply. It counts from a number N down to zero. It is also used to determine the priority level and program number when the priority interrupt process is initiated.

4.1.11 WT REPEAT Switch

The WT (Word Time) REPEAT switch inhibits the shifting of the word time register. This permits repetition of the word time (indicated by the WT indicators) of the instruction whose address appears in the P REGISTER, each time the START button is depressed. The internal clock stops when this switch is depressed.

4.1.12 INSTR REPEAT Switch

When the INSTR REPEAT switch is down, the instruction whose address appears in the P REGISTER

is repeated each time the START button is depressed.

4.1.13 BT OS Switch

When the BT OS (Bit Time One Shot) switch is down, the computer stops after one-bit time. This permits execution of one-bit time for each depression of the START button. The internal clock stops when this switch is depressed.

4.1.14 WT OS Switch

When the WT OS (Word Time One Shot) switch is down, the computer halts after each complete word time. This permits execution of one-word time for each depression of the START button. The internal clock stops when this switch is depressed.

4.1.15 HALT Switch

If the HALT lamp is ON, it indicates that the computer is in a Halt instruction.

4.1.16 I/O Indicator

If the I/O lamp is ON, it indicates that the computer is in an Input/Output instruction.

4.1.17 Sense Switches

The SENSE switches (SS1-SS6) provide a means of making conditional transfers within a program.

4.1.18 WT Indicator

The WT (Word Time) lamps display the status levels or word times of an instruction. Each instruction is processed by means of a specified number of word times. Although the number of word times varies according to the instruction, all instructions have certain word times which are common. These are:

WTO - Moves instruction from the M REGISTER to the Command register.
WTE - Reads the next instruction out of memory and into the M REGISTER.
WTA - When there is address modification, this word time must occur.
WT1 - WT6 - Depending on the instruction, one or more of these word times must occur.
WTS - Moves data to or from HSM in an input/output operation.
WTED - Occurs upon completion of an instruction and continues until WTO of the next instruction.

4.1.19 LOAD TAPE Switch

The LOAD TAPE switch sets up two instructions which cause reading of the paper tape. When the switch is moved to the upper position, a Store Both (STB) instruction is performed. The Left accumulator (L) is automatically set with one's in bits 0-11, and a one is placed in bit 1 of the Right accumulator (R). Other bits are not affected. The contents of these two registers are stored in memory locations zero and one, and they provide the necessary information for the bootstrap loader. (The L and R accumulators must be reset to zero before the LOAD TAPE switch is manipulated.)

When the switch is moved to the lower position, a pseudo instruction (040) is set into the M REGISTER which starts the paper-tape reader and loads the first two instructions on the paper tape into memory locations 0040_8 and 0041_8. Program control is then transferred to location 0040_8. The instructions on the tape being loaded must be in octal format.

Note: The switches as described in sections 4.1.19 through 4.1.24 are active when the internal clock is running.

4.1.20 STORE R Switch

When the STORE R switch is depressed, an STR (422) instruction is automatically performed. This results in the contents of the Right accumulator (R) being placed in the memory location which has been manually set into bits 0-11 of the M REGISTER. The contents of R are not changed in the process.

4.1.21 LOAD R Switch

When the LOAD R switch is depressed, an LDR (402) instruction is automatically performed, placing in the Right accumulator the contents of the memory location which had been set manually into bits 0-11 of the M REGISTER. The contents of R are not changed in the process.

4.1.22 M◄►R Switch

When depressed, the M◄►R switch causes the exchange of data between the M REGISTER and the R accumulator.

4.1.23 M◄►L Switch

The M◄►L switch causes the exchange of data between the M REGISTER and L accumulator.

4.1.24 LOAD P Switch

The LOAD P switch initiates an Unconditional Transfer (TRA) instruction. This results in the contents of bits 0-11 of the M REGISTER being loaded into the program counter, and the contents of the location specified by bits 0-11 of the M REGISTER being loaded into the M REGISTER.

4.1.25 INSTR OS Switch

The INSTR OS switch stops the computer to cease operation at the end of the current instruction, but allows the internal clock to continue running. This permits the execution of one instruction for each depression of the START button.

4.1.26 NO INSTR Indicator

If the NO INSTR lamp is ON, it indicates that the computer has detected a forbidden operation code. A trapping operation occurs, which transfers program control to location 0017_8 and the contents of the PC to 0027_8.

4.1.27 PARITY Alarms

The three conditions for PARITY alarms are:

- HSM — Parity error detected on readout from memory
- DRUM — Parity error detected on transfer to, or from, the drum
- I/O — Parity error detected on transfer to, or from, an input/output device.

On these alarms, trapping occurs to location 0017_8.

4.1.28 OVERFLOW Indicators

The two OVERFLOW alarm conditions are:

- BINARY — Overflow detected on arithmetic operation.
- DIVIDE — Error detected in divide operation.

On these alarms, trapping occurs to location 0013_8.

4.1.29 ALARM RESET Switch

The ALARM RESET switch resets the alarm indicator lamps.

4.1.30 ALARM Switch

The ALARM switch is a three-position switch. When it is in the INHIBIT position (INH), the ALARM light is activated, but the computer does not perform the trap transfer. If it is in the STOP position, the alarm transfer is not performed, but the computer is halted at the end of the word time which causes the alarm. The P REGISTER then contains the address of the instruction which was being executed when the alarm occurred. When the switch is in the JUMP position, the computer performs normal trap operations. The ALARM switch does not affect the POWER FAILURE operation.

4.1.31 DC READY Indicator

The DC READY lamp indicates that the DC power is ON.

4.1.32 DRUM ON Indicator

The DRUM ON lamp indicates that the drum is operating, but not revolving at proper speed.

4.1.33 DRUM READY Indicator

The DRUM READY lamp indicates the drum has reached proper speed.

