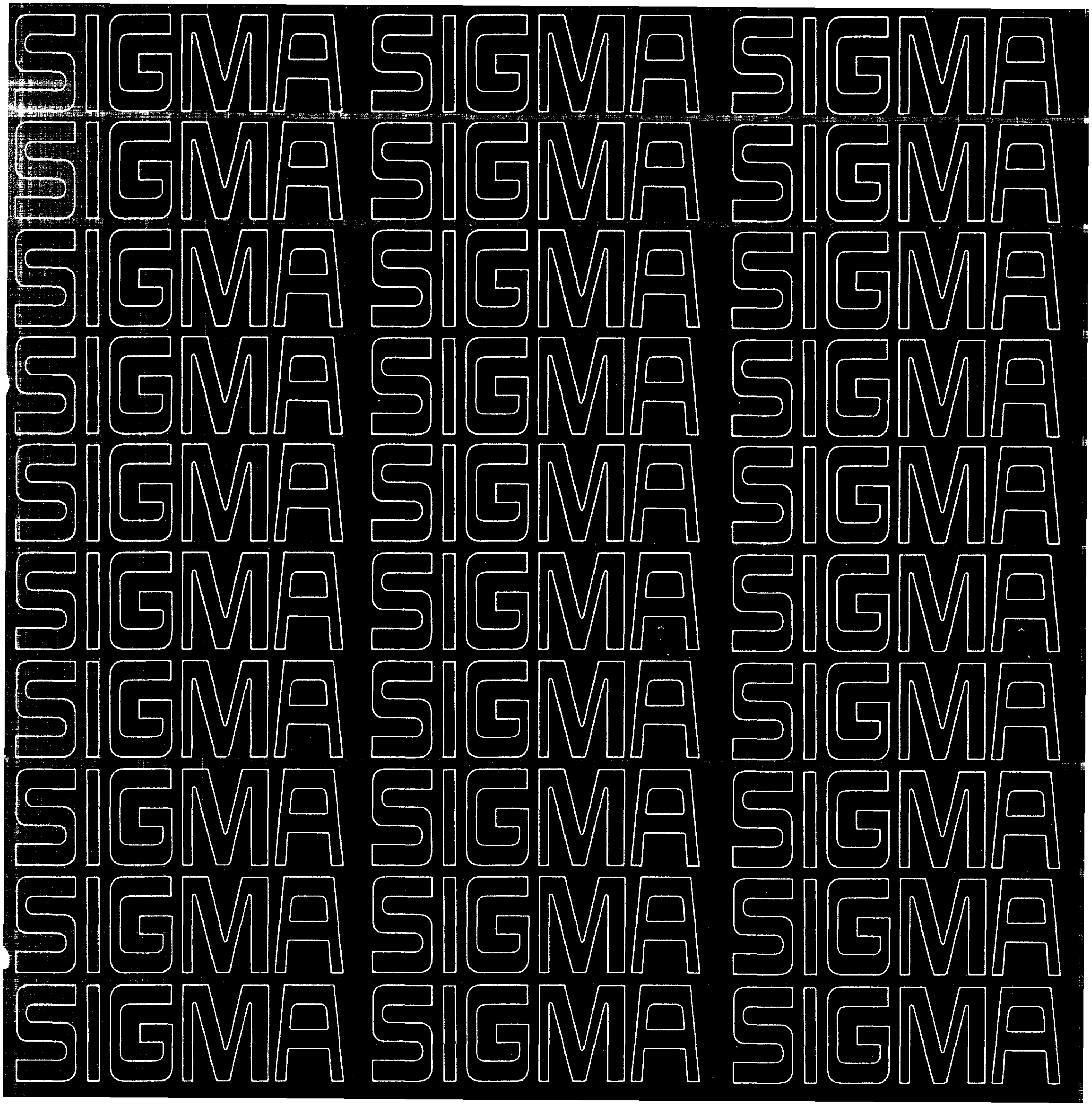
XDS

Xerox Data Systems

XDS SIGMA 3 COMPUTER

Reference Manual



XDS SIGMA 3 INSTRUCTIONS

Instruction name	Mnemonic	Operation code	Page
Add	ADD[†]	1010 RIXS D	16
Logical AND	AND	1001 RIXS D	16
Branch	B	0100 RIXS D	16
Branch if Accumulator Negative	BAN	0110 111S D	18
Branch if Accumulator Zero	BAZ	0110 010S D	18
Branch if Extended Accumulator Negative	BEN	0110 110S D	18
Branch on Incrementing Index	BIX	0110 011S D	18
Branch if No Carry	BNC	0110 001S D	18
Branch if No Overflow	BNO	0110 000S D	18
Branch on Incrementing Index and No Carry	BXNC	0110 101S D	18
Branch on Incrementing Index and No Overflow	BXNO	0110 100S D	18
Compare	CP[†]	1101 RIXS D	17
Divide (optional)	DIV	0101 RIXS D	22
Increment Memory	IM	1111 RIXS D	16
Load Accumulator	LDA[†]	1000 RIXS D	15
Load Index	LDX	1100 RIXS D	15
Multiply (optional)	MUL	0011 RIXS D	21
Read Direct	RD	0001 RIXS D	21
Register Add	RADD	0111 1100 0	19
Register Add and Carry	RADDC	0111 1110 0	20
Register Add and Increment	RADDI	0111 1101 0	20
Register AND	RAND	0111 0000 0	20
Register AND and Carry	RANDC	0111 0010 0	20
Register AND and Increment	RANDI	0111 0001 0	20
Register Copy	RCPY	0111 0100 1	19
Register Copy and Carry	RCPYC	0111 0110 1	20
Register Copy and Increment	RCPYI	0111 0101 1	20
Register Exclusive OR	REOR	0111 1000 0	19
Register Exclusive OR and Carry	REORC	0111 1010 0	20
Register Exclusive OR and Increment	REORI	0111 1001 0	20
Register OR	ROR	0111 0100 0	19
Register OR and Carry	RORC	0111 0110 0	20
Register OR and Increment	RORI	0111 0101 0	20
Shift	S	0010 RIXS D	16
Store Accumulator	STA[†]	1110 RIXS D	15
Subtract	SUB[†]	1011 RIXS D	16
Write Direct	WD	0000 RIXS D	21

[†] Refer also to "Set Multiple Precision Mode Instruction" at end of Chapter 3.

XDS SIGMA 3 OPERATION CODES

Operation code	Mnemonic	Instruction name	Page
0000 RIXS D	WD	Write Direct	21
0001 RIXS D	RD	Read Direct	21
0010 RIXS D	S	Shift	16
0011 RIXS D	MUL	Multiply (optional)	21
0100 RIXS D	B	Branch	16
0101 RIXS D	DIV	Divide (optional)	22
0110 000S D	BNO	Branch if No Overflow	18
0110 001S D	BNC	Branch if No Carry	18
0110 010S D	BAZ	Branch if Accumulator Zero	18
0110 011S D	BIX	Branch on Incrementing Index	18
0110 100S D	BXNO	Branch on Incrementing Index and No Overflow	18
0110 101S D	BXNC	Branch on Incrementing Index and No Carry	18
0110 110S D	BEN	Branch if Extended Accumulator Negative	18
0110 111S D	BAN	Branch if Accumulator Negative	18
0111 0000 0	RAND	Register AND	20
0111 0001 0	RANDI	Register AND and Increment	20
0111 0010 0	RANDC	Register AND and Carry	20
0111 0100 0	ROR	Register OR	19
0111 0100 1	RCPY	Register Copy	19
0111 0101 0	RORI	Register OR and Increment	20
0111 0101 1	RCPYI	Register Copy and Increment	20
0111 0110 0	RORC	Register OR and Carry	20
0111 0110 1	RCPYC	Register Copy and Carry	20
0111 1000 0	REOR	Register Exclusive OR	19
0111 1001 0	REORI	Register Exclusive OR and Increment	20
0111 1010 0	REORC	Register Exclusive OR and Carry	20
0111 1100 0	RADD	Register Add	19
0111 1101 0	RADDI	Register Add and Increment	20
0111 1110 0	RADDC	Register Add and Carry	20
1000 RIXS D	LDA†	Load Accumulator	15
1001 RIXS D	AND	Logical AND	16
1010 RIXS D	ADD†	Add	16
1011 RIXS D	SUB†	Subtract	16
1100 RIXS D	LDX	Load Index	15
1101 RIXS D	CP†	Compare	17
1110 RIXS D	STA†	Store Accumulator	15
1111 RIXS D	IM	Increment Memory	16

†Refer also to "Set Multiple Precision Mode Instruction" at end of Chapter 3.

Price: $2.50

XDS SIGMA 3 COMPUTER REFERENCE MANUAL

90 15 92B

February 1970

XDS

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

©1969, 1970, Xerox Data Systems, Inc.

Printed in U.S.A.

REVISION

This publication, XDS 90 15 92B, is a revision of the XDS Sigma 3 Computer Reference Manual, XDS 90 15 92A (dated August 1969). A change in the text from that of the previous manual is indicated by a vertical line in the margin of the page.

RELATED PUBLICATIONS

Title	Publication No.
XDS Sigma 2/3 Symbol Reference Manual	90 10 51
XDS Sigma 2/3 Extended Symbol Reference Manual	90 10 52
XDS Sigma 2/3 Basic Control Monitor Reference Manual	90 10 64
XDS Sigma 2/3 Real-Time Batch Monitor Reference Manual	90 10 37
XDS Sigma Interface Design Manual	90 09 73

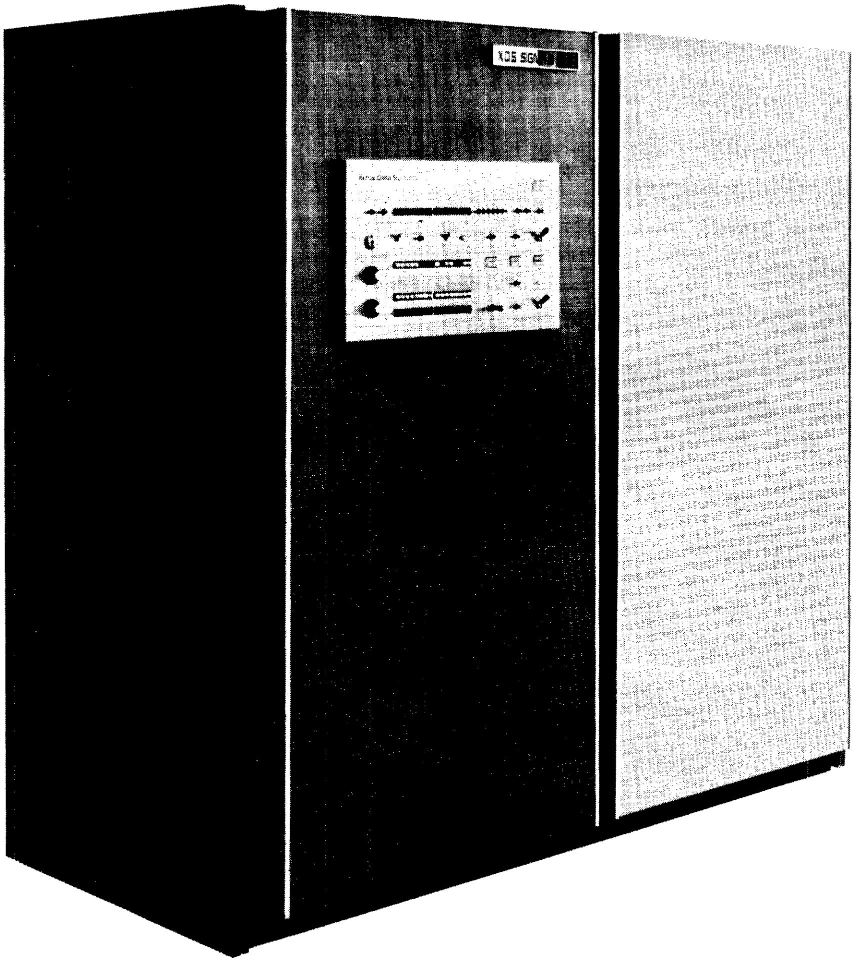
ALL SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

CONTENTS

APPENDIXES

TABLES

ILLUSTRATIONS



SIGMA 3 Computer

1. SYSTEM DESIGN FEATURES

SIGMA 3 is a totally integrated combination of high performance hardware and efficient software. The SIGMA 3 system makes full use of advanced design features first developed for SIGMA 7, and it provides the user with a balanced system that offers advantages normally found only in large computer systems.

Large Capacity, Low-Cost Input/Output. A SIGMA 3 CPU may have either an Internal Input/Output Processor (IIOP), an External Input/Output Processor (EIOP), or both.

The IIOP is a low-cost, medium speed IOP that shares the CPU memory bus. The basic IIOP contains four Input/Output channels. An additional eight channels may be added at low cost. The maximum transfer rate of the IIOP is in excess of 450,000 8-bit bytes per second.

The EIOP is a high performance IOP that has its own registers and memory bus to minimize interference with CPU computation. If the SIGMA 3 CPU and EIOP are accessing the same memory bank at the same time, CPU computation will be delayed by, at most, one memory cycle for each EIOP access to memory. If the CPU and EIOP are accessing different memory banks, there will be no interference between the two. The basic EIOP contains eight Input/Output channels. An additional eight channels may be added at low cost. The maximum transfer rate of the EIOP is in excess of 500,000 8-bit bytes per second. Using the optional two-byte interface, the maximum EIOP transfer rate is in excess of 850,000 bytes per second

Concurrent Foreground/Background Processing. This multiprogramming capability permits the user to operate one or more fully-protected, real-time programs in the foreground while concurrently operating a general-purpose program in the background. Overhead in switching from one task to another is minimized because both hardware and software are specifically designed for rapid context switching. A hardware register permits the software to generate reentrant code efficiently. Thus, routines common to several programs, whether in foreground or background, need be stored in memory only once.

Comprehensive, User-Oriented Software. SIGMA 3 programming systems increase user productivity by providing powerful, easy-to-use programming tools. As a result, user programs are written more quickly at lower cost. The availability of this comprehensive software package makes it possible to exploit the full potential of the hardware. The package includes two operating systems (Monitors), a FORTRAN compiler, two assemblers, and a variety of library and utility programs. To store these extensive software systems yet keep core memory costs at a minimum, XDS has developed its Rapid Access Data (RAD) files. RAD units offer the large capacity and low cost of ordinary disc files. In addition, by using one fixed read/write head for every track of data rather than sharing a movable head among a large group of tracks, the RAD eliminates the access delays associated with head movement. The RAD's fast access time and high data transfer rates produce greater overall system throughput. For basic computer configurations that do not have a RAD unit, a comprehensive group of stand-alone programming systems is provided. For use with larger computer configurations, SIGMA 3 programming systems are RAD-oriented to capitalize on the inherent benefits of this high-performance secondary storage.

Powerful, Multilevel Priority Interrupt System. The real-time oriented SIGMA 3 system provides for quick response to environmental conditions with up to 100 external interrupt levels. The source of each interrupt signal is automatically identified and responded to according to its priority. For further system flexibility, each interrupt level can be individually disarmed (so it stops accepting inputs) and/or disabled (so response is deferred), all under program control. Use of the arm/disarm, enable/disable features makes programmed dynamic reassignment of priorities quick and convenient, even while a real-time process is occurring. In establishing a configuration for any system, each group of 16 interrupt levels can have its group priority assigned differently, to meet the specific needs of an application. The way interrupt levels are programmed is not affected by their priority assignments. The interrupts also can be triggered under program control, allowing hardware queueing of software subroutines or the checkout of real-time software prior to functional interface hardware checkout.

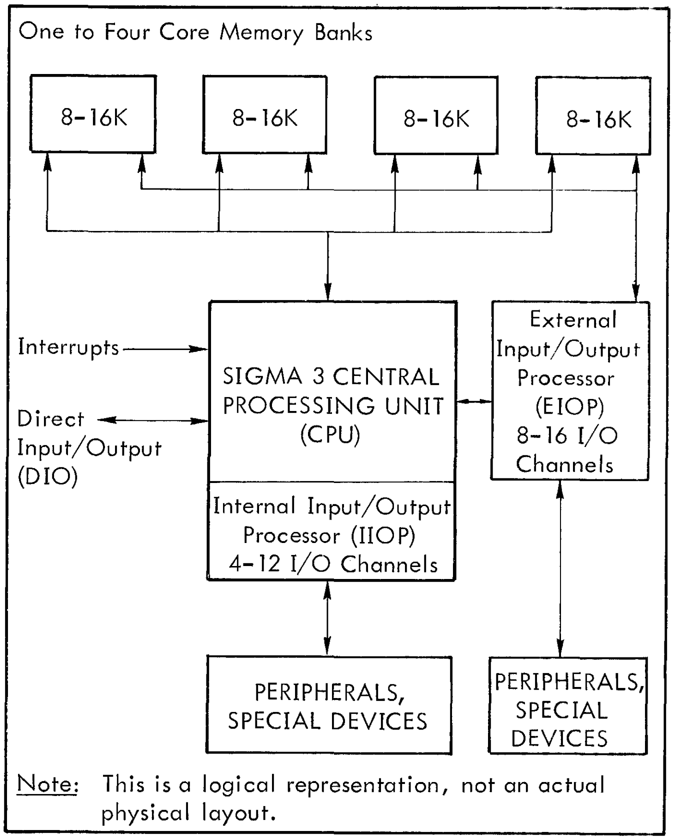

One to Four Core Memory Banks
8-16K
8-16K
8-16K
8-16K
Interrupts
Direct Input/Output (DIO)
SIGMA 3 CENTRAL PROCESSING UNIT (CPU)
Internal Input/Output Processor (IIOP) 4-12 I/O Channels
External Input/Output Processor (EIOP) 8-16 I/O Channels
PERIPHERALS, SPECIAL DEVICES
PERIPHERALS, SPECIAL DEVICES
Note: This is a logical representation, not an actual physical layout.

Figure 1. SIGMA 3 System Configuration

GENERAL CHARACTERISTICS

In its field, the SIGMA 3 computer is unique in its ability to function efficiently in general-purpose, real-time, and multiusage computing environments. The advanced features and operating characteristics contributing to this capability are:

- Both word and byte organization of memory for maximum efficiency. (Words are 16 bits plus parity; bytes are 8 bits.)
- Eight memory sizes available, 8192 to 65,536 words.
- Ability to connect to SIGMA 5 or SIGMA 7 memory systems.
- An extensive instruction set that facilitates efficient programming; SIGMA 3 instruction characteristics include:
  - Only one word of storage required for each instruction.
  - Two levels of indexing and one level of indirect addressing may be invoked individually or simultaneously.
  - Relative addressing (forward and backward).
  - Use of index register 2 as a base address register.
  - Direct reference of up to 1024 addresses; 256 addresses beginning with location zero, 256 addresses beginning with the base address, 256 addresses beginning with the current instruction location (relative forward), 256 addresses backward from the current instruction (relative backward).
- Eight general-purpose registers to control program operations; all are available to the program. They provide:
  - Two hardware index registers for preindexing (base address), post-indexing, or both (double indexing).
  - Hardware register for subroutine linkages.
  - Double precision accumulator.
  - Program address register.
  - Zero register (for a source of zeros).
  - Temporary storage register.
- Rapid context switching, to preserve computer environment when switching from one program to another, including automatic status preservation on interrupt.
- Both word- and byte-oriented I/O systems, for maximum flexibility.
- Up to 28 fully automatic I/O channels operating simultaneously.
- I/O data chaining, for scatter-read and gather-write operations.
- Information transfer rate of approximately 850,000 words per second for each external memory interface.
- Direct input/output of a full word without the use of an I/O channel (optional).
- A real-time priority interrupt system that features:
  - Two to twelve internal interrupt levels and up to 100 external interrupt levels. All of the external and most of the internal levels can be individually armed, enabled, and triggered by program control.
  - Automatic identification, customer-designated priority assignments, and extremely fast response time.
  - Machine fault interrupt (optional).
  - An optional power fail-safe feature, for automatic and safe shutdown in the event of a power failure, and unattended startup when power returns.
  - An optional system protect feature that includes both memory write protection and operation protection for foreground programs.
  - Up to four real-time clocks (with a choice of resolutions) for independent time bases, available as an option.
- A comprehensive array of modular software that expands in capability and speed as the system grows, with no reprogramming required.
  - Free-standing software for small systems includes Symbol and XDS Basic FORTRAN/Basic FORTRAN IV.
  - RBM Monitor for user convenience and increased capability in large systems.
  - Symbol, a basic symbolic assembler.
  - Extended Symbol for expanded features.
  - General loading programs.
  - Utility and RAD editor programs.
  - General Debug for symbolic program troubleshooting.
  - Concordance program for documentation.
  - System Generation program for creating installation master.
  - Mathmetics Library of standard functions.
  - Bootstrap Generator for producing self-loading object programs.

The wide range of standard and special-purpose peripheral equipment, already proven in field operation, includes:

- Rapid-Access Data Files: Capacities for 750,000 to 23,000,000 bytes per control unit, transfer rate of over 170,000 bytes/second, average access time of 17 milliseconds. Fixed read/write head per track eliminates positioning time associated with movable-arm storage devices.
- Removable Disc Storage: Capacities from 49 million to 196 million bytes. Transfer rate of 312,000 bytes per second. Average access time: 87.5 milliseconds.

