523-0561698-101831
15 October 1969
Revised 30 June 1970

COLLINS

instruction book

C-8561A-2 Programming Reference Manual

Printed in United States of America

Collins Radio Company | Dallas, Texas

List of Effective Pages

*The asterisk indicates pages changed, added, or deleted by the current change.

Page No.	Issue
*Title	30 June 70
*A	30 June 70
*Rec of Rev	30 June 70
*i thru iv	30 June 70
*1-1 thru 1-15	30 June 70
*1-16 Blank	30 June 70
*2-1 thru 2-5	30 June 70
*2-6 Blank	30 June 70
*3-1 thru 3-4	30 June 70
*4-1 thru 4-42	30 June 70
*A-1 thru A-4	30 June 70
*B-1 thru B-4	30 June 70
*C-1	30 June 70
*C-2 Blank	30 June 70
*D-1	30 June 70
*D-2 Blank	30 June 70
*E-1 thru E-2	30 June 70
*F-1 thru F-13	30 June 70
*F-14 Blank	30 June 70

Record of Revisions

RETAIN THIS RECORD IN THE FRONT OF MANUAL.
ON RECEIPT OF REVISIONS, INSERT REVISED PAGES IN THE MANUAL, AND ENTER DATE INSERTED AND INITIALS.

ASSIGNED TO (JOB TITLE)

LOCATION

REV. NO.	REVISION DATE	INSERTION DATE	BY	REV. NO.	REVISION DATE	INSERTION DATE	BY
1	30 June 70						

table of contents

table of contents (cont)

table of contents (cont)

list of illustrations

list of tables

section 1
general description

The Collins C-8561A-2 Processor is a medium-scale digital computer used in the C-8500 Computer System. This manual describes the processor from a machine-language programming standpoint. For an overall description of how the processor is used in the computer system, refer to the C-System General Description (523-0561-697).

1.1 FUNCTIONAL ORGANIZATION

The C-8561A-2 Processor (figure 1-1) consists of five major functional sections: arithmetic and control, communications service, main core storage, processor service, and transfer links. Each of these functional sections are described in the following paragraphs.

1.1.1 Arithmetic and Control Section

The arithmetic and control section consists of the transfer link converter, the arithmetic logic and control unit, and the algorithm unit (optional). The transfer link converter transforms the 32-bit data path used by the transfer link into the 16-bit data paths used by the arithmetic logic and control unit and the algorithm unit.

The arithmetic logic and control unit executes 72 different instructions. Of these instructions, 60 are fixed wired, and the remaining 12 are trapped and executed from main core storage. The computer operates with a clock rate of 8 megahertz; a typical instruction execution time is 2.9 or 5.6 microseconds, including memory access. (See Appendix b.)

In the arithmetic logic and control unit, the three main registers communicate with the transfer link through input and output commutators (figure 1-2). These are the function register, F, the memory exchange register, Z, and the memory address register, S. The F register holds the 14 bits of the instruction that includes the operation code and control bits. This register interfaces with the instruction decoding, control, and timing circuits. The Z register, 32 bits in length, receives or transmits one word in parallel to the transfer link. All data to or from the memory passes through the Z register. The S register, 18 bits in length, transmits the address of each memory word to be accessed through the transfer link.

The arithmetic logic and control unit operates into a 32-bit hardware accumulator called the A accumulator, a 32-bit accumulator in the memory called the B accumulator, or the combination of both, called the D accumulator. Three 18-bit index registers provide storage for address modifiers used in indexing operations.

The P register serves as the program counter or instruction address counter. This register, 18 bits in length, holds the address of the next instruction to be executed in the program. The instruction address counter increments by four to skip from one instruction word to the next (memory is organized with byte addressability).

The 32-bit M register serves as temporary storage for data from the other registers during the machine cycle.

Heavy lines in figure 1-2 represent parallel data transfers of 14, 16, or 18 bits; the thin lines represent transfers of 2 bits in parallel. Since the integrated circuits used in the computer logic operate at high speeds, the operations performed serially, 2 bits at a time, do not impair overall instruction execution time.

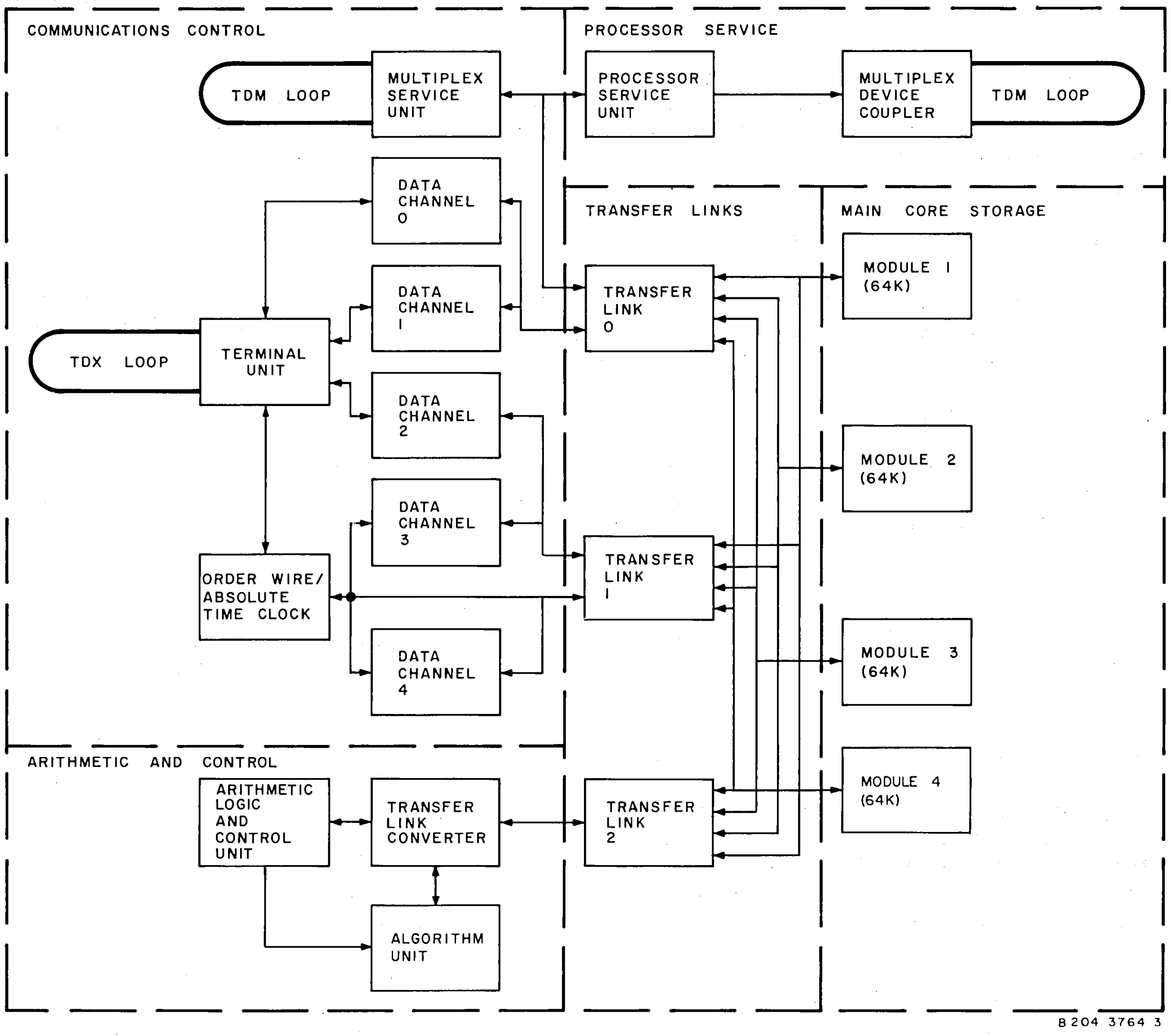


Figure 1-1. C-8561A-2 Processor Block Diagram.

The computer unit includes hardware interrupts that unconditionally branch the program to a fixed memory location in case of memory parity error, invalid instructions or addresses, or the elapse of a selected interval of time. These interrupts are described along with memory-protection features in section 3 of this manual.

The optional algorithm unit effects an increase of operating speed of the processor when performing floating point, fixed point multiply, and fixed point divide instructions. Programming is required in the operation code trap routines to place operands in core and initiate the algorithm unit.