4.1.34 TEMP ALARM Indicator

When the TEMP ALARM lamp is ON, it indicates that the temperature within one of the computer racks is out of tolerance.

4.1.35 START Button

The START button initiates all operations of the computer by starting the timing-pulse generator and/or shifting the contents of the word time register.

4.2 POWER CONTROL PANEL

4.2.1 ELAPSED TIME Meter

The ELAPSED TIME meter records the amount of time that power has been applied to the Power Control panel.

4.2.2 MASTER Switch

The MASTER switch controls the primary AC power to the computer AC circuits.

4.2.3 PS TEST Switch

The PS TEST switch disconnects all loads from all DC power supply outputs and disconnects all DC voltage from the deviation-voltmeter-selector switch.

4.2.4 DRUM ON Switch

The DRUM ON switch controls AC power to the drum motor.

4.2.5 DRUM OFF Switch

The DRUM OFF switch disconnects AC power to the drum motor.

4.2.6 POWER ON Switch

The POWER ON switch turns on AC power to all DC power supplies in proper sequence.

4.2.7 POWER OFF Switch

The POWER OFF switch disconnects AC power to all DC power supplies in proper sequence.

4.3 PAPER-TAPE PUNCH/READER CONTROLS AND INDICATORS

All Paper-Tape Punch/Reader controls and indicators are located on the Paper-Tape Reader, with the exception of the LOAD TAPE switch on the Maintenance and Control panel. The indicators show that power is applied.

4.3.1 ON-OFF Switch

The Punch and Reader each have an ON-OFF switch, which turns on AC power to the individual motors.

4.3.2 PUNCH MODE Control

The Paper-Tape Punch-Mode switch has two positions: BUZZ and AUTOMATIC. In the BUZZ position, paper tape is fed through the punch heads with only sprocket holes being perforated. This permits a leader to be made. In the AUTOMATIC position, the paper tape is fed through the punch heads upon command from the RCA 110 Computer.

4.3.3 READER MODE Control

The Reader-Mode switch has three positions: FORWARD, REVERSE, and AUTOMATIC. In the FORWARD position, paper tape may be run forward through the reader head (without computer action) to permit the operator to start the tape at any point. In the REVERSE position, paper tape may be run in reverse through the head (without computer action) to permit rewinding the paper tape. In the AUTOMATIC position, the tape moves forward through the reader upon commands from the computer.

4.4 OPERATING PROCEDURES

4.4.1 POWER ON Procedure

To turn ON the computer, the following steps must be followed:

1) Turn ON the external source of AC supply voltage.

2) Press MASTER switch on Power Supply panel.

3) Press DRUM ON switch on the Power Supply panel. (The DRUM READY indicator on the Maintenance and Control panel lights when the drum reaches proper speed.)

4) Press POWER ON switch on the Power Supply panel and wait for the DC READY indicator on the Maintenance and Control panel to illuminate.

4.4.2 POWER OFF Procedure

Normal turn-OFF procedure is as follows:

1) Halt the computer by placing the INSTR OS switch down.

2) Press the POWER OFF button on the Power Supply panel.

3) Press the DRUM OFF button on the Power Supply panel.

4) Press the MASTER switch on the Power Supply panel.

Note: Emergency turn-OFF may be accomplished by pressing the MASTER switch only.

If the power-supply voltages drift out of tolerance or fail, and automatic cycle-down of power occurs. The magnetic-core memory voltages are automatically cycled to insure no loss of information.

4.4.3 Preliminary Setup

Before running a program, the computer must be brought out of the MAIN RESET state. This is accomplished by depressing the MAIN RESET switch, which resets all flip-flops and arithmetic registers and stops the clock.

1) Place the INSTR OS switch in the down position.

2) Press the START button. (The NO INSTR and HSM PARITY alarm lamps come ON.)

3) Depress the ALARM RESET button.

These steps should be followed whenever the MAIN RESET switch is depressed.

4.5 LOADING

4.5.1 General Program Loader

The General Data Program Loader (GDPL) tape contains the basic boostrap and loading routine and provides a convenient means of loading programs.

1) Clear L and R

   Depress M RESET.
   Depress M◄►R.
   Depress M RESET.
   Depress M◄►L.

2) Place tape in reader.

3) Depress INSTR OS switch.

4) Depress M RESET.

5) Move LOAD TAPE switch upward, then back to center.

6) Set INSTR OS switch to center position.

7) Move LOAD TAPE switch downward, then back to center.

These movements initiate the loading of the first two instructions of the bootstrap loader and transfer control to that program. After the loader has been read into the computer, the computer halts with the HALT and WT1 indicators ON. See Figure 4-2 for sample of GDPL routine.

4.5.2 Program Tape

After the GDPL, this procedure is to be followed in loading a program tape:

1) Place tape in reader.

2) Depress INSTR OS switch.

3) Press START button.

4) Set 0044 into M REGISTER.

5) Depress LOAD P switch.

6) Raise INSTR OS switch.

7) Press START button.

After the program tape has been loaded, the computer stops with the HALT and WT1 indicators ON.

If the loaded program is then to be executed by the computer, proceed as follows:

1) Place INST OS switch down.

2) Press START button (this causes WT1 to step to WTE).

3) Set starting address of program into bits 0-11 of M REGISTER.