- Magnetic Tape Units: 9-track, IBM-Compatible, 60,000 or 120,000 bytes per second transfer rate; 7-track, IBM-compatible, 20,000 or 60,000 characters per second transfer rates.

- Paper Tape Readers and Punches: readers with speeds of 20 and 300 characters per second, punches with speeds of 10 and 120 characters per second, plus spoolers.

- Keyboard Printers: available with or without a paper tape reader and paper tape punch.

- Card Readers: read cards punched in binary or EBCDIC card code, 200 to 1500 cards per minute.

- Card Punches: binary or EBCDIC card codes, 100 to 300 cards per minute.

- Line Printers: fully buffered, with 132 print positions and carriage control, 225 to 1000 lines per minute.

- Graph Plotter: for two-axis plotting of data under digital control, 300 increments per second.

- Display Equipment: oscilloscope display units, light guns, and character and vector generators.

- Data Communications Equipment: a complete line of character- and message-oriented equipment.

REAL-TIME AND MULTIUSAGE FEATURES

Real-Time applications are characterized by a need for hardware that provides quick response to an external environment, sufficient speed to keep up with the real-time process itself, and input/output flexibility to handle a wide variety of types of data at varying speeds.

Multiusage applications, as implemented in SIGMA 3, are defined as the combining of real-time and background processing techniques into one system. The most difficult general computing problem is the real-time application with its requirements for extreme speed and capacity. Because the SIGMA 3 system has been designed on a real-time base, it is well qualified for the mixture of applications in a multiusage environment. Many of its hardware features that prove valuable for real-time applications are equally useful in background processing, but in different ways.

The major features that make SIGMA 3 uniquely suitable for both real-time and multiusage applications are described in the following paragraphs.

Input/Output Facilities. Three distinct SIGMA 3 input/output systems offer flexibility and capacity to meet the needs of both real-time and general purpose users: the dual byte-oriented system and the direct-to-CPU (DIO) system.

Both the Internal Input/Output Processor (IIOP) and the External Input/Output Processor (EIOP) use the byte-oriented system.

In the byte-oriented I/O system, each automatic I/O channel has its own high-speed registers and operates independently without requiring attention from the program once it has been started. Data is transferred one byte (8 bits) at a time. For high-speed peripherals, bytes are assembled into words in the I/O section and only one memory reference is made for two bytes. For slow-speed peripherals, one reference is made for every byte. All I/O channels may operate concurrently, and parity checking is performed automatically. The optional direct-to-CPU input/output (DIO) system uses only a single instruction to transfer a full 16-bit data word to and from the A register. The same instruction that transfers data also provides a 16-bit control field for external control and selection, and accepts status information returned from the external device to permit rapid sensing of an external condition. The DIO system is generally used for short bursts of asynchronous data transfers to avoid tying up an automatic channel. DIO is also useful when data is to be accepted at medium to high speeds and each input must be examined immediately when received. The direct-to-memory facility allows a real-time user to interface a custom designed piece of equipment directly to a memory bus. Input/output can then be accomodated at approximately one million words per second. In a system with more than one memory bank, this transfer rate can be sustained without interference to computation.

Priority Interrupt System. In a multiusage environment, many elements are operating asynchronously with respect to each other. Thus, having a true priority interrupt system, as the SIGMA 3 does, is especially important. With it the computer system can respond quickly (and in proper order) to the many demands made upon it, without the high overhead cost of complicated programming, lengthy execution time, and extensive storage allocations. Programs that deal with interrupt signals from special equipment must sometimes be checked out before the equipment is actually available. To simulate special equipment, any external SIGMA 3 interrupt level can be triggered by the CPU itself through execution of a single instruction.

Context Switching. When responding to a new set of interrupt-initiated circumstances, a computer system must preserve the current operating environment while it sets up the new environment. In SIGMA 3, relevant information about the current environment is retained as a 32-bit program status doubleword (PSD). When an interrupt occurs, the current PSD is automatically stored at an arbitrary location in memory and the interrupt-servicing routine begins, following the location into which the PSD is stored. At the end of the interrupt-servicing routine, the PSD is restored and the interrupt level cleared.

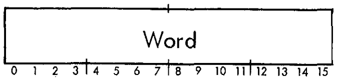
Protection System. Both real-time and background programs can be run concurrently in a SIGMA 3 system because the real-time program can be protected against alteration. The optional protect feature guarantees that protected areas of memory cannot be written into by a program residing in unprotected memory. The protect feature also prevents the execution of unprotected instructions that could change the I/O system or the protection system. The protection pattern can be changed very rapidly.

Real-Time Clocks. In real-time systems, timing information must be provided to cause certain operations to occur at specific instants. Other timing information is also necessary, such as elapsed time after a given event, or the current time of day. SIGMA 3 provides up to four real-time clocks, with varying degrees of resolution, to meet these needs. These clocks also facilitate handling of separate time bases and relative time priorities. Three of the clock counters can be driven from commercial ac line frequency (60 or 50 Hz), from 2- or 8-KHz oscillators, or from an external input. The first counter is normally connected to a 500-Hz clock.

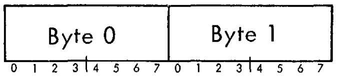
2. SYSTEM ORGANIZATION

INFORMATION FORMAT

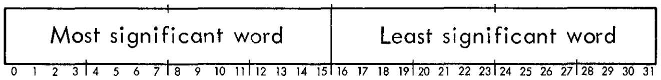
The basic element of SIGMA 3 information is a 16-bit word in which the bit positions are numbered from 0 through 15, as follows:



A SIGMA 3 word can also be divided into two 8-bit parts (called bytes) in which the bit positions of each byte are numbered from 0 through 7, as follows:



Two SIGMA 3 words can be combined to form a 32-bit element (called a doubleword) in which the bit positions are numbered from 0 through 31, as follows:



A doubleword is always referred to by the address of its most significant word.

Binary information in SIGMA 3 computers is generally expressed in hexadecimal notation because four binary digits of information can be expressed by a single hexadecimal digit. Thus, a byte can be expressed with a string of 2 hexadecimal digits, a word with a string of 4 hexadecimal digits, and a doubleword with a string of 8 hexadecimal digits. The following table lists hexadecimal digits and their binary and decimal equivalents.

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

In this reference manual, a hexadecimal number is displayed as a string of hexadecimal digits surrounded by single quotes and preceded by the letter "X". For example, the binary number 01011010 is expressed in hexadecimal notation as X'5A'. Hexadecimal numbers are generally used to denote addresses and data values; however, there are many instances in which decimal numbers are more meaningful or are customary. Because the SIGMA assembler systems perform decimal/hexadecimal conversions, addresses and data values may be expressed as decimal numbers.

In SIGMA 3, fixed-point data is two's complement. It consists of a 15-bit integer and a sign in bit position zero. All arithmetic operations assume that this format is used. Logical operations in SIGMA 3, on the other hand, assume that a logical data word format, consisting of 16 bits without sign, is used.

CORE MEMORIES

A SIGMA 3 computer can be equipped with up to four Basic Memory Units (BMUs). Each BMU contains either 8K or 16K. Therefore the maximum memory size is 64K. (K = 1024 words).

One of the BMUs may be replaced with an adaptor which provides a direct access path to Sigma 5/7 memory. This adaptor allows the SIGMA 3 computer to treat each 32-bit word of Sigma 5/7 memory as two 16-bit words. Special registers allow the programmer to establish a correspondence between any 8192-word portion of SIGMA 3 memory addresses and any 4096-word (32-bit word) portion of Sigma 5/7 memory addresses. A SIGMA 3 memory system may be composed of all SIGMA 3 memory, all Sigma 5/7 memory, and many combinations of the two. However, the memory system must appear to a CPU to be a contiguous block of addresses starting at zero. The Sigma 5/7 memory must be at the upper end of the address field.

A SIGMA 3 memory system has up to four independent access paths (ports), one standard and three optional. Each CPU requires one port and each EIOP requires one port. Therefore a multiprocessor system may be configured where two CPUs and their associated EIOPs all have access to the same memory. Alternately, a memory system may be configured which has attached to it a CPU, an EIOP, and one or two special customer's devices. Each SIGMA 3 BMU is capable of independent operation. Therefore, in a memory system containing four ports and four BMUs, it is possible for four memory accesses to be taking place simultaneously, assuming each port is addressing a different bank of SIGMA 3 memory.

When the SIGMA 3 memory system is 64K words, the memory is "wraparound", or "circular", where the next location after 64K-1 is location 0. If a system has less than 64K words, any fetch operation from a nonexistent storage location causes zeros to be fetched, in which case a memory parity error would occur. An attempt to store information in a nonexistent storage location essentially results in a "no operation".

CENTRAL PROCESSING UNIT AND EIOP

The various elements in a SIGMA 3 system—memories, input/output processors, and device controllers — are organized around a central processing unit (CPU), which is the primary controlling element for most system functions. Not only does the CPU execute instructions, but it also directly controls the byte-oriented IIOP, the direct I/O system, and initializes the EIOP. Basically, the SIGMA 3 CPU consists of a register block and an arithmetic and control unit (see Figure 2).

REGISTER BLOCK

The CPU register block consists of high-speed, integrated-circuit registers that are capable of communicating with the arithmetic and control unit simultaneous with the operation of the core memory. The register block is functionally divided into three parts: general registers, I/O channel registers, and memory protection system registers. Each register of the block is 16 bits in length and is identified by an address code in the range 0 through 7 for general registers, 8 through 63 for I/O channel registers, and 0 through 15 for protection system registers. Specific configurations of the READ DIRECT and WRITE DIRECT instructions are used to transfer information from the accumulator (general register 7) to other registers of the register block, and vice versa (see Chapter 3, "Direct Control Instructions").

General Registers

Eight registers of the register block are used mainly for storage of program control information. These registers are addressable by a COPY instruction (for register-to-register operations) and by certain configurations of the READ DIRECT and WRITE DIRECT instructions (for internal computer control operations). The functions of the general registers are as follows:

Address	Designation	Function
0	Z	Zero Source
1	P	Program address
2	L	Link address
3	T	Temporary storage
4	X1	Index 1 (post-index)
5	X2	Index 2 (pre-index or base)
6	E	Extended accumulator
7	A	Accumulator

A reference to the Z register as a source in a COPY instruction produces a value of zero. The P register contains the address of the next instruction which would be executed in normal sequence. The six remaining registers can be used for various purposes by a program.

I/O Channel Registers

The next eight registers or the register block are used to hold control information for the four basic IIOP SIGMA 3 I/O channels (two registers are used for each channel). Additional I/O channels can be added, in groups of eight (up to a maximum of 28 I/O channels, 56 registers). The I/O channel registers are loaded with control information from the accumulator by a specific configuration of the WRITE DIRECT instruction. The operation of I/O channel registers is described in Chapter 4, "I/O Control Doublewords".

Protection System Registers

Sixteen optional registers are available for both operation protection and memory write protection. Each bit in this 16-register group provides protection for a single 256-word "page" of core memory (see "Protection System" at the end of this chapter).

ARITHMETIC AND CONTROL UNIT

The arithmetic and control unit contains the necessary registers and control circuitry to access general registers or core memory, to modify instruction addresses, to perform arithmetic and logical operations, to provide indications of computational results, and to preserve interrupt status information. Basically, the arithmetic and control unit consists of arithmetic and control registers and program status indicators.

Arithmetic and Control Registers

Three 16-bit registers (S, H, and D) and an adder are used to perform arithmetic and logical manipulations and to modify instruction addresses (see "Effective Address Computation").

Program Status Doubleword

When an interrupt occurs, the current state of the operating program is saved by the automatic storing of a program status doubleword (PSD), which is generated automatically from information in the program status indicators and general registers. When stored in memory, the PSD has the format

0	0	PP	0	II	EI	0	0	O	C	Program address
0 1 2 3	4 5 6 7	8	9	10	11	12	13	14	15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

The first word of the PSD contains five status indicators: protected program (PP), internal interrupt inhibit (II), external interrupt inhibit (EI), overflow (O), and carry (C). The second word of the PSD is the current contents of the program address register (general register 1). (Use of the PSD in interrupt entry and exit is discussed later in this chapter.) If the protect option is installed, the protected program indicator bit is a 1 if the current program is located in an area of core memory that is protected by the memory protection option; otherwise, it is a 0. If the protect option is not installed (PP) is always 1 if the PROTECT switch is in the OFF position.

The internal and external interrupt inhibits determine whether a program interruption can occur. If an interrupt inhibit is 0, the respective interrupt levels are allowed to interrupt the program being executed. Conversely, if an interrupt inhibit is a 1, the respective interrupt levels are inhibited from interrupting the program. Inhibiting interrupt levels also removes them from the interrupt system priority chain, allowing a lower-priority interrupt level to interrupt the program. (Note, however, that the optional override group of internal interrupt levels cannot be inhibited.)

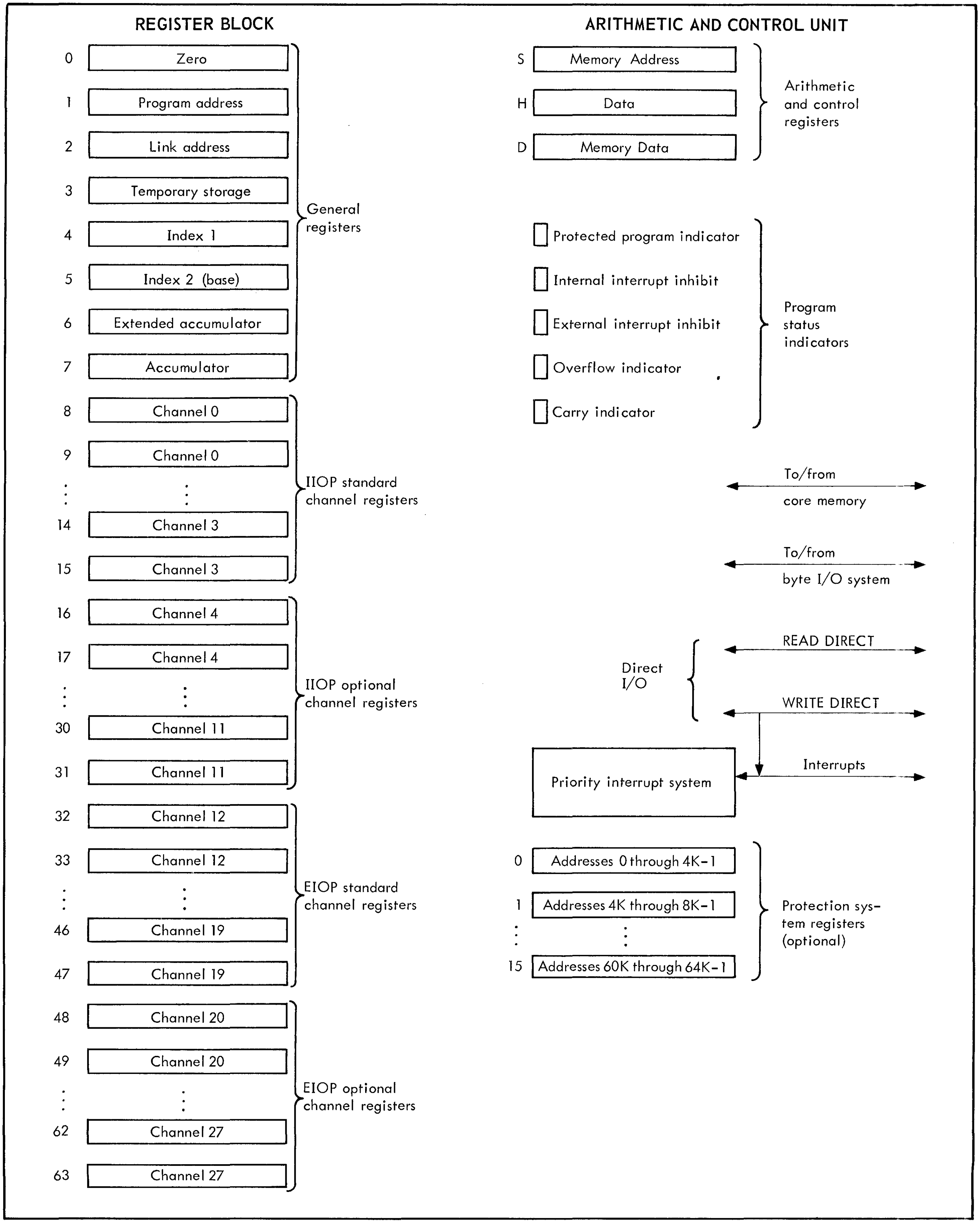


Figure 2. SIGMA 3 Central Processing Unit

The overflow and carry indicators reflect the results of various operations. The overflow indicator is set to 1 if overflow occurs during an arithmetic operation. If, after an arithmetic operation, there is a carry from the most significant position (the sign position) of the adder, the carry indicator is set to 1. Also, on a subtract operation, the carry indicator will be set to 1 if there is a "borrow" from the sign position of the adder. In arithmetic operations, the carry and overflow indicators operate as described above. Some instructions, however, use these indicators to record status information generated as a result of the operation.

INSTRUCTION FORMAT

Most instructions in SIGMA 3 are of the memory reference type and have the following format:

OP	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

In this format, the operation code (OP) occupies the four most significant bits, followed by four address-control bits (R, I, X, and S) and an 8-bit displacement. The R, I, X and S bits control self-relative/nonrelative/base-relative addressing, indirect addressing, and indexing.

Two groups of SIGMA 3 instructions have formats somewhat different from the format of the memory reference type of instructions. The formats of the copy instruction (for register-to-register operations) and the conditional branch instructions (which always invoke self-relative addressing) are described in Chapter 3 (see "Copy Instruction" and "Conditional Branch Instructions").

EFFECTIVE ADDRESS COMPUTATION

The SIGMA 3 computer forms the effective address of a memory reference instruction in three basic steps as follows:

Step 1 (determine reference address)

a. If the R bit (bit 4 of the instruction word) and the S bit (bit 7 of the instruction word) are both 0's, the reference address is equal to the value in the displacement field of the instruction. (This is referred to as "nonrelative" addressing.)

b. If the R bit is a 0 and the S bit is a 1, the reference address is equal to the value in the displacement field in the instruction plus the 16-bit value (base address) in index register 2. (This is referred to as "pre-indexing", or "base-relative" addressing.)

c. If the R bit is a 1, the reference address is equal to the 16-bit value in the H register (address of the instruction) plus the value in the low-order 9 bits of the instruction, interpreted as a 9-bit two's complement integer (this is referred to as "self-relative" addressing).

Step 2 (determine direct address)

a. If the I bit (bit 5 of the instruction word) is a 0, the direct address is equal to the value of the reference address (as determined in step 1).

b. If the I bit is a 1, the reference address is treated as an indirect address; the direct address is the 16-bit value in the location whose address is equal to the reference address. In effect, the indirect address is replaced by the direct address value.

Step 3 (determine effective address)

a. If the X bit (bit 6 of the instruction word) is a 0, the effective address is equal to the value of the direct address (as determined by step 2).

b. If the X bit is a 1, the effective address is equal to the value of the direct address plus the 16-bit value in index register 1. Note that indexing with X1 is applied after indirect addressing. This is referred to as "post-indexing".

The effective address for an instruction, therefore, is the final 16-bit address value developed for that instruction, starting with the displacement value in the instruction itself. The core memory location whose address equals the effective address value is referred to as the "effective location". Similarly, the contents of the effective location are referred to as the "effective word".

The process and timing of effective address computation is summarized in Table 2 at the beginning of Chapter 3. The symbols used in Table 2 are defined as follows:

R	Bit 4 of the instruction
I	Bit 5 of the instruction
X	Bit 6 of the instruction
S	Bit 7 of the instruction
D	Bits 8 through 15 of the instruction (Displacement)
SD	Sign extended displacement value
(D)	Contents of location D
(X1)	Contents of index register 1 (general register 4)
(X2)	Contents of index register 2 (general register 5)
(P)	Contents of the internal P register (the address of the instruction)

INTERRUPT SYSTEM

The SIGMA 3 priority interrupt system is composed of up to 112 interrupt levels, each with a unique location (see Table 1) assigned to core memory, each with a unique priority, and each (except for the override group of interrupt levels) capable of being selectively armed and/or enabled by the CPU (see "Interrupt Level States").

