RC 70 COMPUTER

REFERENCE MANUAL

RC RECOR

CORPORATION

RC 70 INSTRUCTIONS

Mnemonic	Name	Operation Code* Single Word	Operation Code* Double Word	Execution Time**	Reference Manual Page
ADB	Add	00	D820	1.9	3-2
ALB	Arithmetic Left	D2	DA3A	1.9-4.5	3-10
ANB	And	20	D824	1.9	3-2
ARB	Arithmetic Right	D3	DB3A	1.9-4.5	3-10
BBK	Branch Back	F0	D83E	3.5	3-11
BEQ	Branch Equal	80	D830	1.0	3-2
BGE	Branch Greater Than or Equal	90	D832	1.0	3-3
BGT	Branch Greater Than	B0	D836	1.0	3-3
BLE	Branch Less Than or Equal	88	D831	1.0	3-3
BLK	Branch and Link	E8	D83D	3.6	3-4
BLT	Branch Less Than	A8	D835	1.0	3-4
BNE	Branch Unequal	B1	D837	1.0	3-4
BNO	Branch No Overflow	A0	D834	1.0	3-5
BPT	Branch and Put	F8	D83F	5.5	3-5
BUC	Branch Unconditional	98	D833	1.0	3-5
BXB	Branch Index	E0	D83C	3.0	3-6
CMB	Compare	60	D82C	1.9	3-6
DVB	Divide	50	D82A	11.4	3-7
ELB	End Left	D0	D83A	1.9-7.0	3-10
ELD	Double Length Shift	CA	DA39	3.8-9.9	3-10
HLT	Halt	C9	D939	1.0	3-12
IOR	Inclusive Or	CC	DC39	2.1	3-7
LDB	Load	30	D826	1.9	3-7
LDP	Load Page	CE	DE39	2.1	3-12
LDX	Load Index	C0	D838	2.1	3-8
LRB	Logical Right	D1	D93A	1.9-4.5	3-11
MPB	Multiply	10	D822	6.2	3-8
PIP	Parallel Input	D6	DE3A	2.8	3-12
POP	Parallel Output	D7	DF3A	2.8	3-13
SBB	Subtract	40	D828	1.9	3-8
SET	Set	D4	DC3A	2.8	3-13
SNS	Sense	D5	DD3A	2.8	3-13
STB	Store	70	D82E	1.9	3-8
XMB	Exchange Memory and Accumulator	C8	D839	3.0	3-9
XOR	Exclusive Or	CD	DD39	2.1	3-9

* Operation codes are shown with X, I, R, and S bits equal to zero.

** Time is shown for single word instruction. Add 0.86 μs for double word instruction and 0.86 μs for indirect addressing.

RC 70 COMPUTER
REFERENCE MANUAL

March 1969

REDCOR CORPORATION
7800 Deering Avenue, P. O. Box 1031 Canoga Park, California 91304

CONTENTS

ILLUSTRATIONS (Cont.)

TABLES

RC 70 Computer System

SECTION I
GENERAL FEATURES

1.1 INTRODUCTION

The RC 70 Computer is a high-speed, general purpose, digital computer featuring third-generation, state-of-the-art design. Its outstanding price/performance characteristics coupled with its small size, reliability, and flexible input/output capability make it the ideal choice for a variety of systems applications including real-time data acquisition and process control. Figure 1-1 shows a block diagram of the RC 70 Computer.

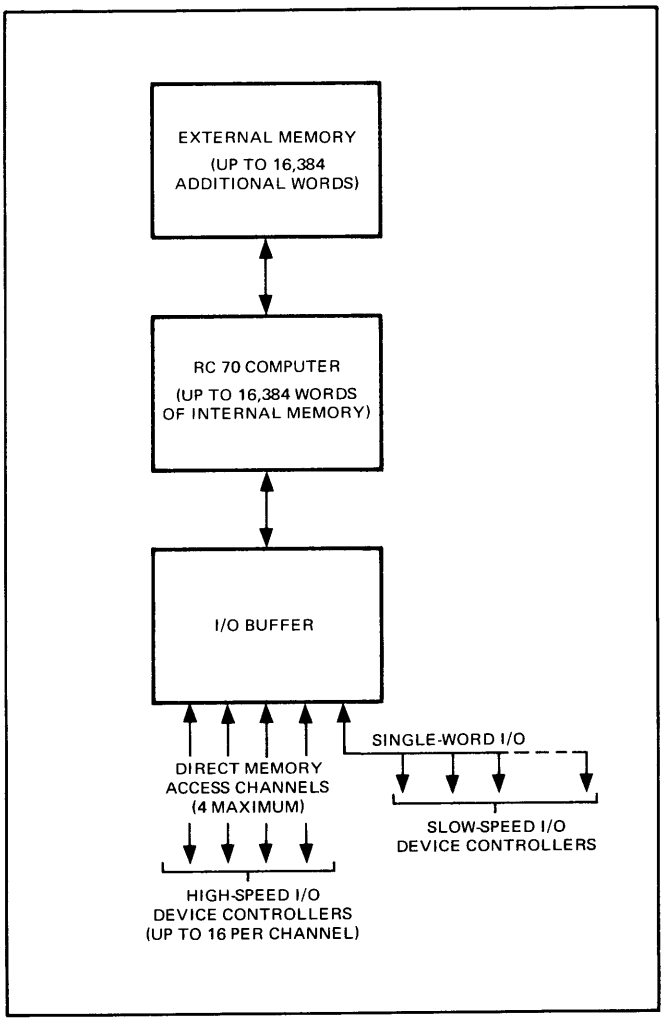


Figure 1-1. RC 70 Computer System Configuration

The following factors are only a few of those that contribute to the overall versatility, flexibility, and capability of the computer:

- Size. Including self-contained power supplies and 16,384 words of internal memory, the computer can fit in an area 19 inches wide, 19 inches high, and 19 inches deep. It requires no special mechanical or environmental facilities.
- Reliability. Maintenance requirements are minimized by the use of highly reliable transistor-transistor integrated logic.
- Memory Expansion. The basic 4096-word memory can be expanded in 4096-word increments up to 16,384 words of internal memory. An additional 16,384 words of external memory can be added in 4096-word increments for a total memory capacity of 32,768 words.
- Speed. Memory cycle time is only 860 nanoseconds.
- Multiple Addressing Modes. Most memory addressing instructions can specify direct addressing, indirect addressing, indexed addressing, and relative addressing (relative to page register or to next instruction address).
- Versatile Instruction Repertoire. All instructions can have either a single word or double word format. The instruction repertoire includes 5 load/store instructions, 8 arithmetic/logical instructions, 12 branch instructions, 5 shift instructions, and 5 input/output and control instructions.
- Flexible Input/Output Capability. Four input/output instructions provide for discrete control and sensing, and single word data transfers. Up to four optional block-transfer channels can be used with the direct memory access feature to control high-speed I/O devices without direct program intervention. Optional priority interrupt levels are available in groups of 8 up to a maximum of 32 levels. Priority interrupt levels feature group enabling and disabling, and individual arming and disarming.

1.2 HARDWARE FEATURES

Standard hardware features of the RC 70 include the following:

- 860-nanosecond memory cycle time
- 4096-word internal memory

- 16-bit memory word
- Memory parity
- Memory write protection
- Direct memory access
- External interrupt
- Hardware bootstrap loader
- Hardware index register for address modification with no increase in execution time
- Roll-arrow register for unlimited nesting of subroutines
- Eight sense switches
- 2-millisecond interrupt clock
- Operator's console
- Maintenance panel

1.3 SOFTWARE FEATURES

Supporting software for the RC 70 includes an assembler, a FORTRAN IV compiler, math subroutines, and a utility package. The RC 70 Assembler operates in a minimum 4096-word memory system. It is a single-pass symbolic assembler that can provide either absolute or relocatable object programs. The FORTRAN IV compiler is USASI standard. It is a single-pass compiler requiring an 8192-word memory configuration. Included in the Math Subroutine Library are routines for logarithmic, exponential, and trigonometric functions, and arithmetic routines for single- and double-precision calculations. The Utility Package provides programs for tape editing, debugging, program loading, and tape copying and verifying.

1.4 OPTIONAL FEATURES

Optional features available with the RC 70 include:

- Memory expansion in 4096-word increments up to a maximum of 32,768 words (16,384 words internal and 16,384 words external)
- High-speed hardware multiply (6.2 μs) and divide (11.4 μs)
- Power shutdown and restart feature for safely operating the computer from an unreliable power source
- Four block-transfer channels which operate through the direct memory access feature to control high-speed I/O devices without direct program intervention. Parallel data transfers of one word per memory cycle can be performed at a rate of up to 1.1 million words per second.
- Priority interrupts expandable in groups of eight up to a maximum of 32 levels. Priority interrupts can be group enabled and disabled, and individually armed and disarmed.
- Real-time clock

1.5 PERIPHERAL I/O EQUIPMENT

A complete line of standard peripheral I/O units is available for use with the computer. This equipment includes both paper and magnetic tape devices, card readers and punches, printers, keyboard devices, and mass storage devices.

1.6 GENERAL SPECIFICATIONS

Table 1-1 lists the general specifications for the computer. Table 1-2 describes the basic equipment model items.

Table 1-1. General Specifications

Specification	Characteristic
Memory	Random access magnetic core; 860 ns full cycle time; 16-bit word length plus parity bit; 4096 words standard, expandable to 32,768 words in blocks of 4096 words; write protection standard
Memory addressing	Direct, indirect, indexed, relative to page register, relative to next instruction address
Arithmetic	Parallel, binary, fixed point, two's complement
Instructions	Single word and double word; 5 load/store, 8 arithmetic/logical, 12 branch, 5 shift, 5 input/output and control

Table 1-1. General Specifications (Cont.)

Specification	Characteristic
Registers	Nine hardware, two memory; lower accumulator, 16-bit flip-flop; instruction register, 16-bit flip-flop; memory address register, 16-bit flip-flop; next-instruction-address register, 16-bit flip-flop; page register, 6-bit flip-flop; index register, 16-bit flip-flop; D-, G-, and J-registers, 16-bit flip-flop; roll-arrow register, memory location X'0007'; upper accumulator, memory location X'0008'
Input/Output	Discrete control and sense; single word to and from lower accumulator; direct memory access (cycle steal); external interrupt; four block-transfer channels through direct memory access (optional); priority interrupt up to 32 levels (optional)
Size	19 inches high, 19 inches wide, 19 inches deep
Weight	90 pounds, including power supply
Environment	0^{o} to 55^{o}C (32^{o} to 131^{o}F), 0 to 90% relative humidity
Power	115 (±10)V, 47 to 63 Hz, 400W

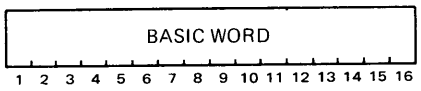
Table 1-2. Equipment Model Descriptions

Model No.	Description
RC 70	General purpose digital computer with 8,192 words of memory, memory parity, hardware multiply/divide, hardware index register, automatic power shutdown and restart, power supply, operator/maintenance panel, and ASR 33 Teletype and controller
70-00	General purpose digital computer with 4,096 words of memory, hardware index register, memory parity, power supply, and operator/maintenance panel
70-01	Hardware multiply/divide
70-02	Automatic power shutdown and restart
70-03	4096-word memory module with parity
70-10	One direct memory access block-transfer channel and eight priority interrupt levels
70-11	Two direct memory access block-transfer channels and 16 priority interrupt levels
70-18	Real-time clock (60 Hz and 1 kHz)

SECTION II
SYSTEM ORGANIZATION

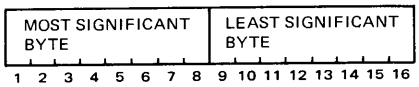
2.1 GENERAL

The RC 70 Computer uses as its basic informational element a 16-bit word with the bit positions numbered 1 through 16 as shown in the following diagram.

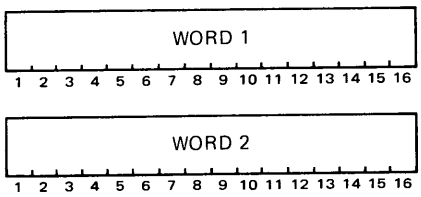


Bit position 1 represents the most significant portion of the word and bit position 16, the least significant. This basic informational scheme is reflected in the operational registers and the adder, which all have 16-bit capacities.

The basic word can be divided into two 8-bit parts, called bytes, with bits 1 through 8 representing the most significant byte and bits 9 through 16, the least significant, as shown in the following diagram.



Although each RC 70 instruction requires only one 16-bit word, the computer provides the added flexibility of using two 16-bit words, called a double word, for each instruction at the option of the programmer. The double word format is as follows:



Double word instructions require two memory locations for storage. They are always stored in consecutive memory locations with the most significant word (word 1) stored at the lower address. Double word instructions are always referenced by the address of their most significant word.

The basic 16-bit structure of the computer makes it convenient to use hexadecimal notation to express the binary information processed by the computer. Only four hexadecimal digits are needed to express a 16-bit binary number. The following table shows the hexadecimal and binary equivalents for decimal numbers 0 through 15:

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Throughout this manual, hexadecimal notation is represented by a string of hexadecimal digits enclosed in single quotation marks and preceded by the letter X. For example, the hexadecimal notation for the decimal number 79 is written as X'4F'. Additional information on hexadecimal notation, including conversion tables, is given in appendix A.

2.2 FUNCTIONAL ORGANIZATION

A functional block diagram of the computer showing the basic operational elements and data flow through these elements is illustrated in figure 2-1. Although the basic elements of the computer have overlapping functions, they can generally be grouped into four major functional sections: memory, control, arithmetic/logic, and input/output.

2.3 MEMORY

The basic memory of the computer consists of 4096 words of internal core storage. Additional core storage is optionally available in blocks of 4096 words each up to a maximum of 32,768 words. The computer is prewired to accept up to 16,384 words of storage without the need for additional cabinets or power supplies. Memory sizes exceeding 16,384 words require another cabinet to house the additional core storage modules.

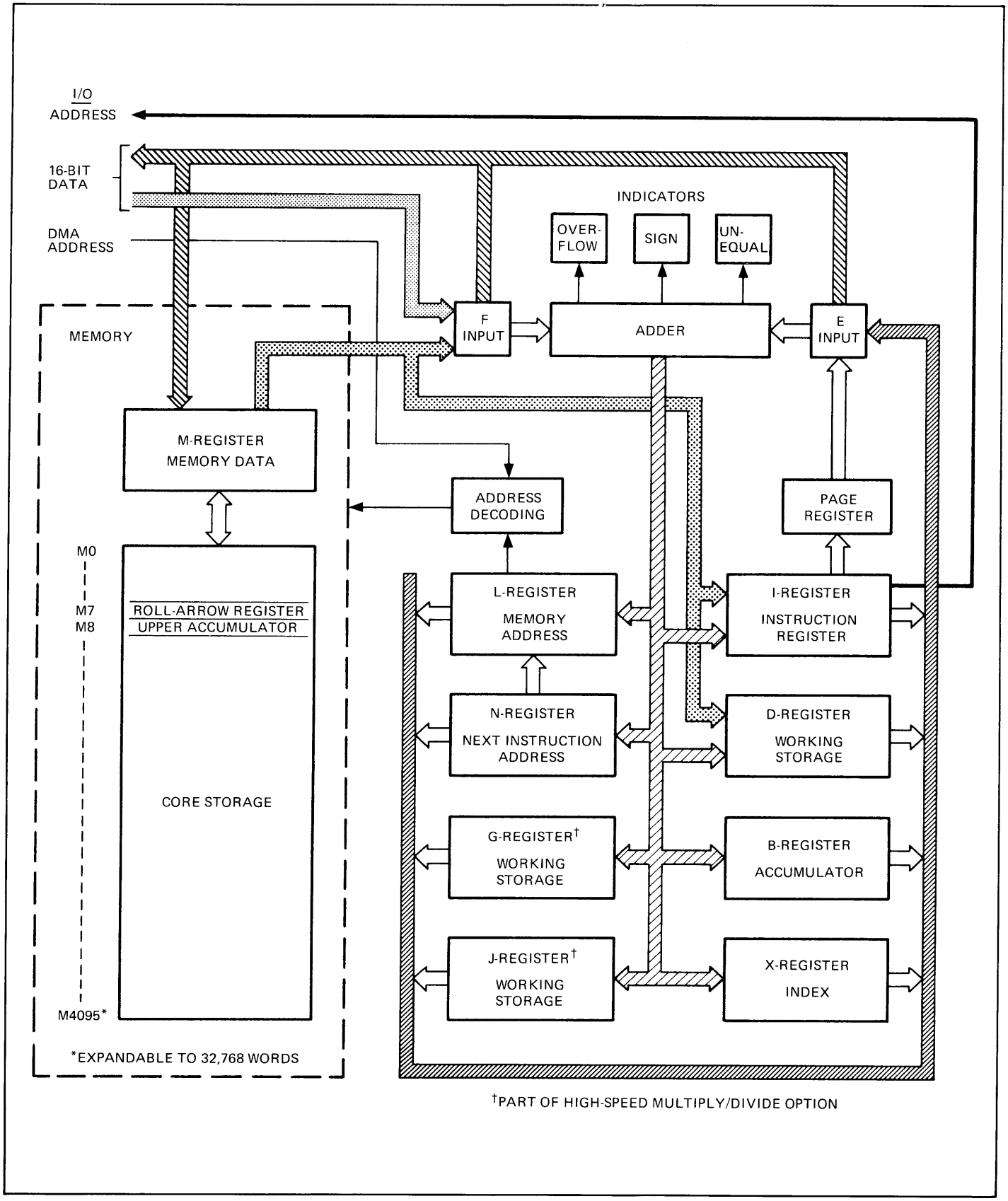
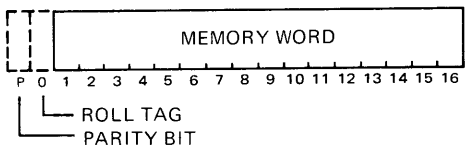


Figure 2-1. Basic Computer Organization

The computer memory word consists of 16 data bits (bit positions 1 through 16), plus a roll tag bit (bit position 0) and a parity bit.



The roll tag bit is meaningful only during the execution of a Branch Back instruction. Its use is explained in the discussion of the roll-arrow register (paragraph 2.12). The parity bit, on the other hand, is significant during each memory read and write cycle. Memory parity is checked each time a word is read from memory, and an indication of a parity error is given by the P indicator on the operator's console. The setting of the console PARITY HLT switch determines whether the computer halts or continues operation when a memory parity error is detected.

Every storage location in memory is directly addressable by the processor. The range of memory addresses extends from X'0000' (first location) to X'7FFF' (last location for maximum size memory of 32,768 words). Regardless of size, the memory is wraparound, or circular; for example, with a 4096-word memory the next location after location X'0FFF' is location X'0000'.

The memory protection feature of the computer makes it possible to protect the contents of memory in 2048-word blocks by inhibiting writing into the protected areas. Memory protection is controlled from the operator's console by the PROTECT MEM and BLOCK switches, which are described in paragraphs 5.16 and 5.17.

Certain storage locations in memory are reserved for special uses either by the processor or by standard software. The addresses of these locations and the purposes for which they are used are given in table 2-1.

As shown in figure 2-1, all data enters and leaves core storage through the memory data register (M-register). The address of the memory location into which data is stored, or from which data is fetched, is provided to the address decoding logic by either the memory address register (L-register) or the direct memory access address lines.

2.4 CONTROL

The orderly execution of a program stored in memory is primarily controlled by the contents of five registers:

a. The next-instruction-address register (N-register)

b. The instruction register (I-register)

Table 2-1. Reserved Memory Locations

Address		Use
Decimal	Hexadecimal	
0	0	Power shutdown storage of index register
1 thru 6	1 thru 6	Displayable on register display indicators (register select switch positions A1 through A6)
7	7	Roll-arrow register
8	8	Upper accumulator
9 thru 14	9 thru E	Temporary working storage
15	F	Basic interrupt address
16 thru 23	10 thru 17	Power shutdown storage
24 thru 31	18 thru 1F	Standard software
32 thru 38	20 thru 27	Communication between direct memory access channels and processor
39 thru 79	28 thru 4F	Storage of instruction string loaded by diode memory fill routine
16,320 thru 16,383	3FC0 thru 3FFF	Optional priority interrupt levels

c. The memory page register (P-register)

d. The index register (X-register)

e. The roll-arrow register (R-register)

The next-instruction-address register is the program sequencer, or counter. It supplies the memory address register with the addresses of the instruction words from which the computer operates. It is initially set to the address of the first instruction of the program when the computer is started. It is then automatically advanced during the execution of each instruction (except for Branch instructions) so that it holds the address of the next instruction in sequence. When a branch instruction is executed, the next-instruction-address register is loaded with the address specified by the branch instruction.

As instruction words are read from memory, they are automatically loaded into the instruction register. The unique codes assigned to each instruction are then decoded to provide the control signals required by the computer to execute the instruction. If the instruction is one that requires the computer to fetch an operand from memory, the address portion of the instruction is used to determine the operand address, which is then loaded into the memory address register for the memory access.

The page and index registers are used to modify the address portion of the instruction. Their operation is described in paragraphs 2.9 and 2.10.

The roll-arrow register is used by the Branch and Put, Branch and Link, and Branch Back instruction to provide unlimited nesting of subroutines. Its operation is described in paragraph 2.12.

2.5 ARITHMETIC/LOGIC

Arithmetic and logical operations in the computer are performed by the 16-bit adder and five accessory storage registers. Two of these registers, the lower accumulator (B-register) and the upper accumulator (UBA-register), can be manipulated by the programmer. The lower accumulator is the primary arithmetic register. Since only one word can be taken from memory or an input/output unit by each instruction, the second operand required for arithmetic or logical operations must be loaded into a register before the instruction is executed. The lower accumulator fulfills this function and also provides temporary storage for the result of the operation. The upper accumulator holds the most significant half of the product after a multiplication operation and holds the remainder after a division operation. In addition to these uses, the two accumulators are used during shifting operation.

The D-, G-, and J-registers are used for temporary working storage and are not under the control of the programmer.

Associated with the adder are three indicators that can be tested by the programmer after an adder operation. They are the overflow indicator (KO), the sign indicator (KS), and the unequal indicator (KU). Seven conditional branch instructions can be used to test the states of these indicators and alter program execution based on the results of the test.

2.6 INPUT/OUTPUT

The input/output section of the computer provides the capability for:

a. Discrete control and sensing of external I/O devices.

b. Single-word transfer (16-bit parallel) between the computer and external I/O devices under program control.

c. Single and multiple block transfers between external I/O devices and memory through the direct memory access feature under control of the optional block-transfer channels.

d. Control of input/output operations through the external interrupt with 32 levels of priority interrupt available as options.

Four input/output instruction provide complete control of all input/output operations: one for transmitting discrete control signals, one for testing discrete responses, and two for single-word transfers. Variations of these four instructions are used to control the optional block-transfer channels and priority interrupts.

2.7 PROGRAMMABLE REGISTERS

The computer contains five operational registers that can be directly manipulated by the programmer. Three of these are hardware registers; the other two are memory locations that are reserved for use as registers. The three programmable hardware registers are the lower accumulator, the index register, and the page register. The memory registers are the upper accumulator and the roll-arrow register.

2.8 LOWER ACCUMULATOR

The lower accumulator (B-register) is a 16-bit, flip-flop register. It is the primary arithmetic register of the computer. It provides one of the operands to the adder for all arithmetic and logical operations and also

stores the result. It contains the multiplicand at the beginning of a multiplication operation and contains the least significant half of the product at the end of the operation. It contains the divisor at the beginning of a division operation and contains the quotient at the end of the operation. It is the source register for a store operation, and the destination register for a load operation. The contents of the lower accumulator are affected by all shift operations specifying shifts of one or more places.

The lower accumulator is used during single-word input/output operations between the computer and peripheral devices. The contents of the lower accumulator are transferred to the I/O device during a parallel output operation. During a parallel input operation, the accumulator receives the data that is transferred into the computer from the I/O device.

2.9 INDEX REGISTER

The index register (X-register) is a 16-bit, flip-flop register that is used for address modification. The contents of the index register are affected by the Load Index and Branch Index instructions. Its contents are also affected by a Store or Exchange Memory and Accumulator instruction that has an effective address of X'0000'.

2.10 PAGE REGISTER

The page register is a six-bit, flip-flop register that is used during page-relative addressing. During the execution of an instruction using page-relative addressing, the contents of the page register are copied into bit positions 3 through 8 of the memory address register (figure 2-2), and the reference address specified by the instruction is copied into bit positions 9 through 16. The effect of using the page register to supply the most significant six bits of the memory address is that of dividing memory (up to 16,384 words) into 64 blocks, or pages, each containing 256 locations. The page register, then, specifies the memory page being addressed, and the reference address of the instruction specifies the address of the memory word within the page. The contents of the page register are affected by the Load Page instruction and by a Branch Back instruction performed after a Branch and Put.

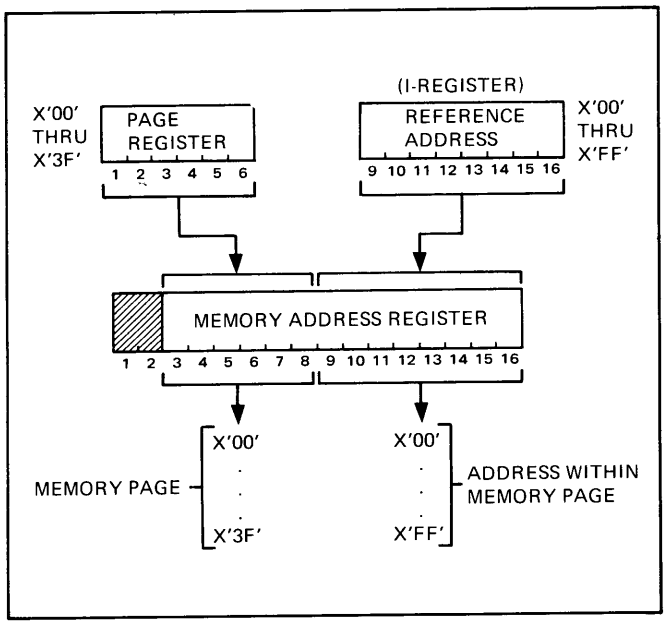


Figure 2-2. Page Relative Addressing

2.11 UPPER ACCUMULATOR

The upper accumulator (UBA-register) resides in memory location X'0008'. It is used as an extension of the lower accumulator for multiplication, division, and double-length shift operations. It contains the most significant half of the product at the end of a multiplication operation and the remainder at the end of a division operation. Its contents are affected by a Store or Exchange Memory and Accumulator instruction with an effective address of X'0008', by the Multiply and Divide instructions, and by the Double Length Shift instruction.