SYMBOLIC LOCATION	ABSOLUTE LOCATION	SYMBOLIC INSTR.	OP CODE	TAG	ABSOLUTE ADDRESS	SYMBOLIC ADDRESS	REMARKS	WORD TIMES
							-1→XR1	
							2→LOC 0000	
							PUSH LOAD P.T.	
	40	SU(0)	600	0	0000	0	BOOT STRAP LOADER	
	41	ST(0)	610	1	0041	41	RTN	
	42	TD(1)	311	1	0040	40		
NEG 1	43	HLT	360	0	7777	7777	-1 BITS 0-11	
	44	SU(0)	600	0	0000	0	READ 1ST ADDRESS	
	45	ST(0)	610	0	0063	ADDRESS	STORE ADDRESS	
MODIFY	46	LDL	401	0	0063	ADDRESS	REDUCE ADDRESS BY 1	
	47	SUB	110	0	0062	V(1)		
	50	STA	430	0	0054	STORE	STORE FOR STORAGE	
	51	LDL	401	0	0043	NEG 1		
	52	STA	430	0	0001	BB1	-1→XR1	
	53	SU(0)	600	0	0000	0	READ DATA	
STORE	54	ST(0)	610	1	0000	( )	STORE DATA	
	55	TD(1)	311	1	0053	*-2		
	56	SU(0)	600	0	0000	0	HERE IF ONE EOM	
	57	ST(0)	610	0	0063	ADDRESS		
	60	TRA	357	0	0046	MODIFY	RETURN FOR NEXT BLOCK	
	61	HLT	360	0	0000	0	HALT WHEN DONE	
V(1)	62		000	0	0001	1		
ADDRESS	63		000	0	0000			

RCA 110 COMPUTER
CODING SHEET

TITLE: GENERAL DATA PROGRAM LOADER ROUTINE DATE: 7/24/62
NAME: RCA 110 PAGE: 1 of 1

Figure 4-2. General Data Program Loader Routine

4) Press LOAD P switch.
5) Raise INST OS switch.
6) Press START button.

4.5.3 General Operations

Shown in Table 4-1 are directions for conducting general operations with the RCA 110 Computer.

TABLE 4-1

GENERAL OPERATING INSTRUCTIONS

OPERATION	DIRECTIONS
To stop	• Move INSTR OS switch to down position
For free run	• Raise the INSTR OS switch • Press the START button
To transfer control to a specific location in memory	• Load M, bits 0-11, with starting address • Depress the LOAD P switch • Press the START button
To stop at a given program address	• Set address in P STOP ADDRESS • Depress P STOP switch
To load all of memory with same word	• Depress HSM CLEAR switch • Hold START button down • Release START button, while depressing HSM CLEAR switch • Load pattern into M REGISTER • Release HSM CLEAR switch
To transfer to loaded program	• Move INSTR OS switch down • Depress START button • Depress M RESET switch • Enter starting address in bits 0-11 of M REGISTER • Depress LOAD P switch • Raise the INSTR OS switch • Press START button
To restart after I/O malfunction	• Depress MAIN RESET and follow "Preliminary Setup" instructions of Section 4.4.3.
To restart after ALARM STOP or HALT command	• Depress the ALARM RESET button • Set ALARM switch to INHIBIT • Depress the INSTR OS switch • Press the START button • Raise the INSTR OS switch • Press the START button

APPENDIX A NUMBER SYSTEM

A number system is a group of symbols and a method used to count the number of units in a collection. Each number system has a base or radix which is equal to the number of different symbols used in the system. Most of us use the decimal number system of counting without realizing that it is but one of many types of number systems. However, most digital computers use three basic number systems:

- Decimal
- Binary
- Octal

A-1 DECIMAL NUMBER SYSTEM

The principle of representing numbers in the decimal number system by symbol and position is familiar to all. That is, we can represent the values zero to nine using the basic set of symbols. If we wish to count above the highest valued symbol, we add one to the column to the left and replace the highest value with the lowest valued symbol in the system.

Example:

9 = Highest Valued Symbol
10 is Highest Valued Symbol + 1
(Thus, 1 is added to the next column to the left and 9 is replaced with 0.)
11 = Highest Valued Symbol + 2

99 = Highest Valued Symbols
100 is Highest Valued Symbols + 1
(Thus, 1 is added to the next column to the left and 9's are replaced with 0's.)

Thus, we see that the symbol in a column indicates the number of times we have counted through the symbol set in the next column to its right. This means that the value of a number is determined by two things: its relative value in the symbol set and its columnar position. This columnar principle applies to all number systems.

We can also represent the value of a column by the value of the radix raised to a particular power.

Column Number	4	3	2	1	
Column Value	R^3	R^2	R^1	R^0	R = Radix or base

The value of column 1 = R^0, column 2 = R^1, column 3 = R^2, etc. We then form a number by placing one of the allowable symbols of the number system into the various columns.

Thus, we think of the value of a number as being equal to the sum of the various symbols multiplied by the values of the respective columns.

If S_n is an allowable symbol in the system, the formula representing the value of the number is

$$S_nR^3 + S_nR^2 + S_nR^1 + S_nR^0 = \text{Value of a number}$$

We normally write a number and assume the columnar values.

Assumed column value =

R^3 R^2 R^1 R^0 becomes $S_nR^3 + S_nR^2 + S_nR^1 + S_nR^0$

S_n S_n S_n S_n

We can represent fractional values of numbers by using negative powers of the radix.

Example:

$1/10 = 10^{-1}$, $1/100 = 10^{-2}$, $1/1000 = 10^{-3}$ and

$2/10 = 2 \cdot 10^{-1}$, $5/100 = 5 \cdot 10^{-2}$, $7/1000 = 7 \cdot 10^{-3}$

The general formula representing a number is

$$R_n \ldots R^3\ R^2\ R^1\ R^0 \cdot R^{-1}\ R^{-2}\ R^{-3} \ldots R^{-n}$$

$$S_n \ldots S_n\ S_n\ S_n\ S_n \cdot S_n\ S_n\ S_n \ldots S$$

We place a "point" (.) between the symbols in column R^0 and R^{-1} to signify that the columns to the right of the "point" are fractional columns.

Thus, we can represent the sum of the following numbers:

$$7,000 + 200 + 50 + 9 + \frac{8}{10} + \frac{3}{100} \text{ as } 7259.83$$

$7 \cdot 10^3 + 2 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0 + 8 \cdot 10^{-1} + 3 \cdot 10^{-2}$ as 7259.83

A-2 BINARY NUMBER SYSTEM

Most digital computers use a number system called "binary". Binary means "two", signifying that we

have two symbols 0 and 1. The symbols 0 and 1 are called binary digits or bits.