Table 1. Core Memory Allocation and Interrupt Priority Groupings

Address Dec.	Address Hex.	Priority Level Within Group	Read Status Register Bits (1)	Set Active WD Register Bits (2)	WRITE DIRECT Register Bit (3)	Assignment	Availability	Group	WRITE DIRECT Group Code (4)
0 1 : 63	0 1 : 3F					First record loaded into memory during a load operation			
64 65 : 251	40 41 : FB					Unassigned			
252	FC	3			0	Counter 4 FC DC	Optional (as a set)	Counter (no inhibit)	X'0'
253	FD	4			1	Counter 3			
254	FE	5			2	Counter 2 8000	Optional (as a set)		
255	FF	6			3	Counter 1 5 00			
256	100	1			none	Power on	Optional (as a set)	Override (no inhibit)	none
257	101	2				Power off			
258	102	7	0		none	Machine fault	Optional (as a set)		
259	103	8	1			Protection violation			
260	104	9	2		none	Multiply exception	Standard (5)		
261	105	10	3			Divide exception			
262	106	1	6	6	6	Input/output	Standard	Input/output Group 0 (inhibited by bit 10 of PSD)	X'0'
263	107	2	7	7	7	Control panel			
264	108	3	8	8	8	Counter 4 = 0	Optional (as a set)		
265	109	4	9	9	9	Counter 3 = 0			
266	10A	5	10	10	10	Counter 2 = 0	Optional (as a set)		
267	10B	6	11	11	11	Counter 1 = 0			
268	10C	7	12	12	12	Integral 1	Optional (as a set)		
269	10D	8	13	13	13	Integral 2			
270	10E	9	14	14	14	Integral 3	Optional (as a set)		
271	10F	10	15	15	15	Integral 4			
272 273 : 287	110 111 : 11F	1 2 : 16			0 1 : 15	Designated by customer	Optional inhibited by bit 11 of PSD)	External Group 5	X'5'
288 289 : 303	120 121 : 12F	1 2 : 16			0 1 : 15			External Group 6	X'6'
:	:	:			:			:	:
352 353 : 357	160 161 : 16F	1 2 : 16			0 1 : 15			External Group 10	X'A'

(1) When the READ DIRECT instruction is used in the interrupt control mode to read interrupt status, the format of the data read is shown in this column. Bits 4 and 5 are indeterminate.

(2) When the WRITE DIRECT instruction is used in the interrupt control mode to perform the set active function, the accumulator bit positions are shown below. Bits 0-5 must be zero.

(3) When the WRITE DIRECT instruction is used in the interrupt control mode to operate on interrupt levels, the interrupt levels are selected by specific bit positions of the accumulator. The decimal numbers in this column indicate the bit positions in the accumulator that correspond to the various interrupt levels. This column applies to all WRITE DIRECT instructions except the set active.

(4) The hexadecimal numbers in this column indicate the group codes (for use with WRITE DIRECT) of the various interrupt levels.

(5) The multiply exception and the divide exception interrupt levels are included only in computers that do not have the extended arithmetic option.

Any interrupt level (except for the override group) can be "triggered" by the CPU; i.e., supplied with a signal at the same physical point where the signal from the external source would enter the interrupt level. The triggering of an interrupt level permits the testing of special systems programs before the special systems equipment is actually attached to the computer. It also permits an interrupt-servicing routine to defer a portion of the processing associated with an interrupt response by processing the urgent part of the interrupt response, triggering a lower-priority level (for a routine that handles the less-urgent part), then clearing the high-priority interrupt level so that other interrupts may be allowed to occur (before the less-urgent part is completed).

INTERNAL INTERRUPT LEVELS

Internal interrupt levels include those that are normally supplied with a SIGMA 3 system, as well as the optional counter (real-time clock), power fail-safe, machine fault, protection violation, and "counter-equals-zero" interrupt levels. The internal interrupt levels are arranged in three groups: the counter group, the override group, and the input/output group.

Counter (real-time clock) Group

These four optional interrupt levels are triggered by pulses from internal or external clock sources. Counter 1 has a constant frequency of 500 Hz; counters 2, 3, and 4 can be individually set to any of four manually switchable frequencies – the commercial line frequency, 2 kHz, 8 kHz, and a user-supplied external signal – that may be different for each counter. When a clock pulse is received by one of the counter interrupt levels (and the level is armed and enabled), the value in the memory location associated with the level is incremented by 1, and the level is cleared and armed. If the value in the affected memory location is zero after being incremented, the corresponding counter-equals-zero interrupt level in the input/output group of internal levels (see below) is then triggered. All other interrupt levels (including the counter-equals-zero interrupt levels) are processed by interrupt-servicing routines and are designated as "normal" interrupt levels. The counter interrupt levels can be armed, disarmed, enabled, disabled, or triggered by means of a specific configuration (interrupt control mode) of the WRITE DIRECT instruction; however, these levels cannot be inhibited. The priority of the counter interrupt levels is immediately below the priority of the power off interrupt level, but above the priority of the machine fault interrupt level.

Override Group

The interrupt levels in this group are associated with independent, optional SIGMA 3 features.

The override interrupt levels are always armed (cannot be disarmed), always enabled (cannot be disabled), cannot be triggered by a WRITE DIRECT instruction, and cannot be inhibited.

Power Fail- Safe. The two optional power fail-safe interrupt levels are used to enter routines that save and restore volatile information in the event of a power failure. The power-off interrupt level is triggered whenever the power supply voltage falls below a safe limit; likewise, the power-on interrupt level is triggered whenever power returns to safe limits. The power fail-safe interrupt levels have a higher priority than the counter interrupt levels.

Machine Fault. This optional interrupt may be triggered by any of the following:

1. Memory parity error. (Interrupt occurs only if the PARITY ERROR switch is in the INTRPT position.)
2. Interface timer runout during external direct I/O (RD or WD).
3. Interface timer runout during integral I/O processor (IIOP) service call.
4. Interface timer runout during external I/O processor (EIOP) service call.

Memory parity error (O and C bits equal zero immediately after entry of fault interrupt routine) may occur upon reading erroneous data or upon reading nonexistent memory. No error will occur upon attempting to write into nonexistent memory.

External direct I/O timer runout error (C bit equals 1 immediately after entry of fault interrupt routine) will occur when an external device does not respond to a RD or a WD within a specified time. The instruction will be aborted and the CPU will go to WAIT.

The IIOP service call timer runout error (O bit equals 1 immediately after entry of fault interrupt routine) may occur either between instructions or during external direct I/O. If this error occurs between instructions, the CPU will delay long enough for the machine fault interrupt to gain control of the interrupt priority logic and then go to the WAIT state. If the error occurs on an IIOP service call during direct I/O, the CPU completes the RD or WD before going to WAIT. For either case of timer runout, if the TIMER switch is in the OVERRIDE position, no interrupt will occur and the CPU will merely wait for the expected response.

The EIOP service call timer runout error triggers the I/O interrupt (if armed and enabled). The interrupt routine must contain an AIO instruction. The AIO is issued to the DIO interface to which the EIOP is connected (EIOP interrupt calls have priority over any IIOP device) but the DIO interface receives no response from the EIOP, thus causing a CPU timer runout (see 2, above) triggering the Machine Fault interrupt. An I/O reset must be programmed to reset the EIOP interface timer.

Protect Violation. The Machine Fault interrupt is part of the Protect Option. The protect option includes the protect violation interrupt level (the protect option is described on page 14). If the option is installed and the PROTECT switch on the processor control panel is in the ON position when a protect violation is encountered, the protect violation interrupt level is triggered.

Multiply/Divide Exception. The extended arithmetic option includes the additional logic required for executing the MULTIPLY and DIVIDE instructions. If the extended arithmetic option is not installed, the multiply exception interrupt levels are provided to allow for simulation of the unimplemented instructions. In this case, the appropriate exception interrupt level is triggered whenever an attempt is made to execute a MULTIPLY or DIVIDE instruction.

Input/Output Group

This interrupt group includes two standard interrupt levels and eight optional levels. The I/O and control panel interrupt levels are standard; the four counter-equals-zero interrupt levels and the four integral interrupt levels are optional.

All interrupt levels in the input/output group can be inhibited by means of the internal interrupt inhibit (bit 10 of the PSD), and can be armed, disarmed, enabled, disabled, and triggered by specific configurations of the WRITE DIRECT instruction.

I/O Interrupt Level. The I/O interrupt level accepts interrupt signals from the standard I/O system. An I/O routine must contain an ACKNOWLEDGE I/O INTERRUPT (AIO) instruction that identifies the source and cause of an I/O interrupt.

Control Panel Interrupt Level. The control panel interrupt level is normally connected to the INTERRUPT switch on the processor control panel. The control panel interrupt level can thus be triggered by the computer operator, allowing him to initiate a specific routine.

Counter-equals-zero Interrupt Levels. The counter-equals-zero interrupt levels are associated with the four optional real-time clocks. For each clock option installed, the CPU automatically increments one of four core memory (counter) locations as the clock pulses are received. When the value in a counter location equals zero, the corresponding counter-equals-zero interrupt level is triggered. Counting continues after the interrupt level is triggered; unless the counter interrupt level is disarmed or disabled, counting will continue until zero is reached again. (See "Counter Interrupt Processing".)

EXTERNAL INTERRUPT LEVELS

A SIGMA 3 system can contain up to seven groups of interrupt levels with up to four levels in the first integral group which can, if desired, be treated as external interrupt levels; and up to 16 levels in each of the six external groups. The integral levels are controlled as part of the input/output group of internal interrupt levels (i.e., inhibited with the internal interrupt inhibit), and have a lower priority than the other levels in the input/output group. All interrupts are controlled separately (i.e., inhibited with the external interrupt inhibit), and can be arranged in almost any priority sequence (see "Interrupt Priority Sequence", below).

INTERRUPT LEVEL STATES

A SIGMA 3 interrupt level is mechanized by means of three flip-flops. Two of the flip-flops are used to define any of four mutually exclusive states: disarmed, armed, waiting, and active. The third flip-flop is used to enable or disable the level. The various states and the condition causing changes in state (see Figure 3) are described in the following paragraphs.

Disarmed

When an interrupt level is in the disarmed state, no signal to that interrupt level is admitted; that is, no record is retained of the signal, nor is any program interrupt caused by it at any time.

Armed

When an interrupt level is in the armed state, it is capable of accepting and remembering an interrupt signal. The receipt of such a signal advances the interrupt level to the waiting state.

Waiting

When an interrupt level in the armed state receives an interrupt signal, it advances to the waiting state, and remains in the waiting state until it is allowed to advance to the active state.

If the level-enable flip-flop is off, the interrupt level can undergo all state changes except that of moving from the waiting to the active state. Furthermore, if this flip-flop is off, the interrupt level is completely removed from the chain that determines the priority of access to the CPU. Thus, an interrupt level in the waiting state with its level-enable in the off condition does not prevent an enabled, uninhibited interrupt level of lower priority from moving the active state.

When an interrupt level is in the waiting state, the following conditions must all exist simultaneously before the level advances to the active state:

1. The level is enabled (i.e., its level enable flip-flop is a 1).

2. The group inhibit (if applicable) is off (i.e., the appropriate inhibit is a 0).

3. No higher-priority interrupt level is in the active state (or is in the enabled, waiting state with its inhibit off).

4. The CPU is in an interruptible phase of operation.

Active

When a normal interrupt level meets all of the conditions necessary to permit it to move from the waiting state to the active state, it is permitted to do so by being acknowledged by the computer, which then stores the current PSD at the location specified by the contents of the location associated with the level. The first instruction of the interrupt-servicing routine is then taken from the location following the stored PSD. A new interrupt cannot occur until after the execution of this first instruction.

A normal interrupt level remains in the active state until it is removed from the active state by a specific configuration of the WRITE DIRECT (WD) instruction, followed by an LDX instruction. An interrupt-servicing routine can itself be interrupted (whenever a higher-priority interrupt level meets

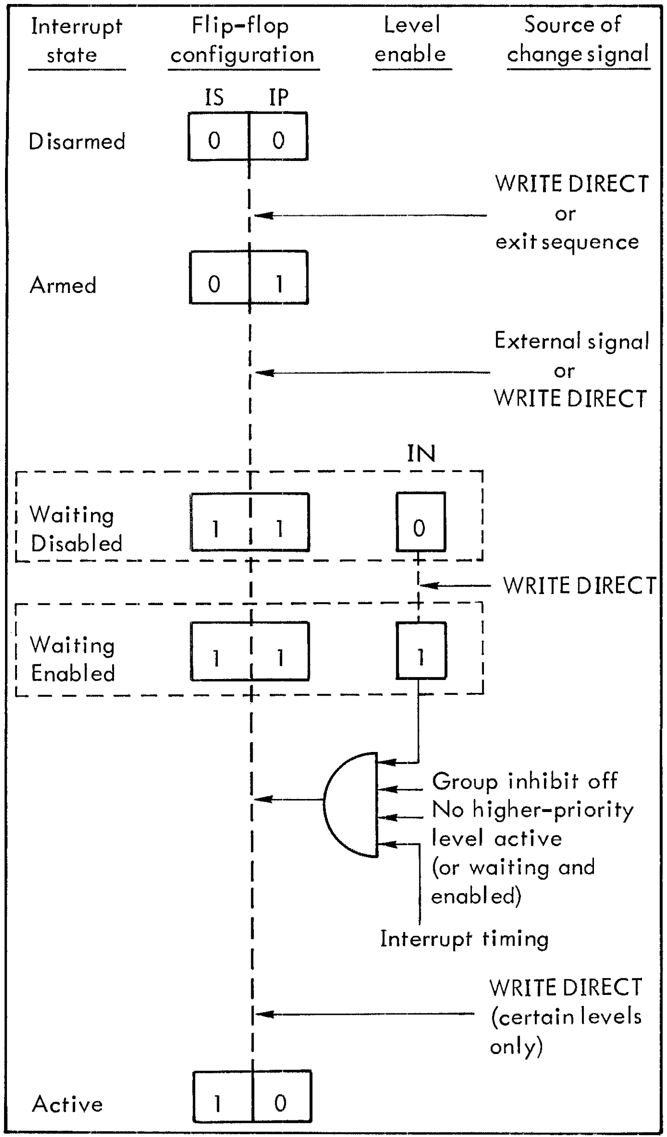


Figure 3. Interrupt Level Operation

all of the conditions for becoming active) and then continued (after the higher-priority interrupt level is removed from the active state). However, an interrupt-servicing routine cannot be interrupted by a lower-priority interrupt level as long as its interrupt level remains in the active state. Normally, the interrupt-servicing routine returns its interrupt level to the armed state and transfers program control back to the point of interrupt by means of an interrupt routine exit sequence (see "Interrupt Routine Entry and Exit").

INTERRUPT SYSTEM CONTROL

The SIGMA 3 interrupt system allows four types of program control over the interrupt system:

1. The ability to read the status of most of the interrupt levels in the override and Input/Output groups with a READ DIRECT (RD) instruction (this capability is part of the Power Fail-Safe option).

2. The ability to set to the Active state most of the interrupts in the override and Input/Output groups on an individual basis with a WRITE DIRECT (WD) instruction.

3. The ability to individually arm, disarm, enable, disable, or trigger any interrupt level in the system except for the override group.

4. The ability to group-inhibit either all of the internal (except the override group) interrupts, all of the external interrupts, or both with a single instruction.

The read-status RD and the set-active WD provide the capability of reading the status of certain key interrupt levels during a power-off interrupt routine and restoring them to the same status during the subsequent power-on interrupt routine.

The format of the effective address of the read-status RD is:

1	0	Code	0	▨
0 1 2 3	4	5 6 7	8 9 10 11	12 13 14 15

Bit positions 0-3 must contain the value X'1' to specify the interrupt control mode. Bit positions 5-7 contain a code specifying the read-status operation to be performed. Bits 4 and 8-11 must contain zeros. Bits 12-15 have no effect.

This instruction tests interrupts in the Override and Input/Output group and sets corresponding bits in the accumulator (see Table 1) according to the state of one of the three flip-flops associated with each interrupt level. The code specifies which of the three flip-flops is to be interrogated, as follows:

Code	Function
001	Set to a 1 the accumulator bits corresponding to each interrupt level that is in the Armed or Waiting state. (Read IP flip-flops.)
010	Set to a 1 the accumulator bits corresponding to each interrupt level that is in the Waiting (or Active) state. (Read IS flip-flops.)
100	Set to a 1 the accumulator bits corresponding to each interrupt level that is Enabled. (Read IN flip-flops.)

The register bit format for the read-status RD is shown in Table 1. Bits 4 and 5 are unused and are indeterminate. An uninstalled interrupt level will respond with a 1 to any of the three read-status instructions.

The WD instruction transmits its 16-bit effective address to all receiving elements of the Sigma 3 system (see Chapter 3, "Direct Control Instructions"). In the case of interrupt system control, the following configuration of a WD effective address is used to control the alteration of the various states of the individual interrupt levels within the interrupt system.

1	0	Code	0	Group Code
0 1 2 3	4	5 6 7	8 9 10 11	12 13 14 15

The format of the effective address is similar to that described above for the read-status RD, though the operation performed is different. This instruction uses the contents of the accumulator (general register 7) to determine which of the interrupt levels in the specified group are to be operated upon. Bit positions 12-15 of the WD effective address must contain the value X'0', so that bit position 0 of the accumulator corresponds to the counter 4 interrupt level, bit position 1 corresponds to the counter 3 interrupt level, and so on through bit position 15, which corresponds to the integral 4 interrupt level (see Table 1).

Each interrupt level in the designated group is operated on according to the function code specified by bits 5 through 7 of the effective address of WD. The defined codes and their associated functions are as follows:

Code — Function

000 — A code field (bits 5-7) of 000 will cause each interrupt level corresponding to the 1's in the accumulator to be set to the Active state if that interrupt level was previously in either the Armed or Waiting state. This operation is not affected by the state of the level enable flip-flops or the group inhibits. Any levels in the Disarmed state and those levels corresponding to the 0's in the accumulator are not affected. If the selected interrupt level is already Active, it will be set to the Disarmed state. The Set Active operation causes the selected level to enter the Active state, without going through the automatic interrupt entry sequence.

For a Set Active WD operation (where group code=X'0'), bits 0-5 of the accumulator must be zero and bits 6-15 shall correspond to the interrupts in the Input/Output group as shown in Table 1.

001 — Disarm all levels corresponding to a 1 in the accumulator; no other levels are affected.

010 — Arm and enable all levels corresponding to a 1 in the accumulator; no other levels are affected.

011 — Arm and disable all levels corresponding to a 1 in the accumulator; no other levels are affected.

100 — Enable all levels corresponding to a 1 in the accumulator; no other levels are affected.

101 — Disable all levels corresponding to a 1 in the accumulator; no other levels are affected.

110 — Enable all levels corresponding to a 1 in the accumulator and disable all other levels.

111 — Trigger all levels corresponding to a 1 in the accumulator. All such levels that are currently armed advance to the waiting state. Those levels currently disarmed are not altered, and all levels corresponding to a 0 in the accumulator are not affected.

Code — Function

111 (cont.) — The interrupt trigger is applied at the same input point as that used by the device connected to the interrupt level.

The recommended method for producing the appropriate configuration of the WRITE DIRECT effective address is to indirectly address a memory location that contains the appropriate bit configuration for the desired effective address.

INTERRUPT PRIORITY SEQUENCE

SIGMA 3 interrupts are arranged in groups so that they may be connected in a predetermined priority arrangement by groups of levels. The priority of each level within a group is fixed, with the first level having the highest-priority and the last level having the lowest-priority. The user has the option of ordering a machine with a priority chain starting with the override group and connecting all remaining groups in any sequence. This allows the user to establish external interrupts above and below the input/output group of internal interrupts. Figure 4 shows a typical interrupt priority chain.

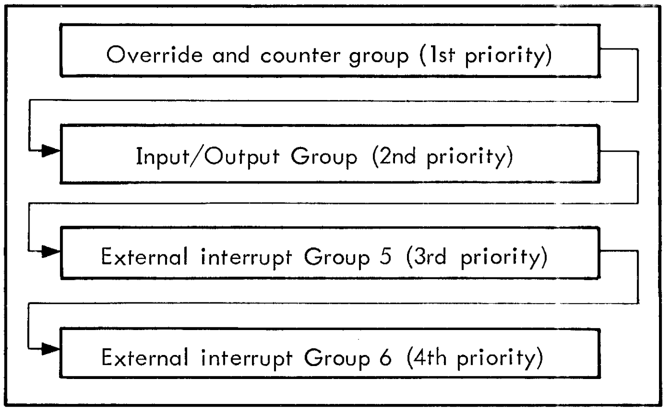

Override and counter group (1st priority)
Input/Output Group (2nd priority)
External interrupt Group 5 (3rd priority)
External interrupt Group 6 (4th priority)

Figure 4. Interrupt Priority Chain

INTERRUPT ROUTINE ENTRY AND EXIT

When a normal interrupt level becomes active, the computer automatically saves the program status doubleword (which contains the protected program indicator, the interrupt inhibits, the carry and overflow indicators, and the program address). The status information is stored in the location whose address is contained in the dedicated interrupt location.

The current value in the program address (P) register is stored in the location following the status information. The significance of the stored program address depends upon the particular interrupt level, as follows:

a. For the machine fault error, protect violation, multiply exception, and divide exception interrupt levels, the stored program address is the address of the instruction that was being executed at the time the interrupt condition occurred, or is the address of a nonexistent or protected location from which an instruction access was attempted.

b. For all other normal interrupt levels, the stored program address is the address of the next instruction in sequence after the instruction whose execution was just completed at the time the interrupt condition occurred.

After the program address is stored, the next instruction to be executed is then taken from the location following the stored program address. The first instruction of the interrupt-servicing routine is always executed before another interrupt can occur. Thus the interrupt-servicing routine may inhibit all other normal interrupt levels so that the routine itself will not be interrupted while in process.

At the end of an interrupt-servicing routine, an exit sequence restores the program status that existed when the interrupt level became active. An exit sequence is a WRITE DIRECT (WD) instruction with an effective address of X'00D8' followed immediately by a LOAD INDEX (LDX) instruction with an effective address equal to the address value in the interrupt location for the routine (no interrupt is allowed to occur between these two instructions). Execution of LDX in an interrupt routine exit sequence does not affect the contents of index register 1.