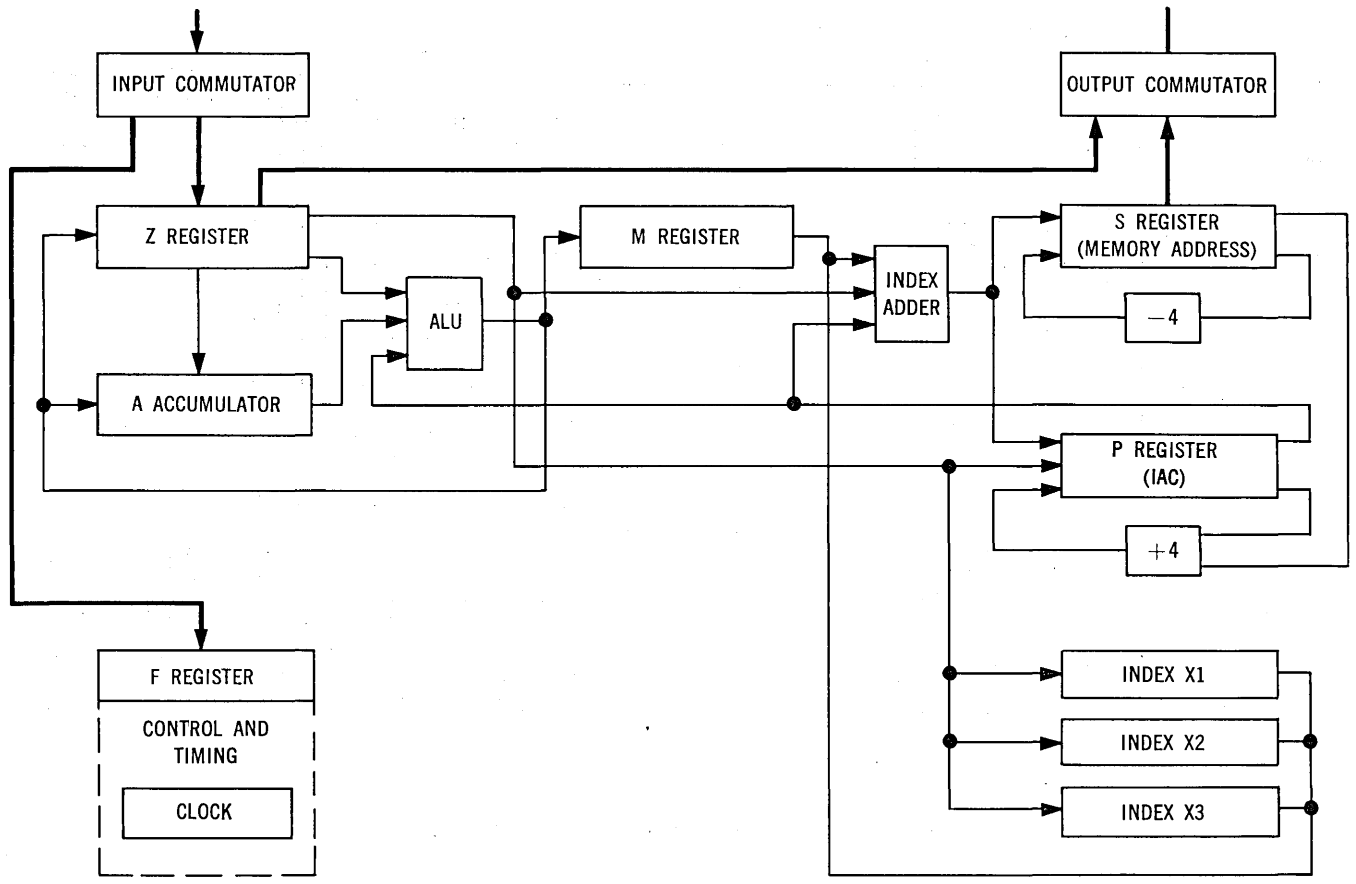


Figure 1-2. Arithmetic Logic and Control Unit Block Diagram.

1.1.2 The Transfer Link

Three 8724C-1 Transfer Links interconnect the main core storage with the arithmetic logic and control unit and the communication control group. All transactions with main core storage take place through the transfer links. To speed processor operations, the transfer links interleave successive accesses to memory among the memory units. This is accomplished by storing odd addresses in one memory unit and even addresses in another. If four memory units are installed, each transfer link addresses all four units in an odd-even-odd-even end-around sequence.

Figure 1-3 is a block diagram of a transfer link. Each transfer link has two source ports and four memory ports. Requests for memory access are handled by the data transfer control and the module decode and priority logic. When requests occur at the same time, the memory access control handles them according to the following priority:

PRIORITY	UNIT
1	Multiplex service unit
1	Processor service unit
1	Data channel 0
1	Data channel 1
2	Data channel 2
2	Data channel 3
2	Data channel 4
2	Orderwire/Absolute Time Clock
3	Arithmetic logic and control unit

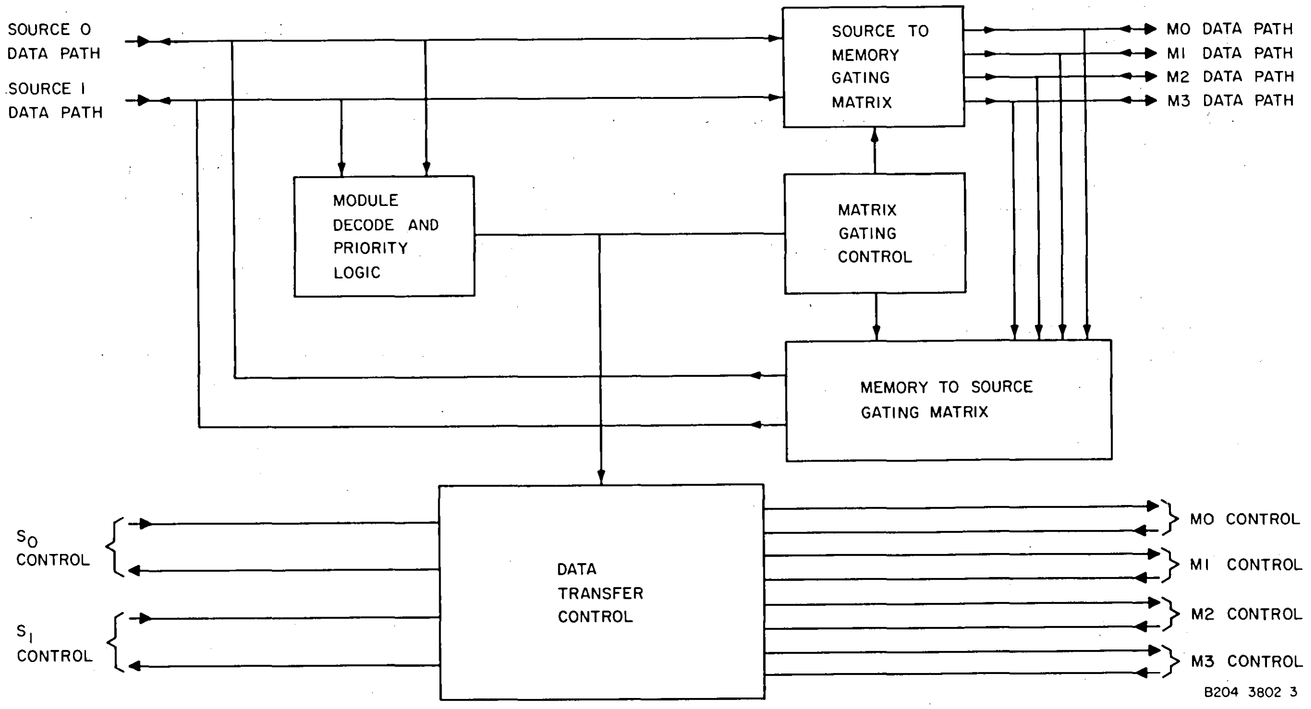


Figure 1-3. Transfer Link Block Diagram.

Requests for access on the same priority level (reflecting connection to the same transfer link) are handled on a first come, first served, alternating basis.

Priorities 1A and 1B are serviced alternately. Each of the three transfer links can access two memory modules simultaneously to provide possible overlap within one priority class.

1.1.3 Main Core Storage

The 8712A-3 memory has a 2-us. cycle time and an access time of 510 ns. It is expandable from one to four modules. Each module has a capacity of 65,536 bytes, giving a maximum capacity of 262,144 bytes for one processor. In addition to load or unload memory cycles, the 8712A-3 has the capability of performing an unload-modify-load function in one memory cycle.

Each module in the memory has its own address and data buffers (figure 1-4). Because of this, and because each transfer link has the capability of servicing two memory accesses simultaneously, it is possible for all four modules to operate concurrently. The alcu, data channels, processor service unit, and multiplex service unit may all request the use of the memory independently. If four units request the access of different memory modules, the entire memory can be in operation at the same time. The transfer links can perform the four transfers of 32 bits in 2 microseconds.

1.1.4 Communications Control Group

Communications control group is a general term applied to the input/output and communication section of the processor. In its simplest form, the communications control group consists of a data channel or a multiplex service unit. In its maximum configuration, the communications control group consists of five data channels, one multiplex service unit, and an orderwire/absolute time clock unit.

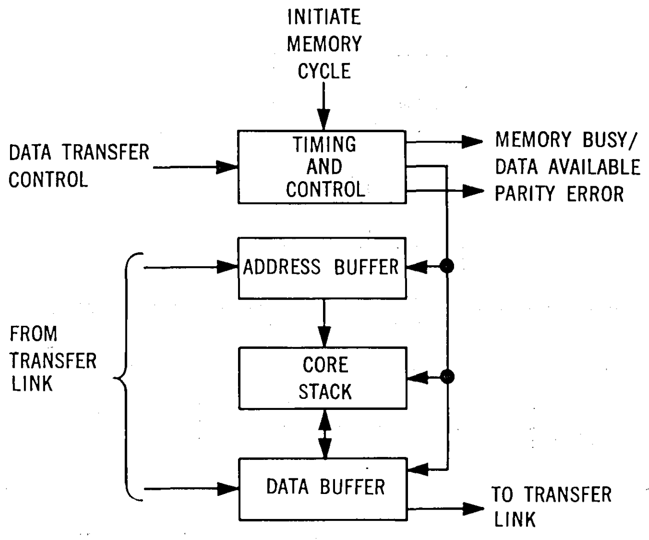


Figure 1-4. Memory Unit Block Diagram.

The multiplex service unit controls the time division multiplex loop. This loop provides multiplex channels for numerous relatively slow-speed devices of various kinds. The data channels and the orderwire/absolute time clock units interface with the Time Division Exchange loop for higher speed communications and input/output operations. These communications loops are explained in more detail in the following paragraphs.