2.12 ROLL-ARROW REGISTER

The roll-arrow register (R-register) resides in memory location X'0007'. It is used during the execution of Branch and Link, Branch and Put, and Branch Back instructions to establish a subroutine linkage table (called a roll table) in memory and to keep track of the linkage information when the program branches through several subroutines (subroutine nesting).

At the start of a program the roll-arrow register is normally loaded with the starting address of the roll table. Thereafter, whenever a Branch and Link or Branch and Put instruction is executed, the information required to allow a branch back from the subroutine is automatically entered into the roll table. For a Branch and Link instruction the information stored in the roll table is the address of the instruction immediately following the Branch and Link instruction. For a Branch and Put instruction the information stored in the roll table includes the next instruction address plus the contents of the page and index registers and the states of indicators KO, KS, and KU. The address in the roll-arrow register is automatically increased as each Branch and Link or Branch and Put instruction is executed to keep track of the linkage and status information as subroutines are nested. The address is automatically decreased as Branch Back instructions are executed to return the program through the nested subroutines.

A typical example showing three levels of subroutine nesting and the corresponding roll table entries is illustrated in figure 2-3. When the Branch and Put instruction in the main program is executed, the next instruction address (X'0201') is entered into the roll

table at the location specified by the roll-arrow register (location n), and the address in the roll-arrow register is increased by one (n+1). Then, the states of indicators KO, KS, and KU, and the contents of the page register are entered in the roll table at location n+1. Again, the contents of the roll-arrow register are increased by one (n+2). Finally, the contents of the index register are entered in the roll table (location n+2), the address in the roll-arrow register is increased by one (n+3), and the program branches to subroutine No. 1.

When the Branch and Link instruction in subroutine No. 1 is executed, the next instruction address (X'0351') is entered into the roll table at location n+3, the address in the roll-arrow register is increased by one (n+4), and the program branches to subroutine No. 2.

When the Branch and Put instruction in subroutine No. 2 is executed, linkage and status information is entered into the roll table at location n+4, n+5, and n+6; the address in the roll-arrow register is increased to n+7, and the program branches to subroutine No. 3.

When the Branch Back instruction at the end of subroutine No. 3 is executed, the address in the roll-arrow register is decreased by one (n+6). The last three entries are fetched from the roll table to restore the conditions existing when the Branch and Put instruction in subroutine No. 2 was executed. Program execution resumes with the instruction stored at location X'0451'.

The Branch Back instructions at the end of subroutines No. 2 and No. 1 eventually return program control to

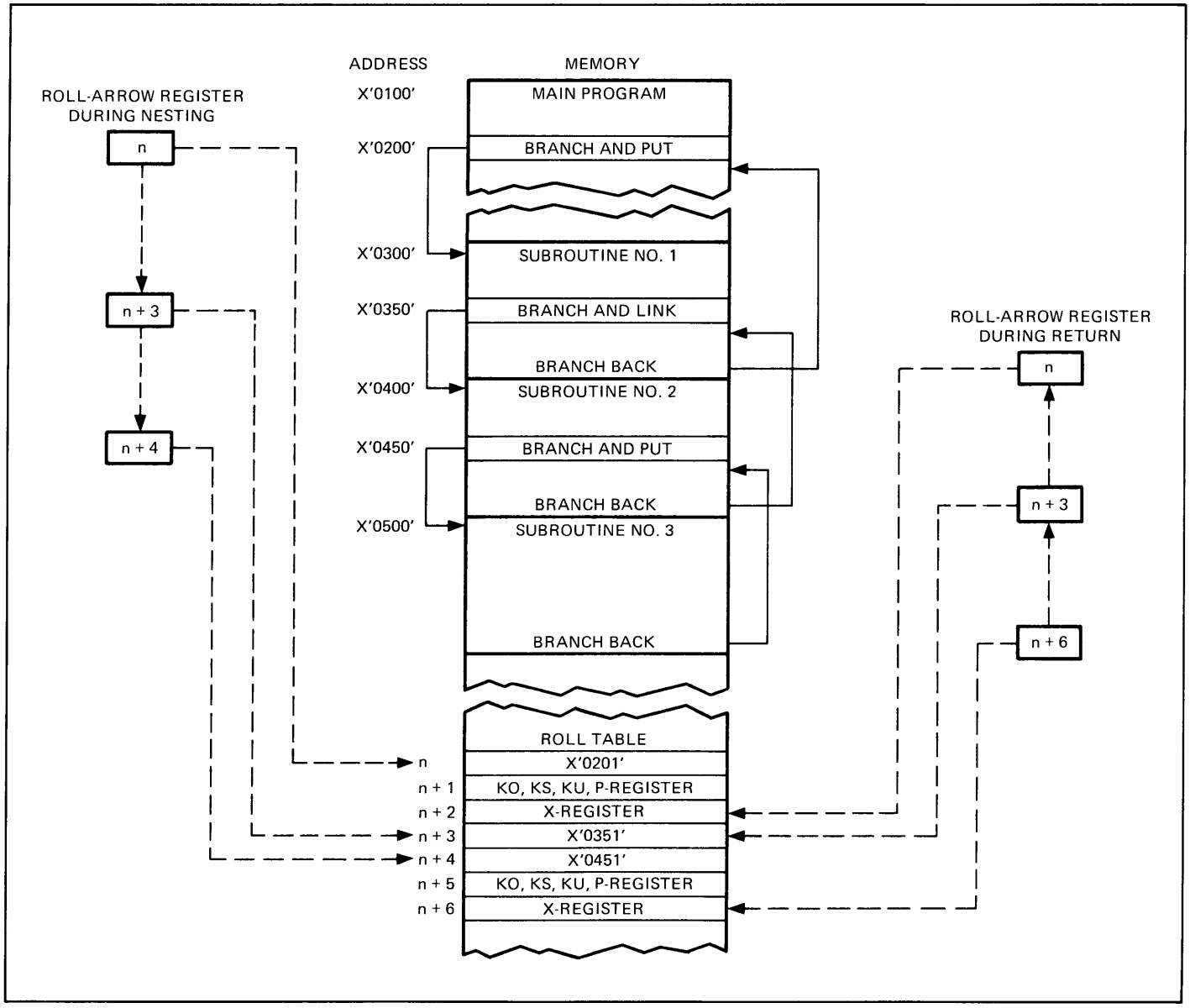


Figure 2-3. Example of Roll-Arrow Register Operation

the main program at location X'0201' after working back through the roll table.

2.13 NONPROGRAMMABLE REGISTERS

The computer contains six hardware registers that cannot be directly manipulated by the programmer. Three of these — the next-instruction-address register, the memory address register, and the instruction register — directly control the automatic execution of the stored program. The other three — the D-, G-, and J-registers — are used for working storage.

2.14 NEXT-INSTRUCTION-ADDRESS REGISTER

The next-instruction-address register (N-register) is a 16-bit, flip-flop register that normally contains the address of the next instruction to be executed. The contents of this register are increased by one at the start of each instruction. At the end of the execution cycle of each instruction, the contents of the N-register are transferred to the memory address register so that the next instruction can be fetched from memory.

2.15 MEMORY ADDRESS REGISTER

The memory address register (L-register) is a 16-bit, flip-flop register that contains the address of the memory location into which instruction or data words are stored, or from which they are fetched. The starting address of a program can be entered into the L-register by setting the data switches on the operator's console to the starting address and then pressing the RESET switch.

2.16 INSTRUCTION REGISTER

The instruction register (I-register) is a 16-bit, flip-flop register that contains the instruction currently being executed. This register automatically receives the instruction word as it is fetched from memory. The instruction is then decoded to provide the control signals required by the computer to execute the instruction and perform the specified operation.

2.17 WORKING REGISTERS

The working registers (D-, G-, and J-registers) are all 16-bit, flip-flop registers. They are used for temporary storage and as working buffers.

2.18 INDICATORS

Three single-bit indicators — Overflow indicator KO, Sign indicator KS, and Unequal indicator KU — can be tested by the programmer to determine the result of an operation involving the adder. Generally, the behavior of these indicators is as follows:

a. Overflow indicator KO is set when the result of an operation exceeds the capacity of the lower accumulator.

b. Sign indicator KS is set when the sign of the result of an operation is negative, and is reset when the sign is positive.

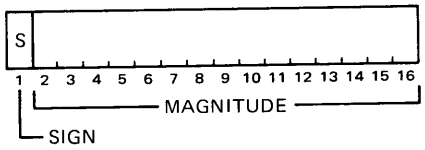
c. Unequal indicator KU is set when the result of an operation is not equal to zero.

Exceptions to this behavior are noted in the instruction descriptions given in section III.

2.19 WORD FORMATS

2.20 DATA WORDS

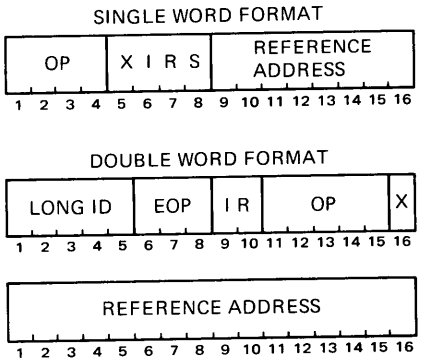
Data is stored in the computer in 16-bit words with the following format:



Bits 2 through 16 of the data word contain the magnitude of the data in two's complement form. Bit 1 specifies the sign (polarity) of the data. A sign bit of 1 represents a negative number; a sign bit of 0, a positive number.

2.21 INSTRUCTION WORDS

Most of the computer instructions are the memory addressing type, which have the following single and double word formats:



In the single word format, the operation code (OP) occupies the four most significant bits and defines the operation to be performed such as add, subtract, multiply, divide. Bits 5 through 8 contain four address mode bits designated X, I, R, and S. These bits determine the addressing mode as described in paragraph 2.22. Bits 9 through 16 contain the reference address which, along with the address mode bits, is used in deriving the effective address of the instruction.

In the double word format, bits 1 through 5 of the first word contain the identification code for a double word instruction. The extended operation code (EOP) and operation code (OP) fields specify the operation to be performed, and the X, I, and R bits control the addressing mode. The second word contains a 16-bit reference address that can be modified as specified by the address mode bits to derive the effective address.

The shift, control, and input/output instructions have slightly different formats from the memory addressing instructions, but like the memory addressing instructions they can have either a single or double word format.

2.22 MEMORY ADDRESSING

Four modes of memory addressing can be specified by memory addressing instructions of the computer:

a. Direct addressing

b. Indirect addressing

c. Indexed addressing

d. Relative addressing

Direct addressing can be performed by both single and double word instructions. Single word instructions can directly address 256 locations of memory storage. Double word instructions can directly address all memory storage locations (32,768 locations).

Both single and double word instructions can indirectly address all memory storage locations.

Both single and double word instructions can use the index register to perform indexed addressing. When both indirect and indexed addressing are specified for the same instruction, indirect addressing is performed first, and then indexing is performed. This is known as postindexing.

Two versions of relative addressing can be performed by single word instructions, but only one version can be performed by double word instructions. Single word instructions can specify addressing relative to the page register (up to 256 locations within each page) or relative to the next instruction address (256 locations forward or backward). Double word instructions can specify addressing relative to the next instruction address, but not relative to the page register.

2.23 EFFECTIVE ADDRESS COMPUTATION

The effective address for an instruction is the final 16-bit address value developed for that instruction, starting with the reference address in the instruction itself and modifying it as specified by the address mode bits. For single word instructions, the effective address is developed as follows:

1. Perform relative addressing.

   a. If bits R and S are both 0's, the relative address is equal to the reference address of the instruction itself.

   b. If bit R is 0 and bit S is 1, the relative address is obtained by combining the contents of the page register and the reference address of the instruction (page-relative addressing).

   c. If bit R is 1 and bit S is 0, the relative address is obtained by adding the contents of the next-instruction-address register to the reference address of the instruction (forward addressing relative to next instruction address).

   d. If bit R is 1 and bit S is 1, the relative address is obtained by subtracting the reference address from the contents of the next-instruction-address register (backward addressing relative to next instruction address).

2. Perform indirect addressing.

   a. If bit I is 0, the direct address is equal to the relative address obtained in step 1.

   b. If bit I is 1, the direct address is the 16-bit value stored at the location specified by the relative address obtained in step 1.

3. Perform indexing.

   a. If bit X is 0, the effective address is the address obtained in step 2.

   b. If bit X is 1, the effective address is obtained by adding the direct address from step 2 to the contents of the index register.

The effective address computation for double word instructions is similar to that for single word instructions except that bit S is not used; therefore, steps 1b and 1d are not valid.

Table 2-2 summarizes effective address computation for both single and double word memory addressing instructions. The symbols used in the table are defined as follows:

X, I, R, S	Address mode control bits
M	Reference address specified by current instruction

(M) Contents of location specified by reference address

(N) Contents of next-instruction-address register

(P) Contents of page register

(X) Contents of index register

2.24 PROTECTION FEATURES

The protective features of the computer include the standard memory protection feature and the optional power shutdown and restart feature. The memory protection feature protects the contents of selected memory locations from inadvertent destruction by inhibiting writing into these locations. The power shutdown and restart feature guarantees the integrity of system operation when the computer is operated from an unreliable power source.

2.25 MEMORY PROTECTION

The PROTECT MEM and BLOCK switches on the operator's console are used to set up protected areas of memory. The use of these controls and the areas of memory that can be selected for protection are described in paragraphs 5.16 and 5.17.

2.26 POWER SHUTDOWN AND RESTART OPTION

When input power to the computer falls below 100V, the power shutdown and restart feature initiates a shutdown sequence causing the contents of all volatile registers to be stored in reserved memory locations. Further

Table 2-2. Effective Address Computation

Single Word / Double Word				Effective Address		Additional Timing*
X	I	R	S			
0	0	0	0	M		-
0	0	0	1	M + (P)	Single word only	-
0	0	1	0	M + (N)		-
0	0	1	1	-M + (N)	Single word only	-
0	1	0	0	(M)		0.86 μs
0	1	0	1	(M + (P))	Single word only	0.86 μs
0	1	1	0	(M + (N))		0.86 μs
0	1	1	1	(-M + (N))	Single word only	0.86 μs
1	0	0	0	M + (X)		-
1	0	0	1	M + (P) + (X)	Single word only	-
1	0	1	0	M + (N) + (X)		0.86 μs
1	0	1	1	-M + (N) + (X)	Single word only	0.86 μs
1	1	0	0	(M) + (X)		0.86 μs
1	1	0	1	(M) + (P)) + (X)	Single word only	0.86 μs
1	1	1	0	(M + (N)) + (X)		0.86 μs
1	1	1	1	(-M + (N)) + (X)	Single word only	0.86 μs

*Add 0.86 μs for all double word instructions

memory access is inhibited and program operation is halted. When input power returns to normal, the operational registers are restored to the conditions existing before the power interruption, and program operation is resumed. The memory locations reserved for shutdown storage and the contents of these locations after shutdown are shown in figure 2-4.

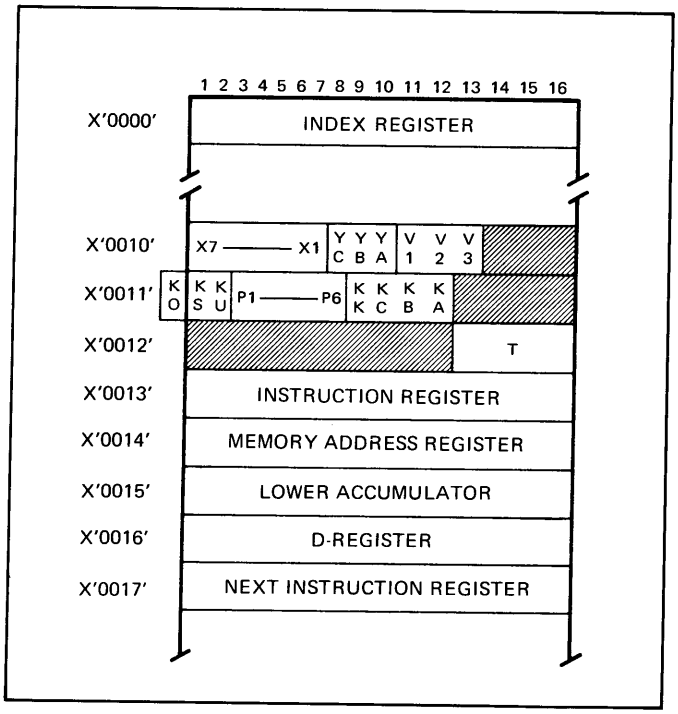


Figure 2-4. Shutdown Storage

SECTION III
COMPUTER INSTRUCTIONS

3.1 GENERAL

This section contains functional descriptions of all the computer instructions arranged in the following order:

a. Memory addressing instructions

b. Shift instructions

c. Input/output and control instructions

Each instruction is described in the following format:

MNEMONIC INSTRUCTION NAME TIMING

FORMAT DIAGRAMS

OPERATION STATEMENT

AFFECTED

DESCRIPTION

MNEMONIC. The mnemonic is a three-letter abbreviation recognized and used by the assembler to produce the instruction's unique operation code.

INSTRUCTION NAME. The instruction name is the descriptive title of the instruction.

TIMING. The execution time for the single word version of the instruction is given in microseconds. The execution times for all shift instructions depend on the number of shifts; therefore, minimum and maximum execution times are given for these instructions. When indirect addressing is specified or when the double word version of the instruction is specified, the execution time is increased by 0.86 microseconds.

FORMAT DIAGRAMS. Format diagrams are given for both the single and double word versions of the instruction. Both hexadecimal and binary notation are used to indicate those fields of the instruction in which the contents are fixed. Symbols or names are used for those fields in which the contents are variable. Shaded fields are ignored by the computer, but should be coded with zeros to avoid conflict in the event of future use. The symbols used on the format diagrams are defined as follows:

X	Index tag
I	Indirect address tag
R,S	Relative address tags

OPERATION STATEMENT. The operation statement states the instruction's operation in abbreviated form or symbolic notation. The symbols used in the statement are defined as follows:

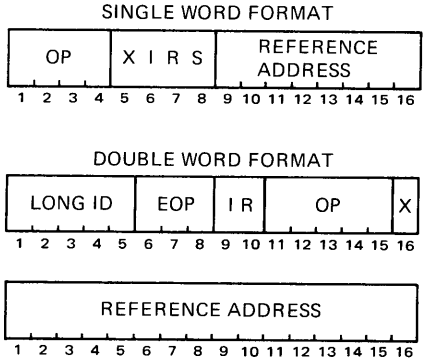
B	Lower accumulator
EA	Effective address
KO	Overflow indicator
KS	Sign indicator
KU	Unequal indicator
M	Reference address field of current instruction
N	Next-instruction-address register
NS	Number of shifts
P	Page register
R	Roll-arrow register
UBA	Upper accumulator
X	Index register
( )	Contents of register, memory location, or field indicated
+	Addition
-	Subtraction
∩	Logical product (AND)
∪	Logical sum (OR)
Ⓤ	Logical difference (exclusive OR)
⟶	Replaces the contents of

AFFECTED. All programmable registers, memory locations, and indicators whose contents can be changed by the instruction are listed.

DESCRIPTION. The description explains the operations performed by the computer as it executes the instruction.

3.2 MEMORY ADDRESSING INSTRUCTIONS

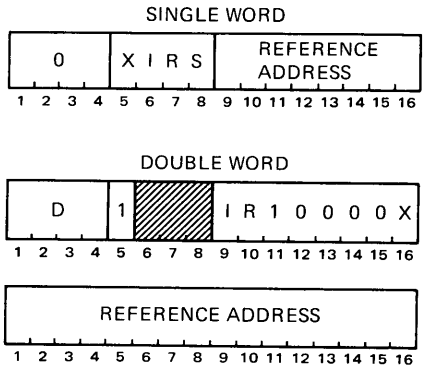
Memory addressing instructions have the following basic single and double word formats:



Three of the instructions – Exclusive Or, Inclusive Or, and Exchange Memory and Accumulator – have slightly different formats: they do not use address mode bits X, I, R, and S. The branch and Load Index instructions also deviate slightly since they are nonindexable instructions. For these instructions, bit 5 of the single word format and bit 16 of the double word format are not regarded as index tags by the computer.

The way in which the processor uses the reference address field and the address mode control bits of the memory addressing instructions to derive the effective address is described in paragraphs 2.22 and 2.23.

ADB ADD **1.9 μs**



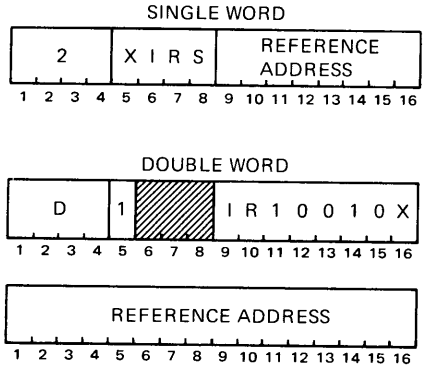
Operation Statement: (B) + (EA) ⟶ (B)

Affected: (B), KO, KS, KU

The Add instruction adds the contents of the effective address to the contents of the lower accumulator and loads the result into the accumulator. Overflow indicator KO is reset before the operation and is then set if the sum exceeds the capacity of the accumulator; Sign indicator KS is set if the sign of the result is negative; and Unequal indicator KU is set if the sum is not equal to zero.

If this instruction is executed with an effective address of X'0000', the contents of the index register are added to the contents of the lower accumulator.

ANB AND **1.9 μs**



Operation Statement: (B) ∩ (EA) ⟶ (B)

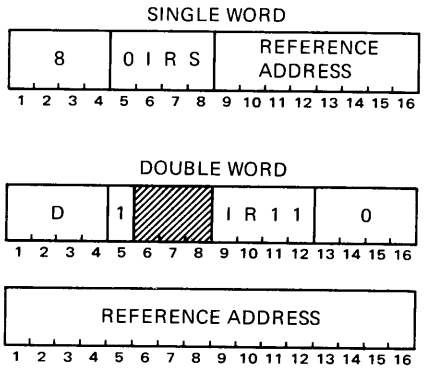
Affected: (B), KO, KS, KU

The And instruction forms the logical product of the contents of the effective address and the contents of the lower accumulator, and loads this product into the accumulator. The logical product contains a 1 in each bit position for which there is a corresponding 1 in both the accumulator and the effective word, and contains a 0 in each bit position for which there is a 0 in the corresponding bit position of either operand.

Overflow indicator KO is reset; Sign indicator KS is set if the sign of the result is negative; and Unequal indicator KU is set if the result is not equal to zero.

Executing this instruction with an effective address of X'0000' forms the logical product of the index register and the accumulator and loads this product into the accumulator.

BEQ BRANCH EQUAL **1.0 μs**

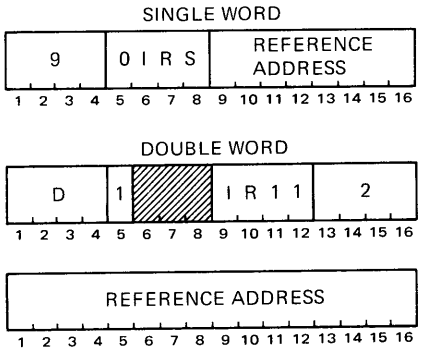


Operation Statement: Branch if tested result equals zero

Affected: None

The Branch Equal instruction causes the program to branch to the effective address if the result of the last instruction tested equals zero (Unequal indicator KU reset). The state of Sign indicator KS is ignored. If the tested result does not equal zero (KU set), program execution continues with the next instruction in sequence. This instruction cannot specify indexed addressing.

BGE BRANCH GREATER THAN OR EQUAL **1.0 μs**



Operation Statement: Branch if tested result is greater than or equal to zero

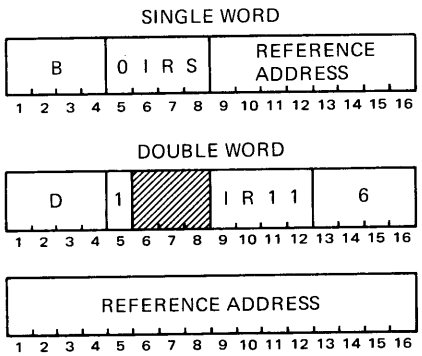
Affected: None

The Branch Greater Than or Equal instruction causes the program to branch to the effective address if the result of the last instruction tested is either greater than or equal to zero according to the states of Sign indicator KS and Unequal indicator KU. If the tested result is less than zero, program execution continues with the next instruction in sequence.

Indicators KS	Indicators KU	Resulting Action
-	0	Branch. Tested result is equal to zero
0	1	Branch. Tested result is greater than zero
1	1	No branch. Tested result is less than zero

This instruction cannot specify indexed addressing.

BGT BRANCH GREATER THAN **1.0 μs**



Operation Statement: Branch if tested result is greater than zero

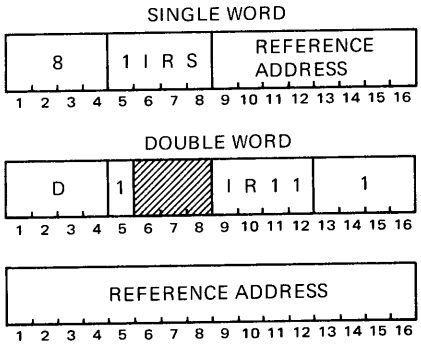
Affected: None

The Branch Greater Than instruction causes the program to branch to the effective address if the result of the last instruction tested is greater than zero according to the states of Sign indicator KS and Unequal indicator KU. If the tested result is equal to or less than zero, program execution continues with the next instruction in sequence.

Indicators KS	Indicators KU	Resulting Action
-	0	No branch. Tested result is equal to zero
0	1	Branch. Tested result is greater than zero
1	1	No branch. Tested result is less than zero

This instruction cannot specify indexed addressing.