Why do computers use binary rather than the decimal system? Most computers are binary in nature. That is, contacts are open or closed, materials are magnetized or not magnetized, flip-flops are ON or OFF. Thus, we can represent an ON condition by a 1 and an OFF condition by a 0.

If we are counting in the binary number system, we use the same principle which applies to the decimal number system.

Example:

0 = Lowest Valued Symbol
1 = High Valued Symbol
10 = Highest Valued Symbol + 1
(Thus, 1 is added to the next column to the left and 1 is replaced with 0.)
11 = Highest Valued Symbol + 2

The general formula applied to the binary system is:

Radix or base (R) = 2
Symbols (S_n) = 0, or 1

$$2^n \ldots 2^2 \; 2^1 \; 2^0 \cdot 2^{-1} \; 2^{-2} \ldots 2^{-n}$$

$$S_n \ldots S_n \; S_n \; S_n \cdot S_n \; S_n \ldots S_n$$

Thus, we can represent the sum of the following binary digits:

$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$ = 101.01 Binary
= 5.25 Decimal

To count in binary, use the following:

BINARY							DECIMAL				
Column Values											
2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}	10^1	10^0	.	10^{-1}	10^{-2}
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			
0	1	1	0					6			
0	1	1	1					7			
1	0	0	0					8			
1	0	0	1					9			
1	0	1	0				1	0			
					.1					.5	
					.0	1				.2	5
					.1	1				.7	5

A-3 BINARY ADDITION

Computation in the binary number system is quite simple. There are three rules for binary addition.

- Zero + Zero = Zero — 0 + 0 = 0
- Zero + One = One — 0 + 1 = 1
- One + One = Zero with a carry of one to the next column to the left — 1 + 1 = 1 0

Add the two binary numbers 10010 and 01010

BINARY						DECIMAL
Column Values						
	2^4	2^3	2^2	2^1	2^0	
(Carry)			(1)			
	1	0	0	1	0	18
+	0	1	0	1	0	10
	1	1	1	0	0	28

In the 2^0 or 1's column we have 0 + 0 = 0
In the 2^1 or 2's column we have 1 + 1 = 0
with a 1 carry to the 2 column
In the 2^2 or 4's column we have 0 + 0 + 1 (carry) = 1
In the 2^3 or 8's column we have 0 + 1 = 1
In the 2^4 or 16's column we have 1 + 0 = 1

A-4 BINARY SUBTRACTION

The rules for binary subtraction are:

- Zero - Zero = Zero — 0 - 0 = 0
- One - One = Zero — 1 - 1 = 0
- One - Zero = One — 1 - 0 = 1
- Zero - One = One with 1 borrowed from the next column to the left — 0 - 1 = 1

Subtract the two binary numbers, 01010 from 10010

BINARY						DECIMAL
Column Values	16	8	4	2	1	
(Borrow)		(1)				
	1	0	0	1	0	18
-	0	1	0	1	0	- 10
	0	1	0	0	0	8

In the 2^0 or 1's column we have $0 - 0 = 0$
In the 2^1 or 2's column we have $1 - 1 = 0$
In the 2^2 or 4's column we have $0 - 0 = 0$
In the 2^3 or 8's column we have $0 - 1 = 1$
with a borrow from the 16's column
In the 2^4 or 16's column we have $1 - 0 - 1$ (Borrow) $= 0$

A-5 RCA 110 SUBTRACTIONS - (Two's Complement Arithmetic)

All negative binary numbers are represented in the RCA 110 Computer in two's complement form. Two's complement of a binary number is formed by:

- Substituting 1's for 0's
- Substituting 0's for 1's (One's complement)
- Adding 1 to the result (Two's complement)

Example:

To express 101 as a negative quantity:

$101 = +5_{10}$
010 = One's complement of 5_{10}
$+001$
011 = Two's complement of $5_{10} = -5_{10}$

Also, the RCA 110 Computer does not perform pure binary subtraction. It considers the subtrahend (the number being subtracted) as a negative number (two's complement) and adds this to the minuend.

Example:

Binary Subtraction	RCA 110	Decimal
111	$111 - 5_{10}$	7
-101	011 (Two's complement	-5
010	(1)010 Difference	2

Note: The final carry is ignored.

A-6 BINARY MULTIPLICATION

The rules for binary multiplication are:

- One X One = One
- Zero X One = Zero
- Zero X Zero = Zero

Example: Multiply 10010 by 01010 (18 by 10)

BINARY	DECIMAL
1 0 0 1 0	18
x 0 1 0 1 0	10
0 0 0 0 0	00
1 0 0 1 0	18
0 0 0 0 0	
1 0 0 1 0	
0 0 0 0 0	
0 1 0 1 1 0 1 0 0	180

The product is derived from a series of multiplying, shifting, and adding operations.

To evaluate the result

	BINARY	DECIMAL
Column Values	128 64 32 16 8 4 2 1	0 X 1 = 0
	1 0 1 1 0 1 0 0	0 X 2 = 0
		1 X 4 = 4
		0 X 8 = 0
		1 X 16 = 16
		1 X 32 = 32
		0 X 64 = 0
		1 X 128 = 128
		180

A-7 BINARY DIVISION

The rules for binary division are:

- One ÷ One = One $1 \div 1 = 1$
- Zero ÷ One = Zero $0 \div 1 = 0$
- Zero ÷ Zero = Zero $0 \div 0 = 0$

Example: Divide 1 1 1 1 by 1 0 1 (15 by 5)

BINARY

```
      0011
101 ) 1111
      101
      0101
      101
      0000
```

DECIMAL

```
    3
5 ) 15
```

Divide .1 0 1 by .0 0 1 (.625 ÷ .125)

BINARY

```
        101
^001 ) ^101.
         1
         001
         001
         000
```

DECIMAL

```
      5.
^125 ) ^625.
       625
```

A-8 OCTAL NUMBER SYSTEM

We have seen that the binary system is quite applicable to computer circuitry but awkward to use outside of the computer. Therefore, we could use some type of binary shorthand to simplify the handling of binary numbers.