1.1.4.1 Time Division Exchange System

The time division exchange (TDX) system provides high-speed communication channels between processors, storage devices, and input/output peripheral equipment. Figure 1-5 is a block diagram of the time division exchange system.

The major components of any time division exchange system are a transmission line for communication, a loop synchronizer, and terminal units that interface devices to the communication path.

The loop provides serial communication between terminal units. It physically consists of a coaxial cable which is driven and terminated by each device on the loop. The signal on the currently available loop is a 32-MHz bi-phase modulated sine wave shown pictorially in figure 1-6. Addressing capability exists for directly utilizing a 512 MHz loop.

The data is encoded one bit per cycle of carrier. Communication channels are bit interlaced symmetrically around the group synchronization pulse. Each channel of a terminal unit has the capability of interfacing with a communication channel of any of the following rates: 7.8125, 15.625, 31.25, 62.5, 125, 250, 2000, 4000, or 8000 kbps. The data format within a channel is shown in figure 1-7.

1.1.4.1.1 Data Channel

The 7521A-1 data channel (figure 1-8) is initiated whenever the first bit of a device control message, contained in core storage, is reset to zero. The data channel indicates completion by setting this same bit to one. Because of the transfer link overlap capabilities the data channel is able to provide communication capability with a minimum of contention with the arithmetic logic and control unit. Each data channel operates with its own independent memory and terminal unit interfaces. Each data channel is capable of

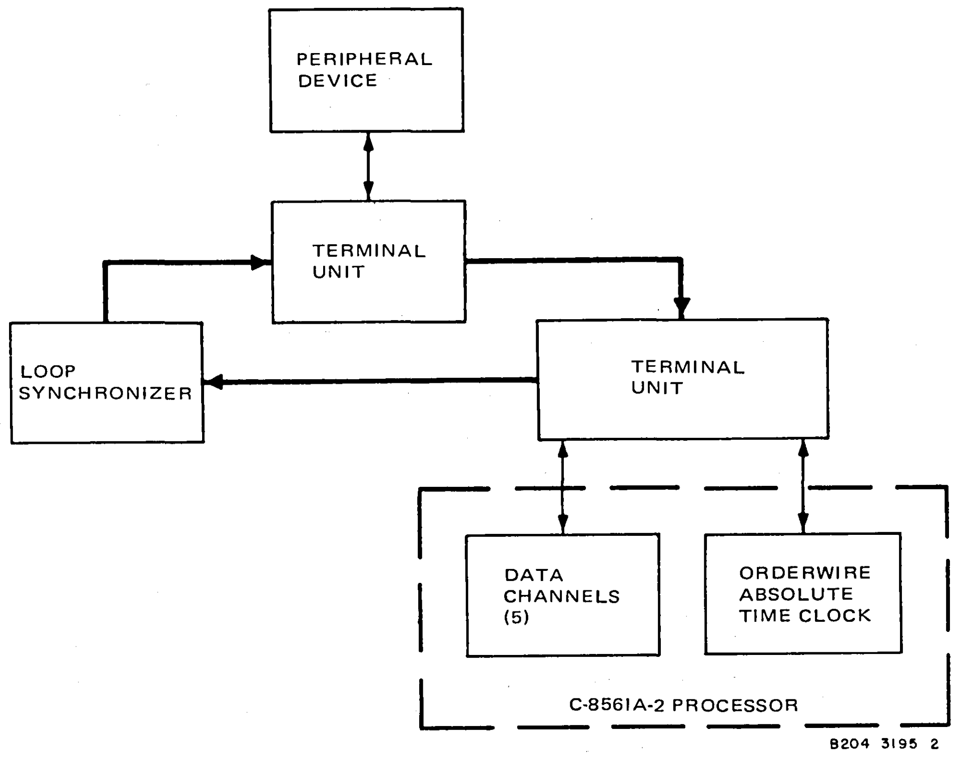


Figure 1-5. Time Division Exchange Communication Facility.

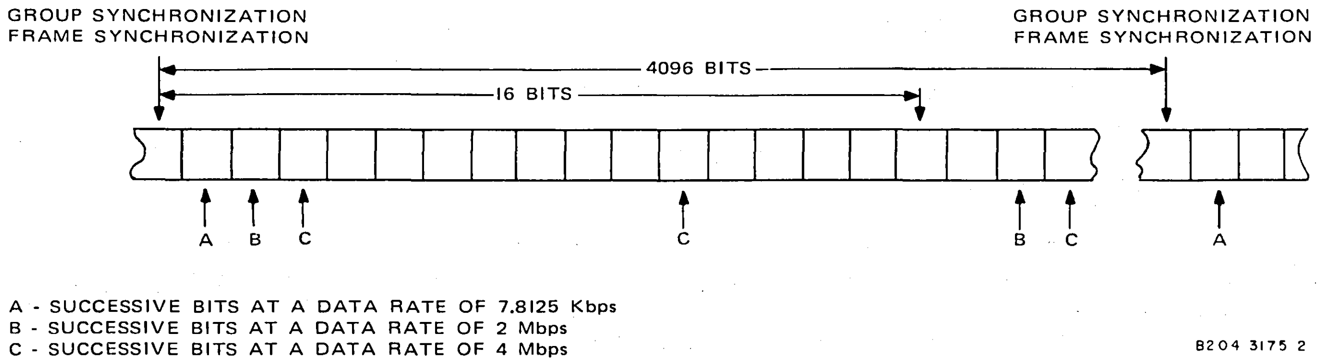
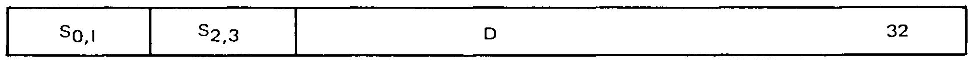

GROUP SYNCHRONIZATION
FRAME SYNCHRONIZATION
GROUP SYNCHRONIZATION
FRAME SYNCHRONIZATION
4096 BITS
16 BITS
A
B
C
C
B
C
A
A - SUCCESSIVE BITS AT A DATA RATE OF 7.8125 Kbps
B - SUCCESSIVE BITS AT A DATA RATE OF 2 Mbps
C - SUCCESSIVE BITS AT A DATA RATE OF 4 Mbps
B204 3175 2

Figure 1-6. Time Division Exchange Loop Format.



$S_{0,1}$ - TWO BIT TERMINAL UNIT SUPERVISORY FIELD ENCODED AS FOLLOWS:

10	POLL
11	BID

$S_{2,3}$ - TWO BIT DEVICE SUPERVISORY FIELD ENCODED AS FOLLOWS:

00	DATA REQUEST
01	NOT USED
10	STATUS
11	DATA

D - THIRTY-TWO BIT DATA FIELD

8204 3192 2

Figure 1-7. Time Division Exchange Loop Word Format.

communicating with one other device (storage, input/output, or processor) on the time division exchange loop. The data rate used for each communication performed is specified in the device control message to be any rate in the range 7.8125 kbps to 8 Mbps.

The device control message is located in the main core storage at a location which is pointed to by a fixed (strapped) core location called the queue address. The format and relationship between these locations is shown in figure 1-9. The device control message contains all of the information necessary for the desired operation including a pointer to the next device control message which is used by the data channel with no program intervention only if no error occurs in this operation. It should be noted that a program can choose to exercise complete control over its communication activity or to generate a large queue of work for the data channel and then allow it to run independently. This control is accomplished by program manipulation of the first bit of the device control message and the address, within one device control message, which chains the data channel to the next message.

All references to data to be handled by the data channel are made by data control words (refer to figure 1-9 for their location in a device control message). A data control word can contain a reference to data to be transmitted, a data area in which to store received data, instructions to ignore incoming data words, instructions to transmit a number of all-zero data words, or pointers to additional data control word lists.

1.1.4.1.2 Orderwire/Absolute Time Clock

The 7531A-1 orderwire absolute time clock (figure 1-10) performs three communication functions; Orderwire 1, Orderwire 2, and Absolute Time Clock.

The Orderwire 1 channel on the time division exchange loop is an assigned 125 kbps channel used for communication between processors.

To perform the orderwire 1 function, the 7531A-1 operates in conjunction with data channel four. Data channel four is initiated only by the orderwire 1 function which is in turn initiated in one of two ways. To call another processor, the orderwire 1 function may be initiated in a manner identical to the way in which a data channel is initiated. To receive a call from another processor, the orderwire 1 function may be initiated by an incoming call from the orderwire 1 time division exchange loop channel. The incoming call must contain the party line address of the called orderwire 1 unit. This party line address is an 8-bit identifier which can be assigned to the orderwire unit by software. Once the orderwire 1 unit has been initiated, it will initiate the data channel and remain idle until the communication has completed.