BLE BRANCH LESS THAN OR EQUAL **1.0 μs**



Operation Statement: Branch if tested result is less than or equal to zero

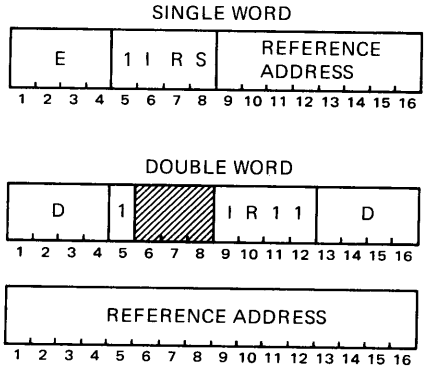
Affected: None

The Branch Less Than or Equal instruction causes the program to branch to the effective address if the result of the last instruction tested is less than or equal to zero according to the states of Sign indicator KS and Unequal indicator KU. If the tested result is greater than zero, program execution continues with the next instruction in sequence.

Indicators KS	Indicators KU	Resulting Action
-	0	Branch. Tested result is equal to zero
0	1	No branch. Tested result is greater than zero
1	1	Branch. Tested result is less than zero

This instruction cannot specify indexed addressing.

BLK BRANCH AND LINK **3.6 μs**



Operation Statement: $(N) \longrightarrow ((R))$

$1 \longrightarrow ((R)_0)$

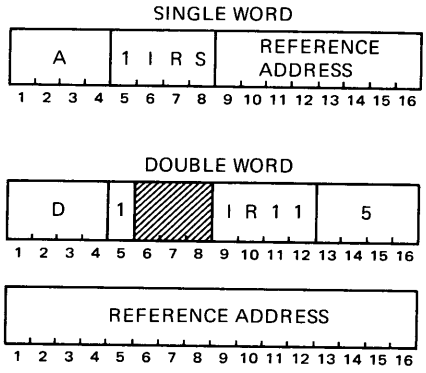
$(R) + 1 \longrightarrow (R)$

Affected: (R)

The Branch and Link instruction enters the next instruction address into the memory roll table at the location specified by the address in the roll-arrow register. The address in the roll-arrow register is increased by one, and then the program branches to the effective address. Bit 0 of the roll table location in which the return link address is stored is set to 1 to indicate that a Branch and Link instruction has been performed.

The link address stored by the Branch and Link instruction permits the program to return to the next instruction when a Branch Back instruction is executed. This instruction cannot specify indexed addressing.

BLT BRANCH LESS THAN **1.0 μs**



Operation Statement: Branch if tested result is less than zero

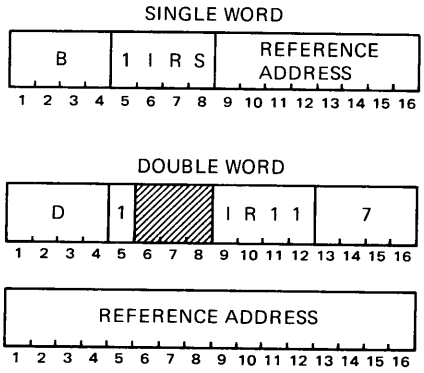
Affected: None

The Branch Less Than instruction causes the program to branch to the effective address if the result of the last instruction tested is less than zero according to the states of Sign indicator KS and Unequal indicator KU. If the tested result is equal to or greater than zero, program execution continues with the next instruction in sequence.

Indicators KS	KU	Resulting Action
-	0	No branch. Tested result is equal to zero
0	1	No branch. Tested result is greater than zero
1	1	Branch. Tested result is less than zero

This instruction cannot specify indexed addressing.

BNE BRANCH UNEQUAL **1.0 μs**



Operation Statement: Branch if tested result is not equal to zero

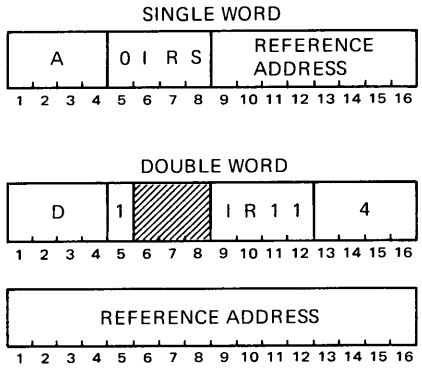
Affected: None

The Branch Unequal instruction causes the program to branch to the effective address if the result of the last instruction tested is not equal to zero according to the states of Sign indicator KS and Unequal indicator KU. If the tested result is equal to zero, program execution continues with the next instruction in sequence.

Indicators KS	KU	Resulting Action
-	0	No branch. Tested result is equal to zero
0	1	Branch. Tested result is greater than zero
1	1	Branch. Tested result is less than zero

This instruction cannot specify indexed addressing.

BNO BRANCH NO OVERFLOW **1.0 μs**



Operation Statement: Branch if tested result has not caused overflow

Affected: None

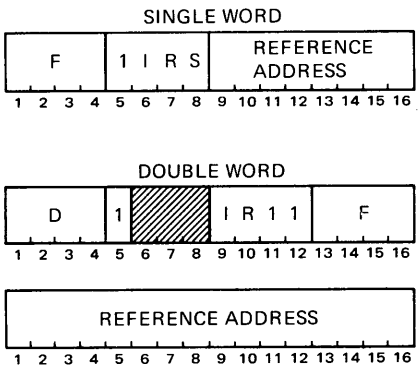
The Branch No Overflow instruction causes the program to branch to the effective address if the last instruction tested does not cause an overflow condition (Overflow indicator KO reset). If the tested result caused an overflow condition (KO set), program execution continues with the next instruction in sequence. Overflow can be caused by any of the following:

a. An addition or subtraction in which the sum or difference exceeds the capacity of the lower accumulator

b. A division in which the contents of the lower accumulator are not less than the magnitude of the divisor

This instruction cannot specify indexed addressing.

BPT BRANCH AND PUT **5.5 μs**



Operation Statement: $(N) \longrightarrow ((R))$

$(KO, KS, KU, P) \longrightarrow ((R) + 1)$

$(X) \longrightarrow ((R) + 2)$

$0 \longrightarrow ((R) + 2)_0$

Affected: (R)

The Branch and Put instruction stores current program status information and a return link address in the memory roll table at the locations specified by roll-arrow register, and then branches to the effective address. The program status stored by this instruction includes the states of indicators KO, KS, and KU; the contents of the page register; and the contents of the index register. The return link address is the address of the next instruction in sequence following the Branch and Put instruction.

Execution of the instruction proceeds as follows (see figure 3-1):

a. The address of the next instruction is entered in the roll table at the location specified by the address in the roll-arrow register. The address in the roll-arrow register is then increased by one.

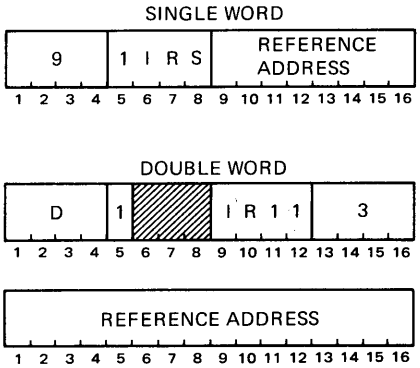
b. The states of indicators KO, KS, and KU, and the contents of the page register are entered into bit positions 0 through 8 of the next roll table location. Again the address in the roll-arrow register is increased by one.

c. The contents of the index register are entered in the roll table, and bit 0 of this location is set to 0 to indicate that these roll table entries were stored by a Branch and Put instruction.

d. The address in the roll-arrow register is increased by one in preparation for the next Branch and Put or Branch and Link instruction. The program then branches to the effective address.

The link address and program status stored by the Branch and Put instruction permits the program to branch to a subroutine and then return to the next instruction in sequence without destroying the contents of the indicators and page and index registers. This instruction cannot specify indexed addressing.

BUC BRANCH UNCONDITIONAL **1.0 μs**



Operation Statement: Branch to effective address

Affected: None

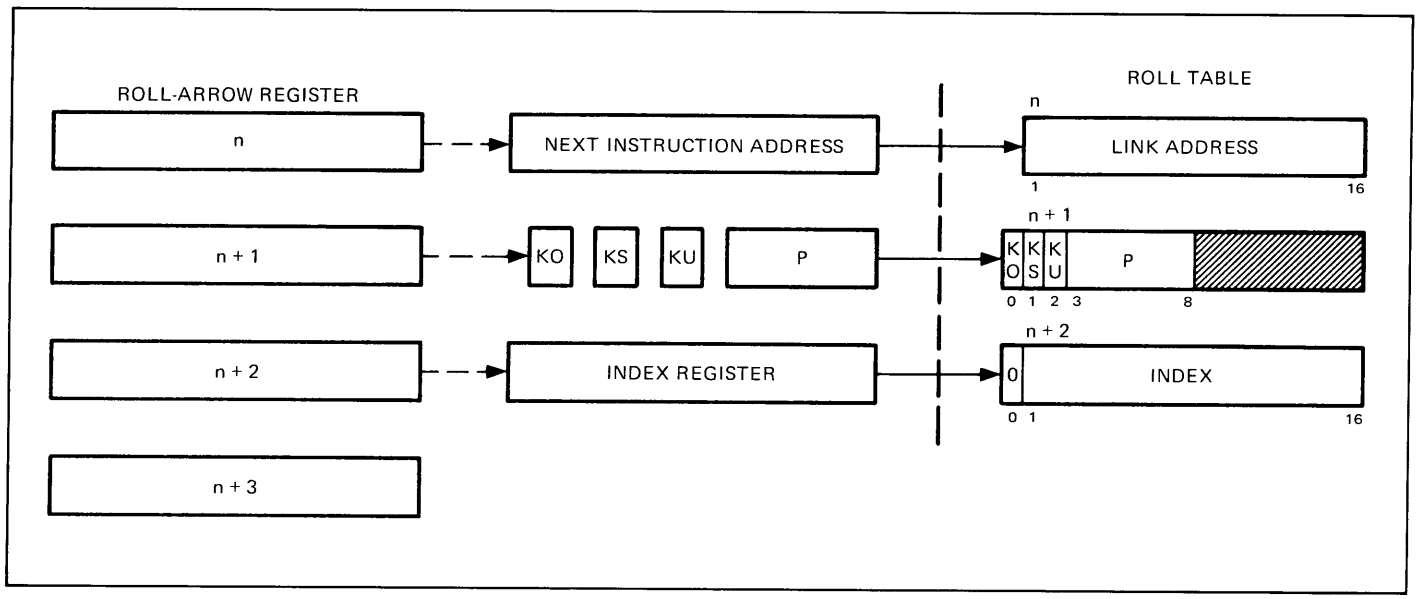
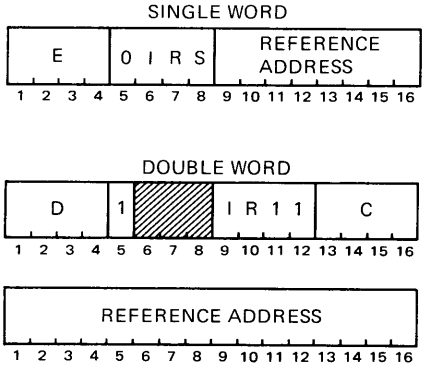


Figure 3-1. Roll Table Storage Sequence for Branch and Put

The Branch Unconditional instruction causes the program to branch to the effective address. This instruction cannot specify indexed addressing.

BXB BRANCH INDEX **3.0 μs**

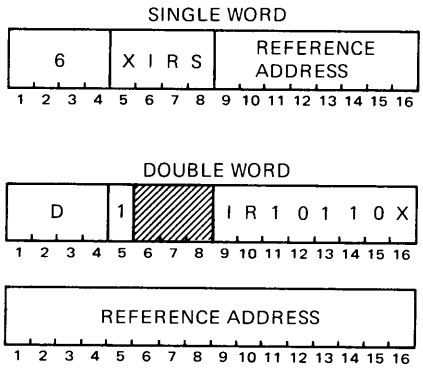


Operation Statement: Branch when index register not equal to zero

Affected: (X)

The Branch Index instruction causes the program to branch to the effective address if the contents of the index register are equal to zero. If the contents are not equal to zero, program execution continues with the next instruction in sequence. Following the test for zero, the contents of the index register are either increased or decreased by a count of one: If the value is negative, it is increased; if the value is positive or equal to zero, it is decreased. This instruction cannot specify indexed addressing.

CMB COMPARE **1.9 μs**



Operation Statement: (B) - (EA)

Affected: KO, KS, KU

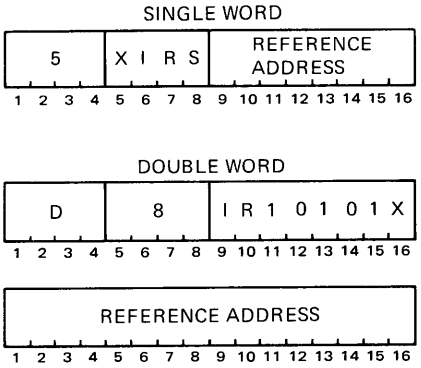
The Compare instruction subtracts the contents of the effective address from the contents of the lower accumulator without affecting the contents of the accumulator. Overflow indicator KO is reset during the operation. Sign indicator KS and Unequal indicator KU are set or reset to indicate the result of the comparison as follows:

Indicators KS	Indicators KU	Result
-	0	Contents of accumulator are equal to contents of effective operand address
0	1	Contents of accumulator are algebraically greater than contents of effective operand address

Indicators KS	KU	Result
1	1	Contents of accumulator are algebraically less than contents of effective operand address

Executing this instruction with an effective address of X'0000' causes the contents of the accumulator to be compared with the contents of the index register.

DVB DIVIDE 11.4 μs



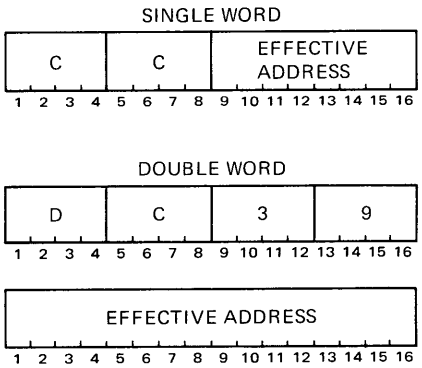
Operation Statement: (B) ÷ (EA) ⟶ (B), (UBA)

Affected: (B), (UBA), KO, KS, KU

The Divide instruction divides the contents of the upper and lower accumulators (treated as one 32-bit register) by the contents of the effective operand address. For integer division, the dividend is loaded into the upper accumulator, and zeros (if the dividend is positive) or ones (if the dividend is negative) are loaded into the lower accumulator. For fractional division, the dividend is loaded into the lower accumulator, and zeros are loaded into the upper accumulator. In either case the operand is compared with the contents of the lower accumulator. If the operand is equal to or smaller than the contents of the lower accumulator, Overflow indicator KO is set and the instruction is terminated with the contents of the upper and lower accumulators unpredictable. If the operand is greater than the contents of the lower accumulator, the division is performed. When instruction execution is completed, the quotient appears in the lower accumulator and the remainder appears in the upper accumulator.

Sign indicator KS is set or reset to match the sign of the quotient (the sign of the remainder is the same as that of the dividend).

IOR INCLUSIVE OR 2.1 μs



Operation Statement: (EA) U (B) ⟶ (B)

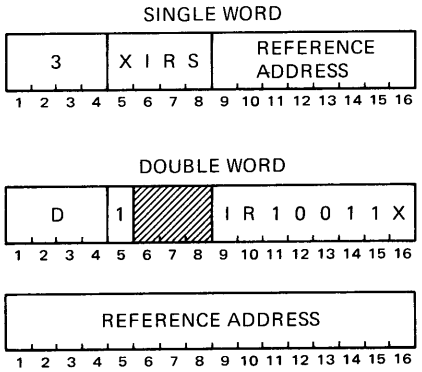
Affected: (B), KO, KS, KU

The Inclusive Or instruction forms the logical sum of the contents of the effective address and the lower accumulator and loads the result into the accumulator. Overflow indicator KO is reset; Sign indicator KS is set if the sign of the result is negative; and Unequal indicator KU is set if the result is not equal to zero.

Executing this instruction with an effective address of X'0000' forms the logical sum of the contents of the index register and the contents of the lower accumulator and loads the result into the accumulator.

This instruction cannot specify indexed, indirect, or relative addressing.

LDB LOAD 1.9 μs



Operation Statement: (EA) ⟶ (B)

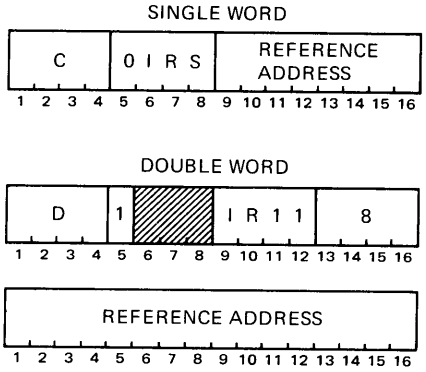
Affected: (B), KO, KS, KU

The Load instruction loads the contents of the effective address into the lower accumulator. Overflow indicator

KO is reset; Sign indicator KS is set if the sign of the effective word is negative; and Unequal indicator is set if the effective word is not equal to zero.

Executing this instruction with an effective address of X'0000' loads the contents of the index register into the lower accumulator.

LDX LOAD INDEX **2.1 μs**

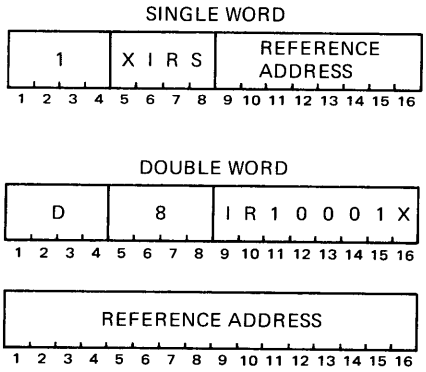


Operation Statement: (EA) ⟶ (X)

Affected: (X)

The Load Index instruction loads the contents of the effective operand address into the index register. Indicators KO, KS, and KU are not affected by this instruction. This instruction cannot specify indexed addressing.

MPB MULTIPLY **6.2 μs**

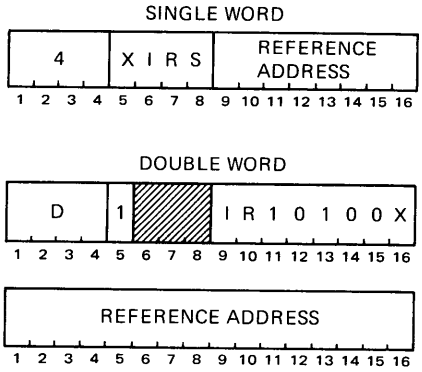


Operation Statement: (B) x (EA) ⟶ (B), (UBA)

Affected: (B), (UBA), KO, KS, KU

The Multiply instruction multiplies the contents of the effective address by the contents of the lower accumulator, loads the most significant half of the product into the upper accumulator, and loads the least significant half of the product into the lower accumulator. Overflow indicator KO is reset, Sign indicator KS is set if the sign of the double word product is negative, and Unequal indicator KU is set if the double word product is not equal to zero.

SBB SUBTRACT **1.9 μs**



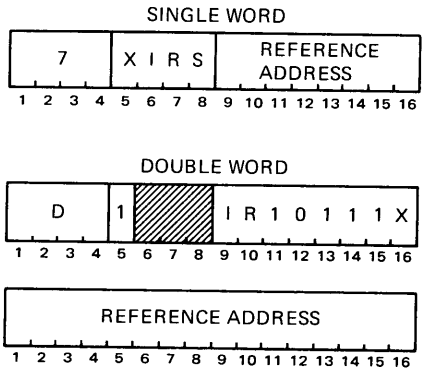
Operation Statement: (B) - (EA) ⟶ (B)

Affected: (B), KO, KS, KU

The Subtract instruction subtracts the contents of the effective address from the contents of the lower accumulator, and loads the difference into the lower accumulator. Overflow indicator KO is reset before the operation and is then set if the difference exceeds the capacity of the accumulator. Sign indicator KS is set if the sign of the difference is negative, and Unequal indicator KU is set if the difference is not equal to zero.

Executing this instruction with an effective address of X'0000' causes the contents of the index register to be subtracted from the contents of the accumulator.

STB STORE **1.9 μs**



Operation Statement: (B) ⟶ (EA)

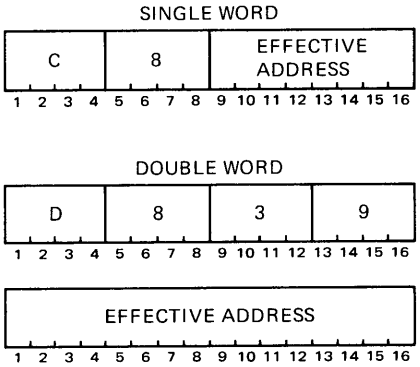
Affected: (EA)

The Store instruction stores the contents of the lower accumulator into the effective address without altering

the contents of the accumulator. The states of indicators KO, KS, and KU are not affected by this instruction.

This instruction can also be used to change the contents of the index register if it is executed with an effective operand address of X'0000'. For this special case the contents of the accumulator are copied into both memory location X'0000' and the index register. The previous contents of the index register are destroyed.

XMB EXCHANGE MEMORY AND LOWER ACCUMULATOR **3.0 μs**



Operation Statement: (EA) ⟶ (B)

(B) ⟶ (EA)

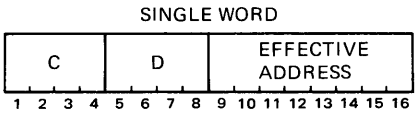
Affected: (EA), (B)

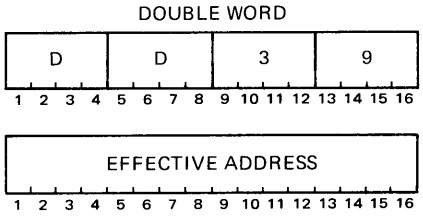
The Exchange Memory and Lower Accumulator instruction exchanges the contents of the effective address with the contents of the lower accumulator. The states of indicators KO, KS, and KU are not affected by this instruction.

This instruction can also be used to change the contents of the index register if it is executed with an effective operand address of X'0000'. For this special case the normal exchange of contents between memory location X'0000' and the lower accumulator occurs; then, the original contents of the accumulator are loaded into the index register. The previous contents of the index register are destroyed.

This instruction cannot specify indexed, indirect, or relative addressing.

XOR EXCLUSIVE OR **2.1 μs**





Operation Statement: (EA) Ⓤ (B) ⟶ (B)

Affected: (B), KO, KS, KU

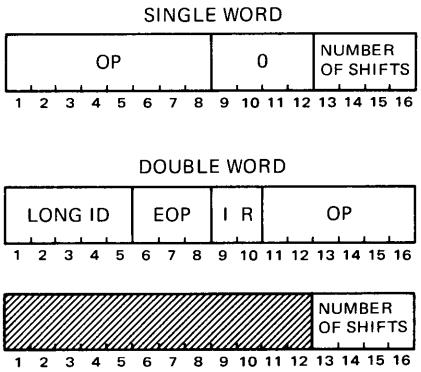
The Exclusive Or instruction forms the logical difference of the contents of the effective address and the lower accumulator and loads the result into the accumulator. Overflow indicator KO is reset; Sign indicator KS is set if the sign of the result is negative; and Unequal indicator KU is set if the result is not equal to zero.

Executing this instruction with an effective address of X'0000' forms the logical difference of the contents of the index register and the lower accumulator and loads the result into the accumulator.

This instruction cannot specify indexed, indirect, or relative addressing.

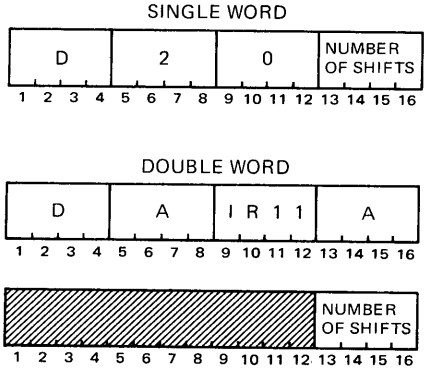
3.3 SHIFT INSTRUCTIONS

Shift instructions have the following basic single and double word formats:



In both formats the operation code fields (OP and EOP) specify the type of shift operation to be performed, and the number-of-shifts field specifies the number of bit positions to be shifted. Both indirect and relative addressing can be specified in the double word format; however, the effective address developed is not used for a memory access. Instead, bits 13 through 16 of the effective address are used to specify the number of bit positions to be shifted. Bits 1 through 12 of the effective address are ignored.

ALB ARITHMETIC LEFT **1.9 to 4.5 μs**



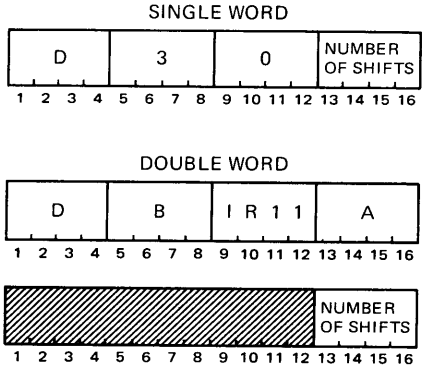
Operation Statement: $2^{(NS)}$ x (B) ⟶ (B)

0's ⟶ $(B)_{16}$

Affected: (B), KO, KS, KU

The Arithmetic Left instruction shifts the contents of the lower accumulator to the left (toward the most significant bit). The number of shifts (0 through 15) is specified by the contents of bits 13 through 16 of the instruction. As the contents are shifted, the bits shifted out of the sign position (bit 1) are lost, and zeros are loaded into the least significant bit position (bit 16). Overflow indicator KO is reset; Sign indicator KS is set if the sign of the shifted result is negative; and Unequal indicator is set if the contents of the accumulator after the shift are not equal to zero.

ARB ARITHMETIC RIGHT **1.9 to 4.5 μs**



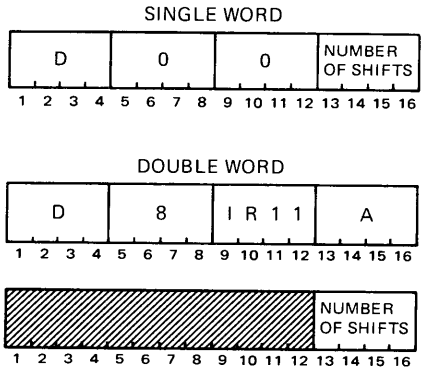
Operation Statement: $(B)/2^{(NS)}$ ⟶ (B)

$(B)_1$ spread right

Affected: (B), KO, KS, KU

The Arithmetic Right instruction shifts the contents of the lower accumulator to the right (toward the most significant bit). The number of shifts (0 through 15) is specified by the number-of-shifts field of the instruction. As the contents are shifted, the sign bit is spread to the right, and the bits shifted out of the least significant bit position (bit 16) are lost. Overflow indicator KO is reset; Sign indicator KS is set if the contents of the accumulator are negative; and Unequal indicator KU is set if the contents are not equal to zero.