COUNTER INTERRUPT PROCESSING

The counter interrupt levels are not associated with interrupt-servicing routines as such. Instead, an active counter interrupt level is serviced by accessing the contents of the memory location assigned to the interrupt level, incrementing the value in the memory location by 1, and restoring the new value in the same memory location. The processing of an active counter interrupt level does not affect the overflow indicator or the carry indicator. Thus, the on-going program is not affected by a counter clock pulse (other than by the time required for processing) unless the result in the assigned memory location is all 0's after being incremented; in that case, the corresponding counter-equals-zero interrupt level is triggered.

CPU INTERRUPT RECOGNITION

If all other conditions are met and an interrupt level is waiting and enabled, the CPU will recognize the interrupt following the completion of any instruction except:

1. WD X'00D8' (precedes LDX for interrupt routine exit).
2. Between the storing of the PSD and the execution of the next instruction upon entering a normal interrupt subroutine.
3. Between an RD Set Multiple Precision Mode instruction and its following instruction.

PROTECTION SYSTEM

The primary purpose of the optional protect feature is to guarantee the integrity of a master- or executive-mode (foreground) program while another (background) program is concurrently being executed. The SIGMA 3 protection system provides both operation protection and memory write protection by means of 16 words of register storage that are installed as part of the protect option. Each bit in these 16 words is associated with a specific block of core memory. A block of core memory is a region of 256 consecutive locations, whose lowest-numbered address is some integer multiple of 256; thus, bit 0 of protection register 0 is associated with core memory locations 0 through X'FF', bit 1 of protection register 0 is associated with core memory locations X'100' through X'1FF', and bit 15 of protection register X'F' is associated with core memory locations X'FF00' through X'FFFF'. A protect bit of 0 designates an unprotected memory block and a protect bit of 1 designates a protected block.

The protect registers can be individually loaded by executing a WRITE DIRECT instruction with an effective address of X'8r', where r is a hexadecimal digit that designates which of the protection registers is to be loaded from the accumulator (A register). Thus, the protect bits for 16 memory pages (4096 words) can be set up by executing a single instruction.

Operation of the protection system is under control of the PROTECT switch and the key-operated lock on the processor control panel (see Chapter 5). If the protection system is operative, the following rules apply:

1. The privileged instructions (READ DIRECT and WRITE DIRECT) can be executed only if they are accessed from protected memory (except for the RD set multiple precision mode). If a privileged instruction is accessed from unprotected memory, the instruction is not executed; instead, the protect violation interrupt level is triggered.
2. If a set multiple precision mode RD instruction is attempted from unprotected memory when the Extended Arithmetic option is not installed, the protect violation interrupt level is triggered.
3. An instruction accessed from unprotected memory can be immediately followed by an instruction accessed from protected memory only in response to an interrupt condition. If an instruction is accessed from protected memory and the immediately preceding instruction was accessed from unprotected memory, the instruction is not executed (unless it is in response to an interrupt condition); instead, the protect violation interrupt level is triggered. This rule applies to branching from unprotected memory to protected memory as well as to executing an instruction in protected memory as the next instruction in normal sequence after an instruction in unprotected memory.
4. A STORE ACCUMULATOR (STA) or an INCREMENT MEMORY (IM) instruction can be used to alter protected memory only if the instruction is accessed from protected memory. If an attempt is made to alter protected memory with an instruction accessed from unprotected memory, the operation is not performed; instead, the protect violation interrupt level is triggered.

3. INSTRUCTION REPERTOIRE

This section contains descriptions of all SIGMA 3 instructions. With each description is a diagram showing the format of the instruction and its operation code (as a hexadecimal digit in the 4 high-order bit positions of the diagram). Some of the instruction diagrams are divided by a horizontal line, as in the SHIFT instruction on page 15. In these cases, the upper portion of the diagram represents the instruction format, while the lower portion represents the format of the instruction's effective address. Bit positions of fields that are shaded represent portions of the instruction's effective address that are ignored. However, these areas should always be coded with 0's to preclude conflict with features that may be implemented in the future.

Above each diagram are the mnemonic code and name of the instructions, sometimes followed by a parenthetic note about optional features or privileged operation. Under each diagram is a description of the instruction, followed by a list of the registers and indicators that can be affected by the instruction.

The following abbreviations are used in the descriptions:

A	Accumulator (general register 7)
E	Extended accumulator (general register 6)
X1	Index 1 (general register 4)
X2	Index 2 (general register 5)
T	Temporary storage (general register 3)
P	Program address register (general register 1)
D	Displacement (bits 8-15 of instruction)
SD	Sign extended displacement value
PP	Protected program indicator (PSD bit 8)
II	Internal interrupt inhibit (PSD bit 10)
EI	External interrupt inhibit (PSD bit 11)
O	Overflow indicator (PSD bit 14)
C	Carry indicator (PSD bit 15)
EL	Effective location
EW	Effective word, or (EL)

INSTRUCTION TIMING

The total time required for an instruction is divided into two parts, preparation and execution. The time required for instruction preparation is determined for most instructions by the RIXS bits as shown in Table 2. The process of effective address computation and the meaning of the symbols appearing in Table 2 were explained under "Effective Address Computation" in Chapter 2. More detailed instruction timing accompanies each instruction description in this chapter and is also presented in Appendix B. The time required for each instruction is expressed as preparation time plus execution time. Thus, an LDA instruction requires from 3 to 7 pulse times of .325 μsec for preparation, plus a fixed execution time of .975 μsec.

Table 2. Effective Address Computation and Timing

R	I	X	S	Effective Address	Preparation Time[†]
0	0	0	0	D	3
0	0	0	1	D + (X2)	4
0	0	1	0	D + (X1)	4
0	0	1	1	D + (X1) + (X2)	5
0	1	0	0	(D)	6
0	1	0	1	(D + (X2))	7
0	1	1	0	(D) + (X1)	6
0	1	1	1	(D + (X2)) + (X1)	7
1	0	0		(P) + SD	3
1	0	1		(P) + SD + (X1)	4
1	1	0		((P) + SD)	6
1	1	1		((P) + SD) + (X1)	6

[†] In multiples of .325 μsec pulse times.

MEMORY REFERENCE INSTRUCTIONS

LDA LOAD ACCUMULATOR

8	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

LOAD ACCUMULATOR loads the effective word into the accumulator (general register 7).

Affected: (A) Time: .325(3-7) + .975 μsec

STA STORE ACCUMULATOR

E	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

STORE ACCUMULATOR stores the contents of the accumulator into the effective location.

Affected: (EL) Time: .325(3-7) + .975 μsec

LDX LOAD INDEX

C	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

LOAD INDEX loads the effective word into index 1 (general register 4).

Affected: (X1) Time: .325(3-7) + .975 μsec

When LOAD INDEX is executed as the next instruction in sequence after a WRITE DIRECT (WD) instruction with an effective address of X'00D8', index register 1 is not affected; instead, LOAD INDEX performs the following operations in order to restore the program environment that existed before the computer acknowledged an interrupt condition:

1. Sets the protected program indicator equal to bit 8 of the effective doubleword.

2. Sets the internal interrupt inhibit equal to bit 10 of the effective doubleword.

3. Sets the external interrupt inhibit equal to bit 11 of the effective doubleword.

4. Sets the overflow indicator equal to bit 14 of the effective doubleword.

5. Sets the carry indicator equal to bit 15 of the effective doubleword.

6. Loads bits 16 through 31 of the effective doubleword into the program address register (general register 1).

7. Clears the highest-priority active interrupt level and returns the interrupt level to the armed state.

8. Resets the exit condition that was set by the preceding WD instruction (that caused LDX to perform the above operations).

Bits 0 through 7, 12, and 13 of the effective doubleword are ignored.

Affected: PP, II, EI, O, C, (P), highest-priority active interrupt level
Time: .325(3-7) +1.950 μsec

ADD ADD

A				R	I	X	S	Displacement							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ADD adds the effective word to the contents of the accumulator and then loads the result into the accumulator. If the signs of the two operands are equal but the sign of the result is different, overflow has occurred, in which case the overflow indicator is set to 1. If overflow does not occur, the overflow indicator is reset to 0. The carry indicator is set to 1 if a carry occurs from the sign bit position of the adder; if such a carry does not occur, the carry indicator is reset to 0.

Affected: (A), O, C Time: .325(3-7) + .975 μsec

SUB SUBTRACT

B				R	I	X	S	Displacement							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

SUBTRACT forms the two's complement of the effective word, adds this value to the contents of the accumulator, and then loads the result into the accumulator. If the sign of the result in the accumulator is equal to the sign of the effective word but the sign of the original operand in the accumulator was different, overflow has occurred, in which case the overflow indicator is set to 1. If overflow does not occur, the overflow indicator is reset to 0. The carry indicator is set to 1 if a carry occurs from the sign bit position of the adder; if no carry occurs, the carry indicator is reset to 0. (A carry occurs if the 16-bit magnitude in the effective location is equal to or less than the 16-bit magnitude in A.)

Affected: (A), O, C Time: .325(3-7) + .975 μsec

AND LOGICAL AND

9				R	I	X	S	Displacement							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

LOGICAL AND forms the logical product of the effective word and the contents of the 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; the logical product contains a 0 in each bit position for which there is a 0 in the corresponding bit position of either operand.

Affected: (A) Time: .325(3-7) + .975 μsec

IM INCREMENT MEMORY

F				R	I	X	S	Displacement							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

INCREMENT MEMORY adds 1 to the effective word and then stores the result in the effective location. Overflow occurs only if the resulting value of the effective word is X'8000' (32,768), in which case the overflow indicator is set to 1; otherwise, the overflow indicator is reset to 0. Carry occurs only if the resulting value of the effective word is X'0000', in which case the carry indicator is set to 1; otherwise, the carry indicator is reset to 0.

Affected: (EL), O, C Time: .325(3-7) + 1.950 μsec

B BRANCH

4				R	I	X	S	Displacement							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

BRANCH loads the effective address into the program address register (general register 1). Thus, the next instruction is accessed from the location pointed to by the effective address of the BRANCH instruction.

Affected: (P) Time: .325(3-7) + 0 μsec

S SHIFT

2				R	I	X	S	Displacement							
▨							Type				N				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

SHIFT uses the 9 low-order bits of the effective address as a specification of the type of shift to be performed. The effective address is not used for a memory access; instead, bits 7, 8, 9, and 10 of the effective address specify the type of shift and bits 11 through 15 of the effective address specify the number of bit positions to be shifted, as follows.

Bit	Specification
7	0 specifies a non-normalize shift. If bit 7 is 1, bits 8 through 15 are ignored and the instruction becomes a normalize shift. If the initial contents of the extended accumulator (E) and the accumulator (A) are both zero, the instruction exits without changing any register. If the initial contents of either E or A are not zero, the instruction performs a double-register arithmetic left shift on E and A (bits shifted out of bit 0 of A shift into bit 15 of E). The double-register shifting continues until bits 0 and 1 of E are different. The contents of the temporary storage register, T (general register 3), are decremented by one for each left shift performed. At the completion of the normalize shift instruction the carry indicator is reset and the overflow indicator is set if the contents of the temporary storage register T overflowed (i.e., was decremented past negative full scale during the normalize operation). If the T register has not overflowed, the overflow indicator is reset.
8	0 specifies a single-register shift; that is, only the contents of the accumulator (general register 7) are to be shifted. The sign bit position is bit position 0 of the accumulator. 1 specifies a double-register shift; that is, the contents of both the extended accumulator (general register 6) and the accumulator are to be shifted simultaneously. The two registers are treated as a single, 32-bit register; bits shifted to the right of bit position 15 of the extended accumulator are copied into bit position 0 of the accumulator. Likewise, bits shifted to the left of bit position 0 of the accumulator are copied into bit position 15 of the extended accumulator. In this case, the sign bit position is bit position 0 of the extended accumulator.
9	0 specifies an arithmetic shift. For single right shifts, the sign of the value in the accumulator (bit 0) is copied into vacated bit positions; for double right shifts, the sign of the value in the extended accumulator is copied into vacated bit positions. (In either case, bits shifted to the right of bit position 15 of the accumulator are lost.) For both single and double left shifts, 0's are copied into vacated bit positions, and bits shifted to the left of the sign bit position are lost. 1 specifies a circular shift. For single shifts, bits shifted to the right of bit position 15 of the accumulator are copied into bit position 0; bits shifted to the left of bit position 0 of the accumulator are copied into bit position 15. For double shifts, bits shifted to the right of bit position 15 of the accumulator are copied into bit position 0 of the extended accumulator; bits shifted to the left of bit position 0 of the extended accumulator are copied into bit position 15 of the accumulator.

Bit	Specification
10	0 specifies a right shift. 1 specifies a left shift.
11-15 (N)	This value specifies the number of bit positions of the shift operation, which may be any number in the range 0 through X'1F' (31).

Bits 0 through 6 of the effective address are ignored.

The overflow indicator is set to 1 only if any bit shifted into the sign bit position during an arithmetic left shift is different from that previously in the sign bit position; otherwise the overflow indicator is reset to 0.

The carry indicator is reset to 0 at the beginning of the shift operation. If the shift is to the right, the carry indicator remains reset; however, if the shift is to the left, the carry indicator is inverted each time a 1 is shifted out of the sign bit position. Hence, the carry bit will represent even parity on the bits shifted out of the sign bit position.

Affected: (T), (E), (A), O, C   Time: See Appendix B

CP COMPARE

D	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

COMPARE algebraically compares the contents of the accumulator and the effective word, with both operands treated as signed quantities. The overflow and carry indicators are set or reset, according to the result of the comparison, as follows:

O	C	Result of comparison
0	0	the operand in the accumulator is algebraically less than the effective word
1	0	the operand in the accumulator is algebraically greater than the effective word
1	1	the operand in the accumulator is equal to the effective word

Affected: O, C   Time: .325(3-7) + .975 μsec

CONDITIONAL BRANCH INSTRUCTIONS

The eight conditional branch instructions specify conditional, relative branching. Each of the conditional branch instructions performs a test to determine whether the branch condition is "true".

If the branch condition is true, the instruction acts as a BRANCH instruction coded for self-relative addressing with neither indirect addressing nor indexing. (The conditional branch instructions automatically invoke self-relative addressing.) Thus, if the branch condition is true, the next instruction is accessed from the location pointed to by the effective address of the conditional branch instruction.

If the branch condition is not true, the instruction acts as a "no operation" instruction. Thus, if the branch condition is not true, the next instruction is accessed from the next location in ascending sequence after the conditional branch instruction.

BAN BRANCH IF ACCUMULATOR NEGATIVE

6	1	1	1	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The branch condition is true only if bit 0 of the accumulator is 1.

Affected: (P) Time: .325(3) + 0 μsec

BAZ BRANCH IF ACCUMULATOR ZERO

6	0	1	0	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The branch condition is true only if the accumulator contains the value X'0000'.

Affected: (P) Time: .325(3) + 0 μsec

BEN BRANCH IF EXTENDED ACCUMULATOR NEGATIVE

6	1	1	0	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The branch condition is true only if bit 0 of the extended accumulator is 1.

Affected: (P) Time: .325(3) + 0 μsec

BNO BRANCH IF NO OVERFLOW

6	0	0	0	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The branch condition is true only if the overflow indicator is reset (0). The overflow indicator is not affected.

Affected: (P) Time: .325(3) + 0 μsec

BNC BRANCH IF NO CARRY

6	0	0	1	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The branch condition is true only if the carry indicator is reset (0). The carry indicator is not affected.

Affected: (P) Time: .325(3) + 0 μsec

BIX BRANCH ON INCREMENTING INDEX

6	0	1	1	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

BIX adds 1 to the current value in index 1 (general register 4) and loads the result into index 1. The branch condition is true only if the result in index 1 is a nonzero value.

Affected: (X1), (P) Time: .325(3) + .650 μsec

BXNO BRANCH ON INCREMENTING INDEX AND NO OVERFLOW

6	1	0	0	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

If the overflow indicator is set (1), no operation is performed and the computer executes the next instruction in sequence. However, if the overflow indicator is reset (0), BXNO adds 1 to the current value in index register 1 (general register 4) and loads the result into index 1; the branch condition is true only if the result in index 1 is a nonzero value. The overflow indicator is not affected by this instruction.

Affected: (X1),(P) Time: .325(3) + 0 μsec (no index)
.325(3) + .650 μsec (index)

BXNC BRANCH ON INCREMENTING INDEX AND NO CARRY

6	1	0	1	±	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

If the carry indicator is set (1), no operation is performed and the computer executes the next instruction in sequence. However, if the carry indicator is reset (0), BXNC adds 1 to the current value in index register 1 and loads the result into index 1; the branch condition is true only if the result in index 1 is a nonzero value. The carry indicator is not affected.

Affected: (X1), (P) Time: .325(3) + 0 μsec (no index)
.325(3) + .650 μsec (index)

COPY INSTRUCTION

The copy instruction specifies operations between any two general registers. The format of the copy instruction is:

7	OP	AC	AI	CD	DR	IS	SR
0 1 2 3	4 5	6	7	8	9 10 11	12	13 14 15

Bit(s)	Function
0-3	Bit positions 0-3 are coded as X'7', to specify the copy instruction.
4-5 (OP)	Bit positions 4-5 specify which of 4 operations is to be performed. The operations are:

4	5	Operation	
0	0	logical AND	overflow and carry indicators not affected
0	1	logical inclusive OR	
1	0	logical exclusive OR	
1	1	arithmetic add (overflow and carry indicators set as described for the instruction ADD)	

Bit(s)	Function
6 (AC)	Bit position 6 specifies whether the current value of the carry indicator is to be added to the result. If this bit is a 1, the carry indicator is added to the low-order bit position of the result. If this bit is a 0, the carry indicator is ignored.
7 (AI)	Bit position 7 specifies whether the value X'0001' is to be added to the result. If this bit is a 1, a 1

Bit(s) Function

is added to the low-order bit position of the result. If bits 6 and 7 are both 1's, the value X'0001' is added to the result (regardless of the current value of the carry indicator).

8 (CD) Bit position 8 specifies whether the destination register (specified by bits 9-11) is to be cleared before the operation called for by bits 4-7 is performed. If bit 8 is a 1, the destination register is initially cleared. If bit 8 is 0, the initial contents of the destination register remain unchanged until the result is loaded into the destination register.

9-11 (DR) Bit positions 9-11 specify the general register that is to contain the result of the instruction. The overflow and carry indicators may be affected.

12 (IS) Bit position 12 specifies whether the source register operand (the value in the register specified by bits 13-15) is to be used as it appears in the source register, or is to be inverted (one's complemented) before the operation is performed. If bit 12 is a 1, the inverse of the value in the source register is to be used as the source register operand; however, the value in the source register is not changed. If bit 12 is a 0, the value in the source register is used as the source register operand.

13-15 (SR) Bit positions 13-15 specify the general register that contains the value to be used (or inverted and used) as the source register operand. A value of 0 in this field designates the value X'0000' as the contents of the source register.

The general registers are identified as follows:

Register	Function	Designation
Z	Zero	0
P	Program address	1
L	Link address	2
T	Temporary storage	3
X1	Index 1	4
X2	Index 2 (base address)	5
E	Extended accumulator	6
A	Accumulator	7

The contents of the P register, at the time the execution of the copy instruction begins, are the address of the next location in sequence after the copy instruction.

Affected: (DR), O, C Time: .325(3) + .975 μsec

Examples:

Instruction	Effect
X'74F0'	Clear the accumulator to all zeros.
X'74FF'	Invert (form the one's complement of) the contents of the accumulator.
X'7DFF'	Negate (form the two's complement of) the contents of the accumulator.
X'7C78'	Subtract 1 from the contents of the accumulator.
X'7D7B'	Subtract the contents of the T register from the contents of the accumulator.
X'75A1'	Copy the contents of the P register plus 1 into the L register.

The basic SIGMA 3 assembly language recognizes the following command mnemonics and generates the appropriate settings for bit positions 0-8 of the copy instruction. The settings for bit positions 9-15 are derived from the argument field of the symbolic line in which the command mnemonic appears. The source register operand is the contents of the source register if the IS bit is 0, or is the inverse (one's complement) of the contents of the source register if the IS bit is 1.

RCPY REGISTER COPY

7	4	1	DR	IS	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RCPY copies the source register operand into the destination register. The overflow and carry indicators are not affected.

RADD REGISTER ADD

7	C	0	DR	IS	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RADD adds the source register operand to the contents of the destination register and loads the result into the destination register. The overflow and carry indicators are set as described for the instruction ADD, based on the register operands and the final result.

ROR REGISTER OR

7	4	0	DR	IS	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

ROR logically inclusive ORs the source register operand with the contents of the destination register and loads the result into the destination register. If the corresponding bits in the source register operand and the destination register are both 0, a 0 remains in the corresponding bit position of the destination register; otherwise, the corresponding bit position of the destination register is set to 1. The overflow and carry indicators are not affected.