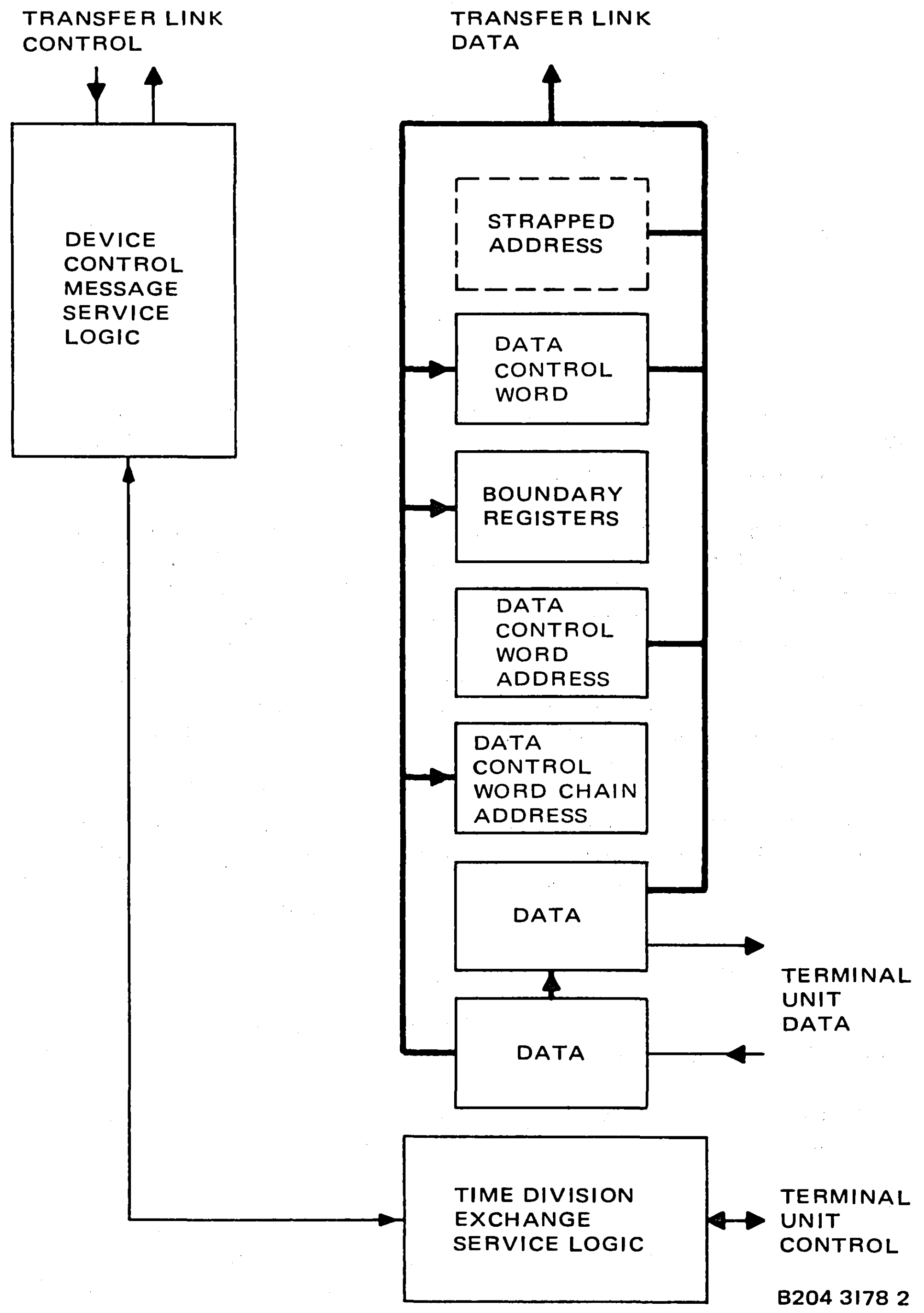


Figure 1-8. Data Channel Block Diagram.

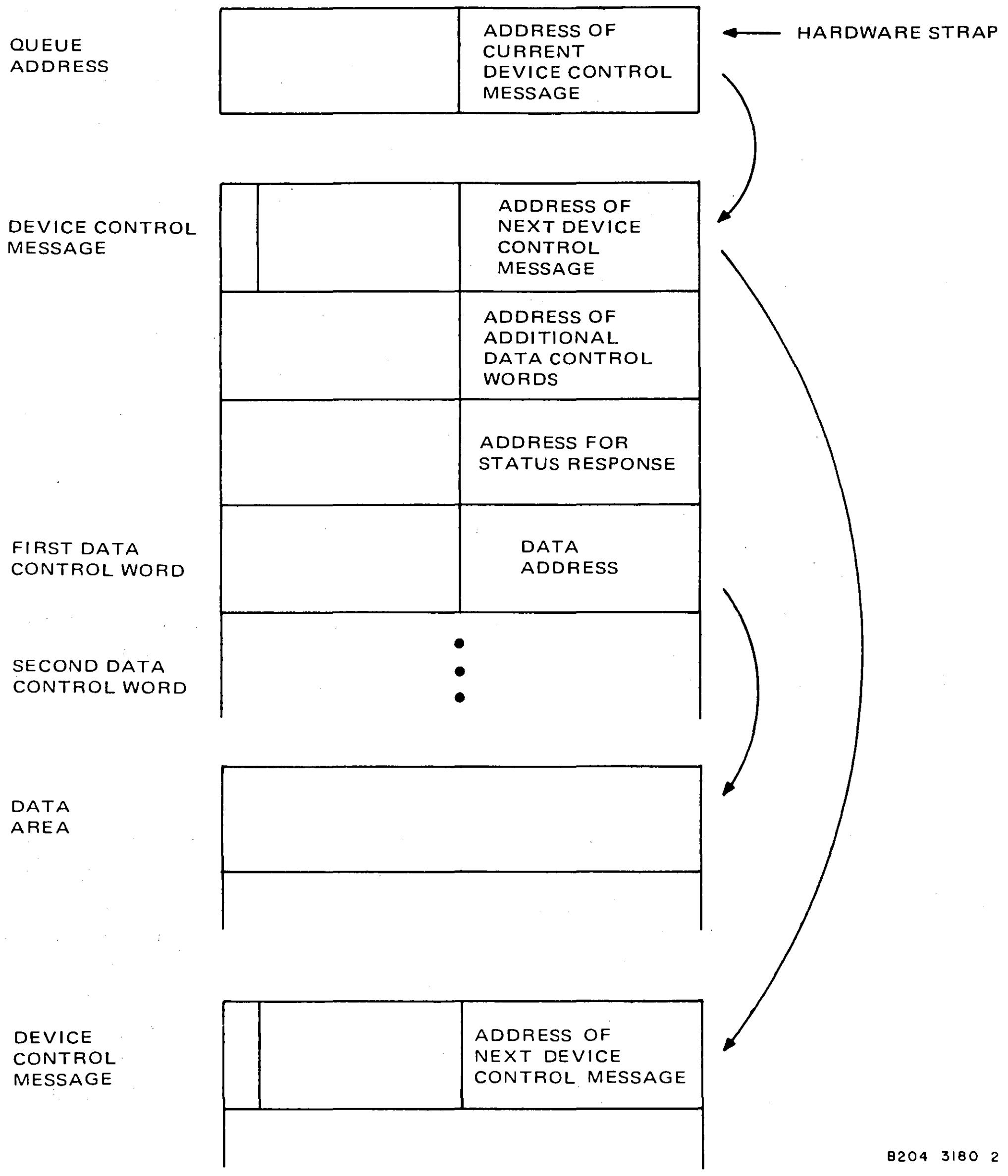

QUEUE ADDRESS
ADDRESS OF CURRENT DEVICE CONTROL MESSAGE
HARDWARE STRAP
DEVICE CONTROL MESSAGE
ADDRESS OF NEXT DEVICE CONTROL MESSAGE
ADDRESS OF ADDITIONAL DATA CONTROL WORDS
ADDRESS FOR STATUS RESPONSE
FIRST DATA CONTROL WORD
DATA ADDRESS
SECOND DATA CONTROL WORD
DATA AREA
DEVICE CONTROL MESSAGE
ADDRESS OF NEXT DEVICE CONTROL MESSAGE
B204 3180 2

Figure 1-9. Data Channel Main Core Storage Interface.

The orderwire 2 channel on the time division exchange loop is an assigned 7.8125 kbps channel used for processor-to-operator communication and for the assignment of peripheral devices to other time division exchange channels for higher speed communication. When initialized, all peripheral units monitor the orderwire 2 channel and accept instructions on that channel to switch to a higher speed channel for communication. To perform the orderwire 2 function, the 7531A-1 operates in conjunction with data channel three. Data channel three has the option, under control of software, of performing the normal data channel function and thus inhibiting the orderwire 2 function or of enabling the orderwire 2 function in which case it may only be initiated by the 7531A-1. In orderwire 2 mode, the orderwire unit and data channel three function in a manner essentially identical to the orderwire 1 unit and data channel four. The only functional difference is that the orderwire 2 unit accepts all incoming calls.

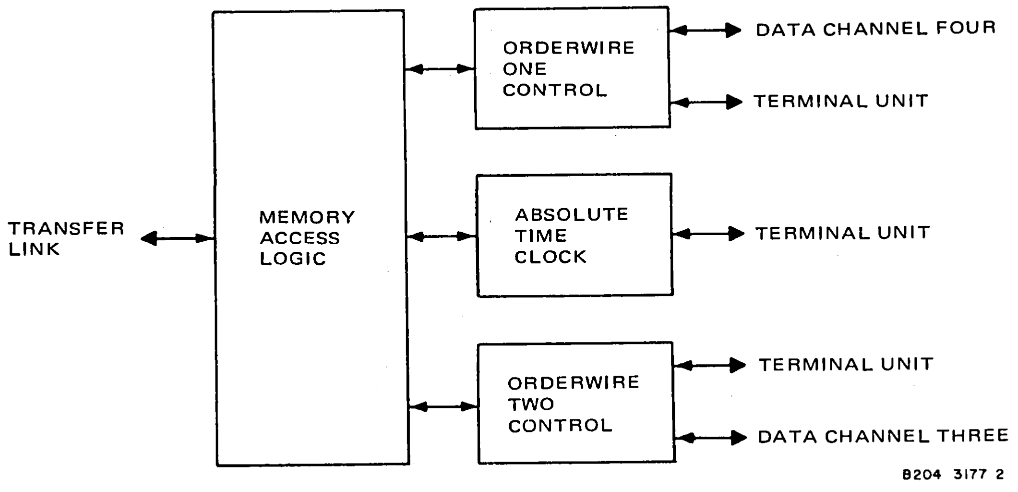


Figure 1-10. Orderwire/Absolute Time Clock Block Diagram.