ELB END LEFT **1.9 to 7.0 μs**



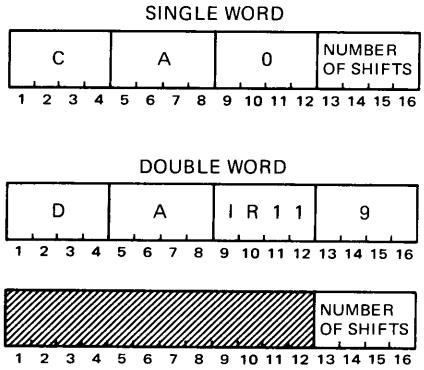
Operation Statement: $2^{(NS)}$ x (B) ⟶ (B)

(B_1) ⟶ (B_{16})

Affected: (B), KO, KS, KU

The End Left instruction shifts the contents of the lower accumulator to the left (toward the most significant bit). The number of shifts (0 through 15) is specified by the number-of-shifts field of the instruction. As the contents are shifted, the bits shifted out of the sign bit position are recirculated and loaded back into the least significant bit position (bit 16). Overflow indicator KO is reset; Sign indicator KS is set if the sign of the shifted result is negative; and Unequal indicator is set if the contents of the accumulator are not equal to zero.

ELD DOUBLE LENGTH SHIFT **3.8 to 9.9 μs**



Operation Statement:

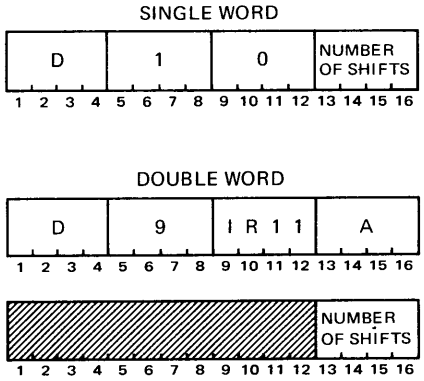
$$2^{(NS)} \times (UBA \text{ and } B) \longrightarrow (UBA \text{ and } B)$$

$$(UBA)_1 \longrightarrow (B)_{16}$$

Affected: (UBA), (B), KO, KS, KU

The Double Length Shift instruction shifts the contents of the upper and lower accumulators to the left (toward the most significant bit). As the contents are shifted, the bits shifted out of the sign bit position of the upper accumulator are recirculated and loaded back into the least significant bit position (bit 16) of the lower accumulator. The number of shifts (0 through 15) is specified by the number-of-shifts field of the instruction. Overflow indicator KO is reset; Sign indicator KS is set if the sign of the result left in the lower accumulator is negative; and Unequal indicator KU is set if the result left in the lower accumulator is not equal to zero.

LRB LOGICAL RIGHT **1.9 to 4.5 μs**



Operation Statement : $(B)/2^{(NS)} \longrightarrow (B)$

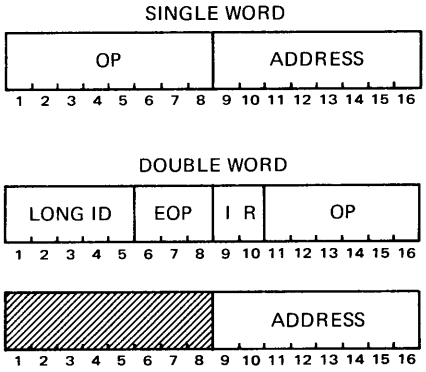
$$0\text{'s} \longrightarrow (B)_1$$

Affected: (B), KO, KS, KU

The Logical Right instruction shifts the contents of the lower accumulator to the right (toward the least significant bit). The number of shifts (0 through 15) is specified by the number-of-shifts field of the instruction. As the contents are shifted, zeros are loaded into the sign bit position (bit 1), and the bits shifted out of the least significant bit position (bit 16) are lost. Overflow indicator KO is reset; Sign indicator KS is reset; and Unequal indicator KU is set if the contents of the accumulator after the shift are not equal to zero.

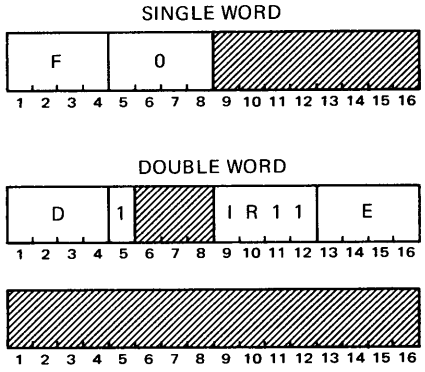
3.4 INPUT/OUTPUT AND CONTROL INSTRUCTIONS

Input/output and control instructions have the following basic single and double word formats:



In both formats the operation code fields (OP and EOP) specify the type of operation (input, output, or control) to be performed. The address field can be interpreted in several ways, depending on the operation being performed. Its specific application for each instruction is explained in the description of each individual instruction.

BBK BRANCH BACK **3.5 μs**



Operation Statement: $(R) - 1 \longrightarrow (R)$

If $((R))_0 = 0$

$((R)) \longrightarrow (X)$

$(R) - 1 \longrightarrow (R)$

$((R))_{0-8} \longrightarrow (KO, KS, KU, P)$

$(R) - 1 \longrightarrow (R)$

$((R)) \longrightarrow (N)$

If $((R))_0 = 1$

$((R)) \longrightarrow (N)$

Affected: (R), (X), (P), KO, KS, KU

The Branch Back instruction returns the program to the instruction immediately following the last executed Branch and Link or Branch and Put instruction. If the last executed instruction was Branch and Link, the Branch Back instruction fetches the return link address from the roll table in memory and returns program control to the instruction stored at the link address. If the last executed instruction was Branch and Put, the Branch Back instruction fetches program status information from the roll table and restores the contents of the index register, page register, and Overflow, Sign, and Unequal indicators to the conditions that existed at the time the Branch and Put instruction was executed. Then, the Branch Back instruction fetches the return link address from the roll table and returns program control to the instruction stored at the link address.

Execution of the Branch Back instruction proceeds as follows:

a. The address in the roll-arrow register is decreased by one count. The new address is now that of the last entry stored in the roll table.

b. Bit 0 of the last entry in the roll table is examined for a 1 (entry was stored by a Branch and Link instruction) or a 0 (entry was stored by a Branch and Put instruction).

c. If bit 0 is a 1, the last entry in the roll table is loaded into the next-instruction-address register, and program execution resumes with the instruction stored at this address.

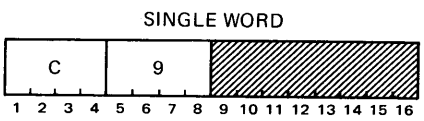
d. If bit 0 is a 0, the following occurs:

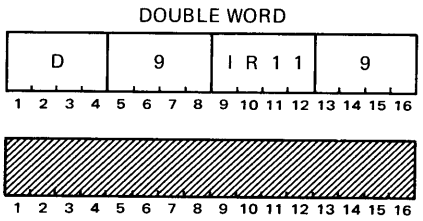
1. The last entry in the roll table is loaded into the index register, and the address in the roll arrow register is decreased by one.

2. Bits 0, 1, and 2 of the next lower entry in the roll table replace the contents of the Overflow, Sign, and Unequal indicators, respectively. Bits 3 through 8 are loaded into the page register. The address in the roll arrow register is then decreased by one.

3. The next lower entry in the roll table is loaded into the next-instruction-address register, and program execution resumes with the instruction stored at this address.

HLT HALT **1.0 μs**



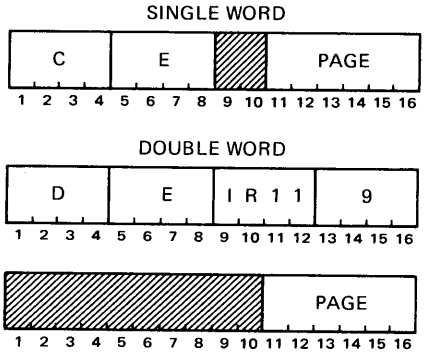


Operation Statement: Set computer to idle mode

Affected: None

The Halt instruction stops instruction execution and places the computer in idle mode. Further instructions cannot be executed until the START switch on the operator's panel is pressed, or until an interrupt (either external or internal) becomes active.

LDP LOAD PAGE **2.1 μs**

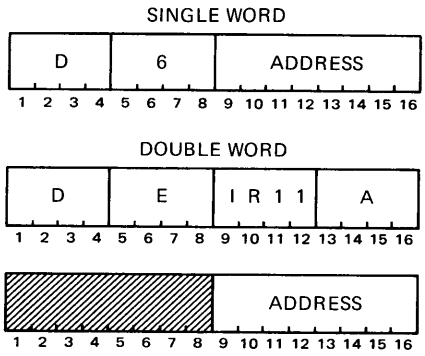


Operation Statement: (Page field) ⟶ (P)

Affected: (P)

The Load Page instruction copies the contents of its page field into the page register. The states of indicators KO, KS, and KU are not affected by this instruction.

PIP PARALLEL INPUT **2.8 μs**



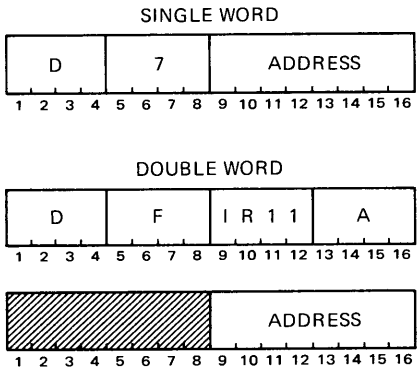
Operation Statement: Input data ⟶ (B)

Affected: (B), KO, KS, KU

The Parallel Input instruction loads data from an input device or from the console data switches into the lower accumulator. Bits 9 through 16 of the effective address specify the input device from which data is loaded. When the effective address is X'01', data is loaded from the console data switches. Overflow indicator KO is reset; Sign indicator KS is set if the sign of the input data is negative; and Unequal indicator KU is set if the input data is not equal to zero.

For more detailed information on input/output operations, refer to section IV.

POP PARALLEL OUTPUT 2.8 μs



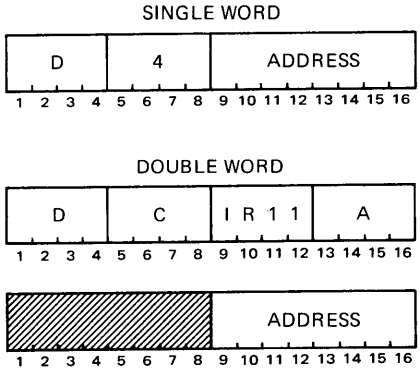
Operation Statement: Transfer data to output device

Affected: None

The Parallel Output instruction transfers the contents of the lower accumulator to an output device, or displays the eight most significant bits of the accumulator on the programmable display indicators of the operator's console. Bits 9 through 16 of the effective address specify the output device to which data is transferred. When the effective address is X'00', bits 1 through 8 of the lower accumulator are displayed by the programmable display indicators on the operator's console. Indicators KO, KS, and KU are not affected by this instruction.

For more detailed information on input/output operations, refer to section IV.

SET SET 2.8 μs



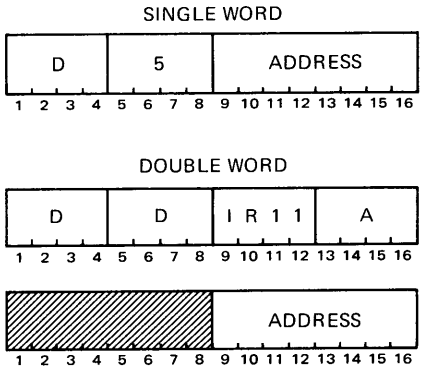
Operation Statement: Transfer discrete control signal to I/O device

Affected: None

The Set instruction transmits a discrete control signal to the I/O device specified by bits 9 through 16 of the effective address. Indicators KO, KS, and KU are not affected by this instruction.

For more detailed information on input/output operations, refer to section IV.

SNS SENSE 2.8 μs



Operation Statement: Test external or internal conditions

Affected: KO, KS, KU

The Sense instruction tests the status of an external device or internal indicator and sets or resets Unequal indicator KU in response to the test. Bits 9 through 16 of the effective address specify the condition to be tested. If the condition is true, indicator KU is set; if the condition is false, KU remains reset. Both Overflow indicator KO and Sign indicator KS are reset.

The following internal conditions can be tested by the Sense instruction:

Address	Indicator
X'00'	Sense switch 8
X'01'	Sense switch 1
X'02'	Sense switch 2
X'03'	Sense switch 3
X'04'	Sense switch 4
X'05'	Sense switch 5
X'06'	Sense switch 6
X'07'	Sense switch 7
X'0A'	Memory parity

For more detailed information on I/O operations, refer to section IV.

SECTION IV
INPUT/OUTPUT OPERATIONS

4.1 GENERAL

The basic input/output system of the computer is characterized by simplicity and flexibility. It permits discrete control and sensing of I/O devices by the Set and Sense instructions, and permits the transfer of single words to and from I/O devices by the Parallel Output and Parallel Input instructions. These four instructions – Set, Sense, Parallel Output, and Parallel Input – are the only ones required to perform any input or output operation, even with an expanded I/O system.

The flexibility of the basic I/O system makes it possible to structure an I/O system to handle the most complex of system applications. Optional block-transfer channels and priority interrupts can be added to the basic system for efficient real-time handling of input/output operations and for single- and multiple-block transfers between the computer and high-speed I/O devices.

4.2 BASIC INPUT/OUTPUT OPERATIONS

Figure 4-1 is a functional diagram of the basic input/output system. Communication and data transfers between the computer and external I/O devices are carried out through the I/O bus to which the I/O devices are connected in party-line fashion. Each I/O device connected to the bus has a unique address to differentiate it from all other devices on the bus. This address is used by each of the input/output instructions to alert a specific device for operation. Since all devices are connected to the same address lines, only the device whose address matches that specified by the input/output instruction responds to the instruction. For control and sensing operations, such as resetting a device or testing that it is ready, discrete control and response signals are transferred between the device and the computer. Responses from an I/O device in answer to a request for status by a Sense instruction are always used to set or reset Unequal indicator KU in

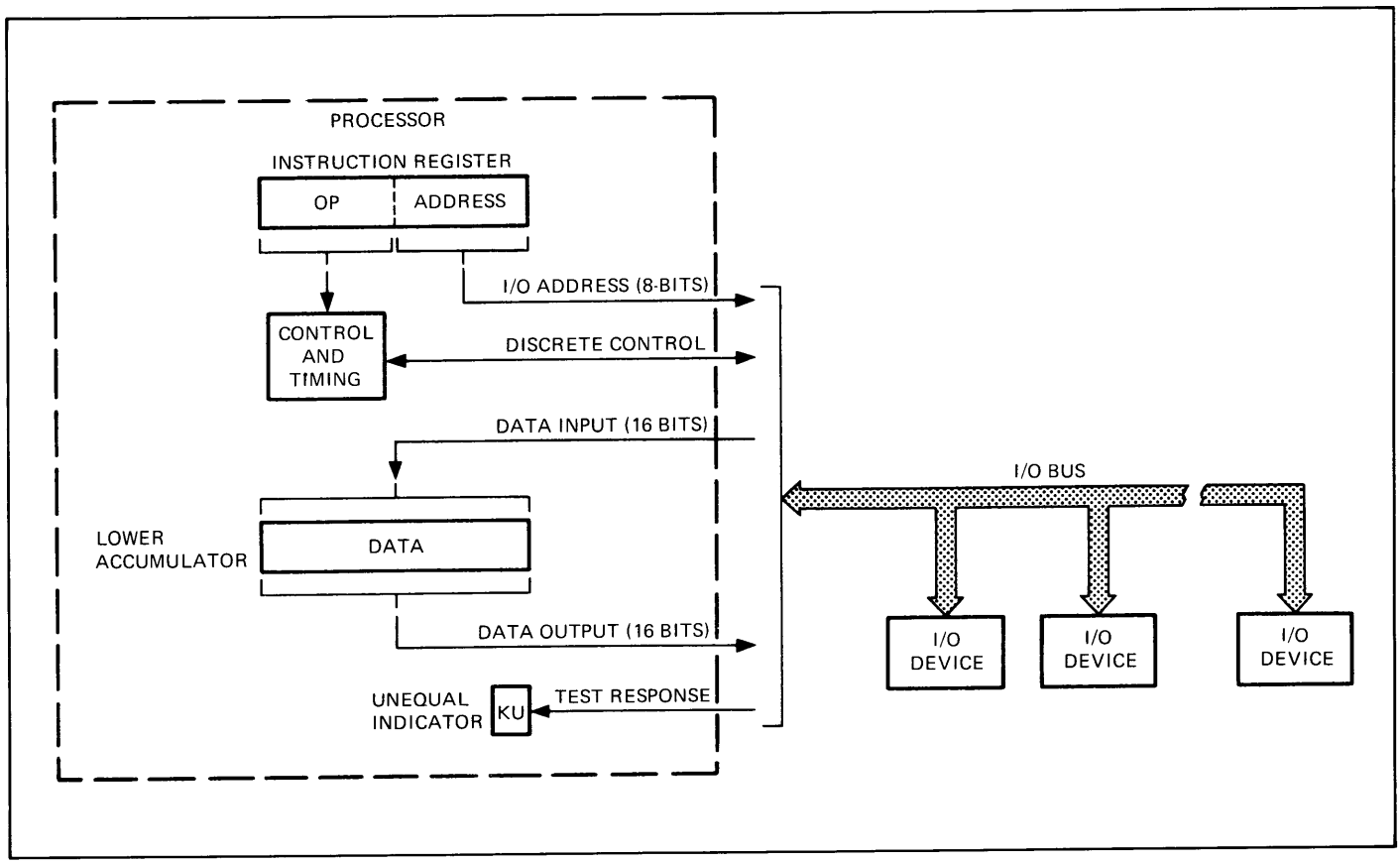


Figure 4-1. Basic Input/Output Functional Diagram

the computer. This indicator can then be tested by the program to determine the status of the device.

Single-word transfers between the computer and the I/O device are carried out through the lower accumulator: A Parallel Input instruction transfers a 16-bit word from the I/O device into the lower accumulator; a Parallel Output instruction transfers a 16-bit word from the lower accumulator to the I/O device.

4.3 I/O DEVICE ADDRESSING

Each input/output instruction has an eight-bit I/O address field to specify the I/O device that is to perform the operation. Although the address field can accommodate 256 different addresses, not all of these are available for assignment to external devices. Some addresses are used for internal control operations that are also performed by the input/output instructions. Table 4-1 contains a list of standard preassigned addresses. All other addresses are assigned to fit each particular application.

4.4 DISCRETE CONTROL AND SENSING

The Set and Sense instructions are used to transmit discrete control signals to an I/O device and to test the status of conditions within the device. These instructions are also used to control and test certain internal conditions of the computer.

The Set instruction is used to transmit a discrete control signal to an I/O device to perform such operations as resetting operating conditions, starting a mechanical operation, rewinding tape, or stopping an operation. When the computer is equipped with the optional block-transfer channels, the Set instruction is used to start or stop a block-transfer operation. It is also used to enable and reset the optional priority interrupts.

The Set instruction is used to enable and disable the internal 2-millisecond clock interrupt. When the instruction is executed with an effective address of X'02', the computer automatically performs an indirect Branch and

Table 4-1. Standard I/O Addresses

Instruction	Address		Function
	Hex	Decimal	
Set	02	2	Enable 2-millisecond interrupt
	03	3	Disable 2-millisecond interrupt
	E2	226	Start block transfer, direct memory access channel 1
	E3	227	Reset direct memory access channel 1
	E4	228	Start block transfer, direct memory access channel 2
	E5	229	Reset direct memory access channel 2
	E6	230	Start block transfer, direct memory access channel 3
	E7	231	Reset direct memory access channel 3
	E8	232	Start block transfer, direct memory access channel 4
	E9	233	Reset direct memory access channel 4
	EA	234	Enable priority interrupts
	EB	235	Reset active priority interrupt
Sense	00	0	Test sense switch 8
	01	1	Test sense switch 1

Table 4-1. Standard I/O Addresses (Cont.)

Instruction	Address		Function
	Hex	Decimal	
Sense (Cont.)	02	2	Test sense switch 2
	03	3	Test sense switch 3
	04	4	Test sense switch 4
	05	5	Test sense switch 5
	06	6	Test sense switch 6
	07	7	Test sense switch 7
	0A	10	Test memory parity
	E2	226	Test chain ready or transfer complete, direct memory access channel 1
	E4	228	Test chain ready or transfer complete, direct memory access channel 2
	E6	230	Test chain ready or transfer complete, direct memory access channel 3
	E8	232	Test chain ready or transfer complete, direct memory access channel 4
Parallel Input	01	1	Read console data switches into lower accumulator
	E2	226	Read word count register, direct memory access channel 1
	E4	228	Read word count register, direct memory access channel 2
	E6	230	Read word count register, direct memory access channel 3
	E8	232	Read word count register, direct memory access channel 4
Parallel Output	00	0	Display most significant byte of lower accumulator on programmable display indicators
	E2	226	Transfer device controller address, direct memory access channel 1
	E4	228	Transfer device controller address, direct memory access channel 2

Table 4-1. Standard I/O Addresses (Cont.)

Instruction	Address		Function
	Hex	Decimal	
Parallel Output (Cont.)	E6	230	Transfer device controller address, direct memory access channel 3
	E8	232	Transfer device controller address, direct memory access channel 4
	EA	234	Arm/disarm priority interrupt levels 2 through 16
	EC	236	Arm/disarm priority interrupt levels 17 through 32

Put instruction to the basic interrupt address X'000F' every 2 milliseconds. Program control is transferred to the address stored in location X'000F' at the time the interrupt occurs. The interrupt is generated every 2 milliseconds until a Set instruction with an effective address of X'03' is executed to disable the 2-millisecond clock.

The Sense instruction is used to test the operational status of an I/O device. It is also used to test the status of the eight console sense switches. When the instruction is executed, Unequal indicator KU is set or reset to indicate the status of the condition specified by the instruction. The Branch Equal or Branch Unequal can then be used to cause the program to take appropriate action based on the test response.

4.5 SINGLE-WORD TRANSFER

The Parallel Input and Parallel Output instructions are used to transfer single 16-bit words between the computer and I/O devices. Single-word transfers are always made to or from the lower accumulator. For a parallel output operation, the word that is to be transferred must be loaded into the accumulator before the Parallel Output instruction is executed.

In addition to transferring data between the computer and external devices, the Parallel Input and Parallel Output instructions can be used to read the console data switches and to display information on the programmable display indicators. Executing a Parallel Input instruction with an effective address of X'01' causes the data set into the console data switches to be entered into the lower accumulator. A Parallel Output instruction with an effective address of X'00' causes the computer to display the most significant byte of the lower accumulator on the programmable display indicators of the console.

Specific applications of the Parallel Input and Parallel Output instructions with the block-transfer and priority interrupt options are described in paragraphs 4.6 through 4.13.

4.6 BLOCK-TRANSFER CHANNELS

The basic input/output system of the computer can be expanded by the addition of optional block-transfer channels which operate through the direct memory access feature. From one to four channels can be added as options, with each channel capable of controlling 16 high-speed I/O device controllers. Each channel can be set up by the program to transfer single blocks or multiple blocks of 16-bit words directly between memory and an I/O device. After setting up the channel, the program initiates the transfer, and from then on the channel controls the transfer operation without further program intervention. Each time the I/O device is ready to transmit or receive data, the channel interrupts the computer after the current memory cycle and uses the next memory cycle to tranfer one word of data. The channel keeps stealing memory cycles until the entire block of words has been transferred or until the program stops the transfer. More than one channel can be set up to operate at the same time, and, if this is done, the channels operate on a priority basis with channel 1 having the highest priority, and channel 4, the lowest. The characteristics of block transfer operation are:

Bits per transfer	16 (one full word)
Maximum words per block	16,384, each channel (chaining feature permits multiple block transfers)
Transfer rate	1.1 million words per second maximum
Memory cycles used per transfer	1

4.7 BLOCK-TRANSFER CONTROL WORDS

Each direct memory access channel requires two control words for a block-transfer operation: one word specifying the starting address of the block to be transferred and one word specifying the number of words in the block. These words must be loaded by the program into memory locations expressly reserved for communication between the channels and the processor before a block-transfer is initiated. The memory locations reserved for use by each direct memory access channel are listed in table 4-2.

Table 4-2. Block-Transfer Control Word Locations

Address		Channel No.	Control Word
Hex	Decimal		
20	32	1	Block starting address
21	33		Block word count
22	34	2	Block starting address
23	35		Block word count
24	36	3	Block starting address
25	37		Block word count
26	38	4	Block starting address
27	39		Block word count

4.8 SINGLE-BLOCK TRANSFER

The sequence of operations for a single-block transfer through a direct memory access channel is as follows:

a. The program loads the block starting address and word count into the memory locations assigned to the channel being used.

b. The program loads the I/O device controller address into the lower accumulator, and then executes a Parallel Output instruction with an effective address of X'E2' (channel 1), X'E4' (channel 2), X'E6' (channel 3), or X'E8' (channel 4) to transfer the address to the appropriate channel.

c. The program executes a Set instruction with an effective address of X'E2' (channel 1), X'E4' (channel 2), X'E6' (channel 3), or X'E8' (channel 4) to initiate the transfer.

d. The direct memory access channel makes two memory accesses to fetch the control words and stores them in two control registers in the channel.

e. As soon as the I/O device is ready to transmit or receive data, the channel stops normal computer operation for one cycle and makes the transfer. After the transfer, the contents of the starting address register in the channel are increased by one, and the word count register is decreased by one in readiness for the next word transfer. When the device is again ready, another cycle of operation takes place. This sequence continues until the contents of the word count register equal zero, or until the program executes a Set instruction with the appropriate channel address (X'E3', X'E5', X'E7', or X'E9') to stop the transfer and reset the channel.