REOR REGISTER EXCLUSIVE OR

7	8	0	DR	IS	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

REOR logically exlusive ORs the source register operand with the contents of the destination register and loads the result into the destination register. If the corresponding bits of the source register operand and the destination register are different, the corresponding bit position of the destination register is set to 1; otherwise, the corresponding bit position of the destination register is reset to 0. The overflow and carry indicators are not affected.

RAND REGISTER AND

7	0	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RAND logically ANDs the source register operand with the contents of the destination register and loads the result into the destination register. If the corresponding bits of the source register operand and the destination register are both 1, a 1 remains in the destination register; otherwise, the corresponding bit position of the destination register is reset to 0. The overflow and carry indicators are not affected.

RCPYI REGISTER COPY AND INCREMENT

7	5	1	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RCPYI copies the source register operand into the destination register and then adds 1 to the new contents of the destination register. The overflow and carry indicators are not affected.

RADDI REGISTER ADD AND INCREMENT

7	D	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RADDI adds the source register operand to the contents of the destination register, increments the result by 1, and loads the final result into the destination register. The overflow and carry indicators are set, as described for the instruction ADD, based on the register operands and the final result.

RORI REGISTER OR AND INCREMENT

7	5	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RORI logically ORs the source register operand with the contents of the destination register, increments the result by 1, and loads the final result into the destination register. The overflow and carry indicators are not affected.

REORI REGISTER EXCLUSIVE OR AND INCREMENT

7	9	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

REORI logically exclusive ORs the source register operand with the contents of the destination register, increments the result by 1, and loads the final result into the destination register. The overflow and carry indicators are not affected.

RANDI REGISTER AND AND INCREMENT

7	1	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RANDI logically ANDs the source register operand with the contents of the destination register, increments the result by 1, and loads the final result into the destination register. The overflow and carry indicators are not affected.

RCPYC REGISTER COPY AND CARRY

7	6	1	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RCPYC copies the source register operand into the destination register and then adds the current value of the carry indicator to the result in the destination register. The overflow and carry indicators are not affected.

RADDC REGISTER ADD AND CARRY

7	E	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RADDC adds the source register operand to the contents of the destination register, adds the current value of the carry indicator to the result and loads the final result into the destination register. The overflow and carry indicators are set, as described for the instruction ADD, based on the register operands and the final result.

RORC REGISTER OR AND CARRY

7	6	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RORC logically inclusive ORs the source register operand with the contents of the destination register, adds the current value of the carry indicator to the result, and loads the result into the destination register. The overflow and carry indicators are not affected.

REORC REGISTER EXCLUSIVE OR AND CARRY

7	A	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

REORC logically exclusive ORs the source register operand with the contents of the destination register, adds the current value of the carry indicator to the result, and loads the final result into the destination register. The overflow and carry indicators are not affected.

RANDC REGISTER AND AND CARRY

7	2	0	DR	I S	SR
0 1 2 3	4 5 6 7	8	9 10 11	12	13 14 15

RANDC logically ANDs the source register operand with the destination register, adds the current value of the carry indicator to the result and loads the final result into the destination register. The overflow and carry indicators are not affected.

DIRECT CONTROL INSTRUCTIONS

The two instructions READ DIRECT (RD)and WRITE DIRECT (WD) are used to perform a variety of operations, such as:

- Transfer the contents of the accumulator into any general register or I/O channel register, and vice versa.
- Start an I/O operation, test an I/O operation, test an I/O device, halt an I/O operation, and acknowledge an I/O interrupt condition.

- Preserve the current program status indicators in the accumulator and conditionally alter the program status indicators.
- Load the optional protection system registers.
- Set a wait condition (stop computation).
- Set an exit condition to prepare for a return to an interrupted program.
- Read the interrupt status of internal interrupts.
- Control the individual levels of the priority interrupt system.
- Perform asynchronous input/output (optional).
- Control special systems equipment (optional).

The values of bits 0 through 3 of the effective address of the READ DIRECT and WRITE DIRECT instructions determine the mode of the instruction, as shown below.

Bit Position				
0	1	2	3	Mode
0	0	0	0	Internal computer control
0	0	0	1	Interrupt control (see Chapter 2)
0	0	1	0	Assigned to various groups of standard SDS products
0	0	1	1	
:				
1	1	1	0	
1	1	1	1	Special systems control (for customer use with specially designed equipment

RD READ DIRECT (Partially privileged, partially optional)

1	R	I	X	S	Displacement
Mode	Function				
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The effective address of the READ DIRECT instruction is used to specify the operation to be performed. In some operations, the contents of the accumulator are used as control or operand information for the specific operation to be performed. Data generated in response to such an operation may replace the previous contents of the accumulator. Two status bits, which may be generated as a result of the operation, are recorded in the overflow and carry indicators.

In the internal control mode, bits 8-15 of the READ DIRECT effective address designate the assigned internal control functions, as shown in Table 3. In this mode, bits 0-7 of the effective address must be zeros. Therefore, the displacement field (bits 8-15) of the instruction designates the control function if the R, I, X, and S bits of the instruction are all coded as zeros.

With the installation of the optional direct I/O feature, the READ DIRECT instruction may be used to communicate directly with an external device. When this feature is installed, the READ DIRECT instruction presents the 16-bit effective address on a connector and holds it there until it receives an acknowledgment from the device. With the response, the device sends 16 bits of data (which are loaded into the accumulator) as well as two status bits (which are recorded in the overflow and carry indicators).

Affected: determined by operation

Timing: See Appendix B

WD WRITE DIRECT (Privileged, partially optional)

0	R	I	X	S	Displacement
Mode	Function				
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

The effective address of the WRITE DIRECT instruction is used to specify the operation to be performed. For some operations, the contents of the accumulator are used as an operand to be transmitted to a receiving section within the central processor. The overflow and carry indicators are used to record two bits of status that may be generated as a result of the WRITE DIRECT instruction.

In the internal control mode, bits 8-15 of the WRITE DIRECT instruction designate the assigned internal control functions, as shown in Table 4. In this mode, bits 0-7 of the effective address must be zeros. Therefore, the displacement field (bits 8-15) of the instruction designate the control function if the R, I, X, and S bits of the instruction are all coded as zeros.

In the interrupt control mode, the effective address of the instruction is used to provide control of the priority interrupt system. (See Chapter 2, "Interrupt System Control".)

The WRITE DIRECT instruction may be used to transmit data directly to an external device. In this case, the optional direct I/O feature must be installed. When this feature is added, the 16-bit effective address and the 16-bit value in the accumulator are both presented in parallel on a connector and held there until the external device acknowledges. As the external unit acknowledges, it returns two bits of status information, which are recorded in the overflow and carry indicators.

Affected: determined by operation

Timing: See Appendix B

EXTENDED ARITHMETIC FEATURE

MUL MULTIPLY (Optional)

3	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

MULTIPLY multiplies the effective word by the contents of the accumulator, loads the 16 high-order bits of the doubleword product into the extended accumulator (general register 6), and loads the 16 low-order bits of the doubleword product into the accumulator. Neither overflow nor carry can occur; however, the carry indicator is set equal to the sign of the doubleword product.

If the Extended Arithmetic option is not implemented and an attempt is made to execute the instruction, the multiply exception interrupt level is triggered instead. The program address stored in memory as a result of the interrupt level becoming active is the address of the MULTIPLY instruction.

Affected: (E), (A), C Time: .325(3-7) + 6.825 μsec

DIV DIVIDE (Optional)

5	R	I	X	S	Displacement
0 1 2 3	4	5	6	7	8 9 10 11 12 13 14 15

DIVIDE divides the contents of the extended accumulator and the accumulator by the effective word.

If the absolute value of the quotient is equal to or greater than 32,768(2^{15}), the overflow indicator is set to 1 and the instruction is terminated with the contents of the extended accumulator and the accumulator unchanged from their previous values, and the carry indicator set equal to the sign of the dividend.

If the absolute value of the quotient is less than 2^{15}, the overflow indicator is reset to 0, the integer quotient is loaded into the accumulator, the integer remainder is loaded into the extended accumulator, and the carry indicator is set equal to the sign of the remainder. (The sign of the remainder is the same as the sign of the dividend.)

If the Extended Arithmetic option is not implemented and an attempt is made to execute the DIVIDE instruction, the divide exception interrupt level is triggered instead. The program address stored in memory as the result of the interrupt level becoming active is the address of the DIVIDE instruction.

Affected: (E),(A),O,C Time: .325(3-7) + 7.150 μsec (no overflow)
.325(3-7) + 2.600 μsec (overflow)

Table 3. READ DIRECT Internal Control Functions

Effective address bits 8	 9	 10	 11	 12	 13	 14	 15	Function
0	0	n	n	n	n	n	n	Copy the contents of general (or I/O channel) register nnnnnn into the accumulator (A). The reading of the EIOP channel registers temporarily interferes with normal I/O processing by the EIOP. Therefore, short program loops which repetitively read the EIOP channel registers should be avoided.
0	1	0	0	0	0	0	1	SIO Input/output instructions (see Chapter 4)
0	1	0	0	0	0	1	0	TIO
0	1	0	0	0	1	0	0	TDV
0	1	0	0	1	0	0	0	HIO
0	1	0	1	0	0	0	0	AIO
0	1	1	0	0	0	0	0	I/O Reset - Sends a reset signal out through the DIO interface and the IIOP 8-bit data path. Resets the entire EIOP. Resets all states of the Input/Output and Override groups of interrupts except the Enabled condition. Resets all states of all interrupt levels in the other interrupt groups.
1	1	0	0	0	0	0	0	Save program status in the accumulator (set bit 8 of A equal to the protected program bit, set bit 10 of A equal to the internal inhibit, bit 11 equal to the external interrupt inhibit, bit 14 equal to the overflow indicator, and bit 15 equal to the carry indicator; reset remainder of accumulator to 0's).
1	1	1	0	I	E	0	0	Save program status in the accumulator; then, if bit 12 of the effective address (I) is 1, reset the internal interrupt inhibit to 0; (if I is 0, the internal interrupt inhibit is not affected); if bit 13 of the effective address (E) is 1, reset the external interrupt inhibit to 0 (if E is 0, the external interrupt inhibit is not affected).
1	1	1	1	I	E	0	0	Save program status in the accumulator; then, if bit 12 of the effective address (I) is 1, set the internal interrupt inhibit to 1 (if I is 0, the internal interrupt inhibit is not affected); if bit 13 of the effective address (E) is 1, set the external interrupt inhibit to 1 (if E is 0, the external interrupt inhibit is not affected).
1	0	0	0	0	0	0	0	Set the bit positions of the accumulator equal to the states of the corresponding DATA switches on the operator control panel.
1	0	X	X	X	Y	Y	Y	Set Multiple Precision Mode (see Extended Arithmetic Feature).

Table 4. WRITE DIRECT Internal Control Functions

Effective address bits 8	 9	 10	 11	 12	 13	 14	 15	Function
0	0	n	n	n	n	n	n	Copy the contents of the accumulator (A) into general (or I/O channel) register nnnnnn.
0	1	n	n	n	n	n	n	Copy bit 0 of general (or I/O channel) register nnnnnn into the overflow indicator and then reset bit 0 of register nnnnnn to 0.
1	0	0	0	x	x	x	x	Copy the contents of the accumulator into protection register xxxx
1	1	0	0	0	0	0	0	Load program status from the accumulator (i.e., copy bit 8 of the accumulator into the protected program indicator, copy bit 10 of the accumulator into the internal interrupt inhibit, copy bit 11 of A into the external interrupt inhibit, copy bit 14 of A into the overflow indicator, and copy bit 15 of A into the carry indicator).
1	1	0	1	0	0	0	0	Set wait flip-flop to 1; this causes the central processor to stop computation. Wait ff is reset to 0 by any interrupt activation (including counter interrupts) or by moving COMPUTE switch to the IDLE position.
1	1	0	1	1	0	0	0	Set exit condition. This effective address configuration prepares the CPU to exit from an interrupt-servicing routine. All normal interrupt levels are inhibited until after the execution of the instruction following WD, which must be a LOAD INDEX (LDX) instruction whose effective address is identical to the address in the interrupt location for that interrupt-servicing routine. In this case, the LDX instruction does not affect index register 1; instead, it loads the PSD from the first two words of the interrupt routine, arms the highest-priority active interrupt level, and resets the exit condition.
1	1	1	0	I	E	0	0	If bit 12 of the effective address (I) is 1, reset the internal interrupt inhibit to 0; (if I is 0 the internal interrupt inhibit is not affected); if bit 13 of the effective address (E) is 1, reset the external interrupt inhibit to 0 (if E is 0, the external interrupt inhibit is not affected).
1	1	1	1	I	E	0	0	If bit 12 of the effective address (I) is 1, set the internal interrupt inhibit to 1 (if I is 0, the internal interrupt inhibit is not affected); if bit 13 of the effective address (E) is 1, set the external interrupt inhibit to I (if E is 0, the external interrupt inhibit is not affected).

SET MULTIPLE PRECISION MODE INSTRUCTION (Optional)

An RD instruction coded as 0000 0000 10XX XYYY will set the multiple precision mode for the next instruction after the RD instruction. This configuration of internal mode RD is not to be considered a privileged instruction if at least one of the X or Y bits is nonzero. If all X and Y bits are zero, the instruction is a privileged instruction that reads the DATA switches into the accumulator and does not set the multiple precision mode. Only the instruction immediately following the set multiple precision mode RD instruction is affected. Interrupts are not allowed to occur between the set multiple precision mode RD instruction and the following instruction; however, if a long string of set multiple precision mode RDs is erroneously programmed, interrupts can be processed after every other instruction. If a set multiple precision mode RD is attempted from protected memory when the Extended Arithmetic option is not installed, results are indeterminate.

If the instruction after the RD is either an ADD (ADD), SUBTRACT (SUB), or a COMPARE (CP) instruction, then that instruction, for any nonzero coding of the X and Y fields of the RD, becomes a double precision operation with the register operand being the extended accumulator and the accumulator and the memory operand being the contents of the effective location and the contents of the effective location + 1. The sign and most significant portion of the operands is contained in the extended accumulator and in the contents of the effective location respectively. For ADD and SUB the O and C indicators after the instruction will refer to the results of the 32-bit operation. For CP the O and C indicators will indicate the results of the 32-bit comparison.

If the instruction after the RD is either a LOAD ACCUMULATOR (LDA) or STORE ACCUMULATOR (STA), then that instruction becomes either a load multiple register or store multiple register operation with the X and Y fields of the RD specifying the number of registers and the initial register address. The 3-bit X field will contain the number of registers to be loaded or stored. The Y field will contain the address of the initial general register. Registers Z and P (general registers 0 and 1) cannot be loaded and stored by this method and therefore only registers L through A (general registers 2 through 7) may be specified by the Y field of the RD. The general register number specified by the Y field of the RD is incremented as each register is loaded or stored until the number of registers specified by the X field has been loaded or stored. The results are unpredictable if the general register address is allowed to increment past 7; however, the instruction will proceed and no I/O channel registers will be modified. The memory locations loaded from or stored into are the one through six consecutive locations starting with the effective location of the LDA or STA instruction. If the number of registers specified is one, the LDA or STA loads or stores one of the general registers L, T, X1, X2, E, or A without affecting any other register. If the number of registers specified is two and the initial register is E, the LDA or STA becomes a double precision load or store for use with the previously described double precision arithmetic instructions. If the number of registers specified is six and the initial register is L, the LDA or STA loads or stores complete register context (with the exception of the P register).

No special testing in the hardware will be performed to determine if a multiple precision operation will cross a memory protection boundary. If a boundary is crossed, the protect violation interrupt level is triggered. In this case, the instruction might be unrecoverable because of the memory locations or registers that have already been modified.

4. INPUT/OUTPUT OPERATIONS

BYTE-ORIENTED I/O SYSTEMS

In a SIGMA 3 system, byte-oriented input/output operations are under control of an input/output processor (IOP). This allows the CPU to concentrate on program execution, free from the time-consuming details of I/O operations. Any I/O events that require CPU intervention are brought to its attention by means of the interrupt system. A SIGMA 3 system may have an External IOP (EIOP), Integral IOP (IIOP), or both. These IOPs operate independently after they have been initialized by the central processor.

The EIOP has its own path to memory so that in a SIGMA 3 system containing more than one BMU, input/output and computation can occur simultaneously without interference. The IIOP shares a common path to memory with the CPU. When an input/output service call is presented to the IIOP, instruction execution is temporarily suspended for the duration of the service call. Thus, for data transfers at the maximum IIOP capacity, the CPU execution rate is effectively zero.

The EIOP can provide simultaneous input/output on up to 16 fully buffered channels at a combined transfer rate of approximately 500,000 8-bit bytes per second. With an optional 2-byte interface feature, the EIOP can transfer two bytes in parallel at an effective rate of approximately 850,000 bytes per second. The IIOP provides simultaneous input/output on up to 12 fully buffered channels at a maximum rate of approximately 450,000 bytes per second.

The EIOP has eight I/O channels standard and can be expanded to 16 channels by adding an optional group of eight. Four I/O channels are standard with the IIOP, with expansion to 12 via an optional group of eight.

Each I/O channel has associated with it two words of register storage located within the CPU which contain the control parameters for operation on that channel. Transfer on a channel is automatic under control of these parameters, with loading of the channel registers and initiation of the transfer from the CPU.

DEVICE NUMBER

Each peripheral device controller attached to the byte I/O system is assigned an 8-bit device number at installation time. This number is manually selected by switches within each device controller, based on the equipment configuration for the specific installation. The device number not only identifies the particular device (and, if appropriate, the control unit) but also designates which I/O channel controls the device. Devices are generally of two types: those that do not share a control unit with other devices (for example, card readers, card punches, or printers), and those that do (for example, magnetic tape units or XDS RAD files). A device that does not share its control unit with other devices has a single-unit device controller number associated with it. A device controller that operates more than one device has a block of 16 device numbers assigned to it. The two forms of device numbers are:

Single devices

0	Device Number
8	9 10 11 12 13 14 15

Multiunit devices

1	DC	D
8	9 10 11	12 13 14 15

For single devices bits 11-15 of the device number are the channel number. Bits 9 and 10 identify one of four devices for that channel (only one may be active at a time). For multiple device controllers, bits 9-11 identify the device controller and bits 12-15 identify the individual device to be used with that controller.

For a SIGMA 3 system with either an EIOP or an IIOP (but not both), each of the first eight I/O channels can accommodate up to four single device controllers (one active at a time) or one multiple device controller. The remaining channels (four IIOP or eight EIOP) can be used only by single device controllers.

Out of a possible 28 I/O channels within a system containing both EIOPs and IIOPs, only the initial eight channels will support either single device controllers (again, up to four with operation of one at a time) or a multiple device controller. When used to service multiple device controllers, eight channels may be distributed on the EIOP, IIOP, or both, depending on their cabling. The remaining channels may only be used for single device controllers.

The device number of a given device determines which I/O channel registers are used to control the data transmission to and from that device. Table 5 illustrates this relationship.

I/O CONTROL DOUBLEWORDS

During an I/O operation, the I/O channel registers contain an I/O Control Doubleword (IOCD), which has the following format:

Word address	E	DC	I	Byte count
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	16	17	18	19 20 21 22 23 24 25 26 27 28 29 30 31

The doubleword is contained in the two registers associated with each I/O channel. The even-numbered register contains the word address of the I/O table being operated on. The odd-numbered register contains a count of the number of bytes involved in the I/O operations, as well as three flags. The first bit (E) is an error flag, which is set to 1 if any parity errors are detected on bytes received during an input operation, or if a memory parity error is detected during an output operation. The remaining two bits called the data chaining flag (DC) and the interrupt flag (I),

Table 5. IOP Channel Register and Channel-Device-Controller Numbers (Hexadecimal)

Contents of Bits 8-15					Channel Register		
Single Device Controllers				Single or Multiple Device Controllers	Word Address	Byte Address	Notes
IIOP							
00	20	40	60	8X	08	09	Basic Channels
01	21	41	61	9X	0A	0B	
02	22	42	62	AX	0C	0D	
03	23	43	63	BX	0E	0F	
04	24	44	64	CX	10	11	Optional
05	25	45	65	DX	12	13	
06	26	46	66	EX	14	15	
07	27	47	67	FX	16	17	
08	28	48	68		18	19	
09	29	49	69		1A	1B	
0A	2A	4A	6A		1C	1D	
0B	2B	4B	6B		1E	1F	
EIOP							
0C	2C	4C	6C	8X	20	21	Basic Channels
0D	2D	4D	6D	9X	22	23	
0E	2E	4E	6E	AX	24	25	
0F	2F	4F	6F	BX	26	27	
10	30	50	70	CX	28	29	
11	31	51	71	DX	2A	2B	
12	32	52	72	EX	2C	2D	
13	33	53	73	FX	2E	2F	
14	34	54	74		30	31	Optional
15	35	55	75		32	33	
16	36	56	76		34	35	
17	37	57	77		36	37	
18	38	58	78		38	39	
19	39	59	79		3A	3B	
1A	3A	5A	7A		3C	3D	
1B	3B	5B	7B		3E	3F	
1C	3C	5C	7C	Device numbers unassigned and unused by SIGMA 3			
1D	3D	5D	7D				
1E	3E	5E	7E				
1F	3F	5F	7F				

specify actions to be taken by the I/O system when the transmission controlled by the IOCD is complete. A data chaining flag of 0 indicates that no further transmission is required after the current operation. When the byte count is reduced to zero, the device is told (via a signal called "count done") that the operation is over and that it should neither send nor receive more data but should terminate its operation. At the conclusion of an I/O operation, when all data has been transmitted and all checking associated with the data record has been performed, the device generates a "channel end" signal. At the time of channel end, the device transmits a byte of status information, called the operational status byte (explained later), that is loaded into the even-numbered I/O channel register associated with the device, replacing the word address in the IOCD. The device controller may also generate an "unusual end" signal in place of or in conjunction with "channel end". The actions caused in SIGMA 3 are the same as for "channel end", except that the Operational Status Byte (see below) contains different information. "Unusual end" may occur at any time during an I/O operation, and causes termination of all I/O operations for the device controller involved; the data chaining flag is ignored.