The absolute time clock channel on the time division exchange loop is an assigned 7.8125 kbps. Every 1/128 second, a word is transmitted around the loop which contains time and date information. Each absolute time clock unit which is on the time division exchange loop will accept this word and store it in a fixed (strapped) location in memory (refer to Appendix d).

1.1.4.2 Time Division Multiplex System

The time division multiplex system provides for the connection of a large number of low-speed devices to a single computer within the C-8500 C-System on a word, time-shared basis. The time division multiplex system (figure 1-11) consists of a serial loop interconnecting the various devices and terminating at the multiplex service unit, which connects the loop to main core storage. The time division multiplex system also includes the multiplex exchange table, multiplex queues, and a multiplex service program resident in the main storage of the processor. The multiplex exchange table, consisting of multiplex status records and multiplex queues, provides the operational linkage between the end devices and controlling software in the processor. The multiplex service program functions as a device independent software interface between time division multiplex devices and application program subroutines within the computer. These subroutines control and service the various devices.

The loop provides the serial communications medium between the multiplex service unit and all connected devices. It physically consists of a coaxial cable which is driven by and terminated in the multiplex service unit. The signal on the loop consists of a 1.2288 MHz bi-phase modulated square-wave where each cycle corresponds to one bit of data. There are 256 thirty-six bit words transmitted serially in one frame. These 256 thirty-six bit words are time division addresses and are identified by their time position relative to a framing pulse. Each time division address corresponds to a device channel. A working channel is a normally unidirectional instantaneous communication connection between hardware and software elements. The framing pulse is generated in synchronization with the first time division address. Data is inserted into and extracted from the channels by the multiplex service unit and devices. A device may be assigned 1, 8, 16, or 32 time division addresses, providing effective channel bit rates of 4.8 kbps, 38.4 kbps, 76.8 kbps, or 153.6 kbps.

The loop word format for each of the 36-bit time division addresses consists of a 4-bit supervisory code field and a 32 bit operand. The supervisory field contains the supervisory codes for multiplex service unit to

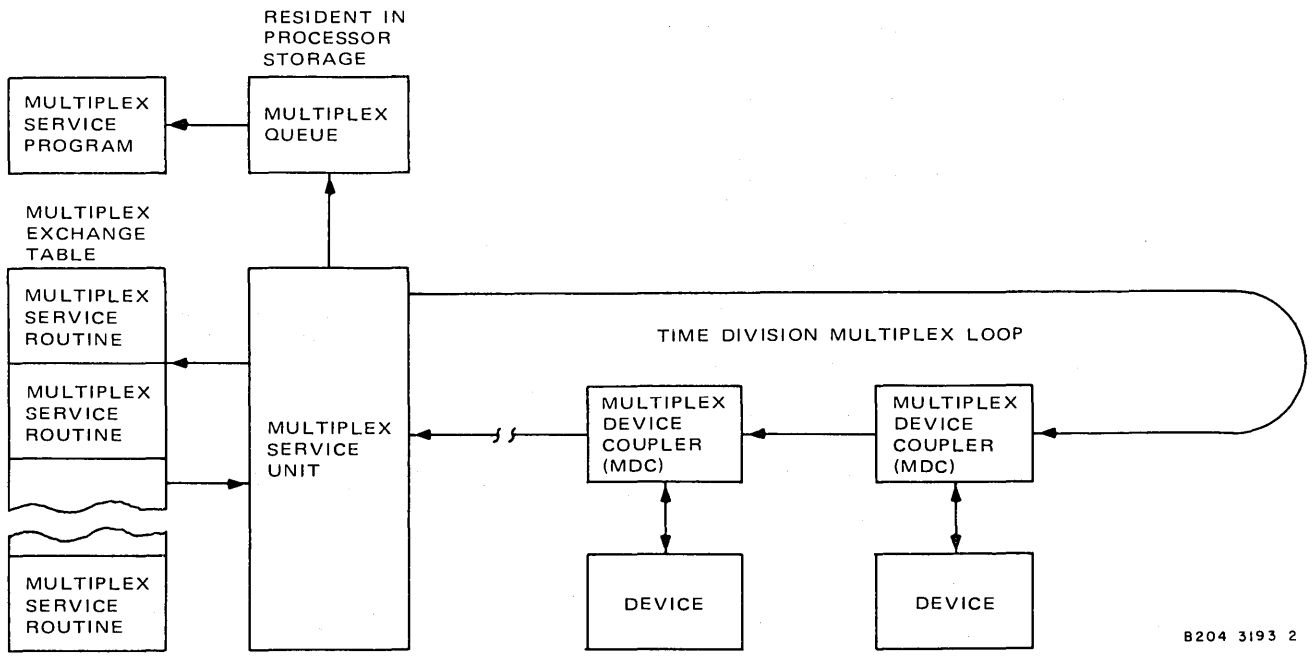

RESIDENT IN PROCESSOR STORAGE
MULTIPLEX SERVICE PROGRAM
MULTIPLEX QUEUE
MULTIPLEX EXCHANGE TABLE
MULTIPLEX SERVICE ROUTINE
MULTIPLEX SERVICE ROUTINE
MULTIPLEX SERVICE ROUTINE
MULTIPLEX SERVICE UNIT
TIME DIVISION MULTIPLEX LOOP
MULTIPLEX DEVICE COUPLER (MDC)
MULTIPLEX DEVICE COUPLER (MDC)
DEVICE
DEVICE
B204 3193 2

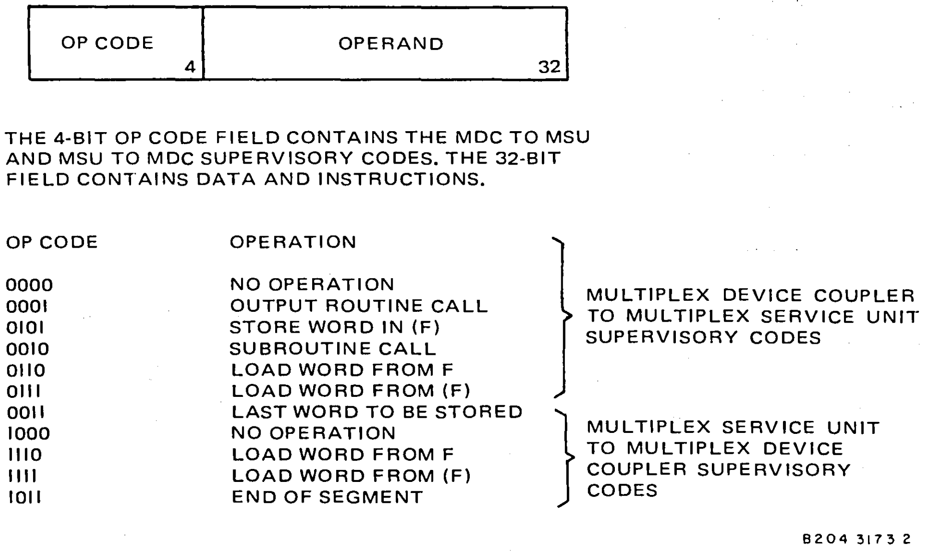
Figure 1-11. Functional Relationship of Time Division Multiplex System Element.

multiplex device coupler communications. The operand contains the instructions, data, parameters, etc. for the device and programs. A description of the word format on the time division multiplex loop is shown in figure 1-12.

The multiplex service unit is a hardware device which serves to connect the multiplex loop to the computer core. It provides all the necessary hardware to drive and terminate the loop and to interface with processor core storage. It contains the logic necessary to allow the independent movement of data and service messages between each device on the loop and data areas in processor storage. The transactions between the multiplex service unit and each device are completely under control of it and the multiplex device coupler during record and/or word transfers once communication between the processor and the device has been established.

The multiplex device coupler provides a standard interface to devices on the time division multiplex loop. It performs the function of time division address recognition, serial extraction and insertion of data into its assigned time division address channel, and logical interpretation and generation of the supervision necessary to communicate with the multiplex service unit.

The multiplex status record serves as the communication interface between programs within the computer, the multiplex service unit, and the end device. Functionally, multiplex status records are sets of 'registers' which are available to both the multiplex service program and the multiplex service unit. These registers are used to hold instructions, data, control, and status information which together define and maintain the current transaction on a working channel. There are as many status records in the processor storage as there are device channels on the loop, thus allowing the multiplex service unit to maintain independent activity with each device upon request in accordance with instructions contained in each status record. Multiplex status records are four words in length and are arranged continuously in storage so that the multiplex service unit can conveniently increment through them in synchronism with the multiplex loop rate. The format of a multiplex status record is illustrated in figure 1-13.



THE 4-BIT OP CODE FIELD CONTAINS THE MDC TO MSU AND MSU TO MDC SUPERVISORY CODES. THE 32-BIT FIELD CONTAINS DATA AND INSTRUCTIONS.

OP CODE	OPERATION	
0000	NO OPERATION	MULTIPLEX DEVICE COUPLER TO MULTIPLEX SERVICE UNIT SUPERVISORY CODES
0001	OUTPUT ROUTINE CALL	
0101	STORE WORD IN (F)	
0010	SUBROUTINE CALL	
0110	LOAD WORD FROM F	
0111	LOAD WORD FROM (F)	
0011	LAST WORD TO BE STORED	MULTIPLEX SERVICE UNIT TO MULTIPLEX DEVICE COUPLER SUPERVISORY CODES
1000	NO OPERATION	
1110	LOAD WORD FROM F	
1111	LOAD WORD FROM (F)	
1011	END OF SEGMENT	

Figure 1-12. Multiplex Service Unit/Multiplex Device Coupler Format and Operand Code Definition.