After the transfer has been initiated, the program can test for transfer completion by executing a Sense instruction with the appropriate channel address (X'E2', X'E4', X'E6', or X'E8'). The program can also determine the number of words that have been transferred at any time by executing a Parallel Input instruction to transfer the contents of the channel word count register into the accumulator.

4.9 MULTIPLE-BLOCK TRANSFER (CHAINING)

The sequence of operations for a multiple-block transfer is similar to that for a single-block transfer with the following exceptions:

a. When the program sets up the control words, it sets the chaining flag (bit 1 of the word count control word) to 1 before initiating the operation.

b. After initiating the transfer, the program executes a Sense instruction with the appropriate channel address to determine when the first block control words have been fetched by the channel. When the control words for the first block have been transferred to the channel, the program sets up two new control words, again setting the chaining flag if more blocks are to follow. When the program sets up the control words for the last block, it signals the channel that the next block is the last by setting the chaining flag to 0. From this point on the channel handles the transfer as though it were a single-block transfer.

4.10 PRIORITY INTERRUPTS

The priority interrupt option provides the computer with the capability to respond quickly to a variety of external stimuli. It is especially useful for efficient program control of input/output operations in a real-time environment. The computer can have up to four groups of priority interrupt levels with each group containing eight interrupt levels for a maximum of 32 levels. Each interrupt level is assigned two locations in memory for

storage of a double word Branch Unconditional instruction. When an interrupt level becomes active, program control is automatically transferred to the location assigned to that interrupt, and the Branch Unconditional instruction stored there then branches the program to the interrupt servicing routine.

The priority interrupt levels are designated 1 through 32 with level 1 having the highest priority. This level is preassigned as the power-on interrupt; all other levels are unassigned and can be used for any purpose. The memory location assignments for the priority interrupt levels are listed in table 4-3.

Table 4-3. Priority Interrupt Level Assignments

Address		Group	Level	Priority	Assignment
Decimal	Hexadecimal				
16,320 16,321	3FC0 3FC1	4	32	32	Unassigned
16,322 16,323	3FC2 3FC3		31	31	Unassigned
16,324 16,325	3FC4 3FC5		30	30	Unassigned
16,326 16,327	3FC6 3FC7		29	29	Unassigned
16,328 16,329	3FC8 3FC9		28	28	Unassigned
16,330 16,331	3FCA 3FCB		27	27	Unassigned
16,332 16,333	3FCC 3FCD		26	26	Unassigned
16,334 16,335	3FCE 3FCF		25	25	Unassigned
16,336 16,337	3FD0 3FD1	3	24	24	Unassigned
16,338 16,339	3FD2 3FD3		23	23	Unassigned
16,340 16,341	3FD4 3FD5		22	22	Unassigned
16,342 16,343	3FD6 3FD7		21	21	Unassigned
16,344 16,345	3FD8 3FD9		20	20	Unassigned
16,346 16,347	3FDA 3FDB		19	19	Unassigned
16,348 16,349	3FDC 3FDD		18	18	Unassigned

Table 4-3. Priority Interrupt Level Assigments (Cont.)

Address		Group	Level	Priority	Assignment
Decimal	Hexadecimal				
16,350 16,351	3FDE 3FDF		17	17	Unassigned
16,352 16,353	3FE0 3FE1	2	16	16	Unassigned
16,354 16,355	3FE2 3FE3		15	15	Unassigned
16,356 16,357	3FE4 3FE5		14	14	Unassigned
16,358 16,359	3FE6 3FE7		13	13	Unassigned
16,360 16,361	3FE8 3FE9		12	12	Unassigned
16,362 16,363	3FEA 3FEB		11	11	Unassigned
16,364 16,365	3FEC 3FED		10	10	Unassigned
16,366 16,367	3FEE 3FEF		9	9	Unassigned
16,368 16,369	3FE0 3FF1	1	8	8	Unassigned
16,370 16,371	3FF2 3FF3		7	7	Unassigned
16,372 16,373	3FF4 3FF5		6	6	Unassigned
16,374 16,375	3FF6 3FF7		5	5	Unassigned
16,376 16,377	3FF8 3FF9		4	4	Unassigned
16,378 16,379	3FFA 3FFB		3	3	Unassigned
16,380 16,381	3FFC 3FFD		2	2	Unassigned
16,382 16,383	3FFE 3FFF		1	1	Power on

4.11 INTERRUPT LEVEL STATES

All priority interrupt levels have four operational states (except the power-on interrupt which is always armed and enabled): disarmed, armed, waiting, and active. The significance of each state is as follows:

a. Disarmed. When an interrupt level is disarmed, it cannot accept an interrupt signal from its assigned source. This is a completely inactive state.

b. Armed. When an interrupt level is armed, it can accept and remember an interrupt signal from its assigned source. On receiving the interrupt signal, the interrupt level advances to the waiting state.

c. Waiting. All interrupt levels that have been armed and have received an interrupt signal remain in the waiting state until the interrupts are enabled. At that time the highest priority waiting interrupt level becomes active. All other waiting interrupt levels remain in the waiting state.

d. Active. When the interrupts are enabled, the highest priority waiting interrupt level becomes active upon completion of the instruction currently being executed. The following sequence then occurs:

1. The interrupt address of the interrupt level is automatically written into basic interrupt location X'000F'.

2. The computer executes an indirect Branch and Put instruction with a reference address of X'000F'. The Branch and Put instruction stores current program status and the link address in the roll table, fetches the interrupt level address from location X'000F', and then transfers program control to that address.

3. The computer executes the double word Branch Unconditional instruction stored at the interrupt level address to enter the interrupt servicing routine.

As soon as an interrupt becomes active, the computer automatically disables all interrupt levels. No interrupt level can advance to the active state until the program again enables the interrupts. If the interrupts are again enabled at the beginning of the current servicing routine and a higher priority interrupt level advances to the waiting state, the higher priority interrupt is allowed to interrupt the servicing routine of the lower priority interrupt. In no case, though, can a lower priority interrupt level interrupt the servicing routine of a higher priority interrupt.

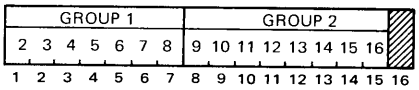
Once an interrupt level becomes active, it remains active until it is reset by the program (normally at the end of the servicing routine). If a lower priority interrupt level is waiting and enabled when the current interrupt level is reset, the lower priority interrupt level immediately advances to the active state.

4.12 INTERRUPT CONTROL

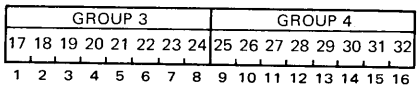
The priority interrupts are controlled by the Parallel Output and Set instructions. Arming and disarming is controlled by the Parallel Output instruction; enabling and resetting are controlled by the Set instruction.

The priority interrupt levels are individually armed or disarmed 16 levels at a time by executing a Parallel Output instruction with an effective address of X'EA' (levels 2 through 16) or X'EC' (levels 17 through 32). The control word in the accumulator at the time the instruction is executed specifies the state to which each level is set. Each bit position of the control word controls one interrupt level. The status of the bit in each position determines whether the level is armed or disarmed. If the bit is 0, the level is disarmed; if the bit is 1, the level is armed. The control words for both versions of the Parallel Output instruction are as follows:

INTERRUPT CONTROL WORD
LEVELS 1 THRU F



INTERRUPT CONTROL WORD
LEVELS 10 THRU 1F



The priority interrupt levels are enabled by executing a Set instruction with an effective address of X'EA'. This instruction enables the interrupt levels as a whole, but only the highest priority waiting interrupt can advance to the active state, and then only if no higher priority interrupt is active.

An active interrupt level is reset by executing a Set instruction with an effective address of X'EB'. This instruction resets the currently active interrupt level and permits the next highest priority waiting and enabled interrupt level to advance to the active state.

4.13 INTERRUPT SERVICING ROUTINE ENTRY AND EXIT

When an interrupt level advances to the active state, the computer automatically stores the status of the program at the time of the interruption (via the Branch and Put instruction executed as a result of the interrupt). It also stores the next instruction address for return linkage from the servicing routine. This permits a Branch Back instruction at the end of the servicing routine to return the program to the point at which the interrupt occurred.

Since the interrupt levels are automatically disabled as soon as an interrupt becomes active, the servicing routine must contain an interrupt enabling instruction (Set X'EA') at the beginning of the routine (if the routine itself is interruptible) or at the end of the routine before control is returned to the point of interruption.

The normal exit instruction sequence is an interrupt reset instruction (Set X'EB') followed by a Branch Back instruction. The Set instruction resets the active interrupt level, and the Branch Back instruction transfers control back to the point at which the interrupt occurred.

SECTION V
OPERATOR'S CONSOLE

5.1 GENERAL

The operator's console contains controls and indicators for applying and removing power, for setting up initial operating conditions, and for displaying or changing the operating status of the computer. As shown in figure 5-1, the console consists of two separate panels. The upper panel contains most of the controls and indicators of interest to the operator or programmer. The lower panel is used mainly for maintenance and is protected by a hinged cover that opens downward for access to the controls.

5.2 CONTROLS AND INDICATORS

5.3 POWER SWITCH-INDICATOR

The POWER switch controls the application of ac power to the computer. It is a push-on, push-off switch and indicator. The face of the switch is back-lighted when ac power is on.

5.4 RESET SWITCH

The RESET switch places the computer in idle mode and sets up initial operating conditions. Pressing the switch causes the following to occur:

a. The computer enters the idle mode at the end of the current memory cycle.

b. The contents of data switches 1 through 16 are loaded into the memory address register.

c. The contents of the page register are set to X'01'. When the computer is in idle mode, all console controls and indicators are operative. The computer is taken out of idle mode when the START switch is pressed or when an internal or external interrupt becomes active.

5.5 FILL SWITCH

The FILL switch activates a read-only diode memory containing a 19-instruction fill routine. Pressing the switch causes the computer to begin executing the fill routine at address X'8000', the starting address of the diode memory. The fill routine begins loading core memory from I/O address X'12' as described in paragraph 5.26. Loading continues until an entire program is loaded or until the RESET or SINGLE CYCLE switch is pressed. The FILL switch is operative only when the computer is in idle mode.

5.6 SINGLE CYCLE SWITCH-INDICATOR

The SINGLE CYCLE switch is used to step the computer through a program one instruction at a time. It is a

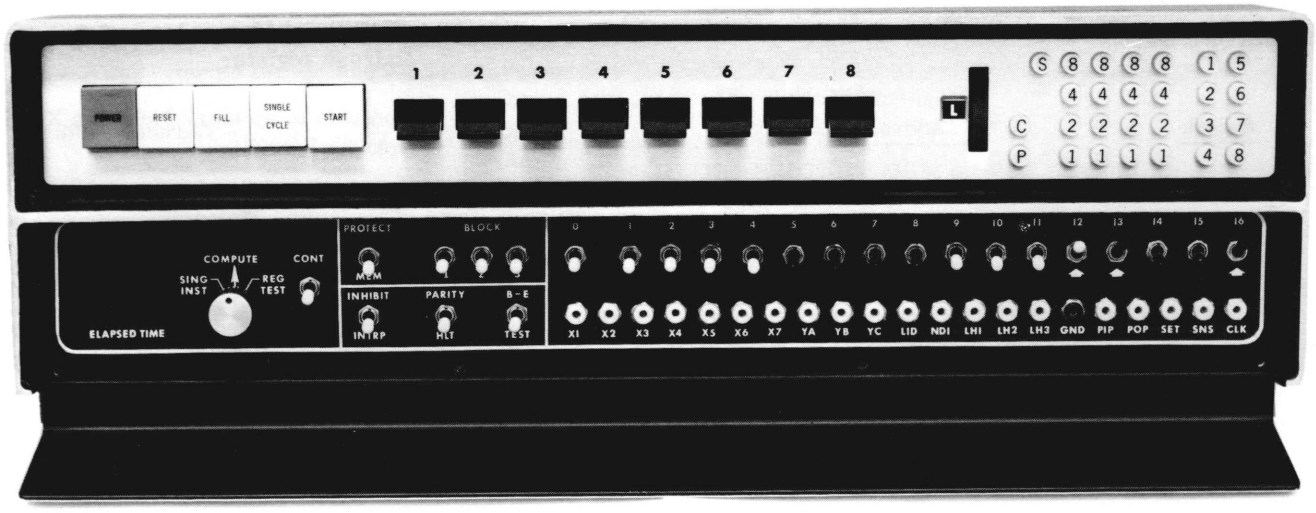


Figure 5-1. Operator's Console

push-on, push-off switch with a back-lighted indicator. If the switch is pressed when the computer is in idle mode, the indicator lights, but nothing else occurs until the START switch is pressed or an external interrupt becomes active. Pressing the START switch then causes the computer to leave idle mode, execute one complete instruction, and return to idle mode again. Pressing the SINGLE CYCLE switch a second time inhibits single cycle operation and causes the indicator to go out. If the SINGLE CYCLE switch is pressed when the computer is in compute mode, the computer returns to idle mode after completely executing the instruction currently in progress, and single cycle operation can then be performed using the START switch. When single cycle operation is in effect, the 2-millisecond internal interrupt is inhibited.

5.7 START SWITCH

The START switch controls instruction execution. Pressing the switch causes the computer to leave idle mode, enter compute mode, and begin executing instructions. If single cycle operation is not in effect when the switch is pressed, the computer continuously executes instructions until either a Halt instruction is executed, or the RESET or SINGLE CYCLE switch is pressed. If single cycle operation is in effect, the computer executes one complete instruction and then returns to idle mode.

5.8 SENSE SWITCHES (1 THROUGH 8)

The eight sense switches are used for external control of a stored program. They are two-position toggle switches with the upper position representing a logical 1 (set) and the lower position representing a logical 0 (reset). The status of each switch can be tested and copied into Unequal indicator KU by a Sense instruction with the appropriate address. Either a Branch Equal or a Branch Unequal instruction can then be used to test the Unequal indicator. The hexadecimal addresses of the sense switches are:

Switch	Hexadecimal Address
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	00

5.9 REGISTER SELECT SWITCH

The register select switch is a 15-position thumbwheel switch that is used primarily to select certain internal registers, indicators, or memory locations for display by the register display indicators. It is also used with the function switch (REG TEST position) to enter data from the 16 data switches into a selected register. The information selected for display by this switch is meaningful only when the computer is in idle mode. The switch positions and the data selected for display by each position are summarized as follows:

Switch Position	Data Displayed
XY	States of minor timing flip-flops X1 through X7 and major timing flip-flops YA through YC. These indications are of interest to maintenance personnel. Figure 5-2 shows the display assignments when XY is selected
PK	Contents of the page register (P1 through P6); states of Overflow indicator KO, Sign indicator KS, and Unequal indicator KU; and the states of carry flip-flop KK and temporary storage flip-flops KA, KB, and KC. Figure 5-2 shows the display assignments when PK is selected
D	Contents of 16-bit D-register
L	Contents of 14-bit memory address register
I	Contents of 16-bit instruction register
R	Contents of roll-arrow register (memory location X'0007')
X	Contents of 16-bit index register
B	Contents of 16-bit lower accumulator
N	Contents of 16-bit next-instruction-address register
A1 through A6	Contents of memory locations X'0001' through X'0006'

5.10 C (COMPUTE) INDICATOR

The C (compute) indicator, located immediately to the right of the register select switch, displays the operating mode of the computer. It is lit whenever the computer is operating in compute mode.

5.11 P (MEMORY PARITY) INDICATOR

The P (memory parity) indicator, located directly below the C indicator, lights to indicate that a memory parity error has been detected.

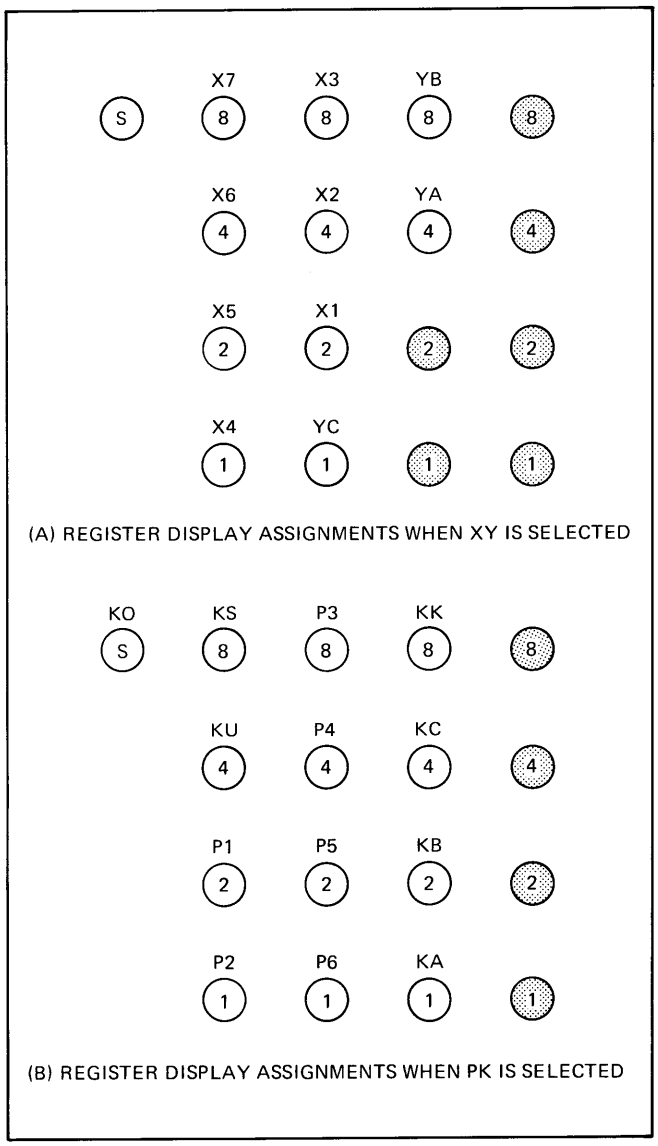


Figure 5-2. Register Display Indicator Assignments for PK and XY

5.12 REGISTER DISPLAY

The 17 register display indicators display the contents of the register, indicators, or memory location selected by the register select switch (paragraph 5.9). The indicators are arranged in four columns of four indicators each (labeled 8-4-2-1) plus a single indicator (labeled S). Each column represents four binary digits or one hexadecimal digit. The information displayed by the indicators is valid only when the computer is operating in idle mode.

5.13 PROGRAMMABLE DISPLAY

The eight programmable display indicators can be programmed to display the eight most significant bits (B1 through B8) of the lower accumulator. The indicators are arranged in two columns of four indicators each (labeled 1-2-3-4 and 5-6-7-8). The display is turned on by executing a Parallel Output instruction with an I/O address of X'00'. After the instruction is executed, each indicator is lit if there is a 1-bit in the corresponding bit position of the accumulator. The information displayed by these indicators is not saved during a shutdown sequence; therefore, when power returns to normal, the display must be reprogrammed to again display the information.

5.14 FUNCTION SWITCH

The function switch, located on the left-hand side of the lower panel, is used mainly for maintenance and program debugging. It is a three-position rotary switch with positions labeled SING INST (single instruction), COMPUTE, AND REG TEST (register test). The switch must set to COMPUTE for normal computer operation. When the switch is set to SING INST and the START switch is pressed, the computer executes the instruction specified by the positions of data switches 1 through 16 and then returns to idle mode. When the switch is set to REG TEST, the data configuration specified by the positions of data switches 1 through 16 is loaded into the register specified by the setting of the register select switch, and is displayed by the register display indicators.

5.15 CONT (CONTINUOUS) SWITCH

The CONT switch is a two-position toggle switch that is used mainly for maintenance and troubleshooting. When the switch is set to the upper (continuous) position, the computer starts executing instructions with the instruction at the address specified by the positions of data switches 1 through 16. One millisecond later the computer halts. After a 1-millisecond halt the computer returns to the address specified by the data switches and again starts computation. This cycle continues until the switch is set to the lower position.

5.16 PROTECT MEM SWITCH

The PROTECT MEM (protect memory) is a two-position toggle switch that is used with the BLOCK switches to prevent writing into protected areas of memory. When the switch is set to the upper (protect) position, writing into the area of memory specified by the setting of the BLOCK switches is prevented. When the switch is set to the lower position (unprotected), writing into any area of memory is permitted.

5.17 BLOCK SWITCHES (1 THROUGH 3)

The three BLOCK switches select the area of memory that is protected when the PROTECT MEM switch is set to the upper position. These are two-position toggle

switches with the upper position representing a logical 1 (set) and the lower position representing a logical 0 (reset). The areas of memory protected by the switches are as follows:

Switch Settings	Protected Memory Area (Hex)
000	Entire memory
001	0800 to highest address
010	1000 to highest address
011	1800 to highest address
100	2000 to highest address
101	2800 to highest address
110	3000 to highest address
111	3800 to highest address

5.18 INHIBIT INTRP SWITCH

The INHIBIT INTRP (inhibit interrupt) switch is a two-position toggle switch that controls whether or not an interrupt can occur. When the switch is set to INHIBIT, no interrupts are allowed. When it is set to INTRP, interrupts are allowed to occur provided that the interrupt level is armed and enabled.

5.19 PARITY HLT SWITCH

The PARITY HLT (parity halt) switch is a two-position toggle switch that controls whether or not the computer halts when a memory parity error is detected. A parity check is made each time a word is read from memory. If the switch is set to HLT and a parity error occurs, the P indicator lights to indicate the error, but the computer continues executing instructions. If the switch is set to PARITY and a parity error occurs, the P indicator lights and the computer halts after executing the current instruction. The RESET and START switches must then be pressed to restart the program.

5.20 B-E TEST SWITCH

The B-E TEST switch is a two-position toggle switch that is used to test the power shutdown and restart feature of the computer. When the switch is set to B-E, the computer continuously cycles through a power shutdown and restart sequence. The sequence consists of a 1-millisecond operation period followed by a 7-millisecond shutdown period. This cycle continues as long as the switch is set to B-E.

5.21 DATA SWITCHES (0 THROUGH 16)

The 16 data switches are used to manually enter data or instructions into the computer. These are two-position toggle switches with the upper position representing a logical or binary 1 and the lower position representing a logical or binary 0. Information is entered into the computer from the data switches in any of the following ways:

a. Pressing the RESET switch loads the address specified by the settings of the switches into the memory address register.

b. Executing a Parallel Input instruction with an I/O address of X'01' loads data from the switches into the lower accumulator.

c. Setting the function switch to SING INST and pressing the START switch causes the computer to execute the instruction set into the switches and then return to idle mode.

d. Setting the function switch to REG TEST loads data from the switches into the register specified by the setting of the register select switch.

5.22 TEST JACKS

Twenty-one test jacks, located on the lower panel, are available for monitoring significant computer timing and control signals. The voltage levels appearing at these test jacks represent inverted logic levels: that is, 0V represents a true logic level and +5V represents a false logic level.

5.23 OPERATING PROCEDURES

5.24 APPLYING POWER AND INITIALIZING

To apply power to the computer and set up initial operating conditions, use the following procedure:

1. Verify that computer power cable is connected to 115V, 60 Hz electrical service.

2. Lower the cover of the bottom panel for access to the switches.

3. Set the function switch to COMPUTE.

4. Set the CONT and B-E TEST switches to the lower position.

5. If an area of memory is to be protected, set the PROTECT MEM and BLOCK switches as required (paragraphs 5.16 and 5.17); otherwise, set them to the lower position.

6. Set the INHIBIT INTRP and PARITY HLT switches for the desired operation (paragraphs 5.18 and 5.19).

7. Set data switches 12, 13, and 16 to the upper position; set all other data switches to the lower position. (NOTE: If a program is already stored in the computer, set the data switches to the starting address of the program.)

8. Press the POWER switch. The POWER indicator should light to indicate that ac power has been applied.

9. Press the RESET switch. The address set into the data switches is loaded into the memory address register, and the page register is set to X'01'.

With the completion of step 9 the computer is in idle mode, ready for operation. If a program is stored in memory, it can be executed by pressing the START switch. If there is no program in memory, one can be loaded using the procedure given in paragraph 5.26.

5.25 REMOVING POWER

To remove power from the computer, press the POWER switch. The POWER indicator should go out to indicate that ac power is off.

5.26 LOADING A PROGRAM

Loading a new program into the computer is initiated by a 19-instruction fill routine permanently stored in a read-only diode memory inside the computer. When the fill routine is executed, it loads a string of instructions, one byte at a time, from the input device into low order memory (locations X'0028' through X'004F'). The storage location of the first word in the string is selected so that the last word loaded by the fill routine is always stored in location X'004F'. Since locations below X'0028' are reserved for special uses, the maximum number of words that can be loaded by the fill routine is 41 (82 bytes). The instruction string read into the computer by the fill routine is normally a bootstrap loader that is used to load the rest of the program into memory.

The input device from which the program is loaded must have X'12' as its I/O address. This address is fixed in the fill routine and cannot be changed. The string of instructions loaded by the fill routine must be headed by one byte containing the one's complement of the number of words to be loaded. This byte is used by the fill routine to determine the storage location of the first word. Normally, the last word stored (location X'004F') is a checksum used to verify correct program loading. When the last word has been loaded by the fill routine, program control is transferred to location X'004E'. This location normally contains a branch instruction to transfer control to the starting address of the bootstrap loader. A flow diagram of the fill routine is shown in figure 5-3.

The procedure for loading a new program into the computer is as follows:

1. Set up initial operating conditions as specified in paragraph 5.24, steps 1 through 9.

2. Press the FILL switch to execute the fill routine. The program is now loaded into memory.

3. To execute the program just loaded, press the RESET switch and then the START switch.

5.27 CHANGING REGISTER CONTENTS

To change the contents of any of the selectable registers, use the following procedure:

1. Place the computer in idle mode by pressing the SINGLE CYCLE switch.

2. Set the register select switch to indicate the register whose contents are to be changed. The register display indicators display the current contents of the register.

3. Set the data that is to be entered into the register into data switches 1 through 16.

4. Set the function switch to REG TEST. The information set into the data switches is loaded into the selected register and is displayed by the register display indicators.

5. To restart the program, set the function switch to COMPUTE, press the SINGLE CYCLE switch, and then press the START switch.

5.28 DISPLAYING REGISTER CONTENTS

To display the contents of any of the selectable registers, use the following procedure:

1. Place the computer in idle mode by pressing the SINGLE CYCLE switch.

2. Set the register select switch to indicate the register whose contents are to be displayed. The register display indicators now display the contents of the selected register.