During normal operation, if data chaining is specified by the DC flag, then (instead of notifying the device, via the "count done" signal, that no further transmission is to take place when the byte count reaches zero) the I/O system automatically fetches a new IOCD from the doubleword location immediately following the current I/O table, and loads it into the I/O channel registers in place of the previous IOCD. Data transmission continues as before, but under control of the new IOCD (see Figure 5).

If the interrupt (I) flag is set (1), the SIGMA 3 I/O system will instruct the device controller to generate an interrupt request under the conditions listed below. This will cause an I/O interrupt. The proper program response must include an AIO instruction to determine which device controller is interrupting (with highest priority), and the reason for the interrupt. The two possible reasons are:

1. "Channel end" or "unusual end" is generated in the Operational Status Byte; or
2. The byte count reaches zero and the data chaining flag is set.

OPERATIONAL STATUS BYTE

At the conclusion of the I/O operation, the device transmits the operational status byte to the CPU, which loads the status byte into bit positions 0-7 of the even-numbered I/O channel register associated with the device and loads zeros into the remainder of the register. (The loading of the operational status byte occurs even if channel end is signalled in the middle of an I/O table transmission.) The operational status byte contains five flags, as shown in the following diagram.

T E	I L	C M	C E	U E	0	0	0
0	1	2	3	4	5	6	7

<u>Bit</u> <u>Function</u>

0[†] The transmission error (TE) flag is set to 1 if the device or the device controller has detected any errors during the operation. This includes such errors as parity check on magnetic tape, the parity check at the end of a RAD sector, and memory parity error on an output operation.

1[†] The incorrect length (IL) flag indicates whether (1) or not (0) the input or output record contained the number of bytes specified by the controlling IOCD's byte count. Incorrect length may or may not be considered an error, depending on the type of operation performed. For example, during a card read operation, if a byte count of 80 is specified, then the length is correct, because only 80 bytes can be read from the card in the EBCDIC format. If, however, a count of 75 bytes is specified, the card reader will receive a count done signal before it reaches the end of the card, which causes the incorrect length flag to be set to 1. Similarly, if the reader detects the end of the card before it reaches a count done signal, the incorrect length flag is set to 1.

2[†] The chaining modifier (CM) flag is set to 1 by some devices to indicate that a special condition has been encountered. For example, the unbuffered card punch requires the output image to be transmitted 12 times, once for each row. After the twelfth row is punched, the punch controller sets the chaining modifier flag to 1 to indicate that the last transmission has been received and that no further transmissions are required for the current card. The chaining modifier may be used in different ways by other devices.

3 The channel end (CE) flag is set to 1 at the conclusion of every error free I/O operation, to indicate that all data involved in the operation have been transmitted and all checking associated with the data has been performed.

4 The unusual end (UE) flag is set to 1 if the operation terminated because of some unusual condition. The unusual condition may or may not be an erroneous or faulty condition; in any event, it is not a normal termination. For example, a magnetic tape Read operation that encountered an end-of-file record instead of a data record would produce an unusual end condition. A faulty operation such as a card jam in the middle of a card-reading operation would also produce unusual end. If the UE flag is set, the state of the CE flag is not specified.

5-7 These bits are always loaded as zeros.

[†] These functions are not necessarily implemented in all peripheral device controllers. Refer to peripheral device reference manuals for more complete information.

DEVICE ORDERS

When a device is started for an input/output operation, it first requests an order from the I/O system to determine what operation is to be performed. A device order is a byte transmitted to the device under control of the channel to which the device is attached. The orders that may be accepted by a device are: Write, Read, Read Backward, Control, Sense, and Stop. The code format for each order is shown in the following table. Bit positions marked "M" specify unique modifications that depend on the device to which the order is sent.

	Bit position of device order byte							
Order	0	1	2	3	4	5	6	7
Write	M	M	M	M	M	M	0	1
Read	M	M	M	M	M	M	1	0
Read Backward	M	M	M	M	1	1	0	0
Control	M	M	M	M	M	M	1	1
Sense	M	M	M	M	0	1	0	0
Stop	I	0	0	0	0	0	0	0

The device orders operate in the following manner:

1. Write. The Write order causes the device controller to initiate an output operation. The controller makes output requests to the I/O system and bytes are transmitted from memory, under control of the IOCD, to the device. The output operation normally continues until no further data chaining is to take place and the byte count of the last IOCD is reduced to zero. At this time, the channel signals count done and the device generates channel end. Channel end occurs when the device has received all information associated with the output operation, has generated all checking information, and (if possible) has performed all post-write checking. It is possible for some devices to generate channel end before count done is received.

2. Read. The Read order causes the device to initiate an input operation. Bytes are transmitted by the device, then stored in memory under control of the IOCD. The input operation continues until the device generates channel end or until the byte count is reduced to zero and count done is signalled to the device. In either case, the operation is eventually terminated by a channel end signal when all checking has been performed on the input record.

3. Read Backward. The Read Backward order can be executed only by certain peripheral devices. The Read Backward order causes the device that can execute it to start operation in a backward direction and to transmit bytes; however, the record appears in memory in reverse sequence from the way it was originally written.

4. Control. The Control order is used to initiate special operations by the device. For some operations, the Control order itself may be sufficient to specify the entire operation to be performed. With magnetic tape operations, for example, the Control order initiates such operations as rewind, backspace record, backspace file, space record, etc. These orders can all be specified by the modifier (M) bits of the Control order. If, however, the controller requires additional information for a particular operation, it is provided by the same IOCD that controls the transmission of the Control order. When all data necessary for the operation have been transmitted (and, in some cases when the operation itself is complete), the device controller signals channel end.

5. Sense. The Sense order causes the device to transmit one or more bytes of information describing its current operational status. These bytes are stored in memory under control of the IOCD. The type of status information that may be transmitted is a function of each individual device.

6. Stop. The Stop order (interpreted by some devices) causes a device to terminate its operation immediately. The I modifier bit (in position 0 of the Stop order) indicates that the device is to trigger the I/O interrupt level at the time it receives the Stop order. Bit positions 1, 2, and 3 of the Stop order are ignored.

I/O OPERATIONS

All I/O operations are performed to or from an I/O table, which may be in any arbitary region of memory. An I/O table consists of two or three parts, depending on the type of operation to be performed. The IOCD controlling the first I/O table must be loaded into the I/O channel registers by the program. A specific configuration of the WRITE DIRECT instruction is used to transfer information from the accumulator to the I/O channel registers (see Chapter 3, "Direct Control Instructions").

The first I/O table always contains an order byte in the first word of the table. If an even number of data bytes is to be transmitted for a given operation, then the order byte must appear in bit positions 8-15 of the first word of the table (in which case bits 0-7 are ignored). If an odd number of bytes is involved in the operation, the order byte must appear in bit positions 0-7 of the first word, and the first data byte in bit positions 8-15. In either case, the data bytes follow the order byte, as shown in Figure 5. The byte count in the first IOCD includes the order byte and all the data in the first I/O table. The data portion of an I/O table is always present for a data transmission operation, but may be absent for an operation initiated by a Control or Stop order.

Note that the interrupt bit should always be set (as shown) if an I/O interrupt is desired in the event of unusual end.

In the example shown in Figure 5, an interrupt will occur when data chaining occurs. A TIO instruction will establish that the controller is still busy, and hence the interrupt is known to signal data chaining (zero byte count) rather than unusual end or channel end.

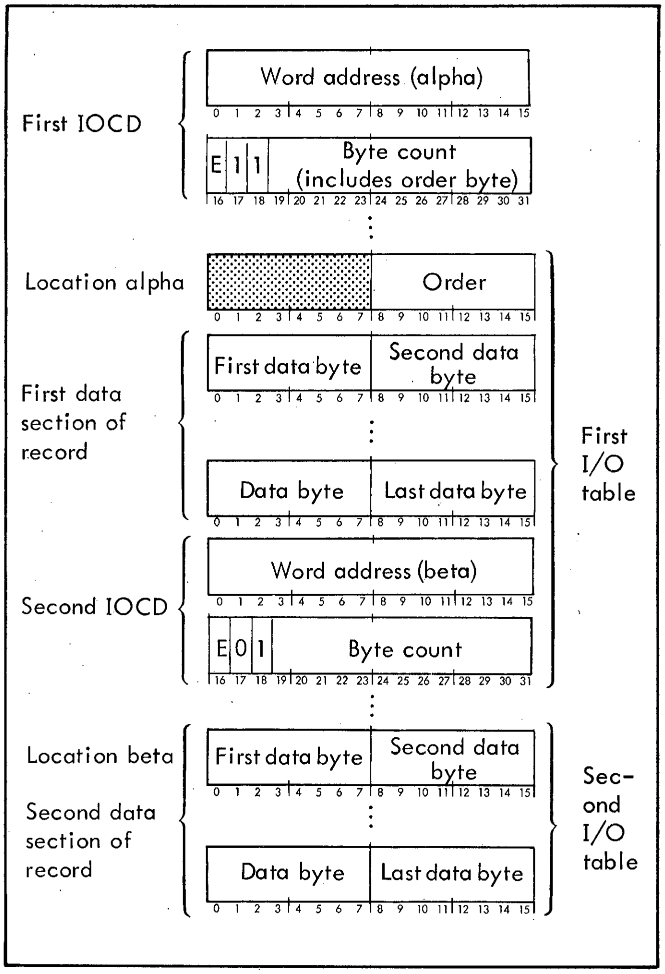


Figure 5. I/O Control Doublewords and I/O Tables

If data chaining is called for, then the I/O table is followed immediately by a second IOCD that specifies a new starting address, new byte count, and new data chaining and interrupt flags. The bytes of the second IOCD are not included in the byte count of the first IOCD. All I/O tables after the first begin with data and do not include an order byte. They may begin in the left- or right-hand byte positions, depending on whether the table contains an even or odd number of bytes, respectively. If data chaining is to take place again, then the second I/O table is assumed to be followed immediately by a new IOCD.

DEVICE INTERRUPTS

All device controllers (and in the case of multiunit devices, the devices themselves) can generate a device interrupt. Each device remembers that it has generated an interrupt so that when the instruction ACKNOWLEDGE I/O INTERRUPT (AIO) is executed, the device with the highest priority identifies itself to the program. Device interrupts are generated by the device at the time of data chaining or at unusual end or channel end if the interrupt (I) flag in the controlling IOCD is set to 1. The interrupt flag is inspected by the I/O system at channel end time, unusual end time, and at data chaining time.

In addition to these normal times for interrupts, some devices can accept a Control order (or even a Read or Write order) that directs the device to interrupt after the transmission operation is completed. This type of interrupt generally occurs at channel end (that time during the operation of the device when all mechanical motion associated with a previously initiated operation has been completed). For example, a magnetic tape unit can be directed (with a Control order) to rewind and to interrupt when the rewind is complete. The order is accepted and channel end is generated immediately after the rewind operation begins. The device remembers the necessity to interrupt and, when the load point is encountered, the tape stops, and channel end occurs; at this time the device generates an interrupt (and holds the interrupt-pending status until it is acknowledged). In this case, the magnetic tape control unit may be busy controlling the operation of another device for a read or write function. The pending device interrupt is a status condition that can be read by I/O instructions.

I/O INSTRUCTIONS

The CPU initiates and controls I/O operations using five instructions:

- Start Input/Output (SIO)
- Test Input/Output (TIO)
- Test Device (TDV)
- Halt Input/Output (HIO)
- Acknowledge I/O Interrupt (AIO)

These instructions are internal control functions of the READ DIRECT instruction. All instructions except AIO require a device number in bit positions 8-15 of the accumulator when the instruction is executed.

If these I/O instructions are executed for devices that are on the EIOP, the execution of these instructions interferes with normal I/O processing by the EIOP. Therefore short program loops which repetitively execute these instructions should be avoided.

SIO START INPUT/OUTPUT

1	0	4	1
0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

SIO is used to initiate an input or output operation with the device selected by the device number contained in bit positions 8-15 of the accumulator. If a device recognizes the number, it returns its device status byte into positions 0-7 of the accumulator.

The overflow and carry indicators are set or reset, according to the result of the instruction, as follows:

O	C	Significance
0	0	I/O address recognized and SIO accepted
0	1	I/O address recognized but SIO not accepted
1	1	I/O address not recognized

Affected: $(A)_{0-7}$, O, C   Timing: See Appendix B

TIO TEST INPUT/OUTPUT

1	0	4	2
0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

TIO causes the device whose device number is in bit positions 8-15 of the accumulator to make the same responses it would make to an SIO instruction, except that the device is not started nor is its state altered. If a device recognizes the device number, it returns its device status byte to positions 0-7 of the accumulator.

The overflow and carry indicators are set or reset, according to the result of the instruction, as follows:

O	C	Significance
0	0	I/O address recognized and SIO can be accepted
0	1	I/O address recognized but SIO can not be accepted
1	1	I/O address not recognized

Affected: $(A)_{0-7}$, O, C Timing: See Appendix B

TDV TEST DEVICE

1	0	4	4
0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

TDV is used to obtain specific information about the device whose device number is contained in bit positions 8-15 of the accumulator. If a device recognizes the device number, it returns its device status byte to positions 0-7 of the accumulator.

The overflow and carry indicators are set or reset, according to the result of the instruction, as follows:

O	C	Significance
0	0	I/O address recognized
0	1	I/O address recognized and device-dependent condition is present
1	1	I/O address not recognized

Affected: $(A)_{0-7}$, O, C Timing: See Appendix B

HIO HALT INPUT/OUTPUT

1	0	4	8
0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

HIO causes the device whose device number is in bit positions 8-15 of the accumulator to stop its current operation immediately. The HIO instruction may cause the device to terminate improperly. In the case of magnetic tape units, for example, the device is forced to stop whether it has reached an inter-record gap or not. A pending interrupt within the device will be reset. If a device recognizes the device number, it returns its device status byte to positions 0-7 of the accumulator.

The overflow and carry indicators are set or reset, according to the result of the instruction, as follows:

O	C	Significance
0	0	I/O address recognized and the device controller is not "busy".
0	1	I/O address recognized and the device controller was "busy" at the time of the hal-
1	1	I/O address not recognized

Affected: $(A)_{0-7}$, O, C Timing: See Appendix B

AIO ACKNOWLEDGE I/O INTERRUPT

1	0	5	0
0 1 2 3	4 5 6 7	8 9 10 11	12 13 14 15

AIO is used to acknowledge an interrupt generated by an I/O device. It causes the highest-priority device to identify itself and return not only status, but its device number. If any devices have interrupts pending, the highest-priority device clears its pending interrupt and returns its status (which is loaded into positions 0-7 of the accumulator) and its device number (which is loaded into positions 8-15).

The overflow and carry indicators are set or reset, according to the result of the instruction as follows:

O	C	Significance
0	0	Normal interrupt recognition
0	1	Unusual interrupt recognition
1	1	No interrupt recognition

Affected: (A), O, C Timing: See Appendix B

DEVICE STATUS BYTE

As the result of executing an I/O instruction, if there is a device whose number corresponds to the number in the accumulator, its Device Status Byte is loaded into positions 0-7 of the accumulator. (The device number in bits 8-15 is not altered.)

The AIO instruction does not require the device number, since one of its functions is to obtain the number of the device that triggered the I/O interrupt level.

The overflow and carry indicators are set to record the nature of the response to all I/O instructions. The I/O status loaded into the accumulator by the I/O instructions is summarized in Table 6.

For the instructions SIO, TIO, and HIO the status indicators have the following meaning:

Device Interrupt Pending. Bit 0 indicates whether (if it is a 1) or not (if it is a 0) the device has generated an interrupt signal that has not yet been acknowledged. A new I/O operation cannot be initiated on this device until the pending interrupt signal has been acknowledged by means of an AIO instruction.

Table 6. Device Status Byte

Position and state in A								
0	1	2	3	4	5	6	7	Significance for SIO, HIO, TIO[†]
1	-	-	-	-	-	-	-	device interrupt pending
-	0	0	-	-	-	-	-	device ready
-	0	1	-	-	-	-	-	device not operational
-	1	0	-	-	-	-	-	device unavailable
-	1	1	-	-	-	-	-	device busy
-	-	-	0	-	-	-	-	device manual
-	-	-	1	-	-	-	-	device automatic
-	-	-	-	1	-	-	-	device unusual end
-	-	-	-	-	0	0	-	device controller ready
-	-	-	-	-	0	1	-	device controller not operational
-	-	-	-	-	1	0	-	device controller unavailable
-	-	-	-	-	1	1	-	device controller busy
-	-	-	-	-	-	-	0	unassigned

Position and state in A								
0	1	2	3	4	5	6	7	Significance for TDV, AIO
1	-	-	-	-	-	-	-	unique to the device (see peripheral device reference manuals)
-	1	-	-	-	-	-	-	
-	-	1	-	-	-	-	-	
-	-	-	1	-	-	-	-	
-	-	-	-	1	-	-	-	
-	-	-	-	-	1	-	-	
-	-	-	-	-	-	1	-	
-	-	-	-	-	-	-	1	

Device Condition.[†] Bits 1 and 2 describe which of four possible conditions the device is currently in. The device conditions are:

00 Device ready. The device can accept and act upon an SIO instruction if no device interrupt is pending.

01 Device not operational. A nonoperational device does not accept an SIO instruction. It requires operator intervention before any action can be taken with regard to its operation.

10 Device unavailable.

11 Device busy. The device has accepted an SIO instruction and has not yet concluded the operation.

Device Mode. Bit 3, the mode status indicator, is a 1 if the operator has cleared the device for operation and has actuated the START switch, placing the device in the "automatic" mode. If the mode status indicator is a 0, the device is in the "manual" mode and requires operator intervention before it can operate. A ready device in the "manual" mode can accept an SIO instruction even though it cannot begin to operate until it is placed in the "automatic" mode. Some devices are "permanently" in the automatic mode.

Unusual End Termination. Bit 4 is set to 1 if the previous operation on this device resulted in an unusual end; otherwise, bit 4 is reset to 0.

Device Controller Condition. Bits 5 and 6 describe which of four possible conditions the device controller is currently in. These conditions are identical in meaning to the device conditions. The controller need not be in the same condition as the device, in the case of a multi-unit device controller. The device controller conditions are:

00 Device controller ready. If the controller is ready and the device is ready, an SIO instruction can be accepted.

01 Device controller not operational.

10 Device controller unavailable.

11 Device controller busy. The controller and the device connected to it (or one of the devices connected to it) have accepted an SIO instruction and the I/O operation thus initiated has not terminated.

Note that, in addition to the Device Status Byte in positions 0-7, the instruction AIO also causes the device number to be loaded into positions 8-15 of the accumulator.

EXTERNAL INTERFACE SYSTEM

In addition to the byte-oriented I/O system, SIGMA 3 has an additional I/O capability. With the incorporation of the optional External Interface System, the READ DIRECT and WRITE DIRECT instructions can be used to communicate with special system devices. WRITE DIRECT can be used to transmit a control signal, along with 16 data bits, to a device. Similarly, READ DIRECT can be used to transmit a control signal and then accept 16 data bits from the external unit. Both instructions can be used to obtain a 2-bit status response from the device.

[†] For single-device controllers, bits 1-2 and 5-6 are identical. Some devices only differentiate between the "ready" and "busy" states, rather than identifying four distinct states.

When the External Interface Feature is installed, the WRITE DIRECT instruction can set up the 16 control lines plus the 16 data lines; these remain stable until an acknowledgment signal is received from the device. A delay by the device in responding to WRITE DIRECT does not have any adverse effect on the operation of the byte-oriented I/O system.

The READ DIRECT instruction operates in a similar fashion. The 16 control lines are held stable and the device responds with its acknowledge signal and 16 data bits. The interface is sometimes referred to as the Direct Input/Output (DIO) interface. XDS publication 90 09 73 (Interface Design Manual) describes this interface in detail.

PROTECT OFF position, the central processor ignores the physical positions of certain switches and, instead, operates as if the switches were in specific positions. The affected switches and their "locked" positions are:

Switch	Locked Condition
COMPUTE	RUN
MEMORY MODE	NORMAL
ADDRESS	NORMAL
CLOCK	CONT
RESET	NOT RESET
INTERRUPT	NORMAL
PARITY ERROR	INTRPT
LOAD	NOT LOAD
TIMER	NORMAL
AUTO RESET	OFF
INSTRUCTION	OFF
OPERAND	OFF

PROTECT

The PROTECT switch is a 2-position latching toggle switch that controls the operation of the memory protect option. The protection interrupt system is operative only if the option is installed and the PROTECT switch is in the ON position. If the PROTECT switch is in the OFF position, the protection system is inoperative. The PROTECT switch does not affect the operation of the computer in any way if the protect option is not installed.