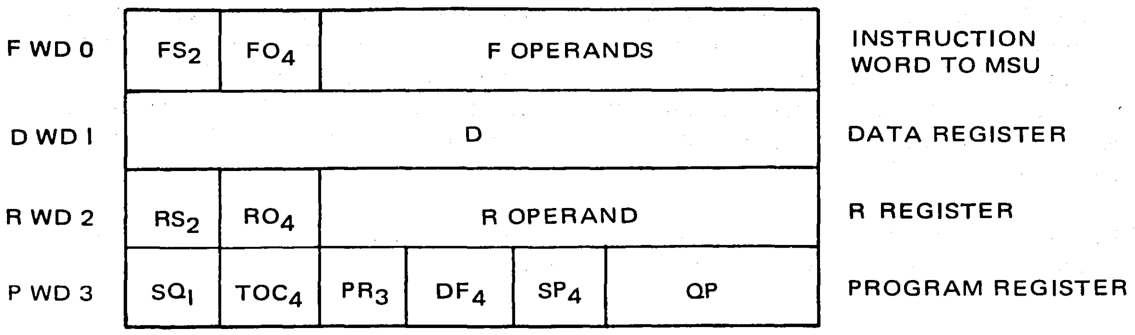
Two multiplex queues, an output queue and an input queue, are maintained in the processor storage. In either case, a queue entry indicates that device program service is required. There is space for one queue entry per status record. A queue entry is made by the multiplex service unit in multiplex queue 1 whenever a device requires program intervention in order to continue, when the multiplex service unit detects that additional processor storage allocation is required, and/or when a multiplex service unit instruction is complete. An entry is made in multiplex queue 2 when the device must initiate an output transaction. A queue entry is normally accompanied by a multiplex service unit transfer to program control for this particular device.

The multiplex service program functions as an application-independent software program which effectively services each device on the time division multiplex loop. Its purpose is to manage input/output transactions between application systems and devices which are connected to the multiplex loop. The multiplex service program is independent of any particular device type or application. It provides linkage between multiplex loop devices and secondary storage and/or multiplex loop devices and multiplex channel subroutines.

1.1.5 Processor Service Unit

The C-8561A-2 Processor has only three manual operating controls which are contained on the front panel of the 7508C-2 Processor Service Unit (figure 1-14). The IPL buttons start the automatic initial program load sequence, which loads programs from the time division exchange loop. During initial program load, the memory protection feature is overridden (refer to section 3). The INIT button initializes the arithmetic logic and control unit without loading a program. Lamps on the panel of the unit indicate that the processor is operating normally (RUN), or that it has failed, (machine failure monitor). An 8-bit lamp-bank indicates which processor function has failed during an IPL. Upon successful completion of an IPL, these lights are under program control.

The 7508C-2 controls certain operations of all processor units. These include enabling the arithmetic logic and control unit and communication control equipment, enabling diagnostic mode to all units, controlling marginal voltage tests, forcing memory parity errors in all memory modules, disabling protected memory,



THE F WORD DEFINES A GIVEN I/O TRANSACTION USING THE FOLLOWING:

FS: THIS FIELD SPECIFIES MULTIPLEX SYSTEM CONTROL
00 MSU IS IN CONTROL
01 MSP IN IN CONTROL
11 MSU IS IN CONTROL
10 DEVICE IS IN CONTROL

FO: THIS FIELD IS THE OP CODE FIELD AND SPECIFIES DATA MOVE OPERATIONS AS FOLLOWS:

0111	FIELD STORE
1111	FIELD LOAD
0100	STORE D
0110	STORE D IF IB*=NON ZERO
1110	LOAD D IF IB*=NON ZERO
0101	STORED IF IB* IS OUTSIDE LIMITS
1101	LOAD D IF IB* IS OUTSIDE LIMITS
	*IB=OPERAND RECEIVED FROM DEVICE
0000	NO OPERATION
0011	FIELD STORE AND LINK
1011	FIELD LOAD AND LINK

THE OPERAND FIELD IS CONDITIONAL ON FO AND CONTAINS POINTERS OR LIMIT CONDITIONS USED IN EXECUTION OF THE INSTRUCTIONS.

THE D WORD IS USED AS DATA STORAGE FOR SINGLE-WORD TRANSACTIONS WITH THE DEVICE. CONTROL PARAMETERS ARE STORED IN D FOR USE BY THE MULTIPLEX SERVICE PROGRAM WITH DEVICE-GENERATED SUBROUTINE CALLS.

THE R WORD IS USED AS STORAGE FOR A SECOND INSTRUCTION IN LINK MODE OPERATIONS. IF FO CONTAINS A LINK INSTRUCTION, R CONTAINS THE NEXT MULTIPLEX SERVICE UNIT INSTRUCTION AND THE CONTENTS OF R AND F ARE INTERCHANGED ONCE THE FIRST INSTRUCTION IS EXECUTED. THIS ACTION IS PERFORMED BY THE MULTIPLEX SERVICE UNIT THUS ALLOWING TWO SUCCESSIVE TRANSACTIONS TO BE LINKED TOGETHER WITHOUT PROGRAM INTERVENTION.

THE P WORD CONTAINS AN OUTPUT FILE ACTIVITY INDICATOR BIT SQ, USED BY THE MULTIPLEX SERVICE UNIT. IT ALSO CONTAINS DEVICE CHANNEL STATUS POINTERS AND INDICATORS REQUIRED AND USED ONLY BY THE MULTIPLEX SERVICE PROGRAM.

B204 3174 2

Figure 1-13. Multiplex Status Records.

controlling interleaved memory mode, and displaying status as the 8-bit lamp-bank on the front of the processor. These functions are controlled by the processor service unit during an IPL or INIT sequence. After a successful IPL or INIT, these functions are under the control of software. Every time the arithmetic logic and control unit executes a reset machine failure monitor instruction, the processor service unit will set the above functions to reflect the bits in memory location hexadecimal 40. This word is referred to as the processor control word.

The 7508C-2 maintains a processor status word in memory location hexadecimal 44. This word contains the current status of the fault alarms of all processor hardware.

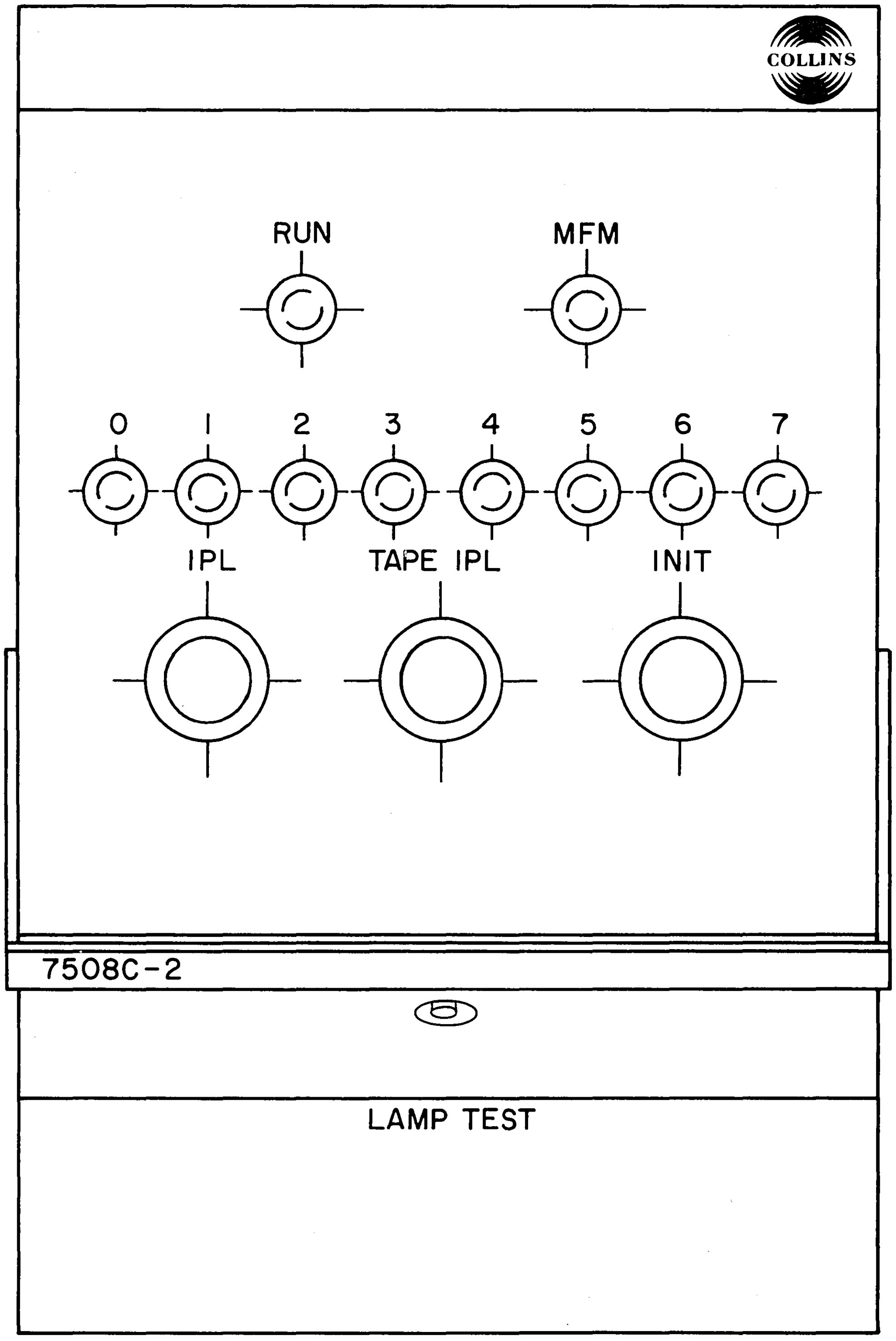


Figure 1-14. 7508C-2 Processor Service Unit.

section 2
internal operations

2.1 DATA FORMATS

The C-8561A-2 computer has a basic data format of 8-bit bytes and 32-bit words. A byte can represent a character in some 8-bit code, or a byte can be simply one-fourth of a 32-bit word. Two bytes make a half-word, and four bytes make a full-word. In some instances, a double-word of 8 bytes is used.