3. To restart the program, press the SINGLE CYCLE switch, and then press the START switch.

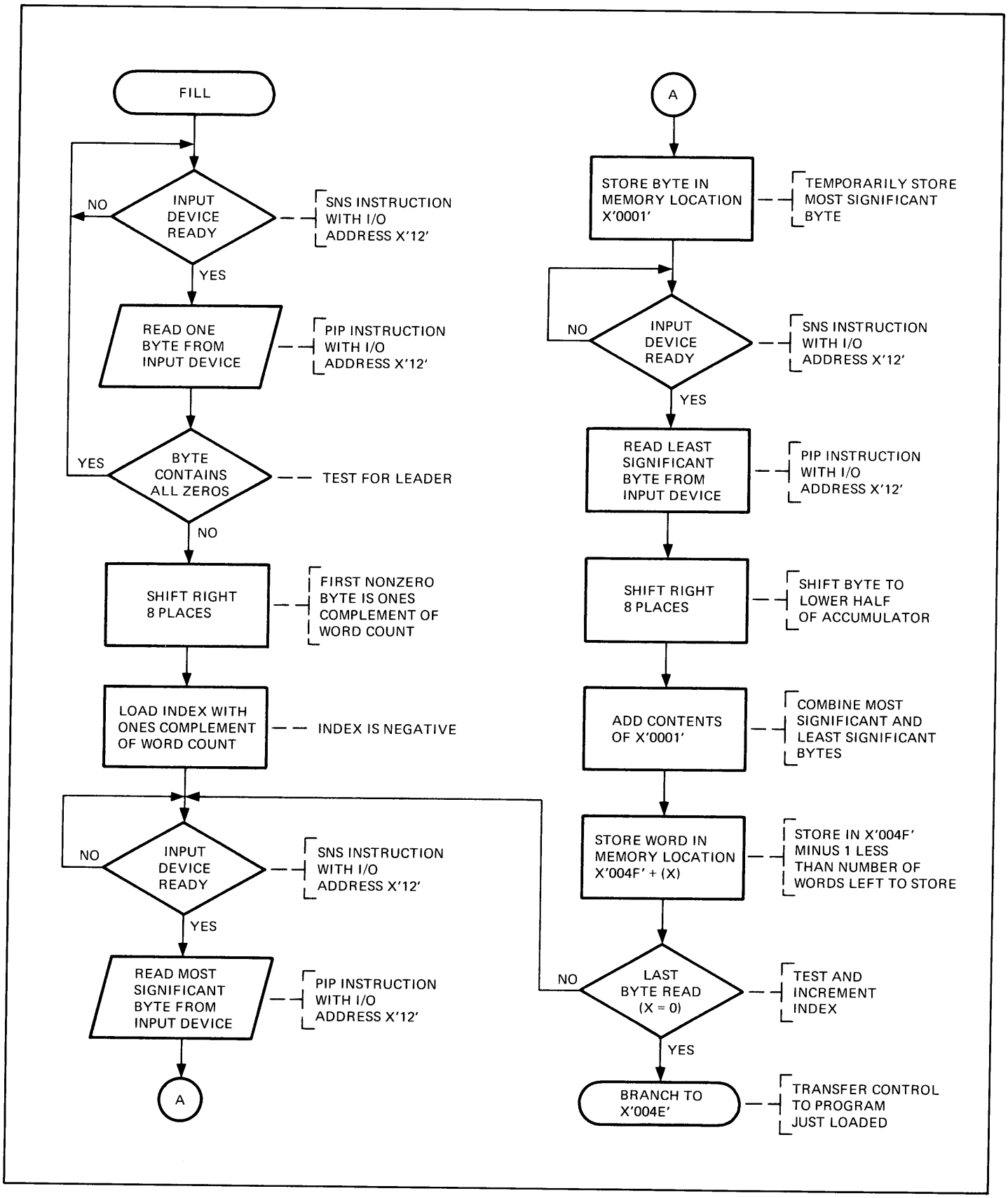


Figure 5-3. Fill Routine, Flow Diagram

5.29 CHANGING MEMORY CONTENTS

To change the contents of a location in memory, use the following procedure:

1. Place the computer in idle mode by pressing the SINGLE CYCLE switch.

2. Set the register select switch to X.

3. Set the memory address into data switches 1 through 16.

4. Set the function switch to REG TEST and then back to COMPUTE. The memory address is now loaded into the index register.

5. Set the register select switch to B.

6. Set the data to be stored in memory into data switches 1 through 16.

7. Set the function switch to REG TEST and then back to COMPUTE. The data is now loaded into the B-register.

8. Set X'7800' (Store instruction) into data switches 1 through 16. Verify that the PROTECT MEM switch is off before proceeding with the next step.

9. Set the function switch to SING INST and then press the START switch. The computer now executes the Store instruction to write the data into the selected memory location.

10. To restart the program, reset the function switch to COMPUTE, press the SINGLE CYCLE switch, and then press the RESET switch.

5.30 DISPLAYING MEMORY CONTENTS

To display the contents of a location in memory, use the following procedure:

1. Place the computer in idle mode by pressing the SINGLE CYCLE switch.

2. Set the register select switch to X.

3. Set the memory address into data switches 1 through 16.

4. Set the function switch to REG TEST and then back to COMPUTE. The memory address is now loaded into the index register.

5. Set X'3800' (Load instruction) into data switches 1 through 16.

6. Set the function switch to SING INST and then press the START switch. The contents of the selected memory location are now loaded into the accumulator.

7. Set the register select switch to B to display the contents of the memory location.

8. To restart the program, reset the function switch to COMPUTE, press the SINGLE CYCLE switch, and then press the START switch.

APPENDIX A
CONVERSION TABLES

This appendix contains the following reference tables:

HEXADECIMAL ARITHMETIC

ADDITION TABLE

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
3	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12
4	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
5	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14
6	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
7	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
8	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17
9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18
A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A
C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

MULTIPLICATION TABLE

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
3	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
4	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	1E	2B	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

TABLE OF POWERS OF SIXTEEN$_{10}$

16^n	n	16^{-n}
1	0	0.10000 00000 00000 00000 × 10
16	1	0.62500 00000 00000 00000 × 10^{-1}
256	2	0.39062 50000 00000 00000 × 10^{-2}
4 096	3	0.24414 06250 00000 00000 × 10^{-3}
65 536	4	0.15258 78906 25000 00000 × 10^{-4}
1 048 576	5	0.95367 43164 06250 00000 × 10^{-6}
16 777 216	6	0.59604 64477 53906 25000 × 10^{-7}
268 435 456	7	0.37252 90298 46191 40625 × 10^{-8}
4 294 967 296	8	0.23283 06436 53869 62891 × 10^{-9}
68 719 476 736	9	0.14551 91522 83668 51807 × 10^{-10}
1 099 511 627 776	10	0.90949 47017 72928 23792 × 10^{-12}
17 592 186 044 416	11	0.56843 41886 08080 14870 × 10^{-13}
281 474 976 710 656	12	0.35527 13678 80050 09294 × 10^{-14}
4 503 599 627 370 496	13	0.22204 46049 25031 30808 × 10^{-15}
72 057 594 037 927 936	14	0.13877 78780 78144 56755 × 10^{-16}
1 152 921 504 606 846 976	15	0.86736 17379 88403 54721 × 10^{-18}

TABLE OF POWERS OF 10_{16}

10^n	n	10^{-n}
1	0	1.0000 0000 0000 0000
A	1	0.1999 9999 9999 999A
64	2	0.28F5 C28F 5C28 F5C3 × 16^{-1}
3E8	3	0.4189 374B C6A7 EF9E × 16^{-2}
2710	4	0.68DB 8BAC 710C B296 × 16^{-3}
1 86A0	5	0.A7C5 AC47 1B47 8423 × 16^{-4}
F 4240	6	0.10C6 F7A0 B5ED 8D37 × 16^{-4}
98 9680	7	0.1AD7 F29A BCAF 4858 × 16^{-5}
5F5 E100	8	0.2AF3 1DC4 6118 73BF × 16^{-6}
3B9A CA00	9	0.44B8 2FA0 9B5A 52CC × 16^{-7}
2 540B E400	10	0.6DF3 7F67 5EF6 EADF × 16^{-8}
17 4876 E800	11	0.AFEB FF0B CB24 AAFF × 16^{-9}
E8 D4A5 1000	12	0.1197 9981 2DEA 1119 × 16^{-9}
918 4E72 A000	13	0.1C25 C268 4976 81C2 × 16^{-10}
5AF3 107A 4000	14	0.2D09 370D 4257 3604 × 16^{-11}
3 8D7E A4C6 8000	15	0.480E BE7B 9D58 566D × 16^{-12}
23 8652 6FC1 0000	16	0.734A CA5F 6226 F0AE × 16^{-13}
163 4578 5D8A 0000	17	0.B877 AA32 36A4 B449 × 16^{-14}
DE0 B6B3 A764 0000	18	0.1272 5DD1 D243 ABA1 × 16^{-14}
8AC7 2304 89E8 0000	19	0.1D83 C94F B6D2 AC35 × 16^{-15}

HEXADECIMAL-DECIMAL INTEGER CONVERSION

The table below provides for direct conversions between hexadecimal integers in the range 0—FFF and decimal integers in the range 0—4095. For conversion of larger integers, the table values may be added to the following figures:

Hexadecimal	Decimal	Hexadecimal	Decimal
01 000	4 096	20 000	131 072
02 000	8 192	30 000	196 608
03 000	12 288	40 000	262 144
04 000	16 384	50 000	327 680
05 000	20 480	60 000	393 216
06 000	24 576	70 000	458 752
07 000	28 672	80 000	524 288
08 000	32 768	90 000	589 824
09 000	36 864	A0 000	655 360
0A 000	40 960	B0 000	720 896
0B 000	45 056	C0 000	786 432
0C 000	49 152	D0 000	851 968
0D 000	53 248	E0 000	917 504
0E 000	57 344	F0 000	983 040
0F 000	61 440	100 000	1 048 576
10 000	65 536	200 000	2 097 152
11 000	69 632	300 000	3 145 728
12 000	73 728	400 000	4 194 304
13 000	77 824	500 000	5 242 880
14 000	81 920	600 000	6 291 456
15 000	86 016	700 000	7 340 032
16 000	90 112	800 000	8 388 608
17 000	94 208	900 000	9 437 184
18 000	98 304	A00 000	10 485 760
19 000	102 400	B00 000	11 534 336
1A 000	106 496	C00 000	12 582 912
1B 000	110 592	D00 000	13 631 488
1C 000	114 688	E00 000	14 680 064
1D 000	118 784	F00 000	15 728 640
1E 000	122 880	1 000 000	16 777 216
1F 000	126 976	2 000 000	33 554 432

Hexadecimal fractions may be converted to decimal fractions as follows:

1. Express the hexadecimal fraction as an integer times 16^{-n}, where n is the number of significant hexadecimal places to the right of the hexadecimal point.

$$0.\,CA9BF3_{16} = CA9\,BF3_{16} \times 16^{-6}$$

2. Find the decimal equivalent of the hexadecimal integer

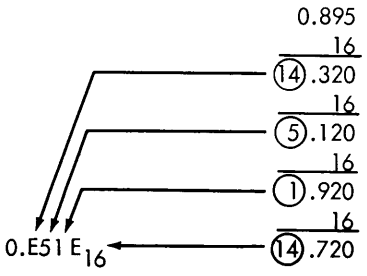
$$CA9\,BF3_{16} = 13\,278\,195_{10}$$

3. Multiply the decimal equivalent by 16^{-n}

$$\begin{array}{l} 13\,278\,195 \\ \times\ 596\,046\,448 \times 10^{-16} \\ \hline 0.791\,442\,096_{10} \end{array}$$

Decimal fractions may be converted to hexadecimal fractions by successively multiplying the decimal fraction by 16_{10}. After each multiplication, the integer portion is removed to form a hexadecimal fraction by building to the right of the hexadecimal point. However, since decimal arithmetic is used in this conversion, the integer portion of each product must be converted to hexadecimal numbers.

Example: Convert 0.895_{10} to its hexadecimal equivalent



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
010	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
020	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
030	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
040	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
050	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
060	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
070	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
080	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
090	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A0	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B0	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C0	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D0	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E0	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F0	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
100	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
110	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
120	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
130	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
140	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
150	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
160	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
170	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
180	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
190	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A0	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B0	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C0	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D0	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E0	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F0	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
200	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
210	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
220	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
230	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
240	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
250	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
260	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
270	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
280	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
290	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A0	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B0	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C0	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D0	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E0	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F0	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
300	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
310	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
320	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
330	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
340	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
350	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
360	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
370	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
380	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
390	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A0	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B0	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C0	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D0	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E0	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE0	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF0	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB0	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC0	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF0	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F00	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB0	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD0	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE0	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

HEXADECIMAL-DECIMAL FRACTION CONVERSION

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.40 00 00 00	.25000 00000	.80 00 00 00	.50000 00000	.C0 00 00 00	.75000 00000
.01 00 00 00	.00390 62500	.41 00 00 00	.25390 62500	.81 00 00 00	.50390 62500	.C1 00 00 00	.75390 62500
.02 00 00 00	.00781 25000	.42 00 00 00	.25781 25000	.82 00 00 00	.50781 25000	.C2 00 00 00	.75781 25000
.03 00 00 00	.01171 87500	.43 00 00 00	.26171 87500	.83 00 00 00	.51171 87500	.C3 00 00 00	.76171 87500
.04 00 00 00	.01562 50000	.44 00 00 00	.26562 50000	.84 00 00 00	.51562 50000	.C4 00 00 00	.76562 50000
.05 00 00 00	.01953 12500	.45 00 00 00	.26953 12500	.85 00 00 00	.51953 12500	.C5 00 00 00	.76953 12500
.06 00 00 00	.02343 75000	.46 00 00 00	.27343 75000	.86 00 00 00	.52343 75000	.C6 00 00 00	.77343 75000
.07 00 00 00	.02734 37500	.47 00 00 00	.27734 37500	.87 00 00 00	.52734 37500	.C7 00 00 00	.77734 37500
.08 00 00 00	.03125 00000	.48 00 00 00	.28125 00000	.88 00 00 00	.53125 00000	.C8 00 00 00	.78125 00000
.09 00 00 00	.03515 62500	.49 00 00 00	.28515 62500	.89 00 00 00	.53515 62500	.C9 00 00 00	.78515 62500
.0A 00 00 00	.03906 25000	.4A 00 00 00	.28906 25000	.8A 00 00 00	.53906 25000	.CA 00 00 00	.78906 25000
.0B 00 00 00	.04296 87500	.4B 00 00 00	.29296 87500	.8B 00 00 00	.54296 87500	.CB 00 00 00	.79296 87500
.0C 00 00 00	.04687 50000	.4C 00 00 00	.29687 50000	.8C 00 00 00	.54687 50000	.CC 00 00 00	.79687 50000
.0D 00 00 00	.05078 12500	.4D 00 00 00	.30078 12500	.8D 00 00 00	.55078 12500	.CD 00 00 00	.80078 12500
.0E 00 00 00	.05468 75000	.4E 00 00 00	.30468 75000	.8E 00 00 00	.55468 75000	.CE 00 00 00	.80468 75000
.0F 00 00 00	.05859 37500	.4F 00 00 00	.30859 37500	.8F 00 00 00	.55859 37500	.CF 00 00 00	.80859 37500
.10 00 00 00	.06250 00000	.50 00 00 00	.31250 00000	.90 00 00 00	.56250 00000	.D0 00 00 00	.81250 00000
.11 00 00 00	.06640 62500	.51 00 00 00	.31640 62500	.91 00 00 00	.56640 62500	.D1 00 00 00	.81640 62500
.12 00 00 00	.07031 25000	.52 00 00 00	.32031 25000	.92 00 00 00	.57031 25000	.D2 00 00 00	.82031 25000
.13 00 00 00	.07421 87500	.53 00 00 00	.32421 87500	.93 00 00 00	.57421 87500	.D3 00 00 00	.82421 87500
.14 00 00 00	.07812 50000	.54 00 00 00	.32812 50000	.94 00 00 00	.57812 50000	.D4 00 00 00	.82812 50000
.15 00 00 00	.08203 12500	.55 00 00 00	.33203 12500	.95 00 00 00	.58203 12500	.D5 00 00 00	.83203 12500
.16 00 00 00	.08593 75000	.56 00 00 00	.33593 75000	.96 00 00 00	.58593 75000	.D6 00 00 00	.83593 75000
.17 00 00 00	.08984 37500	.57 00 00 00	.33984 37500	.97 00 00 00	.58984 37500	.D7 00 00 00	.83984 37500
.18 00 00 00	.09375 00000	.58 00 00 00	.34375 00000	.98 00 00 00	.59375 00000	.D8 00 00 00	.84375 00000
.19 00 00 00	.09765 62500	.59 00 00 00	.34765 62500	.99 00 00 00	.59765 62500	.D9 00 00 00	.84765 62500
.1A 00 00 00	.10156 25000	.5A 00 00 00	.35156 25000	.9A 00 00 00	.60156 25000	.DA 00 00 00	.85156 25000
.1B 00 00 00	.10546 87500	.5B 00 00 00	.35546 87500	.9B 00 00 00	.60546 87500	.DB 00 00 00	.85546 87500
.1C 00 00 00	.10937 50000	.5C 00 00 00	.35937 50000	.9C 00 00 00	.60937 50000	.DC 00 00 00	.85937 50000
.1D 00 00 00	.11328 12500	.5D 00 00 00	.36328 12500	.9D 00 00 00	.61328 12500	.DD 00 00 00	.86328 12500
.1E 00 00 00	.11718 75000	.5E 00 00 00	.36718 75000	.9E 00 00 00	.61718 75000	.DE 00 00 00	.86718 75000
.1F 00 00 00	.12109 37500	.5F 00 00 00	.37109 37500	.9F 00 00 00	.62109 37500	.DF 00 00 00	.87109 37500
.20 00 00 00	.12500 00000	.60 00 00 00	.37500 00000	.A0 00 00 00	.62500 00000	.E0 00 00 00	.87500 00000
.21 00 00 00	.12890 62500	.61 00 00 00	.37890 62500	.A1 00 00 00	.62890 62500	.E1 00 00 00	.87890 62500
.22 00 00 00	.13281 25000	.62 00 00 00	.38281 25000	.A2 00 00 00	.63281 25000	.E2 00 00 00	.88281 25000
.23 00 00 00	.13671 87500	.63 00 00 00	.38671 87500	.A3 00 00 00	.63671 87500	.E3 00 00 00	.88671 87500
.24 00 00 00	.14062 50000	.64 00 00 00	.39062 50000	.A4 00 00 00	.64062 50000	.E4 00 00 00	.89062 50000
.25 00 00 00	.14453 12500	.65 00 00 00	.39453 12500	.A5 00 00 00	.64453 12500	.E5 00 00 00	.89453 12500
.26 00 00 00	.14843 75000	.66 00 00 00	.39843 75000	.A6 00 00 00	.64843 75000	.E6 00 00 00	.89843 75000
.27 00 00 00	.15234 37500	.67 00 00 00	.40234 37500	.A7 00 00 00	.65234 37500	.E7 00 00 00	.90234 37500
.28 00 00 00	.15625 00000	.68 00 00 00	.40625 00000	.A8 00 00 00	.65625 00000	.E8 00 00 00	.90625 00000
.29 00 00 00	.16015 62500	.69 00 00 00	.41015 62500	.A9 00 00 00	.66015 62500	.E9 00 00 00	.91015 62500
.2A 00 00 00	.16406 25000	.6A 00 00 00	.41406 25000	.AA 00 00 00	.66406 25000	.EA 00 00 00	.91406 25000
.2B 00 00 00	.16796 87500	.6B 00 00 00	.41796 87500	.AB 00 00 00	.66796 87500	.EB 00 00 00	.91796 87500
.2C 00 00 00	.17187 50000	.6C 00 00 00	.42187 50000	.AC 00 00 00	.67187 50000	.EC 00 00 00	.92187 50000
.2D 00 00 00	.17578 12500	.6D 00 00 00	.42578 12500	.AD 00 00 00	.67578 12500	.ED 00 00 00	.92578 12500
.2E 00 00 00	.17968 75000	.6E 00 00 00	.42968 75000	.AE 00 00 00	.67968 75000	.EE 00 00 00	.92968 75000
.2F 00 00 00	.18359 37500	.6F 00 00 00	.43359 37500	.AF 00 00 00	.68359 37500	.EF 00 00 00	.93359 37500
.30 00 00 00	.18750 00000	.70 00 00 00	.43750 00000	.B0 00 00 00	.68750 00000	.F0 00 00 00	.93750 00000
.31 00 00 00	.19140 62500	.71 00 00 00	.44140 62500	.B1 00 00 00	.69140 62500	.F1 00 00 00	.94140 62500
.32 00 00 00	.19531 25000	.72 00 00 00	.44531 25000	.B2 00 00 00	.69531 25000	.F2 00 00 00	.94531 25000
.33 00 00 00	.19921 87500	.73 00 00 00	.44921 87500	.B3 00 00 00	.69921 87500	.F3 00 00 00	.94921 87500
.34 00 00 00	.20312 50000	.74 00 00 00	.45312 50000	.B4 00 00 00	.70312 50000	.F4 00 00 00	.95312 50000
.35 00 00 00	.20703 12500	.75 00 00 00	.45703 12500	.B5 00 00 00	.70703 12500	.F5 00 00 00	.95703 12500
.36 00 00 00	.21093 75000	.76 00 00 00	.46093 75000	.B6 00 00 00	.71093 75000	.F6 00 00 00	.96093 75000
.37 00 00 00	.21484 37500	.77 00 00 00	.46484 37500	.B7 00 00 00	.71484 37500	.F7 00 00 00	.96484 37500
.38 00 00 00	.21875 00000	.78 00 00 00	.46875 00000	.B8 00 00 00	.71875 00000	.F8 00 00 00	.96875 00000
.39 00 00 00	.22265 62500	.79 00 00 00	.47265 62500	.B9 00 00 00	.72265 62500	.F9 00 00 00	.97265 62500
.3A 00 00 00	.22656 25000	.7A 00 00 00	.47656 25000	.BA 00 00 00	.72656 25000	.FA 00 00 00	.97656 25000
.3B 00 00 00	.23046 87500	.7B 00 00 00	.48046 87500	.BB 00 00 00	.73046 87500	.FB 00 00 00	.98046 87500
.3C 00 00 00	.23437 50000	.7C 00 00 00	.48437 50000	.BC 00 00 00	.73437 50000	.FC 00 00 00	.98437 50000
.3D 00 00 00	.23828 12500	.7D 00 00 00	.48828 12500	.BD 00 00 00	.73828 12500	.FD 00 00 00	.98828 12500
.3E 00 00 00	.24218 75000	.7E 00 00 00	.49218 75000	.BE 00 00 00	.74218 75000	.FE 00 00 00	.99218 75000
.3F 00 00 00	.24609 37500	.7F 00 00 00	.49609 37500	.BF 00 00 00	.74609 37500	.FF 00 00 00	.99609 37500