Either position of the PROTECT toggle switch may be overridden by the keylock switch. If the keylock switch is in the PROTECT ON position, the protection function is enabled, regardless of the state of the PROTECT toggle switch. If the keylock switch is in the PROTECT OFF position, the protection function is disabled, regardless of the state of the PROTECT switch.

DISPLAY

The DISPLAY switch is a 2-position latching toggle switch that controls the display of selected data from either the CPU or EIOP. The EIOP display is for maintenance only. The data is displayed in the row of 16 indicator lights below the DISPLAY switch and is selected by the two rotary switches SELECT and REG ADDRESS explained below.

SELECT

The SELECT rotary switch is used to inspect data in various registers. To inspect the general registers (A, E, X1, etc.), the switch is moved to the REG position and the appropriate register is selected from the REG ADDRESS switch. Registers H, S, and D may also be displayed by moving the SELECT switch directly to their respective positions (see discussion below of DATA switch and DATA toggle switches for means of manipulating data in these registers). All other positions of the SELECT switch are primarily for maintenance use.

The 4 x 16 matrix of abbreviations and acronyms immediately above the DATA toggle switches and indicator lights is primarily for maintenance use.

REG ADDRESS

REG ADDRESS is a latching rotary switch that can be used to display the contents of the various fast memory registers (A, E, X, L, etc.) and the channel registers for channels zero and one (CH0 and CH1). The SELECT rotary switch must be in the REG position before the REG ADDRESS switch will work effectively. This switch is only operative when the COMPUTE switch is in the IDLE position.

PARITY ERROR

The parity error feature is composed of a 3-position latching toggle switch and an indicator light. If the switch is in the CONT position, all memory parity errors are ignored.

If the PARITY ERROR switch is in the HALT position when a memory parity error occurs, the CPU is halted and the PARITY ERROR indicator is lit. The CPU cannot be interrupted while in a halted condition. This condition may be removed by either moving the PARITY ERROR switch to the CONT position or by pressing the RESET pushbutton switch.

If the PROTECT option is installed and the PARITY ERROR switch is in the INTERRUPT position, the occurrence of a memory parity error will cause the CPU to enter a wait state and the machine fault interrupt to be set. An instruction may be aborted. The indicator will be lit. If the CPU is not equipped with the PROTECT option, placing the PARITY ERROR switch in the INTERRUPT position will only cause the CPU to enter a wait state when a parity error occurs.

When a parity error occurs during IIOP or EIOP memory accesses, the only action taken is the setting of the TE bit in the appropriate IOCD. This is the only action taken, regardless of the setting of the PARITY ERROR switch.

INTERFACE TIMER

The TIMER switch is a 2-position latching toggle switch. If the switch is in the NORMAL position, timer runouts (5.2 μsec for DC (device controller) communications and 42 μsec for direct I/O communications) will initiate remedial actions (if the PROTECT option is installed). A direct I/O communication can be interrupted by an IIOP service-call if no response has been received from the DIO device, and if the interface timer has not expired. In this instance, the DIO communication is renewed after the I/O service is completed and is treated as a new command with another 42 μsec allowed before timer runout. The effect of certain control panel switches at timer runout is shown in Table 7.

Table 7. Interface Timer Actions

TIMER Switch	Key Switch			Action at Timer Runout
	UNLOCK	PROTECT OFF	PROTECT ON	
OVERRIDE	X			CPU/IIOP hang-up
NORMAL	X			Normal interface timer action taken
-		X		
-			X	

If the Protect option is not installed, the CPU will enter a wait if the timer runs out and the TIMER switch is in the NORMAL position.

INTERRUPT (Toggle)

The INTERRUPT switch is immediately to the right of the TIMER switch. It is a 3-position latching switch. When in the DIAGNOSTIC position, a memory parity error will trigger the console interrupt if it is armed and enabled. This position also enables the RUN-DEPOSIT-INCREMENT operation (see Table 8).

If the switch is in the TRACE position, a console interrupt will be triggered (if armed and enabled) by the end of each instruction.

If the switch is in the NORMAL position, a console interrupt will be triggered (if armed and enabled) whenever the INTERRUPT (see below) momentary switch is pushed.

INTERRUPT (Pushbutton)

The INTERRUPT switch is a momentary pushbutton switch that will trigger a console interrupt when pushed if the console interrupt has been previously armed and enabled.

CLOCK

The CLOCK switch is a 3-position toggle switch that controls the clock to the CPU and to the EIOP. If in the CONTINUE position (latching), the clocks run continuously if the COMPUTE switch is in the RUN position. When the CLOCK switch is moved to the STOP position (latching), the CPU and EIOP clocks are stopped at the next normal machine step. If the COMPUTE switch is in the RUN position and the CLOCK switch is depressed to the STEP (momentary) position, the CPU and EIOP clocks are advanced one phase (single clocked).

RESET

The RESET switch is a momentary pushbutton switch that will, when pressed, produce a signal that causes the initialization of the CPU, memory, IOPs, and peripheral devices.

RESET has no effect on the S register, but resets the P register. To start a program at location 0 after RESET, move the COMPUTE switch from IDLE to RUN. To start a program at the location addressed by the S register after RESET, manually fetch from memory the contents of S and then move the COMPUTE switch to RUN.

LOAD

The LOAD switch is a momentary pushbutton switch used in the bootstrap loading program (see below). This switch is only operative when the COMPUTE switch is in the IDLE position.

NORMAL MODE

The NORMAL MODE indicator light will be lit if all of the following conditions exist:

Switch	Position
POWER	ON
TIMER	NORMAL
MEMORY MODE	NORMAL
ADDRESS	NORMAL
CLOCK	CONT
INTERRUPT	NORMAL
AUTO RESET	OFF
INSTRUCTION	OFF
OPERAND	OFF
PT16B MARGIN	NORMAL
PT17B BREAKER	ON

SIGMA 3 OPERATIONS

The ADDRESS, MEMORY MODE, and COMPUTE switches operate as a functional group. All switches are toggle latching with the exception of the STEP position of the COMPUTE switch, which is momentary. The interaction of these switches is illustrated in Table 8 below.

RUN

The RUN indicator, when lit, indicates that the machine clocks are running.

DATA

The DATA switch is a momentary switch that is operative only when the COMPUTE switch is in the IDLE position. When depressed to the ENTER position, the value represented by the data switches (a bank of sixteen 2-position latching switches located between REG ADDRESS switch and the DATA switch) will be entered into the register selected for display. Elevating the switch to the CLEAR position (momentary) will clear the contents of the register selected. DATA will clear or enter data in the following: H, S, D and all general registers in the CPU and the first two I/O channels in the IIOP.

Table 8. ADDRESS, MEMORY MODE, and COMPUTE Switch Group

COMPUTE Switch	MEMORY Switch	ADDRESS Switch	SIGMA 3 Operation
RUN	NORMAL	NORMAL	Normal program execution.
RUN	NORMAL	HOLD	Repetitively execute the same instruction.
RUN	NORMAL	INCREMENT	Normal program execution.
RUN	FETCH	NORMAL	Repetitively read into the D register the contents of the memory location addressed by the S register.
RUN	FETCH	HOLD	
RUN	FETCH	INCREMENT	Scan memory, in wraparound fashion reading location contents into the D register on each memory cycle. Stop on memory parity error if PARITY ERROR switch is in HALT position or when RUN is switched to IDLE. Ignore memory parity check if PARITY ERROR switch is in INTERRUPT or CONT position.
RUN	DEPOSIT	NORMAL	Repetitively write the contents of the DATA switches into the memory location addressed by the address previously entered into the S register.
RUN	DEPOSIT	HOLD	
RUN	DEPOSIT	INCREMENT	If the INTERRUPT switch is on NORMAL or TRACE, memory is not altered. If the INTERRUPT switch is on DIAGNOSTIC, the contents of data switches are written throughout memory, in a wraparound fashion. Stop when COMPUTE is switched to IDLE.
STEP	-	-	Same as RUN except that all operations are performed one instruction at a time with each depression to STEP.
IDLE	-	-	No instructions or interrupts or I/O service calls are executed by the IIOP. If the clock switch is in CONTINUE position, the machine is in IDLE state, i.e., the next instruction has been fetched into the D register and the program register (P) and register S are pointing to this address.

INDICATOR LIGHTS

There are 10 permanent indicator lights located between the REG ADDRESS switch and the RESET pushbutton. They are in vertical alignment with the data toggle switch below. The function they indicate when lighted and their bit position in the PSD are listed in Table 9 below.

Table 9. Indicator Lights

Name	Bit Position	Function
IO	0	I/O Service Call to the IIOP is taking place.
INT	1	CPU is entering an interrupt routine or processing a single instruction interrupt.
WAIT	2	CPU is in WAIT state.
SW1 and SW2	3 and 4	Machine internal indicators.
PP	8	Machine is operating in a protected area of memory (always lit if the Protect Option is not installed and the PROTECT toggle switch is OFF).
II	10	Internal interrupt inhibit.
EI	11	External interrupt inhibit.
O	14	Overflow.
C	15	Carry.

OPERATING PROCEDURES

BOOTSTRAP LOADING PROGRAM

The operator may cause an initial loading operation to be performed by the CPU in order to set up a new program in the machine. To do so, the operator performs the following actions:

1. Move the key-operated switch to UNLOCKED and the PROTECT switch to OFF. Set data switches 8-15 to the number of the device from which the initial program is to be loaded. Data switch 7 is set if the device is on the EIOP.
2. Move the COMPUTE switch to IDLE. This action stops the computer from further execution of instructions.
3. Actuate the RESET switch. This action clears all internal CPU indicators, sets up the SIO instruction in the D register, stops all peripheral devices, and causes devices such as mass memories or disc files to clear their starting address registers.

4. Actuate the LOAD switch, setting the load condition indicator within the computer.

5. Move the COMPUTE switch to RUN. This action causes the computer to execute the SIO instruction in the D register and then enter the "wait" condition. The P and S registers are cleared. This SIO uses bits 8-15 of the data switches as the device number, and then loads it and the I/O status information into the accumulator. No memory reference is made to fetch an order from an I/O table; instead, the central processor, or the EIOP if data switch 7 is set, generates a read order (X'02') and an input/output control doubleword (IOCD) of the form X'0000 0080' which specifies location 0 as a starting address and a byte count of X'80' (128 bytes).

6. Wait for the first record to be read from the selected input device. While the operator waits for the first record to be loaded, the following action takes place:

   The device selected by the device number in the accumulator has started and has received its first order, which is a Read order. The device then transmits the initial record, which is stored in core memory beginning at location 0 and continuing through location X'3F' (a total of 64 words, if the first record on the selected device is that long). When the first record has been read, the device generates channel end and stops. No data chaining occurs and the I/O interrupt level is not triggered; however, the operational status byte is loaded into the even-numbered I/O channel register associated with the device number, and bit 0 of the odd-numbered I/O channel register is set to 1 if a parity error occurred during the input operation. When the operator observes the input device stop, he may then proceed to step 7.

7. Move the COMPUTE switch to IDLE and then back to RUN for execution of the loaded program, beginning with location 0. From this point on, the computer is under control of the program just loaded into memory.

FETCHING AND STORING DATA

To fetch data from a memory location and display it:

1. Set COMPUTE switch to IDLE.
2. Set DATA toggle switches to desired address.
3. Set SELECT rotary switch to S.
4. Depress DATA switch to ENTER.
5. Set MEMORY MODE switch to FETCH.
6. Depress COMPUTE switch to STEP.
7. Turn SELECT switch to D; contents of the designated memory location will be displayed in the lights immediately above the DATA toggle switches.

To fetch and display data from successive memory locations:

1. Execute steps 1-7 listed above.
2. Set ADDRESS switch to INCR.
3. Momentarily depress COMPUTE switch to STEP.

Memory locations are incremented by one with each depression to STEP.

To store data in a designated memory location:

1. Set COMPUTE switch to IDLE.
2. Set DATA toggle switches to desired address.
3. Set SELECT rotary switch to S.
4. Depress DATA switch to ENTER.
5. Turn SELECT rotary switch to D.
6. Set DATA toggle switches to desired storage value.
7. Depress DATA switch to ENTER.
8. Set MEMORY MODE switch to DEPOSIT.
9. Depress COMPUTE switch to STEP.

Care should be taken to return the ADDRESS and MEMORY MODE switches to their NORMAL positions before returning the COMPUTE switch to RUN. Otherwise, successive locations are altered.

APPENDIX A. REFERENCE TABLES

This appendix contains the following reference material:

XDS STANDARD SYMBOLS AND CODES

The symbol and code standards described in this publication are applicable to all XDS products, both hardware and software. They may be expanded or altered from time to time to meet changing requirements.

The symbols listed here include two types: graphic symbols and control characters. Graphic symbols are displayable and printable; control characters are not. Hybrids are SP, the symbol for a blank space, and DEL, the delete code which is not considered a control command.

Three types of code are shown: (1) the 8-bit XDS Standard Computer Code, i.e., the XDS Extended Binary-Coded-Decimal Interchange Code (EBCDIC); (2) the 7-bit United States of America Standard Code for Information Interchange (USASCII); and (3) the XDS standard card code.

XDS STANDARD CHARACTER SETS

1. EBCDIC

   57-character set: uppercase letters, numerals, space, and & - / . < > ( ) + | $ * : ; , % # @ ' =

   63-character set: same as above plus ¢ ! _ ? " ¬

   89-character set: same as 63-character set plus lowercase letters

2. USASCII

   64-character set: upper case letters, numerals, space, and ! " $ % & ' ( ) * + , - . / \ ; : = < > ? @ _ [ ] ^ ' #

   95-character set: same as above plus lowercase letters and { } ¦ ~ `

CONTROL CODES

In addition to the standard character sets listed above, the XDS symbol repertoire includes 37 control codes and the hybrid code DEL (hybrid code SP is considered part of all character sets). These are listed in the table titled XDS Standard Symbol-Code Correspondences.

SPECIAL CODE PROPERTIES

The following two properties of all XDS standard codes will be retained for future standard code extensions:

1. All control codes, and only the control codes, have their two high-order bits equal to "00". DEL is not considered a control code.

2. No two graphic EBCDIC codes have their seven low-order bits equal.

XDS STANDARD 8-BIT COMPUTER CODES (EBCDIC)

			Most Significant Digits															
	Hexadecimal		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Least Significant Digits	0	0000	NUL	DLE	ds		SP	&	-	▨								0
	1	0001	SOH	DC1	ss		▨	▨	/	▨	a	j		\ [1]	A	J		1
	2	0010	STX	DC2	fs		▨	▨	▨	▨	b	k	s	{ [1]	B	K	S	2
	3	0011	ETX	DC3	si		▨	▨	▨	▨	c	l	t	} [1]	C	L	T	3
	4	0100	EOT	DC4			▨	▨	▨	▨	d	m	u	[[1]	D	M	U	4
	5	0101	HT	LF NL			Will not be assigned	▨	▨	▨	e	n	v	] [1]	E	N	V	5
	6	0110	ACK	SYN			▨	▨	▨	▨	f	o	w		F	O	W	6
	7	0111	BEL	ETB			▨	▨	▨	▨	g	p	x		G	P	X	7
	8	1000	EOM BS	CAN			▨	▨	▨	▨	h	q	y		H	Q	Y	8
	9	1001	ENQ	EM			▨	▨	▨	▨	i	r	z		I	R	Z	9
	A	1010	NAK	SS			¢ [2]	!	^ [1]	:					▨	▨	▨	▨
	B	1011	VT	ESC			.	$	,	#					▨	▨	▨	▨
	C	1100	FF	FS			<	*	%	@					Will not be assigned	▨	▨	▨
	D	1101	CR	GS			(	)	_	'					▨	▨	▨	▨
	E	1110	SO	RS			+	;	>	=					▨	▨	▨	▨
	F	1111	SI	US	PE		| [2]	¬ [2]	?	"					▨	▨	▨	DEL
			3				4				5							

NOTES:

1 The characters ^ \ { } [ ] are USASCII characters that do not appear in any of the XDSEBCDIC-based character sets, though they are shown in the EBCDIC table.

2 The characters ¢ | ¬ appear in the XDS 63- and 89-character EBCDIC sets but not in either of the XDS USASCII-based sets, However, XDS software translates the characters ¢ | ¬ into USASCII characters as follows:

EBCDIC	=	UASCII
¢		` (6-0)
|		¦ (7-12)
¬		~ (7-14)

3 The EBCDIC control codes in columns 0 and 1 and their binary representation are exactly the same as those in the USASCII table, except for two interchanges: LF/NL with NAK, and HT with ENQ.

4 Characters enclosed in heavy lines are included only in the XDS standard 63- and 89-character EBCDIC sets.

5 These characters are included only in the XDS standard 89-character EBCDIC set.

XDS STANDARD 7-BIT COMMUNICATION CODES (USASCII)

			Most Significant Digits							
	Decimal (rows)	(col's.) →	0	1	2	3	4	5	6	7
	↓	Binary [1]	x000	x001	x010	x011	x100	x101	x110	x111
Least Significant Digits	0	0000	NUL	DLE	SP	0	@	P	`	p
	1	0001	SOH	DC1	! [5]	1	A	Q	a	q
	2	0010	STX	DC2	"	2	B	R	b	r
	3	0011	ETX	DC3	#	3	C	S	c	s
	4	0100	EOT	DC4	$	4	D	T	d	t
	5	0101	ENQ	NAK	%	5	E	U	e	u
	6	0110	ACK	SYN	&	6	F	V	f	v
	7	0111	BEL	ETB	'	7	G	W	g	w
	8	1000	BS	CAN	(	8	H	X	h	x
	9	1001	HT	EM	)	9	I	Y	i	y
	10	1010	LF NL	SS	*	:	J	Z	j	z
	11	1011	VT	ESC	+	;	K	[[5]	k	{
	12	1100	FF	FS	,	<	L	\	l	¦
	13	1101	CR	GS	-	=	M	] [5]	m	}
	14	1110	SO	RS	.	>	N	[4] ^ [5]	n	~ [4]
	15	1111	SI	US	/	?	O	_ [4]	o	DEL
			2		3					

NOTES:

1 Most significant bit, added for 8-bit format, is either 0 or an odd-parity bit for the remaining 7 bits.

2 Columns 0-1 are control codes.

3 Columns 2-5 correspond to the XDS 64-character USASCII set.
Columns 2-7 correspond to the XDS 95-character USASCII set.

4 On many current teletypes, the symbol

^	is	↑	(5-14)
_	is	←	(5-15)
~	is	ESC or ALTMODE control	(7-14)

and none of the symbols appearing in columns 6-7 are provided. Except for the three symbol differences noted above, therefore, such teletypes provide all the characters in the XDS 64-character USASCII set. (The XDS 7015 Remote Keyboard Printer provides the 64-character USASCII set also, but prints ^ as ∧ .)

5 On the XDS 7670 Remote Batch Terminal, the symbol

!	is	|	(2-1)
[	is	¢	(5-11)
]	is	!	(5-13)
^	is	¬	(5-14)

and none of the symbols appearing in columns 6-7 are provided. Except for the four symbol differences noted above, therefore, this terminal provides an the characters in the XDS 64-character USASCII set.

XDS STANDARD SYMBOL-CODE CORRESPONDENCES

EBCDIC[t]	Symbol	Card Code	USASCII[tt]	Meaning	Remarks
00	NUL	12-0-9-8-1	0-0	null	00 through 23 and 2F are control codes.
01	SOH	12-9-1	0-1	start of header	
02	STX	12-9-2	0-2	start of text	
03	ETX	12-9-3	0-3	end of text	
04	EOT	12-9-4	0-4	end of transmission	
05	HT	12-9-5	0-9	horizontal tab	
06	ACK	12-9-6	0-6	acknowledge (positive)	
07	BEL	12-9-7	0-7	bell	
08	BS or EOM	12-9-8	0-8	backspace or end of message	EOM is used only on XDS Keyboard/ Printers Models 7012, 7020, 8091, and 8092.
09	ENQ	12-9-8-1	0-5	enquiry	
0A	NAK	12-9-8-2	1-5	negative acknowledge	
0B	VT	12-9-8-3	0-11	vertical tab	
0C	FF	12-9-8-4	0-12	form feed	
0D	CR	12-9-8-5	0-13	carriage return	
0E	SO	12-9-8-6	0-14	shift out	
0F	SI	12-9-8-7	0-15	shift in	
10	DLE	12-11-9-8-1	1-0	data link escape	
11	DC1	11-9-1	1-1	device control 1	
12	DC2	11-9-2	1-2	device control 2	
13	DC3	11-9-3	1-3	device control 3	
14	DC4	11-9-4	1-4	device control 4	
15	LF or NL	11-9-5	0-10	line feed or new line	
16	SYN	11-9-6	1-6	sync	
17	ETB	11-9-7	1-7	end of transmission block	
18	CAN	11-9-8	1-8	cancel	
19	EM	11-9-8-1	1-9	end of medium	
1A	SS	11-9-8-2	1-10	start of special sequence	
1B	ESC	11-9-8-3	1-11	escape	
1C	FS	11-9-8-4	1-12	file separator	
1D	GS	11-9-8-5	1-13	group separator	
1E	RS	11-9-8-6	1-14	record separator	
1F	US	11-9-8-7	1-15	unit separator	
20	ds	11-0-9-8-1		digit selector	20 through 23 are used with Sigma 7 EDIT BYTE STRING (EBS) instruction — not input/output control codes.
21	ss	0-9-1		significance start	
22	fs	0-9-2		field separation	
23	si	0-9-3		immediate significance start	
24		0-9-4			24 through 2E are unassigned.
25		0-9-5			
26		0-9-6			
27		0-9-7			
28		0-9-8			
29		0-9-8-1			
2A		0-9-8-2			
2B		0-9-8-3			
2C		0-9-8-4			
2D		0-9-8-5			
2E		0-9-8-6			
2F	PE	0-9-8-7		parity error	If parity checking is requested.
30		12-11-0-9-8-1			30 through 3F are unassigned.
31		9-1			
32		9-2			
33		9-3			
34		9-4			
35		9-5			
36		9-6			
37		9-7			
38		9-8			
39		9-8-1			
3A		9-8-2			
3B		9-8-3			
3C		9-8-4			
3D		9-8-5			
3E		9-8-6			
3F		9-8-7			

[t] Hexadecimal notation.