1 byte	=	8 bits
1 half-word	=	16 bits or 2 bytes
1 word	=	32 bits or 4 bytes
1 double-word	=	64 bits or 8 bytes

Main core storage (mcs) is byte addressable, and certain instructions manipulate bytes or characters. However, mcs is accessed by words, and numerous instructions manipulate words. Word manipulations provide maximum speed for arithmetic and data handling operations. Byte manipulations, on the other hand, provide the ability to operate on variable length data and coded characters.

Words and half-words are referenced (addressed) by the first (leftmost) byte. The length of the data is implied by the operation.

Words, half-words, and bytes must be properly aligned. Even addresses are half-word boundaries, and addresses divisible by four are word and double-word boundaries. All addresses are byte boundaries. Table 2-1 illustrates the alignment restrictions.

Table 2-1. Alignment Restrictions.

DATA UNIT	ADDRESS OF LEFTMOST BYTE
Byte	Any address
Half-word	Even addresses
Word	Addresses divisible by 4
Double-word	Addresses divisible by 4
Instruction word	Addresses divisible by 4

Two 32-bit registers (A and B) provide the accumulators for the implementation of the C-8561A-2 operations. The two accumulators can be used together to form a single 64-bit accumulator (D) where the A accumulator forms the left 32 bits and the B accumulator forms the right 32 bits. Three 18-bit index registers (X1, X2, and X3) are provided for addressing purposes.

2.2 INSTRUCTIONS

Each machine instruction for the C-8561A-2 occupies one word and has the following format (note that the bit positions are numbered 0 through 31).

I	X		L	OP-CODE	M		C	ADDRESS
0	1	2	3	4 5 6 7 8 9 10	11	12	13	14 15 . . . 31

The interpretation of fields within the instruction word is described in the following two paragraphs. For specific instructions, some of the fields are either restricted or extended, but the principles described below still apply.

2.2.1 Addressing

The fields of the instruction word that determine the interpretation of the address field are the I, X, and L fields.

I — indirect addressing indicators; binary 0 or 1.
0 specifies that the address field contains a direct address; 1 specifies that the address field contains an indirect address.
X — index register designator; binary 0, 1, 2, or 3.
0 specifies that no indexing is to be performed; 1 specifies indexing with index register 1; 2 specifies indexing with index register 2, and 3 specifies indexing with index register 3.
L — literal data indicator; binary 0 or 1.
0 specifies that the address field contains a memory address; 1 specifies that the address field contains the data on which the associated instruction operates.

The effects of the I, X, and L indicators upon the address field interpretation are best described with an example. Consider a load A accumulator instruction (see paragraph 4.1.1) for the following discussion.

If the I, X, and L fields each contain zero, and the address field contains m, the A accumulator contains the contents of address m, c (m), after the instruction is executed. In this instance, m is the effective address, and c (m) is the effective operand. If only the L field contains 1 (literal address mode), and the address field contains m, the A accumulator contains m itself after the instruction is executed. In this instance, m is the effective operand. If the I and L fields each contain zero, and the X field contains 1, 2, or 3 (designated with X_n), the A accumulator contains the contents of the address formed by the algebraic sum of m and the contents of X_n, c (m+c(X_n)). In this instance m + c (X_n) is the effective address, and c (m + c (X_n)) is the effective operand. If only the I field contains 1 (indirect addressing) m is not the address of the effective operand, but of a second address. If the I field of the second address is equal to 1, the second address is the address of a third address, and so forth. When an address is found where I=0, that address is the effective address (designated with p). The X field is inspected at each level of indirect addressing, and indexing is performed as specified. If the word that contains p also contains an index register designator, c (p + c (X_n)) is the effective operand. If both I and L contain 1, indirect addresses are accessed until a direct address (p) is found. In this instance (I=1 and L=1), p is the effective operand. Again, indexing is performed at each level where indexing is specified. So if the X field contains n in the word location containing p, the effective address is the address of p, and the effective operand is p + c (X_n).

2.2.2 Operations

In addition to the operation code (OP-CODE) field, the M field and the C field determine the exact operation to be performed on the data.

Generally, the M field performs one of four functions (although special interpretations apply to some operations):

a. M may indicate the accumulator involved in word or double-word operations (A, B, or D, where D is the 64-bit accumulator formed by both A and B).

b. In byte or half-word operations, M indicates the position of the data in accumulator A. For half-word operations, M indicates left half-word or right half-word; for byte operations, a byte (0 through 3).

c. In operations that load, store, or modify the index registers, M denotes the register involved — X1, X2, or X3. Note that indexing (X field) on the address of these instructions is permitted as in any instruction.

d. For conditional branches, the M-field indicates the condition or combination of conditions under which control is to be transferred. The condition indicators that are tested are set as a result of some previous operation.

The M field augments and clarifies the operation specified by the operation code. When the M field contains an entry that is not permitted for the specified operation, the results are unpredictable.

The C field, when applicable and equal to 1, causes the condition-code to be set following instruction execution. The paragraph on each operation code in this manual describes the applicability of the C field.

The meaning of the condition-code value is determined by the operation code used. For accumulator and index register loading, storing, and modification, the condition-code indicates less than zero, equal to zero, or greater than zero. A single operation can thus add data to a register and test the sign of the result. For logical operations, the condition-code indicates all ones, all zeros, or mixed ones and zeros. The compare operations generally compare the contents of a register to the contents of a memory location and set the condition-code to indicate whether the register is greater than, equal to, or less than the data in memory. For the compare operation, the C field must be 1. In general, the meaning of the condition-code is dependent on the operation that causes it to be set.

2.3 FLOATING-POINT ARITHMETIC

The C-8500 single-word precision floating-point number has the following format:

S	EXPONENT	FRACTION
0	1 2 3 4 5 6 7	8 31

The double-word precision floating-point number has the following format:

S	EXPONENT	FRACTION
0	1 2 3 4 5 6 7	8 63

The two formats are identical except for the difference in length. Floating-point numbers are expressed as the product of a signed hexadecimal fraction and the base number 16 raised to a signed power.

Bit 0 of the floating-point number designates the sign, positive or negative, of the fraction.

Bits 1 through 7 of the internal floating-point number form the exponent. The exponent implies the signed power to which the base number 16 is raised. This is done by excess 64 representation; the value in the exponent field of the floating-point number is obtained by adding +64 to the actual signed exponent. The exponent field values of 0 through 127 (decimal) correspond to the signed power -64 through +63 respectively.

An exponent overflow condition occurs when the exponent of the result of a floating-point operation exceeds 127. An exponent underflow occurs when the exponent of the result is less than 0. If the exponent underflows during a floating-point operation, the overflow indicator is set and the referenced accumulator is cleared. If the exponent overflows, the overflow indicator is set, but the accumulator is not cleared. In either case, the operation is terminated.

Bits 8 through 31 in the single-word precision format (bits 8 through 63 in double-word precision) contain the hexadecimal fraction. Each hexadecimal digit occupies 4 bits. Fractions are always represented in a positive form. The sign bit of the floating point number distinguishes between negative and positive exponents. When the leading or highest order digit of the fraction is nonzero, the floating-point number is said to be normalized. Six hexadecimal digits can be contained in the single precision fraction (14 digits in double precision). Add, subtract, negate and normalize operations can be performed on both unnormalized and normalized numbers. The radix point of the floating-point fraction immediately precedes the high-order fraction digit.

Operations are always performed on operands having the same (single-precision or double-precision) format. Although the results of single-precision operations contain 6-hexadecimal fraction digits, the intermediate result extends to 7 digits. The low-order digit of the intermediate result is a significant digit and is used to increase the precision of the final result.

2.4 TRAPPED OPERATION CODES

Certain instructions in the C-8561A-2 repertoire are implemented by a trapping mechanism. Trapped operations provided with the standard system are listed separately in appendix B. Since these functions are software implemented, additional operations can be provided to meet customer requirements.

When a trapped operation code is detected, the status of the arithmetic logic and control unit is stored in fixed locations of protected memory in the following format:

WORD 1

	NEW IAC (P REG)
0 13	14 31

WORD 2

ZEROS OR ONES	EFFECTIVE ADDRESS
0 13	14 31

WORD 3

ZEROS	FUNCTION WORD
0 17	18 31

WORD 4

ZEROS	C	O	CURRENT IAC (P REG)
0 5	6 7	8	14 31

The component parts of the trapped instructions are stored in words 2 and 3 in the format shown. Word 2 holds either the effective address of the actual operand, depending on whether the literal mode is specified and applicable. Word 4 holds the condition code, the overflow indicator, and the contents of the P register at trap time. Word 1 holds the address of the trapped operation code handling routine, resident in protected memory; this is transferred to the P register.