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 40 00 00	.00097 65625	.00 80 00 00	.00195 31250	.00 C0 00 00	.00292 96875
.00 01 00 00	.00001 52587	.00 41 00 00	.00099 18212	.00 81 00 00	.00196 83837	.00 C1 00 00	.00294 49462
.00 02 00 00	.00003 05175	.00 42 00 00	.00100 70800	.00 82 00 00	.00198 36425	.00 C2 00 00	.00296 02050
.00 03 00 00	.00004 57763	.00 43 00 00	.00102 23388	.00 83 00 00	.00199 89013	.00 C3 00 00	.00297 54638
.00 04 00 00	.00006 10351	.00 44 00 00	.00103 75976	.00 84 00 00	.00201 41601	.00 C4 00 00	.00299 07226
.00 05 00 00	.00007 62939	.00 45 00 00	.00105 28564	.00 85 00 00	.00202 94189	.00 C5 00 00	.00300 59814
.00 06 00 00	.00009 15527	.00 46 00 00	.00106 81152	.00 86 00 00	.00204 46777	.00 C6 00 00	.00302 12402
.00 07 00 00	.00010 68115	.00 47 00 00	.00108 33740	.00 87 00 00	.00205 99365	.00 C7 00 00	.00303 64990
.00 08 00 00	.00012 20703	.00 48 00 00	.00109 86328	.00 88 00 00	.00207 51953	.00 C8 00 00	.00305 17578
.00 09 00 00	.00013 73291	.00 49 00 00	.00111 38916	.00 89 00 00	.00209 04541	.00 C9 00 00	.00306 70166
.00 0A 00 00	.00015 25878	.00 4A 00 00	.00112 91503	.00 8A 00 00	.00210 57128	.00 CA 00 00	.00308 22753
.00 0B 00 00	.00016 78466	.00 4B 00 00	.00114 44091	.00 8B 00 00	.00212 09716	.00 CB 00 00	.00309 75341
.00 0C 00 00	.00018 31054	.00 4C 00 00	.00115 96679	.00 8C 00 00	.00213 62304	.00 CC 00 00	.00311 27929
.00 0D 00 00	.00019 83642	.00 4D 00 00	.00117 49267	.00 8D 00 00	.00215 14892	.00 CD 00 00	.00312 80517
.00 0E 00 00	.00021 36230	.00 4E 00 00	.00119 01855	.00 8E 00 00	.00216 67480	.00 CE 00 00	.00314 33105
.00 0F 00 00	.00022 88818	.00 4F 00 00	.00120 54443	.00 8F 00 00	.00218 20068	.00 CF 00 00	.00315 85693
.00 10 00 00	.00024 41406	.00 50 00 00	.00122 07031	.00 90 00 00	.00219 72656	.00 D0 00 00	.00317 38281
.00 11 00 00	.00025 93994	.00 51 00 00	.00123 59619	.00 91 00 00	.00221 25244	.00 D1 00 00	.00318 90869
.00 12 00 00	.00027 46582	.00 52 00 00	.00125 12207	.00 92 00 00	.00222 77832	.00 D2 00 00	.00320 43457
.00 13 00 00	.00028 99169	.00 53 00 00	.00126 64794	.00 93 00 00	.00224 30419	.00 D3 00 00	.00321 96044
.00 14 00 00	.00030 51757	.00 54 00 00	.00128 17382	.00 94 00 00	.00225 83007	.00 D4 00 00	.00323 48632
.00 15 00 00	.00032 04345	.00 55 00 00	.00129 69970	.00 95 00 00	.00227 35595	.00 D5 00 00	.00325 01220
.00 16 00 00	.00033 56933	.00 56 00 00	.00131 22558	.00 96 00 00	.00228 88183	.00 D6 00 00	.00326 53808
.00 17 00 00	.00035 09521	.00 57 00 00	.00132 75146	.00 97 00 00	.00230 40771	.00 D7 00 00	.00328 06396
.00 18 00 00	.00036 62109	.00 58 00 00	.00134 27734	.00 98 00 00	.00231 93359	.00 D8 00 00	.00329 58984
.00 19 00 00	.00038 14697	.00 59 00 00	.00135 80322	.00 99 00 00	.00233 45947	.00 D9 00 00	.00331 11572
.00 1A 00 00	.00039 67285	.00 5A 00 00	.00137 32910	.00 9A 00 00	.00234 98535	.00 DA 00 00	.00332 64160
.00 1B 00 00	.00041 19873	.00 5B 00 00	.00138 85498	.00 9B 00 00	.00236 51123	.00 DB 00 00	.00334 16748
.00 1C 00 00	.00042 72460	.00 5C 00 00	.00140 38085	.00 9C 00 00	.00238 03710	.00 DC 00 00	.00335 69335
.00 1D 00 00	.00044 25048	.00 5D 00 00	.00141 90673	.00 9D 00 00	.00239 56298	.00 DD 00 00	.00337 21923
.00 1E 00 00	.00045 77636	.00 5E 00 00	.00143 43261	.00 9E 00 00	.00241 08886	.00 DE 00 00	.00338 74511
.00 1F 00 00	.00047 30224	.00 5F 00 00	.00144 95849	.00 9F 00 00	.00242 61474	.00 DF 00 00	.00340 27099
.00 20 00 00	.00048 82812	.00 60 00 00	.00146 48437	.00 A0 00 00	.00244 14062	.00 E0 00 00	.00341 79687
.00 21 00 00	.00050 35400	.00 61 00 00	.00148 01025	.00 A1 00 00	.00245 66650	.00 E1 00 00	.00343 32275
.00 22 00 00	.00051 87988	.00 62 00 00	.00149 53613	.00 A2 00 00	.00247 19238	.00 E2 00 00	.00344 84863
.00 23 00 00	.00053 40576	.00 63 00 00	.00151 06201	.00 A3 00 00	.00248 71826	.00 E3 00 00	.00346 37451
.00 24 00 00	.00054 93164	.00 64 00 00	.00152 58789	.00 A4 00 00	.00250 24414	.00 E4 00 00	.00347 90039
.00 25 00 00	.00056 45751	.00 65 00 00	.00154 11376	.00 A5 00 00	.00251 77001	.00 E5 00 00	.00349 42626
.00 26 00 00	.00057 98339	.00 66 00 00	.00155 63964	.00 A6 00 00	.00253 29589	.00 E6 00 00	.00350 95214
.00 27 00 00	.00059 50927	.00 67 00 00	.00157 16552	.00 A7 00 00	.00254 82177	.00 E7 00 00	.00352 47802
.00 28 00 00	.00061 03515	.00 68 00 00	.00158 69140	.00 A8 00 00	.00256 34765	.00 E8 00 00	.00354 00390
.00 29 00 00	.00062 56103	.00 69 00 00	.00160 21728	.00 A9 00 00	.00257 87353	.00 E9 00 00	.00355 52978
.00 2A 00 00	.00064 08691	.00 6A 00 00	.00161 74316	.00 AA 00 00	.00259 39941	.00 EA 00 00	.00357 05566
.00 2B 00 00	.00065 61279	.00 6B 00 00	.00163 26904	.00 AB 00 00	.00260 92529	.00 EB 00 00	.00358 58154
.00 2C 00 00	.00067 13867	.00 6C 00 00	.00164 79492	.00 AC 00 00	.00262 45117	.00 EC 00 00	.00360 10742
.00 2D 00 00	.00068 66455	.00 6D 00 00	.00166 32080	.00 AD 00 00	.00263 97705	.00 ED 00 00	.00361 63330
.00 2E 00 00	.00070 19042	.00 6E 00 00	.00167 84667	.00 AE 00 00	.00265 50292	.00 EE 00 00	.00363 15917
.00 2F 00 00	.00071 71630	.00 6F 00 00	.00169 37255	.00 AF 00 00	.00267 02880	.00 EF 00 00	.00364 68505
.00 30 00 00	.00073 24218	.00 70 00 00	.00170 89843	.00 B0 00 00	.00268 55468	.00 F0 00 00	.00366 21093
.00 31 00 00	.00074 76806	.00 71 00 00	.00172 42431	.00 B1 00 00	.00270 08056	.00 F1 00 00	.00367 73681
.00 32 00 00	.00076 29394	.00 72 00 00	.00173 95019	.00 B2 00 00	.00271 60644	.00 F2 00 00	.00369 26269
.00 33 00 00	.00077 81982	.00 73 00 00	.00175 47607	.00 B3 00 00	.00273 13232	.00 F3 00 00	.00370 78857
.00 34 00 00	.00079 34570	.00 74 00 00	.00177 00195	.00 B4 00 00	.00274 65820	.00 F4 00 00	.00372 31445
.00 35 00 00	.00080 87158	.00 75 00 00	.00178 52783	.00 B5 00 00	.00276 18408	.00 F5 00 00	.00373 84033
.00 36 00 00	.00082 39746	.00 76 00 00	.00180 05371	.00 B6 00 00	.00277 70996	.00 F6 00 00	.00375 36621
.00 37 00 00	.00083 92333	.00 77 00 00	.00181 57958	.00 B7 00 00	.00279 23583	.00 F7 00 00	.00376 89208
.00 38 00 00	.00085 44921	.00 78 00 00	.00183 10546	.00 B8 00 00	.00280 76171	.00 F8 00 00	.00378 41796
.00 39 00 00	.00086 97509	.00 79 00 00	.00184 63134	.00 B9 00 00	.00282 28759	.00 F9 00 00	.00379 94384
.00 3A 00 00	.00088 50097	.00 7A 00 00	.00186 15722	.00 BA 00 00	.00283 81347	.00 FA 00 00	.00381 46972
.00 3B 00 00	.00090 02685	.00 7B 00 00	.00187 68310	.00 BB 00 00	.00285 33935	.00 FB 00 00	.00382 99560
.00 3C 00 00	.00091 55273	.00 7C 00 00	.00189 20898	.00 BC 00 00	.00286 86523	.00 FC 00 00	.00384 52148
.00 3D 00 00	.00093 07861	.00 7D 00 00	.00190 73486	.00 BD 00 00	.00288 39111	.00 FD 00 00	.00386 04736
.00 3E 00 00	.00094 60449	.00 7E 00 00	.00192 26074	.00 BE 00 00	.00289 91699	.00 FE 00 00	.00387 57324
.00 3F 00 00	.00096 13037	.00 7F 00 00	.00193 78662	.00 BF 00 00	.00291 44287	.00 FF 00 00	.00389 09912

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 00 40 00	.00000 38146	.00 00 80 00	.00000 76293	.00 00 C0 00	.00001 14440
.00 00 01 00	.00000 00596	.00 00 41 00	.00000 38743	.00 00 81 00	.00000 76889	.00 00 C1 00	.00001 15036
.00 00 02 00	.00000 01192	.00 00 42 00	.00000 39339	.00 00 82 00	.00000 77486	.00 00 C2 00	.00001 15633
.00 00 03 00	.00000 01788	.00 00 43 00	.00000 39935	.00 00 83 00	.00000 78082	.00 00 C3 00	.00001 16229
.00 00 04 00	.00000 02384	.00 00 44 00	.00000 40531	.00 00 84 00	.00000 78678	.00 00 C4 00	.00001 16825
.00 00 05 00	.00000 02980	.00 00 45 00	.00000 41127	.00 00 85 00	.00000 79274	.00 00 C5 00	.00001 17421
.00 00 06 00	.00000 03576	.00 00 46 00	.00000 41723	.00 00 86 00	.00000 79870	.00 00 C6 00	.00001 18017
.00 00 07 00	.00000 04172	.00 00 47 00	.00000 42319	.00 00 87 00	.00000 80466	.00 00 C7 00	.00001 18613
.00 00 08 00	.00000 04768	.00 00 48 00	.00000 42915	.00 00 88 00	.00000 81062	.00 00 C8 00	.00001 19209
.00 00 09 00	.00000 05364	.00 00 49 00	.00000 43511	.00 00 89 00	.00000 81658	.00 00 C9 00	.00001 19805
.00 00 0A 00	.00000 05960	.00 00 4A 00	.00000 44107	.00 00 8A 00	.00000 82254	.00 00 CA 00	.00001 20401
.00 00 0B 00	.00000 06556	.00 00 4B 00	.00000 44703	.00 00 8B 00	.00000 82850	.00 00 CB 00	.00001 20997
.00 00 0C 00	.00000 07152	.00 00 4C 00	.00000 45299	.00 00 8C 00	.00000 83446	.00 00 CC 00	.00001 21593
.00 00 0D 00	.00000 07748	.00 00 4D 00	.00000 45895	.00 00 8D 00	.00000 84042	.00 00 CD 00	.00001 22189
.00 00 0E 00	.00000 08344	.00 00 4E 00	.00000 46491	.00 00 8E 00	.00000 84638	.00 00 CE 00	.00001 22785
.00 00 0F 00	.00000 08940	.00 00 4F 00	.00000 47087	.00 00 8F 00	.00000 85234	.00 00 CF 00	.00001 23381
.00 00 10 00	.00000 09536	.00 00 50 00	.00000 47683	.00 00 90 00	.00000 85830	.00 00 D0 00	.00001 23977
.00 00 11 00	.00000 10132	.00 00 51 00	.00000 48279	.00 00 91 00	.00000 86426	.00 00 D1 00	.00001 24573
.00 00 12 00	.00000 10728	.00 00 52 00	.00000 48875	.00 00 92 00	.00000 87022	.00 00 D2 00	.00001 25169
.00 00 13 00	.00000 11324	.00 00 53 00	.00000 49471	.00 00 93 00	.00000 87618	.00 00 D3 00	.00001 25765
.00 00 14 00	.00000 11920	.00 00 54 00	.00000 50067	.00 00 94 00	.00000 88214	.00 00 D4 00	.00001 26361
.00 00 15 00	.00000 12516	.00 00 55 00	.00000 50663	.00 00 95 00	.00000 88810	.00 00 D5 00	.00001 26957
.00 00 16 00	.00000 13113	.00 00 56 00	.00000 51259	.00 00 96 00	.00000 89406	.00 00 D6 00	.00001 27553
.00 00 17 00	.00000 13709	.00 00 57 00	.00000 51856	.00 00 97 00	.00000 90003	.00 00 D7 00	.00001 28149
.00 00 18 00	.00000 14305	.00 00 58 00	.00000 52452	.00 00 98 00	.00000 90599	.00 00 D8 00	.00001 28746
.00 00 19 00	.00000 14901	.00 00 59 00	.00000 53048	.00 00 99 00	.00000 91195	.00 00 D9 00	.00001 29342
.00 00 1A 00	.00000 15497	.00 00 5A 00	.00000 53644	.00 00 9A 00	.00000 91791	.00 00 DA 00	.00001 29938
.00 00 1B 00	.00000 16093	.00 00 5B 00	.00000 54240	.00 00 9B 00	.00000 92387	.00 00 DB 00	.00001 30534
.00 00 1C 00	.00000 16689	.00 00 5C 00	.00000 54836	.00 00 9C 00	.00000 92983	.00 00 DC 00	.00001 31130
.00 00 1D 00	.00000 17285	.00 00 5D 00	.00000 55432	.00 00 9D 00	.00000 93579	.00 00 DD 00	.00001 31726
.00 00 1E 00	.00000 17881	.00 00 5E 00	.00000 56028	.00 00 9E 00	.00000 94175	.00 00 DE 00	.00001 32322
.00 00 1F 00	.00000 18477	.00 00 5F 00	.00000 56624	.00 00 9F 00	.00000 94771	.00 00 DF 00	.00001 32918
.00 00 20 00	.00000 19073	.00 00 60 00	.00000 57220	.00 00 A0 00	.00000 95367	.00 00 E0 00	.00001 33514
.00 00 21 00	.00000 19669	.00 00 61 00	.00000 57816	.00 00 A1 00	.00000 95963	.00 00 E1 00	.00001 34110
.00 00 22 00	.00000 20265	.00 00 62 00	.00000 58412	.00 00 A2 00	.00000 96559	.00 00 E2 00	.00001 34706
.00 00 23 00	.00000 20861	.00 00 63 00	.00000 59008	.00 00 A3 00	.00000 97155	.00 00 E3 00	.00001 35302
.00 00 24 00	.00000 21457	.00 00 64 00	.00000 59604	.00 00 A4 00	.00000 97751	.00 00 E4 00	.00001 35898
.00 00 25 00	.00000 22053	.00 00 65 00	.00000 60200	.00 00 A5 00	.00000 98347	.00 00 E5 00	.00001 36494
.00 00 26 00	.00000 22649	.00 00 66 00	.00000 60796	.00 00 A6 00	.00000 98943	.00 00 E6 00	.00001 37090
.00 00 27 00	.00000 23245	.00 00 67 00	.00000 61392	.00 00 A7 00	.00000 99539	.00 00 E7 00	.00001 37686
.00 00 28 00	.00000 23841	.00 00 68 00	.00000 61988	.00 00 A8 00	.00001 00135	.00 00 E8 00	.00001 38282
.00 00 29 00	.00000 24437	.00 00 69 00	.00000 62584	.00 00 A9 00	.00001 00731	.00 00 E9 00	.00001 38878
.00 00 2A 00	.00000 25033	.00 00 6A 00	.00000 63180	.00 00 AA 00	.00001 01327	.00 00 EA 00	.00001 39474
.00 00 2B 00	.00000 25629	.00 00 6B 00	.00000 63776	.00 00 AB 00	.00001 01923	.00 00 EB 00	.00001 40070
.00 00 2C 00	.00000 26226	.00 00 6C 00	.00000 64373	.00 00 AC 00	.00001 02519	.00 00 EC 00	.00001 40666
.00 00 2D 00	.00000 26822	.00 00 6D 00	.00000 64969	.00 00 AD 00	.00001 03116	.00 00 ED 00	.00001 41263
.00 00 2E 00	.00000 27418	.00 00 6E 00	.00000 65565	.00 00 AE 00	.00001 03712	.00 00 EE 00	.00001 41859
.00 00 2F 00	.00000 28014	.00 00 6F 00	.00000 66161	.00 00 AF 00	.00001 04308	.00 00 EF 00	.00001 42455
.00 00 30 00	.00000 28610	.00 00 70 00	.00000 66757	.00 00 B0 00	.00001 04904	.00 00 F0 00	.00001 43051
.00 00 31 00	.00000 29206	.00 00 71 00	.00000 67353	.00 00 B1 00	.00001 05500	.00 00 F1 00	.00001 43647
.00 00 32 00	.00000 29802	.00 00 72 00	.00000 67949	.00 00 B2 00	.00001 06096	.00 00 F2 00	.00001 44243
.00 00 33 00	.00000 30398	.00 00 73 00	.00000 68545	.00 00 B3 00	.00001 06692	.00 00 F3 00	.00001 44839
.00 00 34 00	.00000 30994	.00 00 74 00	.00000 69141	.00 00 B4 00	.00001 07288	.00 00 F4 00	.00001 45435
.00 00 35 00	.00000 31590	.00 00 75 00	.00000 69737	.00 00 B5 00	.00001 07884	.00 00 F5 00	.00001 46031
.00 00 36 00	.00000 32186	.00 00 76 00	.00000 70333	.00 00 B6 00	.00001 08480	.00 00 F6 00	.00001 46627
.00 00 37 00	.00000 32782	.00 00 77 00	.00000 70929	.00 00 B7 00	.00001 09076	.00 00 F7 00	.00001 47223
.00 00 38 00	.00000 33378	.00 00 78 00	.00000 71525	.00 00 B8 00	.00001 09672	.00 00 F8 00	.00001 47819
.00 00 39 00	.00000 33974	.00 00 79 00	.00000 72121	.00 00 B9 00	.00001 10268	.00 00 F9 00	.00001 48415
.00 00 3A 00	.00000 34570	.00 00 7A 00	.00000 72717	.00 00 BA 00	.00001 10864	.00 00 FA 00	.00001 49011
.00 00 3B 00	.00000 35166	.00 00 7B 00	.00000 73313	.00 00 BB 00	.00001 11460	.00 00 FB 00	.00001 49607
.00 00 3C 00	.00000 35762	.00 00 7C 00	.00000 73909	.00 00 BC 00	.00001 12056	.00 00 FC 00	.00001 50203
.00 00 3D 00	.00000 36358	.00 00 7D 00	.00000 74505	.00 00 BD 00	.00001 12652	.00 00 FD 00	.00001 50799
.00 00 3E 00	.00000 36954	.00 00 7E 00	.00000 75101	.00 00 BE 00	.00001 13248	.00 00 FE 00	.00001 51395
.00 00 3F 00	.00000 37550	.00 00 7F 00	.00000 75697	.00 00 BF 00	.00001 13844	.00 00 FF 00	.00001 51991

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 00 00 40	.00000 00149	.00 00 00 80	.00000 00298	.00 00 00 C0	.00000 00447
.00 00 00 01	.00000 00002	.00 00 00 41	.00000 00151	.00 00 00 81	.00000 00300	.00 00 00 C1	.00000 00449
.00 00 00 02	.00000 00004	.00 00 00 42	.00000 00153	.00 00 00 82	.00000 00302	.00 00 00 C2	.00000 00451
.00 00 00 03	.00000 00006	.00 00 00 43	.00000 00155	.00 00 00 83	.00000 00305	.00 00 00 C3	.00000 00454
.00 00 00 04	.00000 00009	.00 00 00 44	.00000 00158	.00 00 00 84	.00000 00307	.00 00 00 C4	.00000 00456
.00 00 00 05	.00000 00011	.00 00 00 45	.00000 00160	.00 00 00 85	.00000 00309	.00 00 00 C5	.00000 00458
.00 00 00 06	.00000 00013	.00 00 00 46	.00000 00162	.00 00 00 86	.00000 00311	.00 00 00 C6	.00000 00461
.00 00 00 07	.00000 00016	.00 00 00 47	.00000 00165	.00 00 00 87	.00000 00314	.00 00 00 C7	.00000 00463
.00 00 00 08	.00000 00018	.00 00 00 48	.00000 00167	.00 00 00 88	.00000 00316	.00 00 00 C8	.00000 00465
.00 00 00 09	.00000 00020	.00 00 00 49	.00000 00169	.00 00 00 89	.00000 00318	.00 00 00 C9	.00000 00467
.00 00 00 0A	.00000 00023	.00 00 00 4A	.00000 00172	.00 00 00 8A	.00000 00321	.00 00 00 CA	.00000 00470
.00 00 00 0B	.00000 00025	.00 00 00 4B	.00000 00174	.00 00 00 8B	.00000 00323	.00 00 00 CB	.00000 00472
.00 00 00 0C	.00000 00027	.00 00 00 4C	.00000 00176	.00 00 00 8C	.00000 00325	.00 00 00 CC	.00000 00474
.00 00 00 0D	.00000 00030	.00 00 00 4D	.00000 00179	.00 00 00 8D	.00000 00328	.00 00 00 CD	.00000 00477
.00 00 00 0E	.00000 00032	.00 00 00 4E	.00000 00181	.00 00 00 8E	.00000 00330	.00 00 00 CE	.00000 00479
.00 00 00 0F	.00000 00034	.00 00 00 4F	.00000 00183	.00 00 00 8F	.00000 00332	.00 00 00 CF	.00000 00481
.00 00 00 10	.00000 00037	.00 00 00 50	.00000 00186	.00 00 00 90	.00000 00335	.00 00 00 D0	.00000 00484
.00 00 00 11	.00000 00039	.00 00 00 51	.00000 00188	.00 00 00 91	.00000 00337	.00 00 00 D1	.00000 00486
.00 00 00 12	.00000 00041	.00 00 00 52	.00000 00190	.00 00 00 92	.00000 00339	.00 00 00 D2	.00000 00488
.00 00 00 13	.00000 00044	.00 00 00 53	.00000 00193	.00 00 00 93	.00000 00342	.00 00 00 D3	.00000 00491
.00 00 00 14	.00000 00046	.00 00 00 54	.00000 00195	.00 00 00 94	.00000 00344	.00 00 00 D4	.00000 00493
.00 00 00 15	.00000 00048	.00 00 00 55	.00000 00197	.00 00 00 95	.00000 00346	.00 00 00 D5	.00000 00495
.00 00 00 16	.00000 00051	.00 00 00 56	.00000 00200	.00 00 00 96	.00000 00349	.00 00 00 D6	.00000 00498
.00 00 00 17	.00000 00053	.00 00 00 57	.00000 00202	.00 00 00 97	.00000 00351	.00 00 00 D7	.00000 00500
.00 00 00 18	.00000 00055	.00 00 00 58	.00000 00204	.00 00 00 98	.00000 00353	.00 00 00 D8	.00000 00502
.00 00 00 19	.00000 00058	.00 00 00 59	.00000 00207	.00 00 00 99	.00000 00356	.00 00 00 D9	.00000 00505
.00 00 00 1A	.00000 00060	.00 00 00 5A	.00000 00209	.00 00 00 9A	.00000 00358	.00 00 00 DA	.00000 00507
.00 00 00 1B	.00000 00062	.00 00 00 5B	.00000 00211	.00 00 00 9B	.00000 00360	.00 00 00 DB	.00000 00509
.00 00 00 1C	.00000 00065	.00 00 00 5C	.00000 00214	.00 00 00 9C	.00000 00363	.00 00 00 DC	.00000 00512
.00 00 00 1D	.00000 00067	.00 00 00 5D	.00000 00216	.00 00 00 9D	.00000 00365	.00 00 00 DD	.00000 00514
.00 00 00 1E	.00000 00069	.00 00 00 5E	.00000 00218	.00 00 00 9E	.00000 00367	.00 00 00 DE	.00000 00516
.00 00 00 1F	.00000 00072	.00 00 00 5F	.00000 00221	.00 00 00 9F	.00000 00370	.00 00 00 DF	.00000 00519
.00 00 00 20	.00000 00074	.00 00 00 60	.00000 00223	.00 00 00 A0	.00000 00372	.00 00 00 E0	.00000 00521
.00 00 00 21	.00000 00076	.00 00 00 61	.00000 00225	.00 00 00 A1	.00000 00374	.00 00 00 E1	.00000 00523
.00 00 00 22	.00000 00079	.00 00 00 62	.00000 00228	.00 00 00 A2	.00000 00377	.00 00 00 E2	.00000 00526
.00 00 00 23	.00000 00081	.00 00 00 63	.00000 00230	.00 00 00 A3	.00000 00379	.00 00 00 E3	.00000 00528
.00 00 00 24	.00000 00083	.00 00 00 64	.00000 00232	.00 00 00 A4	.00000 00381	.00 00 00 E4	.00000 00530
.00 00 00 25	.00000 00086	.00 00 00 65	.00000 00235	.00 00 00 A5	.00000 00384	.00 00 00 E5	.00000 00533
.00 00 00 26	.00000 00088	.00 00 00 66	.00000 00237	.00 00 00 A6	.00000 00386	.00 00 00 E6	.00000 00535
.00 00 00 27	.00000 00090	.00 00 00 67	.00000 00239	.00 00 00 A7	.00000 00388	.00 00 00 E7	.00000 00537
.00 00 00 28	.00000 00093	.00 00 00 68	.00000 00242	.00 00 00 A8	.00000 00391	.00 00 00 E8	.00000 00540
.00 00 00 29	.00000 00095	.00 00 00 69	.00000 00244	.00 00 00 A9	.00000 00393	.00 00 00 E9	.00000 00542
.00 00 00 2A	.00000 00097	.00 00 00 6A	.00000 00246	.00 00 00 AA	.00000 00395	.00 00 00 EA	.00000 00544
.00 00 00 2B	.00000 00100	.00 00 00 6B	.00000 00249	.00 00 00 AB	.00000 00398	.00 00 00 EB	.00000 00547
.00 00 00 2C	.00000 00102	.00 00 00 6C	.00000 00251	.00 00 00 AC	.00000 00400	.00 00 00 EC	.00000 00549
.00 00 00 2D	.00000 00104	.00 00 00 6D	.00000 00253	.00 00 00 AD	.00000 00402	.00 00 00 ED	.00000 00551
.00 00 00 2E	.00000 00107	.00 00 00 6E	.00000 00256	.00 00 00 AE	.00000 00405	.00 00 00 EE	.00000 00554
.00 00 00 2F	.00000 00109	.00 00 00 6F	.00000 00258	.00 00 00 AF	.00000 00407	.00 00 00 EF	.00000 00556
.00 00 00 30	.00000 00111	.00 00 00 70	.00000 00260	.00 00 00 B0	.00000 00409	.00 00 00 F0	.00000 00558
.00 00 00 31	.00000 00114	.00 00 00 71	.00000 00263	.00 00 00 B1	.00000 00412	.00 00 00 F1	.00000 00561
.00 00 00 32	.00000 00116	.00 00 00 72	.00000 00265	.00 00 00 B2	.00000 00414	.00 00 00 F2	.00000 00563
.00 00 00 33	.00000 00118	.00 00 00 73	.00000 00267	.00 00 00 B3	.00000 00416	.00 00 00 F3	.00000 00565
.00 00 00 34	.00000 00121	.00 00 00 74	.00000 00270	.00 00 00 B4	.00000 00419	.00 00 00 F4	.00000 00568
.00 00 00 35	.00000 00123	.00 00 00 75	.00000 00272	.00 00 00 B5	.00000 00421	.00 00 00 F5	.00000 00570
.00 00 00 36	.00000 00125	.00 00 00 76	.00000 00274	.00 00 00 B6	.00000 00423	.00 00 00 F6	.00000 00572
.00 00 00 37	.00000 00128	.00 00 00 77	.00000 00277	.00 00 00 B7	.00000 00426	.00 00 00 F7	.00000 00575
.00 00 00 38	.00000 00130	.00 00 00 78	.00000 00279	.00 00 00 B8	.00000 00428	.00 00 00 F8	.00000 00577
.00 00 00 39	.00000 00132	.00 00 00 79	.00000 00281	.00 00 00 B9	.00000 00430	.00 00 00 F9	.00000 00579
.00 00 00 3A	.00000 00135	.00 00 00 7A	.00000 00284	.00 00 00 BA	.00000 00433	.00 00 00 FA	.00000 00582
.00 00 00 3B	.00000 00137	.00 00 00 7B	.00000 00286	.00 00 00 BB	.00000 00435	.00 00 00 FB	.00000 00584
.00 00 00 3C	.00000 00139	.00 00 00 7C	.00000 00288	.00 00 00 BC	.00000 00437	.00 00 00 FC	.00000 00586
.00 00 00 3D	.00000 00142	.00 00 00 7D	.00000 00291	.00 00 00 BD	.00000 00440	.00 00 00 FD	.00000 00589
.00 00 00 3E	.00000 00144	.00 00 00 7E	.00000 00293	.00 00 00 BE	.00000 00442	.00 00 00 FE	.00000 00591
.00 00 00 3F	.00000 00146	.00 00 00 7F	.00000 00295	.00 00 00 BF	.00000 00444	.00 00 00 FF	.00000 00593