In the binary system, we can count from zero to seven using just three bits:

Example:

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

This means that we can represent eight different symbols with three bits. Thus, one of eight symbols could be substituted for the corresponding group of bits.

These eight symbols could then be used to form an Octonary or Octal number system.

Examples:

BINARY									OCTAL		DECIMAL	
256	128	64	32	16	8	4	2	1	64	8	1	10	1
						1	1	0			6		6
				0	0	1	0	0	0		1	0	8
				0	0	1	1	0	1		1	5	1	3
				0	1	0	0	0	0		2	0	1	6
0		0	1	0	1	0	0	0	1	1	2	1	8	1

Thus, by dividing binary digits into groups of three, we can calculate the equivalent octal value.

A-9 NUMBER CONVERSIONS

Since computer operations involve the use of decimal, binary, and octal number systems, it is frequently necessary to convert a number from one number system (Radix) to another. This may usually be done by referring to number tables. However, the programmer may not have access to a table when conversions are required and will find it practical to be familiar with basic conversion techniques.

DECIMAL-TO-BINARY - Binary numbers are formed from successive powers of two. Thus, to change a decimal number to its equivalent binary form, begin by dividing the decimal number by two, and repeatedly dividing each quotient by two. The remainder at each step of division forms the binary digit.

Example: Convert decimal 157 to binary:

```
2 |157
  2 | 78 (remainder) = 1  least significant bit
  2 | 39             = 0
  2 | 19             = 1
    2 | 9            = 1
    2 | 4            = 1
    2 | 2            = 0
    2 | 1            = 0   ← read up
        0            = 1  most significant bit
      Decimal 157  = Binary 10 011 101
```

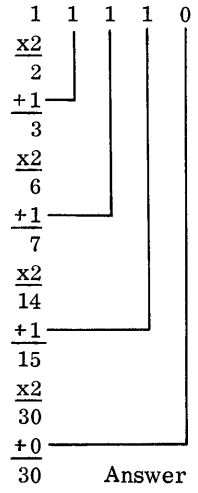
BINARY-TO-DECIMAL - The ascending powers of two are summed to obtain the decimal equivalent.

Example: Convert binary 11110 to decimal:

$$11110 = 1(2^4) + 1(2^3) + 1(2^2) + 1(2^1) + 0(2^0)$$
$$= 16 + 8 + 4 + 2 + 0 = 30$$

A second method is to multiply the most significant bit by 2, add the next most significant bit, multiply the resulting sum by 2, add the next most significant bit, and repeat the operation until all bits are used.

Example: Convert binary 11110 to decimal:



Binary 11110 = Decimal 30

DECIMAL-TO-OCTAL - To convert a decimal number to octal, divide the number repeatedly by 8 and write the resulting remainder. The number composed of the remainders will equal the equivalent octal number. The most significant digit in the octal number is the last remainder.

Example: Convert decimal 69 to octal:

$8 \overline{)69}$

$8 \overline{)8}$ (remainder) = 5 least significant digit

$8 \overline{)1}$ = 0 read up

1 = 1 most significant digit

Decimal 69 = Octal 105

OCTAL-TO-DECIMAL - Since octal numbers represent successive powers of 8, the ascending of powers of 8 are multiplied by their coefficient and summed to obtain the decimal quantity.

Example: Convert octal 126 to decimal:

$1(8^2) + 2(8^1) + 6(8^0) =$

$64 + 16 + 6 = 86$

Octal 126 = Decimal 86

A second method is to multiply the most significant digit by 8, add the next most significant digit, multiply the resulting sum by 8, and repeat this series of operations through the final addition.

Example: Convert octal 126 to decimal:

1 2 6

x8

8

+2

10

x8

80

+6

86 Answer

Octal 126 = Decimal 86

BINARY-TO-OCTAL - Since a binary number represents powers of 2 and an octal number represents powers of 8, thus, $2^3 = 8^1$. One octal digit can be used to represent three binary digits. This means that octal numbers can be used as a shorthand system for binary. This principle applies to fractional quantities as well as whole numbers.

A binary number is converted to octal by grouping bits in units of 3 and writing the equivalent octal number for each group.

Example:

110 010 100

6 2 4 = 624_8

and .111 011

.7 3 = $.73_8$

Conversely, any octal number can be converted to binary form by writing groups of 3 binary digits for each octal.

Example:

5 4 3

101 100 011

A-10 FRACTION CONVERSIONS

DECIMAL-TO-OCTAL - The decimal fraction is repeatedly multiplied by 8. Each time it is multiplied, the carry digit is placed on the left hand side to be used to form the corresponding octal number.

Example: Convert decimal $.157_{10}$ to an octal:

Read down

	.157
	x 8
1	.256
	x 8
2	.048
	x 8
0	.384
	x 8
3	.072

$= .1203_8 +$

Decimal .157 = Octal .1203

Example: Convert octal .1203 to a decimal

$.1203 = 1(8^{-1}) + 2(8^{-2}) + 0(8^{-3}) + 3(8^{-4})$

$= \frac{1}{8} + \frac{2}{64} + \frac{0}{512} + \frac{3}{4096}$

$= \frac{643}{4096}$

$= .1569$

Octal .1203 = Decimal .1569

BINARY-TO-DECIMAL

Example: Convert binary .101 to a decimal:

$.101 = 1(2^{-1}) + 0(2^{-2}) + 1(2^{-3})$

$.101 = 1\ \frac{1}{2} + 0\ \frac{1}{2^2} + 1\ \frac{1}{2^3}$

$= \frac{1}{2} + 0 + \frac{1}{8}$

$= .5 + .125$

$= .625$

Binary .101 = Decimal .625

DECIMAL-TO-BINARY - The same procedure for converting to octal fractions is followed for binary except the multiplier is 2.