[tt] Decimal notation (column-row).

XDS STANDARD SYMBOL-CODE CORRESPONDENCES (cont.)

EBCDIC[†]	Symbol	Card Code	USASCII[††]	Meanirg	Remarks
40	SP	blank	2-0	blank	
41		12-0-9-1			41 through 49 will not be assigned.
42		12-0-9-2			
43		12-0-9-3			
44		12-0-9-4			
45		12-0-9-5			
46		12-0-9-6			
47		12-0-9-7			
48		12-0-9-8			
49		12-8-1			
4A	¢ or `	12-8-2	6-0	cent or accent grave	Accent grave used for left single quote. On model 7670, ` not available, and ¢ = USASCII 5-11.
4B	.	12-8-3	2-14	period	
4C	<	12-8-4	3-12	less than	
4D	(	12-8-5	2-8	left parenthesis	
4E	+	12-8-6	2-11	plus	
4F	| or ¦	12-8-7	7-12	vertical bar or broken bar	On Model 7670, ¦ not available, and | = ASASCII 2-1.
50	&	12	2-6	ampersand	
51		12-11-9-1			51 through 59 will not be assigned.
52		12-11-9-2			
53		12-11-9-3			
54		12-11-9-4			
55		12-11-9-5			
56		12-11-9-6			
57		12-11-9-7			
58		12-11-9-8			
59		11-8-1			
5A	!	11-8-2	2-1	exclamation point	On Model 7670, ! is |.
5B	$	11-8-3	2-4	dollars	
5C	*	11-8-4	2-10	asterisk	
5D	)	11-8-5	2-9	right parenthesis	
5E	;	11-8-6	3-11	semicolon	
5F	~ or ¬	11-8-7	7-14	tilde or logical not	On Model 7670, ~ is not available, and ¬ = USASCII 5-14.
60	-	11	2-13	minus, dash, hyphen	
61	/	0-1	2-15	slash	
62		11-0-9-2			62 through 69 will not be assigned.
63		11-0-9-3			
64		11-0-9-4			
65		11-0-9-5			
66		11-0-9-6			
67		11-0-9-7			
68		11-0-9-8			
69		0-8-1			
6A	^	12-11	5-14	circumflex	On Model 7670 ^ is ¬. On Model 7015 ^ is ∧ (caret).
6B	,	0-8-3	2-12	comma	
6C	%	0-8-4	2-5	percent	
6D	_	0-8-5	5-15	underline	Underline is sometimes called "break character"; may be printed along bottom of character line.
6E	>	0-8-6	3-14	greater than	
6F	?	0-8-7	3-15	question mark	
70		12-11-0			70 through 79 will not be assigned.
71		12-11-0-9-1			
72		12-11-0-9-2			
73		12-11-0-9-3			
74		12-11-0-9-4			
75		12-11-0-9-5			
76		12-11-0-9-6			
77		12-11-0-9-7			
78		12-11-0-9-8			
79		8-1			
7A	:	8-2	3-10	colon	
7B	#	8-3	2-3	number	
7C	@	8-4	4-0	at	
7D	'	8-5	2-7	apostrophe (right single quote)	
7E	=	8-6	3-13	equals	
7F	"	8-7	2-2	quotation mark	

[†] Hexadecimal notation

[††] Decimal notation (column-row).

XDS STANDARD SYMBOL-CODE CORRESPONDENCES (cont.)

EBCDIC[†]	Symbol	Card Code	USASCII[††]	Meaning	Remarks
80		12-0-8-1			80 is unassigned.
81	a	12-0-1	6-1		81-89, 91-99, A2-A9 comprise the lowercase alphabet. Available only in SDS standard 89- and 95-character sets.
82	b	12-0-2	6-2		
83	c	12-0-3	6-3		
84	d	12-0-4	6-4		
85	e	12-0-5	6-5		
86	f	12-0-6	6-6		
87	g	12-0-7	6-7		
88	h	12-0-8	6-8		
89	i	12-0-9	6-9		
8A		12-0-8-2			8A through 90 are unassigned.
8B		12-0-8-3			
8C		12-0-8-4			
8D		12-0-8-5			
8E		12-0-8-6			
8F		12-0-8-7			
90		12-11-8-1			
91	j	12-11-1	6-10		
92	k	12-11-2	6-11		
93	l	12-11-3	6-12		
94	m	12-11-4	6-13		
95	n	12-11-5	6-14		
96	o	12-11-6	6-15		
97	p	12-11-7	7-0		
98	q	12-11-8	7-1		
99	r	12-11-9	7-2		
9A		12-11-8-2			9A through A1 are unassigned.
9B		12-11-8-3			
9C		12-11-8-4			
9D		12-11-8-5			
9E		12-11-8-6			
9F		12-11-8-7			
A0		11-0-8-1			
A1		11-0-1			
A2	s	11-0-2	7-3		
A3	t	11-0-3	7-4		
A4	u	11-0-4	7-5		
A5	v	11-0-5	7-6		
A6	w	11-0-6	7-7		
A7	x	11-0-7	7-8		
A8	y	11-0-8	7-9		
A9	z	11-0-9	7-10		
AA		11-0-8-2			AA through B0 are unassigned.
AB		11-0-8-3			
AC		11-0-8-4			
AD		11-0-8-5			
AE		11-0-8-6			
AF		11-0-8-7			
B0		12-11-0-8-1			
B1	\	12-11-0-1	5-12	backslash	
B2	{	12-11-0-2	7-11	left brace	
B3	}	12-11-0-3	7-13	right brace	
B4	[	12-11-0-4	5-11	left bracket	On Model 7670, [is ¢.
B5	]	12-11-0-5	5-13	right bracket	On Model 7670,] is !.
B6		12-11-0-6			B6 through BF are unassigned.
B7		12-11-0-7			
B8		12-11-0-8			
B9		12-11-0-9			
BA		12-11-0-8-2			
BB		12-11-0-8-3			
BC		12-11-0-8-4			
BD		12-11-0-8-5			
BE		12-11-0-8-6			
BF		12-11-0-8-7			

[†] Hexadecimal notation.

[††] Decimal notation (column-row).

XDS STANDARD SYMBOL-CODE CORRESPONDENCES (cont.)

EBCDIC[†]	Symbol	Card Code	USASCII[††]	Meaning	Remarks
C0		12-0			C0 is unassigned.
C1	A	12-1	4-1		C1-C9, D1-D9, E2-E9 comprise the uppercase alphabet.
C2	B	12-2	4-2		
C3	C	12-3	4-3		
C4	D	12-4	4-4		
C5	E	12-5	4-5		
C6	F	12-6	4-6		
C7	G	12-7	4-7		
C8	H	12-8	4-8		
C9	I	12-9	4-9		
CA		12-0-9-8-2			CA through CF will not be assigned.
CB		12-0-9-8-3			
CC		12-0-9-8-4			
CD		12-0-9-8-5			
CE		12-0-9-8-6			
CF		12-0-9-8-7			
D0		11-0			D0 is unassigned.
D1	J	11-1	4-10		
D2	K	11-2	4-11		
D3	L	11-3	4-12		
D4	M	11-4	4-13		
D5	N	11-5	4-14		
D6	O	11-6	4-15		
D7	P	11-7	5-0		
D8	Q	11-8	5-1		
D9	R	11-9	5-2		
DA		12-11-9-8-2			DA through DF will not be assigned.
DB		12-11-9-8-3			
DC		12-11-9-8-4			
DD		12-11-9-8-5			
DE		12-11-9-8-6			
DF		12-11-9-8-7			
E0		0-8-2	11-0-9-1		E0, E1 are unassigned.
E1		11-0-9-1			
E2	S	0-2	5-3		
E3	T	0-3	5-4		
E4	U	0-4	5-5		
E5	V	0-5	5-6		
E6	W	0-6	5-7		
E7	X	0-7	5-8		
E8	Y	0-8	5-9		
E9	Z	0-9	5-10		
EA		11-0-9-8-2			EA through EF will not be assigned.
EB		11-0-9-8-3			
EC		11-0-9-8-4			
ED		11-0-9-8-5			
EE		11-0-9-8-6			
EF		11-0-9-8-7			
F0	0	0	3-0		
F1	1	1	3-1		
F2	2	2	3-2		
F3	3	3	3-3		
F4	4	4	3-4		
F5	5	5	3-5		
F6	6	6	3-6		
F7	7	7	3-7		
F8	8	8	3-8		
F9	9	9	3-9		
FA		12-11-0-9-8-2			FA through FE will not be assigned.
FB		12-11-0-9-8-3			
FC		12-11-0-9-8-4			
FD		12-11-0-9-8-5			
FE		12-11-0-9-8-6			
FF	DEL	12-11-0-9-8-7		delete	Special – neither graphic nor control symbol.

[†] Hexadecimal notation.

[††] Decimal notation (column-row).

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

TABLE OF POWERS OF TEN$_{16}$

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	x	16^{-1}
3E8	3	0.4189	374B	C6A7	EF9E	x	16^{-2}
2710	4	0.68DB	8BAC	710C	B296	x	16^{-3}
1 86A0	5	0.A7C5	AC47	1B47	8423	x	16^{-4}
F 4240	6	0.10C6	F7A0	B5ED	8D37	x	16^{-4}
98 9680	7	0.1AD7	F29A	BCAF	4858	x	16^{-5}
5F5 E100	8	0.2AF3	1DC4	6118	73BF	x	16^{-6}
3B9A CA00	9	0.44B8	2FA0	9B5A	52CC	x	16^{-7}
2 540B E400	10	0.6DF3	7F67	5EF6	EADF	x	16^{-8}
17 4876 E800	11	0.AFEB	FF0B	CB24	AAFF	x	16^{-9}
E8 D4A5 1000	12	0.1197	9981	2DEA	1119	x	16^{-9}
918 4E72 A000	13	0.1C25	C268	4976	81C2	x	16^{-10}
5AF3 107A 4000	14	0.2D09	370D	4257	3604	x	16^{-11}
3 8D7E A4C6 8000	15	0.480E	BE7B	9D58	566D	x	16^{-12}
23 86F2 6FC1 0000	16	0.734A	CA5F	6226	F0AE	x	16^{-13}
163 4578 5D8A 0000	17	0.B877	AA32	36A4	B449	x	16^{-14}
DE0 B6B3 A764 0000	18	0.1272	5DD1	D243	ABA1	x	16^{-14}
8AC7 2304 89E8 0000	19	0.1D83	C94F	B6D2	AC35	x	16^{-15}

HEXADECIMAL-DECIMAL INTEGER CONVERSION TABLE

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.

$$0.\text{CA9BF3}_{16} = \text{CA9 BF3}_{16} \times 16^{-6}$$

2. Find the decimal equivalent of the hexadecimal integer

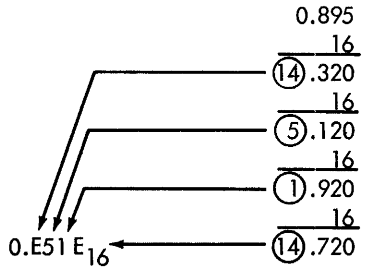
$$\text{CA9 BF3}_{16} = 13\ 278\ 195_{10}$$

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

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

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

HEXADECIMAL - DECIMAL INTEGER CONVERSION TABLE (cont.)

	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

HEXADECIMAL - DECIMAL INTEGER CONVERSION TABLE (cont.)

	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

HEXADECIMAL - DECIMAL INTEGER CONVERSION TABLE (cont.)

	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

HEXADECIMAL - DECIMAL INTEGER CONVERSION TABLE (cont.)

	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 TABLE

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

HEXADECIMAL - DECIMAL FRACTION CONVERSION TABLE (cont.)

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 FRACTION CONVERSION TABLE (cont.)

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimcl
.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	.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

TABLE OF POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625

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

APPENDIX B. INSTRUCTION TIMING

Table B-1. Instruction Preparation and Execution Time

Instruction	Prep. Timing[1]	Timing Formula for Instruction Execution[1]	All SIGMA 3 Memory	
			Minimum	Maximum
ADD	.325(3-7)	.325 (3)	1.950	3.250
ADDD[2]	.325(3-7)	.325 (7)	3.250	4.550
AND	.325(3-7)	.325 (3)	1.950	3.250
B	.325(3-7)	0	0.975	2.275
BAN, BAZ, BEN, BNC, BNO	.325(3)	0	0.975	0.975
BXNC No Index	.325(3)	0	0.975	0.975
BXNO Index	.325(3)	.325 (2)	1.625	1.625
BIX	.325(3)	.325 (2)	1.625	1.625
CP	.325(3-7)	.325 (3)	1.950	3.250
CPD[2]	.325(3-7)	.325 (7)	3.250	4.550
DIV[2]	.325(3-7)	.325 (22) If overflow, .325 (8)	8.125	9.425
IM	.325(3-7)	.325 (6)	2.925	4.225
LDA	.325(3-7)	.325 (3)	1.950	3.250
LDM[2]	.325(3-7)	.325(1+3x) where $1<x<6$[3]	2.275	8.450
LDX (Normal)	.325(3-7)	.325 (3)	1.950	3.250
LDX (Interrupt Exit)	.325(3-7)	.325 (6)	2.925	4.225
MUL[2]	.325(3-7)	.325 (21)	7.800	9.100
All Register Copy Instructions	.325(3)	.325 (3)	1.950	1.950
RD				
Internal	.325(3-7)	.325 (3)	1.950	3.250
(Except for the following:)				
Set MM	.325(3)	0	0.975	0.975
Data Switches	.325(3-7)	.325 (2)	1.625	2.925
I/O Reset	.325(3-7)	.325 (22)	8.125	9.425
EIOP Registers	.325(3-7)	.325(8+FSA Delay)[4]	3.575	-
IIOP Instructions	.325(3-7)	.325(7+FSL Delay)[4]	3.250	-
EIOP Instructions	.325(3-7)	Single Device: .325(8+FSA Delay)[4]	3.575	-
		Multiple Device:		
		.325(5+AVO Delay)+5+FSA Delay[4]	4.225	-
Interrupt	.325(4-7)	.325 (4)	2.600	3.575
External	.325(4-7)	.325(8+FSA Delay)[4]	3.900	-
SHIFT (Non-normalize)	.325(3-7)	.325(6-20)[5]	2.925	8.775
SHIFT (Normalize)	.325(3-7)	.325(3-22)[5]	3.575	9.425
STA	.325(3-7)	.325 (3)	1.950	3.250
STM[2]	.325(3-7)	.325(1+3x) where $1<x<6$[3]	2.275	8.450
SUB	.325(3-7)	.325 (3)	1.950	3.250
SUBD[2]	.325(3-7)	.325 (7)	3.250	4.550

Table B-1. Instruction Preparation and Execution Time (cont.)

Instruction	Prep. Timing[1]	Timing Formula for Instruction Execution[1]	All SIGMA 3 Memory	
			Minimum	Maximum
WD				
Internal	.325(3-7)	.325 (3)	1.950	3.250
(Except for the following:)				
Set WAIT	.325(3-7)	.325 (1)	1.300	2.600
PROTECT Register	.325(3-7)	.325 (2)	1.625	2.925
EIOP Register	.325(3-7)	.325(8+FSA Delay)[4]	3.575	-
Interrupt	.325(4-7)	.325 (5)	2.925	3.900
External	.325(4-7)	.325(8+FSA Delay)[4]	3.900	-

Notes: All times given are ± 0.1% and assume absence of memory access conflicts.

[1] Numbers in parentheses represent multiplier for computation of total time.

R	I	X	S	Effective Address	Preparation Time[†]
0	0	0	0	D	3
0	0	0	1	D + (X2)	4
0	0	1	0	D + (X1)	4
0	0	1	1	D + (X1) + (X2)	5
0	1	0	0	(D)	6
0	1	0	1	(D + (X2))	7
0	1	1	0	(D) + (X1)	6
0	1	1	1	(D + (X2)) + (X1)	7
1	0	0		(P) + SD	3
1	0	1		(P) + SD + (X1)	4
1	1	0		((P) + SD)	6
1	1	1		((P) + SD) + (X1)	6

[†] In multiples of .325 μsec pulse times.

[2] Optional feature. See "Extended Arithmetic Feature" at end of Chapter 3.

[3] x is the number of general registers to be loaded or stored.

[4] FSA – Function Strobe Acknowledge
FSL – Function Strobe Leading Acknowledge
AVO – Available Output Priority Signal

For further information see SIGMA Interface Design Manual, XDS No. 90 09 73.

Table B-1. Instruction Preparation and Execution Time (cont.)

5 SHIFT execution timing – non-normalize. The multiples of .325 μsec pulse time required for execution are shown below under the Single Register and Double Register columns. Preparation time required is .325(3-7) μsec.

No. of Bits Shifted	Single Register (A)	Double Register (E and A)	No. of Bits Shifted	Single Register (A)	Double Register (E and A)
0	5	6	16	5	6
1	5	6	17	5	6
2	6	7	18	6	7
3	7	8	19	7	8
4	8	9	20	8	9
5	9	10	21	9	10
6	10	11	22	10	11
7	11	12	23	11	12
8	12	13	24	12	13
9	13	14	25	13	14
10	14	15	26	14	15
11	15	16	27	15	16
12	16	17	28	16	17
13	17	18	29	17	18
14	18	19	30	18	19
15	19	20	31	19	20

SHIFT execution timing – normalize. The following A and E register drawings show the multiples of .325 μsec pulse time required for normalize execution. The number required corresponds to the position of the last bit in a consecutive string of 1's or 0's starting with bit position 0 of register E. Preparation time required is .325(3-7) μsec.

E Register

Bit Positions	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pulse Times	3	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

A Register

Bit Positions	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Pulse Times	8	8	9	10	11	12	13	14	15	16	17	18	19	20	21	8	← Last Zero
																22	← Last One

APPENDIX C. WATCHDOG TIMER

The optional Watchdog Timer (Model 8072) performs three functions:

1. System hangup monitoring.
2. Monitoring of power within the Watchdog Timer chassis.
3. Direct Input/Output (DIO) monitoring.

A typical use of the Watchdog Timer would be in a process control system, detecting and signaling malfunctions due either to program hangups or system failure to respond to a DIO signal. The system must include the optional DIO feature to implement the Watchdog Timer. In addition, if a CPU signal is desired for DIO response failure (no function strobe acknowledge), an optional priority interrupt must be installed.

To detect a system hangup, the Watchdog Timer monitors program continuation signals (see Reset Timer instruction) within predetermined time constraints. Failure to detect a continuation signal within the specified time causes a relay in the Watchdog Timer chassis to close and a system hangup signal to be produced. As an example, this relay may be connected to an audible alarm, so that an operator may take corrective action. The timing interval is selected by manual switch settings. These activate the Watchdog Timer to expect a Write Direct (WD) instruction within either 8 ms, 128 ms, or 1.024 seconds, according to the switch settings. The Watchdog Timer recognizes three WD instructions:

Enable

0	R	I	X	S	Displacement	
0	2				0	0
0 3	4			7	8 11	12 15

Disable

0	R	I	X	S	Displacement	
0	1				0	0
0 3	4			7	8 11	12 15

Reset Timer

0	R	I	X	S	Displacement	
0	3				0	0
0 3	4			7	8 11	12 15

An Enable WD starts the Timer and must be followed by Reset Timer WDs within the selected time intervals to avoid the system hangup signal. The Disable WD disables and resets the Timer.

Power monitoring is accomplished through a hardware relay in the Watchdog Timer. The relay drops out in case of power failure. This relay, too, may be connected to an alarm or it may be wired in conjunction with the timing feature to provide a fail-safe capability.

DIO monitoring prevents excessive and indefinite delays in CPU operations due to delayed function strobe acknowledge (FSA) signals generated by the controlled device. If the Watchdog Timer fails to detect an FSA within approximately 64 microseconds of the function strobe, it generates an FSA enabling the CPU to continue operations. The DIO instruction associated with the missing FSA is aborted.

The Timer signal may be used to initiate an optional priority interrupt. This interrupt will occur after the system has completed the instruction following the aborted DIO instruction.

INDEX

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.