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125

MATHEMATICAL CONSTANTS

Constant	Decimal Value	Hexadecimal Value
π	3.14159 26535 89793	3.243F 6A89
π^{-1}	0.31830 98861 83790	0.517C C1B7
$\sqrt{\pi}$	1.77245 38509 05516	1.C5BF 891C
$\ln \pi$	1.14472 98858 49400	1.250D 048F
e	2.71828 18284 59045	2.B7E1 5163
e^{-1}	0.36787 94411 71442	0.5E2D 58D9
$\sqrt{e}$	1.64872 12707 00128	1.A612 98E2
$\log_{10} e$	0.43429 44819 03252	0.6F2D EC55
$\log_2 e$	1.44269 50408 88963	1.7154 7653
γ	0.57721 56649 01533	0.93C4 67E4
$\ln \gamma$	-0.54953 93129 81645	-0.8CAE 9BC1
$\sqrt{2}$	1.41421 35623 73095	1.6A09 E668
ln 2	0.69314 71805 59945	0.B172 17F8
$\log_{10} 2$	0.30102 99956 63981	0.4D10 4D42
$\sqrt{10}$	3.16227 76601 68379	3.298B 075C
ln 10	2.30258 40929 94046	2.4D76 3777

INDEX

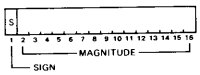
T

U

W

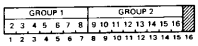
X

DATA WORD FORMAT

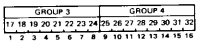


INTERRUPT CONTROL WORD FORMAT

LEVELS 2 THRU 16



LEVELS 17 THRU 32



MEMORY PROTECTION CONTROL

BLOCK Switch Settings	Protected Memory Area (Hex)
000	Entire memory
001	0800 to highest address
010	1000 to highest address
011	1800 to highest address
100	2000 to highest address
101	2800 to highest address
110	3000 to highest address
111	3800 to highest address

RESERVED MEMORY LOCATIONS

Addr (Hex)	Use
0000	Shutdown storage, X-register
0001 thru 0006	Register display A1 thru A6
0007	Roll-arrow register
0008	Upper accumulator
0009 thru 000E	Temporary working storage
000F	Basic interrupt address
0010 thru 0017	Shutdown storage
0018 thru 001F	Standard software
0020 thru 0027	DMA control words
0028 thru 004F	Fill routine
3FC0, 3FC1 thru 3FFE, 3FFF	Priority interrupt level 32 thru Priority interrupt level 1

DMA CONTROL WORD LOCATIONS

Addr (Hex)	Channel	Control Word
20	1	Block starting address
21	1	Block word count
22	2	Block starting address
23	2	Block word count
24	3	Block starting address
25	3	Block word count
26	4	Block starting address
27	4	Block word count

EFFECTIVE ADDRESS COMPUTATION

XIRS	Effective Address	Add'l Time*
0000	M	-
0001	M + (P)	-
0010	M + (N)	-
0011	-M + (N)	-
0100	(M)	0.86 μs
0101	(M + (P))	0.86 μs
0110	(M +(N))	0.86 μs
0111	(-M + (N))	0.86 μs
1000	M + (X)	-
1001	M + (P) + (X)	-
1010	M + (N) + (X)	0.86 μs
1011	-M + (N) + (X)	0.86 μs
1100	(M) + (X)	0.86 μs
1101	(M + (P)) + (X)	0.86 μs
1110	(M + (N)) + (X)	0.86 μs
1111	(-M + (N)) + (X)	0.86 μs

*Add add'l 0.86 μs for double word instruction

STANDARD I/O ADDRESSES

Instruction	Addr (Hex)	Operation
Set	02	Enable 2-ms interrupt
	03	Disable 2-ms interrupt
	E2	Start transfer, DMA 1
	E3	Reset DMA 1
	E4	Start transfer, DMA 2
	E5	Reset DMA 2
	E6	Start transfer, DMA 3
	E7	Reset DMA 3
	E8	Start transfer, DMA 4
	E9	Reset DMA 4
	EA	Enable priority interrupts
	EB	Reset priority interrupt
Sense	00	Test sense switch 8
	01	Test sense switch 1
	02	Test sense switch 2
	03	Test sense switch 3
	04	Test sense switch 4
	05	Test sense switch 5
	06	Test sense switch 6
	07	Test sense switch 7
	0A	Test memory parity
	E2	Chaining or transfer complete, DMA 1
	E4	Chaining or transfer complete, DMA 2
	E6	Chaining or transfer complete, DMA 3
	E8	Chaining or transfer complete, DMA 4
Parallel Input	01	Read data switches
	E2	Read word count, DMA 1
	E4	Read word count, DMA 2
	E6	Read word count, DMA 3
	E8	Read word count, DMA 4
Parallel Output	00	Display accumulator
	E2	Transfer address, DMA 1
	E4	Transfer address, DMA 2
	E6	Transfer address, DMA 3
	E8	Transfer address, DMA 4
	EA	Arm/disarm interrupt levels 2 thru 16
	EB	Arm/disarm interrupt levels 17 thru 32

REFERENCE CARD

RC 70 COMPUTER

REDCOR CORPORATION
7800 Deering Avenue, Canoga Park, California 91304

REDCOR CORPORATION

DATA WORD FORMAT

S
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
MAGNITUDE
SIGN

INTERRUPT CONTROL WORD FORMAT

LEVELS 2 THRU 16

GROUP 1
GROUP 2
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

LEVELS 17 THRU 32

GROUP 3
GROUP 4
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

MEMORY PROTECTION CONTROL

BLOCK Switch Settings	Protected Memory Area (Hex)
000	Entire memory
001	0800 to highest address
010	1000 to highest address
011	1800 to highest address
100	2000 to highest address
101	2800 to highest address
110	3000 to highest address
111	3800 to highest address

RESERVED MEMORY LOCATIONS

Addr (Hex)	Use
0000	Shutdown storage, X-register
0001 thru 0006	Register display A1 thru A6
0007	Roll-arrow register
0008	Upper accumulator
0009 thru 000E	Temporary working storage
000F	Basic interrupt address
0010 thru 0017	Shutdown storage
0018 thru 001F	Standard software
0020 thru 0027	DMA control words
0028 thru 004F	Fill routine
3FC0, 3FC1 thru 3FFE, 3FFF	Priority interrupt level 32 thru Priority interrupt level 1

DMA CONTROL WORD LOCATIONS

Addr (Hex)	Channel	Control Word
20	1	Block starting address
21	1	Block word count
22	2	Block starting address
23	2	Block word count
24	3	Block starting address
25	3	Block word count
26	4	Block starting address
27	4	Block word count

EFFECTIVE ADDRESS COMPUTATION

XIRS	Effective Address	Add'l Time*
0000	M	-
0001	M + (P)	-
0010	M + (N)	-
0011	-M + (N)	-
0100	(M)	0.86 μs
0101	(M + (P))	0.86 μs
0110	(M +(N))	0.86 μs
0111	(-M + (N))	0.86 μs
1000	M + (X)	-
1001	M + (P) + (X)	-
1010	M + (N) + (X)	0.86 μs
1011	-M + (N) + (X)	0.86 μs
1100	(M) + (X)	0.86 μs
1101	(M + (P)) + (X)	0.86 μs
1110	(M + (N)) + (X)	0.86 μs
1111	(-M + (N)) + (X)	0.86 μs

*Add add'l 0.86 μs for double word instruction

STANDARD I/O ADDRESSES

Instruction	Addr (Hex)	Operation
Set	02	Enable 2-ms interrupt
	03	Disable 2-ms interrupt
	E2	Start transfer, DMA 1
	E3	Reset DMA 1
	E4	Start transfer, DMA 2
	E5	Reset DMA 2
	E6	Start transfer, DMA 3
	E7	Reset DMA 3
	E8	Start transfer, DMA 4
	E9	Reset DMA 4
	EA	Enable priority interrupts
	EB	Reset priority interrupt
Sense	00	Test sense switch 8
	01	Test sense switch 1
	02	Test sense switch 2
	03	Test sense switch 3
	04	Test sense switch 4
	05	Test sense switch 5
	06	Test sense switch 6
	07	Test sense switch 7
	0A	Test memory parity
	E2	Chaining or transfer complete, DMA 1
	E4	Chaining or transfer complete, DMA 2
	E6	Chaining or transfer complete, DMA 3
	E8	Chaining or transfer complete, DMA 4
Parallel Input	01	Read data switches
	E2	Read word count, DMA 1
	E4	Read word count, DMA 2
	E6	Read word count, DMA 3
	E8	Read word count, DMA 4
Parallel Output	00	Display accumulator
	E2	Transfer address, DMA 1
	E4	Transfer address, DMA 2
	E6	Transfer address, DMA 3
	E8	Transfer address, DMA 4
	EA	Arm/disarm interrupt levels 2 thru 16
	EB	Arm/disarm interrupt levels 17 thru 32

REFERENCE CARD

RC 70 COMPUTER

REDCOR CORPORATION
7800 Deering Avenue, Canoga Park, California 91304

REDCOR CORPORATION

© 1969, REDCOR Corporation

RC 70 INSTRUCTIONS

Mnemonic	Name	Operation Code* Single Word	Operation Code* Double Word	Execution Time**	Reference Manual Page
ADB	Add	00	D820	1.9	3-2
ALB	Arithmetic Left	D2	DA3A	1.9-4.5	3-10
ANB	And	20	D824	1.9	3-2
ARB	Arithmetic Right	D3	DB3A	1.9-4.5	3-10
BBK	Branch Back	F0	D83E	3.5	3-11
BEQ	Branch Equal	80	D830	1.0	3-2
BGE	Branch Greater Than or Equal	90	D832	1.0	3-3
BGT	Branch Greater Than	B0	D836	1.0	3-3
BLE	Branch Less Than or Equal	88	D831	1.0	3-3
BLK	Branch and Link	E8	D83D	3.6	3-4
BLT	Branch Less Than	A8	D835	1.0	3-4
BNE	Branch Unequal	B1	D837	1.0	3-4
BNO	Branch No Overflow	A0	D834	1.0	3-5
BPT	Branch and Put	F8	D83F	5.5	3-5
BUC	Branch Unconditional	98	D833	1.0	3-5
BXB	Branch Index	E0	D83C	3.0	3-6
CMB	Compare	60	D82C	1.9	3-6
DVB	Divide	50	D82A	11.4	3-7
ELB	End Left	D0	D83A	1.9-7.0	3-10
ELD	Double Length Shift	CA	DA39	3.8-9.9	3-10
HLT	Halt	C9	D939	1.0	3-12
IOR	Inclusive Or	CC	DC39	2.1	3-7
LDB	Load	30	D826	1.9	3-7
LDP	Load Page	CE	DE39	2.1	3-12
LDX	Load Index	C0	D838	2.1	3-8
LRB	Logical Right	D1	D93A	1.9-4.5	3-11
MPB	Multiply	10	D822	6.2	3-8
PIP	Parallel Input	D6	DE3A	2.8	3-12
POP	Parallel Output	D7	DF3A	2.8	3-13
SBB	Subtract	40	D828	1.9	3-8
SET	Set	D4	DC3A	2.8	3-13
SNS	Sense	D5	DD3A	2.8	3-13
STB	Store	70	D82E	1.9	3-8
XMB	Exchange Memory and Accumulator	C8	D839	3.0	3-9
XOR	Exclusive Or	CD	DD39	2.1	3-9

* Operation codes are shown with X, I, R, and S bits equal to zero.
** Time is shown for single word instruction. Add 0.86 μs for double word instruction and 0.86 μs for indirect addressing.

RC 70 INSTRUCTION FORMATS

MNEMONIC	NAME	OP	X I R S	REFERENCE ADDRESS	ID	EOP	IR	OP	X	REFERENCE ADDRESS
ADB	Add	0	XIRS	00-FF	D	8	IR10	0-1		0000-FFFF
ALB	Arithmetic Left	D	2	00-0F	D	A	IR11	A		0000-FFFF
ANB	And	2	XIRS	00-FF	D	8	IR10	4-5		0000-FFFF
ARB	Arithmetic Right	D	3	00-0F	D	B	IR11	A		0000-FFFF
BBK	Branch Back	F	0	-----	D	8	IR11	E		----------
BEQ	Branch Equal	8	0IRS	00-FF	D	8	IR11	0		0000-FFFF
BGE	Branch Greater Than or Equal	9	0IRS	00-FF	D	8	IR11	2		0000-FFFF
BGT	Branch Greater Than	B	0IRS	00-FF	D	8	IR11	6		0000-FFFF
BLE	Branch Less Than Or Equal	8	1IRS	00-FF	D	8	IR11	1		0000-FFFF
BLK	Branch and Link	E	1IRS	00-FF	D	8	IR11	D		0000-FFFF
BLT	Branch Less Than	A	1IRS	00-FF	D	8	IR11	5		0000-FFFF
BNE	Branch Unequal	B	1IRS	00-FF	D	8	IR11	7		0000-FFFF
BNO	Branch No Overflow	A	0IRS	00-FF	D	8	IR11	4		0000-FFFF
BPT	Branch and Put	F	1IRS	00-FF	D	8	IR11	F		0000-FFFF
BUC	Branch Unconditional	9	1IRS	00-FF	D	8	IR11	3		0000-FFFF
BXB	Branch Index	E	0IRS	00-FF	D	8	IR11	C		0000-FFFF
CMB	Compare	6	XIRS	00-FF	D	8	IR10	C-D		0000-FFFF
DVB	Divide	5	XIRS	00-FF	D	8	IR10	A-B		0000-FFFF
ELB	End Left	D	0	00-0F	D	8	IR11	A		0000-FFFF
ELD	Double Length Shift	C	A	00-0F	D	A	IR11	9		0000-FFFF
HLT	Halt	C	9	-----	D	9	IR11	9		----------
IOR	Inclusive Or	C	C	00-FF	D	C	2	9		0000-FFFF
LDB	Load	3	XIRS	00-FF	D	8	IR10	6-7		0000-FFFF
LDP	Load Page	C	E	00-3F	D	E	IR11	9		0000-FFFF
LDX	Load Index	C	0IRS	00-FF	D	8	IR11	8		0000-FFFF
LRB	Logical Right	D	1	00-0F	D	9	IR11	A		0000-FFFF
MPB	Multiply	1	XIRS	00-FF	D	8	IR10	2-3		0000-FFFF
PIP	Parallel Input	D	6	00-3F	D	E	IR11	A		0000-FFFF
POP	Parallel Output	D	7	00-FF	D	F	IR11	A		0000-FFFF
SBB	Subtract	4	XIRS	00-FF	D	8	IR10	8-9		0000-FFFF
SET	Set	D	4	00-FF	D	C	IR11	A		0000-FFFF
SNS	Sense	D	5	00-FF	D	D	IR11	A		0000-FFFF
STB	Store	7	XIRS	00-FF	D	8	IR10	E-F		0000-FFFF
XMB	Exchange Memory and Accumulator	C	8	00-FF	D	8	3	9		0000-FFFF
XOR	Exclusive Or	C	D	00-FF	D	D	3	9		0000-FFFF

OP = Operation Code; X = Index Flag; I = Indirect Address Flag; RS = Relative Address Flags;
ID = Double Word Instruction Identification; EOP = Extended Operation Code

RC 70 INSTRUCTIONS

Mnemonic	Name	Operation Code* Single Word	Operation Code* Double Word	Execution Time**	Reference Manual Page
ADB	Add	00	D820	1.9	3-2
ALB	Arithmetic Left	D2	DA3A	1.9-4.5	3-10
ANB	And	20	D824	1.9	3-2
ARB	Arithmetic Right	D3	DB3A	1.9-4.5	3-10
BBK	Branch Back	F0	D83E	3.5	3-11
BEQ	Branch Equal	80	D830	1.0	3-2
BGE	Branch Greater Than or Equal	90	D832	1.0	3-3
BGT	Branch Greater Than	B0	D836	1.0	3-3
BLE	Branch Less Than or Equal	88	D831	1.0	3-3
BLK	Branch and Link	E8	D83D	3.6	3-4
BLT	Branch Less Than	A8	D835	1.0	3-4
BNE	Branch Unequal	B1	D837	1.0	3-4
BNO	Branch No Overflow	A0	D834	1.0	3-5
BPT	Branch and Put	F8	D83F	5.5	3-5
BUC	Branch Unconditional	98	D833	1.0	3-5
BXB	Branch Index	E0	D83C	3.0	3-6
CMB	Compare	60	D82C	1.9	3-6
DVB	Divide	50	D82A	11.4	3-7
ELB	End Left	D0	D83A	1.9-7.0	3-10
ELD	Double Length Shift	CA	DA39	3.8-9.9	3-10
HLT	Halt	C9	D939	1.0	3-12
IOR	Inclusive Or	CC	DC39	2.1	3-7
LDB	Load	30	D826	1.9	3-7
LDP	Load Page	CE	DE39	2.1	3-12
LDX	Load Index	C0	D838	2.1	3-8
LRB	Logical Right	D1	D93A	1.9-4.5	3-11
MPB	Multiply	10	D822	6.2	3-8
PIP	Parallel Input	D6	DE3A	2.8	3-12
POP	Parallel Output	D7	DF3A	2.8	3-13
SBB	Subtract	40	D828	1.9	3-8
SET	Set	D4	DC3A	2.8	3-13
SNS	Sense	D5	DD3A	2.8	3-13
STB	Store	70	D82E	1.9	3-8
XMB	Exchange Memory and Accumulator	C8	D839	3.0	3-9
XOR	Exclusive Or	CD	DD39	2.1	3-9

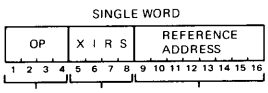
* Operation codes are shown with X, I, R, and S bits equal to zero.
** Time is shown for single word instruction. Add 0.86 μs for double word instruction and 0.86 μs for indirect addressing.

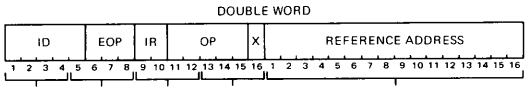
RC 70 INSTRUCTION FORMATS

MNEMONIC	NAME	OP	X I R S	REFERENCE ADDRESS	ID	EOP	IR	OP	X	REFERENCE ADDRESS
ADB	Add	0	XIRS	00-FF	D	8	IR10	0-1		0000-FFFF
ALB	Arithmetic Left	D	2	00-0F	D	A	IR11	A		0000-FFFF
ANB	And	2	XIRS	00-FF	D	8	IR10	4-5		0000-FFFF
ARB	Arithmetic Right	D	3	00-0F	D	B	IR11	A		0000-FFFF
BBK	Branch Back	F	0	-----	D	8	IR11	E		----------
BEQ	Branch Equal	8	0IRS	00-FF	D	8	IR11	0		0000-FFFF
BGE	Branch Greater Than or Equal	9	0IRS	00-FF	D	8	IR11	2		0000-FFFF
BGT	Branch Greater Than	B	0IRS	00-FF	D	8	IR11	6		0000-FFFF
BLE	Branch Less Than Or Equal	8	1IRS	00-FF	D	8	IR11	1		0000-FFFF
BLK	Branch and Link	E	1IRS	00-FF	D	8	IR11	D		0000-FFFF
BLT	Branch Less Than	A	1IRS	00-FF	D	8	IR11	5		0000-FFFF
BNE	Branch Unequal	B	1IRS	00-FF	D	8	IR11	7		0000-FFFF
BNO	Branch No Overflow	A	0IRS	00-FF	D	8	IR11	4		0000-FFFF
BPT	Branch and Put	F	1IRS	00-FF	D	8	IR11	F		0000-FFFF
BUC	Branch Unconditional	9	1IRS	00-FF	D	8	IR11	3		0000-FFFF
BXB	Branch Index	E	0IRS	00-FF	D	8	IR11	C		0000-FFFF
CMB	Compare	6	XIRS	00-FF	D	8	IR10	C-D		0000-FFFF
DVB	Divide	5	XIRS	00-FF	D	8	IR10	A-B		0000-FFFF
ELB	End Left	D	0	00-0F	D	8	IR11	A		0000-FFFF
ELD	Double Length Shift	C	A	00-0F	D	A	IR11	9		0000-FFFF
HLT	Halt	C	9	-----	D	9	IR11	9		----------
IOR	Inclusive Or	C	C	00-FF	D	C	2	9		0000-FFFF
LDB	Load	3	XIRS	00-FF	D	8	IR10	6-7		0000-FFFF
LDP	Load Page	C	E	00-3F	D	E	IR11	9		0000-FFFF
LDX	Load Index	C	0IRS	00-FF	D	8	IR11	8		0000-FFFF
LRB	Logical Right	D	1	00-0F	D	9	IR11	A		0000-FFFF
MPB	Multiply	1	XIRS	00-FF	D	8	IR10	2-3		0000-FFFF
PIP	Parallel Input	D	6	00-FF	D	E	IR11	A		0000-FFFF
POP	Parallel Output	D	7	00-FF	D	F	IR11	A		0000-FFFF
SBB	Subtract	4	XIRS	00-FF	D	8	IR10	8-9		0000-FFFF
SET	Set	D	4	00-FF	D	C	IR11	A		0000-FFFF
SNS	Sense	D	5	00-FF	D	D	IR11	A		0000-FFFF
STB	Store	7	XIRS	00-FF	D	8	IR10	E-F		0000-FFFF
XMB	Exchange Memory and Accumulator	C	8	00-FF	D	8	3	9		0000-FFFF
XOR	Exclusive Or	C	D	00-FF	D	D	3	9		0000-FFFF

OP = Operation Code; X = Index Flag; I = Indirect Address Flag; RS = Relative Address Flags;
ID = Double Word Instruction Identification; EOP = Extended Operation Code

RC 70 INSTRUCTION FORMATS





MNEMONIC	NAME	OP	X I R S	REFERENCE ADDRESS	ID	EOP	IR	OP	X	REFERENCE ADDRESS
ADB	Add	0	XIRS	00-FF	D	8	IR10	0-1		0000-FFFF
ALB	Arithmetic Left	D	2	00-0F	D	A	IR11	A		0000-FFFF
ANB	And	2	XIRS	00-FF	D	8	IR10	4-5		0000-FFFF
ARB	Arithmetic Right	D	3	00-0F	D	B	IR11	A		0000-FFFF
BBK	Branch Back	F	0	-----	D	8	IR11	E		----------
BEQ	Branch Equal	8	0IRS	00-FF	D	8	IR11	0		0000-FFFF
BGE	Branch Greater Than or Equal	9	0IRS	00-FF	D	8	IR11	2		0000-FFFF
BGT	Branch Greater Than	B	0IRS	00-FF	D	8	IR11	6		0000-FFFF
BLE	Branch Less Than Or Equal	8	1IRS	00-FF	D	8	IR11	1		0000-FFFF
BLK	Branch and Link	E	1IRS	00-FF	D	8	IR11	D		0000-FFFF
BLT	Branch Less Than	A	1IRS	00-FF	D	8	IR11	5		0000-FFFF
BNE	Branch Unequal	B	1IRS	00-FF	D	8	IR11	7		0000-FFFF
BNO	Branch No Overflow	A	0IRS	00-FF	D	8	IR11	4		0000-FFFF
BPT	Branch and Put	F	1IRS	00-FF	D	8	IR11	F		0000-FFFF
BUC	Branch Unconditional	9	1IRS	00-FF	D	8	IR11	3		0000-FFFF
BXB	Branch Index	E	0IRS	00-FF	D	8	IR11	C		0000-FFFF
CMB	Compare	6	XIRS	00-FF	D	8	IR10	C-D		0000-FFFF
DVB	Divide	5	XIRS	00-FF	D	8	IR10	A-B		0000-FFFF
ELB	End Left	D	0	00-0F	D	8	IR11	A		0000-FFFF
ELD	Double Length Shift	C	A	00-0F	D	A	IR11	9		0000-FFFF
HLT	Halt	C	9	-----	D	9	IR11	9		----------
IOR	Inclusive Or	C	C	00-FF	D	C	2	9		0000-FFFF
LDB	Load	3	XIRS	00-FF	D	8	IR10	6-7		0000-FFFF
LDP	Load Page	C	E	00-3F	D	E	IR11	9		0000-FFFF
LDX	Load Index	C	0IRS	00-FF	D	8	IR11	8		0000-FFFF
LRB	Logical Right	D	1	00-0F	D	9	IR11	A		0000-FFFF
MPB	Multiply	1	XIRS	00-FF	D	8	IR10	2-3		0000-FFFF
PIP	Parallel Input	D	6	00-FF	D	E	IR11	A		0000-FFFF
POP	Parallel Output	D	7	00-FF	D	F	IR11	A		0000-FFFF
SBB	Subtract	4	XIRS	00-FF	D	8	IR10	8-9		0000-FFFF
SET	Set	D	4	00-FF	D	C	IR11	A		0000-FFFF
SNS	Sense	D	5	00-FF	D	D	IR11	A		0000-FFFF
STB	Store	7	XIRS	00-FF	D	8	IR10	E-F		0000-FFFF
XMB	Exchange Memory and Accumulator	C	8	00-FF	D	8	3	9		0000-FFFF
XOR	Exclusive Or	C	D	00-FF	D	D	3	9		0000-FFFF

OP – Operation Code; X – Index Flag; I – Indirect Address Flag; RS – Relative Address Flags;

ID – Double Word Instruction Identification; EOP – Extended Operation Code