Example: Convert decimal $.157_{10}$ to binary:

Read down →	.157
	x 2
0	.314
	x 2
0	.628
	x 2
1	.256
	x 2
0	.512
	x 2
1	.024

= .00101

Decimal .157 = Binary .00101

OCTAL-TO-BINARY - The same rules apply for fractions as those for the conversion of whole numbers.

Example: Octal .1203 to binary:

.1	2	0	3
.001	010	000	011

BINARY-TO-OCTAL

Example:

.001	010	000	011
.1	2	0	3

APPENDIX B POWERS OF TWO

2^{η}	η	$2^{-\eta}$
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	0.000 000 000 456 661 287 307 739 257 812 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	0.000 000 000 014 551 915 228 336 851 806 640 625
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 C OCTAL-DECIMAL INTEGER CONVERSION TABLE

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

	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
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
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

APPENDIX C OCTAL-DECIMAL INTEGER CONVERSION TABLE (Cont)

2000 to 2777 (Octal) 1024 to 1535 (Decimal)

	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
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

3000 to 3777 (Octal) 1536 to 2047 (Decimal)

	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	1595	1596	1597	1598
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
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	1921	1922	1923	1924	1925	1926
3610	1927	1928	1929	1930	1931	1932	1933	1934
3620	1935	1936	1937	1938	1939	1940	1941	1942
3630	1943	1944	1945	1946	1947	1948	1949	1950
3640	1951	1952	1953	1954	1955	1956	1957	1958
3650	1959	1960	1961	1962	1963	1964	1965	1966
3660	1967	1968	1969	1970	1971	1972	1973	1974
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

4000 to 4777 (Octal) 2048 to 2559 (Decimal)

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	0253	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	2304
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	2331
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
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

APPENDIX C OCTAL-DECIMAL INTEGER CONVERSION TABLE (Cont)

6000 to 6777 (Octal) 3072 to 3583 (Decimal)

	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
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	3475
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
7261	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
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 D OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042958	.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

APPENDIX D OCTAL-DECIMAL FRACTION CONVERSION TABLE (Cont)

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

APPENDIX D OCTAL-DECIMAL FRACTION CONVERSION TABLE (Cont)

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

APPENDIX E ADDITION AND MULTIPLICATION TABLES

BINARY

ADDITION

	0	1
0	0	1
1	1	10

MULTIPLICATION

	0	1
0	0	0
1	0	1

OCTAL

ADDITION

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

MULTIPLICATION

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

APPENDIX F STANDARD MEMORY LOCATIONS

HSM LOCATION (OCTAL)	DESCRIPTION	DATA STORAGE
0000	Drum Interrupt	Restart Location
0001	Index Register 1	Modification of Y
0002	Index Register 2	Modification of Y
0003	Index Register 3	Modification of Y
0004	Index Register 4	Modification of Y
0005	Index Register 5	Modification of Y
0006	Index Register 6	Modification of Y
0007	Index Register 7	Modification of Y
0010	Interrupt Level 1	Stores PC, bits 0-11 Stores JS, bits 15-22
0011	Interrupt Level 1	Stores L, bits 0-23
0012	Interrupt Level 1	Stores R, bits 0-23
0013	Overflow Location	Contains next instruction on overflow
0014	Interrupt Level 2	Stores PC, bits 0-11 Stores JS, bits 15-22
0015	Interrupt Level 2	Stores L, bits 0-23
0016	Interrupt Level 2	Stores R, bits 0-23
0017	Alarm Location	Address of next instruction on alarm
0020	Interrupt Level 3	Stores PC, bits 0-11 Stores JS, bits 15-22
0021	Interrupt Level 3	Stores L, bits 0-23
0022	Interrupt Level 3	Stores R, bits 0-23
0023	Overflow Location	Contains PC on overflow
0024	Interrupt Level 4	Stores PC, bits 0-11 Stores JS, bits 15-22
0025	Interrupt Level 4	Stores L, bits 0-23
0026	Interrupt Level 4	Stores R, bits 0-23
0027	Alarm Location	Contains PC on alarm
0030	Priority Interrupt 1	Program 1 First Instruction
0031	Priority Interrupt 2	Program 2 First Instruction
0032	Priority Interrupt 3	Program 3 First Instruction
0033	Priority Interrupt 4	Program 4 First Instruction
0034	Priority Interrupt 5	Program 5 First Instruction
0035	Priority Interrupt 6	Program 6 First Instruction
0036	Priority Interrupt 7	Program 7 First Instruction
0037	Priority Interrupt 8	Program 8 First Instruction
0040	Power Failure	Stores PC and JS
0041	Power Failure	Stores L Contents
0042	Power Failure	Stores R Contents

V BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	SSW1	SSW2	SSW3	SSW4	SSW5	SSW6					MTS REWIND	MTS LOAD POINT	MTS EOF		MTS OPERATOR ERROR	MTS PARITY ERROR	MTS INOPER-ABLE	MTS EOT	MTS BUFFER BUSY	PTR REWIND	IOR 0 BUSY	PTP OUT	PTP OUT	TW POWER OFF	
1	← ANALOG POINT 1 →												A/D BUSY	← ANALOG POINT 2 →											
2	← MS 24 BITS AIRBORNE COMPUTER INPUT REGISTER →																						LS 3 BITS ACIR		A_0 A_1
3	← EASTERN STANDARD TIME →																								A_0
	← COUNTDOWN CLOCK →																								A_1
	← TIMER →																								A_2
4	← 0-23 →																								A_0
	DISPLAY SENSE LINES																								
	← 24-47 →																								A_1
5																	INPUT REGISTER READY	ABC OUTPUT REG RCD	ABC INPUT REG RDY	REMOTE OUTPUT REG RCD	REMOTE INPUT REG RDY	PAYLOAD OUTPUT REG RCD	PAYLOAD INPUT REG RDY	TELM INPUT REG RDY	
6	← DISCRETE INPUTS →																								A_0 ⋮ A_{20} $1A_0$ ⋮ $1A_{20}$
7																									

V BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	PTR MODE III	TW COLOR RED	TW COLOR BLACK					MTS REWIND TO LOAD POINT & UNLOAD	MTS REWIND	MTS WRITE EOF	MTS SKIP FILE	MTS SKIP RECORD	MTS BACK-FILE	MTS BACK-SPACE RECORD	MTS WRITE	MTS READ	MTS BCD MODE	TW MODE II	PTP MODE II	PTR REWIND	PTR MODE II	PTP MODE I (OCTAL)	PTR MODE I	TW MODE I	
1				START ANALOG CONT LOGIC	← ANALOG ADDRESS 2 →														← ANALOG ADDRESS 1 →							
2		DISCRETE OUTPUT ENABLE	← INPUT/OUTPUT ADDRESS →																							
3	← MS 24 BITS OF AIRBORNE COMPUTER OUTPUT REGISTER →																				OUTPUT READY		LS 3 BITS ACOR		A_0 A_1
4												TELM INPUT REG RCD	PAYLOAD INPUT REG RCD	PAYLOAD OUTPUT REG RDY	TELM INPUT IOR STROBE	PAYLOAD INPUT IOR STROBE	24 BIT INPUT RCD	24 BIT OUTPUT REG RDY	ABC INPUT REG STROBE	ABC INPUT RCD	24 BIT INPUT IOR STROBE	IOR 7 INPUT STROBE	IOR 7 INPUT RCD	
5	STEER BIT D/A 2	STEER BIT D/A 1	← ANALOG OUTPUT 2 →											← ANALOG OUTPUT 1 →											
6																									
7																									

APPENDIX I IOR ASSIGNMENTS

BIT	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	←																							→
1	←																							→
2	←																							→
3	←																							→
4	←																							→
5	←																							→
6	←																							→
7																								

0 ← INPUT/OUTPUT (TYPEWRITER, MAGNETIC TAPE & PAPER TAPE EQUIPMENT) →

1 ← REMOTE OPERATIONS INPUT REGISTER →

2 ← COMMAND REGISTER →

3 ← REMOTE OPERATIONS OUTPUT REGISTER →

4 ← PAYLOAD OUTPUT REGISTER →

5 ← TELEMETRY INPUT REGISTER →

6 ← PAYLOAD INPUT REGISTER →

7

APPENDIX J RCA 110 COMPUTER OPERATION CODES (MNEMONIC)

MNEMONIC	OCTAL	DESCRIPTION	WORD TIMES	PAGE NO.
ACI	342	Activate Interrupt	2	2-11
ADD	100	Add	2	2-2
ADL	104	Add Long	3	2-3
ADR	101	Add and Replace	3	2-2
ADT	102	Add and Trap on Overflow	3	2-2
ALR	105	Add Long and Replace	5	2-3
ALT	106	Add Long and Trap on Overflow	4	2-3
ALW	107	Add Long, Replace, and Trap on Overflow	5	2-4
ART	103	Add, Replace, and Trap on Overflow	3	2-3
BGR	551	Binary to Gray and Replace	3	2-8
BTG	550	Binary to Gray	2	2-8
CBM	163	Complement Both on Minus	3	2-7
CLM	162	Complement L on Minus	2	2-7
CMB	161	Complement Both	3	2-7
CML	160	Complement L	2	2-7
DVD	130	Divide	30	2-6
DVT	132	Divide and Trap on Overflow	31	2-6
GBR	541	Gray to Binary and Replace	3	2-8
GTB	540	Gray to Binary	2	2-7
HLT	360	Halt	1	2-11
HTI	361	Halt Interruptable	1	2-11
INI	341	Inhibit Interrupt	2	2-11
LAN	140	Logical AND	2	2-6
LAR	141	Logical AND and Replace	3	2-6
LDA	410	Load Address	2	2-2
LDB	403	Load Both	3	2-1
LDL	401	Load L	2	2-1
LDN	62N	Load I/O Register N	2	2-13
LDR	402	Load R	2	2-1
LDZ	400	Load Zero into L	2	2-1
LEO	170	Logical Exclusive OR	2	2-7
LER	171	Logical Exclusive OR and Replace	3	2-7
LIO	150	Logical Inclusive OR	2	2-6
LIR	151	Logical Inclusive OR and Replace	3	2-7
MPY	120	Multiply	27	2-6
NOP	350	No Operation	2	2-11
NRM	520	Normalize	5	2-9
RAI	330	Return after Interrupt	4	2-10
RBA	500	Rotate Both Algebraic	4	2-8
RBL	517	Rotate Both Logical	4	2-9
RDR	441	Read Drum	*	2-12
RLL	515	Rotate L Logical	3	2-9
SBA	501	Shift Both Algebraic	4	2-8
SBL	513	Shift Both Logical	4	2-8
SDN	63N	Sense Device N	2	2-13
SLL	511	Shift L Logical	3	2-8
SLR	115	Subtract Long and Replace	4	2-5
SLT	116	Subtract Long and Trap on Overflow	4	2-5
SLW	117	Subtract Long, Replace, and Trap on Overflow	4	2-5
SRT	113	Subtract, Replace, and Trap on Overflow	3	2-4
STA	430	Store Address	3	2-2
STB	423	Store Both	4	2-2
STL	421	Store L	3	2-2
STN	61N	Store I/O Register N	3	2-12
STP	320	Store PC Complemented	3	2-1
STR	422	Store R	3	2-2
STZ	420	Store Zero into Y	3	2-2
SUB	110	Subtract	2	2-4
SUL	114	Subtract Long	2	2-5
SUN	60N	Set Up I/O Device	2	2-12

* Transfer Time = 8.32 + (K + n + 1) 0.128 milliseconds

APPENDIX J RCA 110 COMPUTER OPERATION CODES (MNEMONIC)

MNEMONIC	OCTAL	DESCRIPTION	WORD TIMES	PAGE NO.
SUR	111	Subtract and Replace	3	2-4
SUT	112	Subtract and Trap on Overflow	3	2-4
TNZ	353	Transfer on Negative or Zero	2	2-10
TPN	355	Transfer on Positive or Negative	2	2-10
TPZ	356	Transfer on Positive or Zero	2	2-10
TRA	357	Transfer Unconditional	2	2-10
TRN	351	Transfer on Negative	2	2-9
TRP	354	Transfer on Positive	2	2-10
TRZ	352	Transfer on Zero	2	2-10
TXD	31K	Transfer on XR Not Zero and Decrement	3	2-9
TXI	30K	Transfer on XR Not Zero and Increment	3	2-9
WDR	440	Write Drum	*	2-11

* Transfer Time = 8.32 + (K + n + 1) 0.128 milliseconds

Where

K = number of word times required to find specified starting address
n = number of words to be transferred

APPENDIX K RCA 110 COMPUTER OPERATION CODES (OCTAL)

OCTAL	MNEMONIC	DESCRIPTION	WORD TIMES	PAGE NO.
100	ADD	Add	2	2-2
101	ADR	Add and Replace	3	2-2
102	ADT	Add and Trap on Overflow	3	2-2
103	ART	Add, Replace, and Trap on Overflow	3	2-3
104	ADL	Add Long	3	2-3
105	ALR	Add Long and Replace	5	2-3
106	ALT	Add Long and Trap on Overflow	4	2-3
107	ALW	Add Long, Replace, and Trap on Overflow	5	2-4
110	SUB	Subtract	2	2-4
111	SUR	Subtract and Replace	3	2-4
112	SUT	Subtract and Trap on Overflow	3	2-4
113	SRT	Subtract, Replace, and Trap on Overflow	3	2-4
114	SUL	Subtract Long	3	2-5
115	SLR	Subtract Long and Replace	4	2-5
116	SLT	Subtract Long and Trap on Overflow	4	2-5
117	SLW	Subtract Long, Replace, and Trap on Overflow	4	2-5
120	MPY	Multiply	27	2-6
130	DVD	Divide	30	2-6
132	DVT	Divide and Trap on Overflow	31	2-6
140	LAN	Logical AND	2	2-6
141	LAR	Logical AND and Replace	3	2-6
150	LIO	Logical Inclusive OR	2	2-6
151	LIR	Logical Inclusive OR and Replace	3	2-7
160	CML	Complement L	2	2-7
161	CMB	Complement Both	3	2-7
162	CLM	Complement L on Minus	2	2-7
163	CBM	Complement Both on Minus	3	2-7
170	LEO	Logical Exclusive OR	2	2-7
171	LER	Logical Exclusive OR and Replace	3	2-7
30K	TXI	Transfer on XR Not Zero and Increment	3	2-9
31K	TXD	Transfer on XR Not Zero and Decrement	3	2-9
320	STP	Store PC Complemented	3	2-1
330	RAI	Return after Interrupt	4	2-10
341	INI	Inhibit Interrupt	2	2-11
342	ACI	Activate Interrupt	2	2-11
350	NOP	No Operation	2	2-11
351	TRN	Transfer on Negative	2	2-9
352	TRZ	Transfer on Zero	2	2-10
353	TNZ	Transfer on Negative or Zero	2	2-10
354	TRP	Transfer on Positive	2	2-10
355	TPN	Transfer on Positive or Negative	2	2-10
356	TPZ	Transfer on Positive or Zero	2	2-10
357	TRA	Unconditional Transfer	2	2-10
360	HLT	Halt	1	2-11
361	HTI	Halt Interruptable	1	2-11
400	LDZ	Load Zero into L	2	2-1
401	LDL	Load L	2	2-1
402	LDR	Load R	2	2-1
403	LDB	Load Both	3	2-1
410	LDA	Load Address	2	2-2
420	STZ	Store Zero into Y	3	2-2
421	STL	Store L	3	2-2
422	STR	Store R	3	2-2
423	STB	Store Both	4	2-2
430	STA	Store Address	3	2-2
440	WDR	Write Drum	*	2-11
441	RDR	Read Drum	*	2-12

* Transfer Time = 8.32 + (K + n + 1) 0.128 milliseconds

Where

K = number of drum word times required to find specified starting address
n = number of words to be transferred

APPENDIX K RCA 110 COMPUTER OPERATION CODES (OCTAL)

OCTAL	MNEMONIC	DESCRIPTION	WORD TIMES	PAGE NO.
500	RBA	Rotate Both Algebraic	4	2-8
501	SPA	Shift Both Algebraic	4	2-8
511	SLL	Shift L Logical	3	2-8
513	SBL	Shift Both Logical	4	2-8
515	RLL	Rotate L Logical	3	2-9
517	RBL	Rotate Both Logical	4	2-9
520	NRM	Normalize	5	2-9
540	GTB	Gray to Binary	2	2-7
541	GBR	Gray to Binary and Replace	3	2-7
550	BTG	Binary to Gray	2	2-8
551	BGR	Binary to Gray and Replace	3	2-8
60N	SUN	Set Up I/O Device	2	2-12
61N	STN	Store I/O Register N	3	2-12
62N	LDN	Load I/O Register N	2	2-13
63N	SDN	Sense Device N	3	2-13