The trapped operation code handler routine branches on the operation code in word 3 to execute the specified function, then restores the arithmetic logic and control unit status as specified in word 4.

section 3
privileged mode and interrupts

3.1 PRIVILEGED MODE

To inhibit interrupts or to access an area in memory known as protected memory, the processor must be placed in privileged mode. This mode is entered whenever an interrupt occurs.

3.1.1 Protected Memory

Protected memory is an area in main core storage that cannot be altered unless the processor is operating in privileged mode. The protected area includes at least the first 512 word locations (00000 through 007FF in hexadecimal). It can be expanded in modules of 512 words up to 16,384 words. The use of protected memory depends on system requirements, but some areas are fixed in any C-8561A-2 Processor; these are listed in appendix D of this manual.

3.1.2 Entering Privileged Mode

Privileged mode is entered by the processor when a branch and set return link to protected area instruction is executed. This instruction (described in detail in section 4) is similar to the branch and set return link instruction used for sub-routine entry, but it branches the program to a fixed memory location in protected memory. The contents of this location are established in the initial program load operation; the word may branch the program back to the next instruction in the originating program, or it may be the entry point to a subroutine that handles all privileged mode operations, depending on the program system.

3.1.3 Exiting Privileged Mode

The Branch and Enable Protection instruction (described in detail in section 4) removes the processor from the privileged mode of operation.

3.2 INTERRUPTS

When an interrupt condition occurs, the contents of the instruction address counter (P register), the overflow indicator, and the condition code are stored in a fixed location of protected memory, and a new address is loaded into the instruction address counter from a second fixed location. The fixed locations have the following format:

WORD 1	ZEROS	NEW IAC (P REG)
	0 13	14 31

WORD 2	ZEROS	C	O	T	OLD IAC (P REG)
	0 5	6 7	8	9	14 31

Where C indicates condition code, 0 is the overflow indicator, and T is the timer interval specification bit (0=timer zero, 1=timer one).

The address of each pair of fixed locations is determined by the type of interrupt. The machine enters the privileged mode when interrupted, and the next instruction to be executed is determined by the new instruction address. Three classes of interrupt can occur; program interrupt, memory parity interrupt, and interval timer interrupt. The conditions that cause these interrupts and the conditions that inhibit them are described in the following paragraphs.

3.2.1 Program Interrupt

A program interrupt occurs under any of the following conditions:

An invalid address is detected.
An invalid operation code is detected.
A direct control instruction is attempted with the processor in the non-privileged mode.
A write into protected memory is attempted with the processor in the non-privileged mode.

An invalid address is detected when a memory address exceeds the highest address in the implemented memory, or when a write to protected memory is attempted with the processor in the nonprivileged mode. An invalid operation code is detected when bits 4 through 7 of the instruction word specify a code that is not defined by the C-8500 instruction set. A program interrupt cannot be inhibited.

3.2.2 Memory Parity Interrupt

A memory parity interrupt occurs when a memory parity error occurs during a memory read initiated by the arithmetic logic and control unit. A memory parity interrupt cannot be inhibited.

3.2.3 Interval Timer Interrupt

The timer interrupt can be inhibited by masking as described in paragraph 3.2.5. A timer interrupt occurs when a fixed time interval determined by the processor service unit elapses. The timer intervals available are 24 and 2.4 milliseconds.

3.2.4 Memory Interrupt Locations

When an interrupt occurs, machine conditions are stored in a fixed memory location. The instruction address counter content is stored in bits 14 through 31; the overflow indicator is stored in bit 8; the condition indicator is stored in bits 6 and 7; bit 9 is set to 1 when a timer interrupt is received from the 2.4 ms timer. For all other interrupts bit 9 is a 0. Bits 0 through 5 and 10 through 13 normally contain 0. The fixed locations determined by the classes of interrupt are shown in table 3-1.

3.2.5 Interrupt Priority and Masking

Interrupt priorities are as follows:

Program interrupt — priority 1
Memory parity interrupt — priority 2
Timer interrupt — priority 3

Table 3-2 summarizes interrupt priorities and interrupt inhibit masking. Each interrupt is listed according to priority in its class.

Table 3-1. Memory Interrupt Locations.

CLASS	LOCATION (hexadecimal)	CONTENT
Program	A0	New instruction address
	A4	Previous instruction address, overflow indicator, and condition indicator
Memory parity	A8	New instruction address
	AC	Previous instruction address, overflow indicator, and condition indicator
Timer	B0	New instruction address
	B4	Previous instruction address, overflow indicator, and condition indicator

Table 3-2. Interrupt Priority and Masking.

CLASS	INTERRUPT	MEMORY LOCATION		PRIORITY	MASK CONDITION
		PREV	NEW		
Program	Invalid address	A4	A0	1A	Cannot be masked
	Illegal operation	A4	A0	1A	Cannot be masked
	Direct control instructions without privileged mode	A4	A0	1A	Cannot be masked
Parity	Memory parity error on alcu read	AC	A8	2	Cannot be masked
Timer	Timer zero (24 ms)	B4	B0	3A	*P register bit 16 masked and privileged mode enabled
	Timer one (2.4 ms)	B4	B0	3B	*P register bit 17 masked

*Bits 16 and 17 of the P register (instruction address counter) contain the least significant bits of the instruction address. These bits are not used for addressing purposes, and can be set and reset with corresponding 1's and 0's in the address field of a branch instruction.

section 4
operation repertoire

The C-8500 operation repertoire is divided into ten functional instruction groups, as follows:

- Data transfer instructions
- Shift instructions
- Logical instructions
- Arithmetic instructions
- Floating-point instructions
- Branch instructions
- Field instructions
- Input-output instructions
- Miscellaneous instructions

Each instruction is described in this section.

4.1 DATA TRANSFER INSTRUCTIONS

4.1.1 Load Accumulator (1C)

The content of the word(s) whose leftmost byte is specified by the effective address is inserted into the accumulator specified in the M field.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

All addressing modes are applicable. If the literal mode is specified, the 18-bit effective operand is loaded into the specified accumulator; the leftmost bit is treated as a sign bit and is propagated to the left across the accumulator.

4.1.2 Load Half-Word (11)

The content of the half-word whose leftmost byte is specified by the effective address is inserted into the half of the A accumulator specified in the M field.

The M field can also specify that the half-word is to be loaded into the right half of accumulator A with the sign bit propagated across the accumulator.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

All addressing modes are applicable. If the literal mode is specified, the rightmost 16 bits of the effective operand are inserted in the specified position of the A accumulator.

4.1.3 Load Byte (00)

The content of the byte specified by the effective address is inserted into the A accumulator in the position specified in the M field.

The unspecified bytes of the A accumulator are not affected.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

All addressing modes are applicable. If the literal mode is specified, the rightmost byte of the effective operand is inserted into the specified position of the A accumulator.

4.1.4 Load Byte and Clear (01)

The content of the byte specified by the effective address is inserted into the A accumulator in the position specified in the M field.

Each bit in the remaining bytes of the A accumulator is set to zero.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

All addressing modes are applicable. If the literal mode is specified, the rightmost byte of the effective operand is inserted into the specified position of the A accumulator.

4.1.5 Load Selective (37)

Selected bits in the A accumulator are replaced by the corresponding bits of the word whose leftmost byte is specified by the effective address. Where a one occurs in the B accumulator, the corresponding bit in the A accumulator is set to the same state as the bit in the memory word. Otherwise the bit in the A accumulator is undisturbed. Thus, the content of B is a mask that controls the loading of A.

The M field is ignored for this operation.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the condition indicator is set as follows:

1 — all bits are one in accumulator A
2 — all bits are zero in accumulator A
3 — neither of the above; accumulator A contains mixed zeros and ones

All addressing modes are applicable. When the literal addressing mode is specified the leftmost bit of the effective operand is propagated to the left to form a 32-bit operand.

4.1.6 Load Magnitude Accumulator (1D)

The absolute value of the word(s) specified by the effective address is loaded into the accumulator specified in the M field.

If the specified memory location contains a negative operand, the two's complement is formed before insertion into the accumulator.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

If the two's complement operation is performed on the maximum 32-bit or 64-bit negative number before insertion into the accumulator, overflow occurs and the overflow indicator is set.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero (overflow)
2 — equal to zero
3 — greater than zero

All addressing modes are applicable. If L equals one, the leftmost bit of the absolute value of the 18-bit effective operand is propagated to the left to form the 32-bit or 64-bit operand. If a negative literal is found, the leftmost bit of the literal is propagated to the left before the literal is negated.

Note

If the literal mode is specified, overflow cannot occur, since the maximum 32-bit negative number cannot be expressed as an 18-bit literal.

4.1.7 Load Index Register (09)

The rightmost 18-bits of the word whose leftmost byte is specified by the effective address is inserted into the index register specified in the M field.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified index register is compared to zero, and the condition indicator is set as follows:

1 — leftmost bit is one
2 — index register content equal to zero
3 — index register content nonzero and leftmost bit is zero

It should be noted that the index register content is always treated as an unsigned integer.

All addressing modes are applicable.

4.1.8 Store Accumulator (54)

Either the content of the accumulator specified in the M field or implied zeros are stored in the word(s) whose leftmost byte is specified by the effective address.

The M field is set as follows:

00 — zeros
01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire accumulator (word or double-word) is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

Note

If the literal and indirect modes of this instruction are both specified, the effective operand is stored in the address field of the first word found having the indirect bit equal to zero. If the literal mode is not specified, the operand address is obtained from this location as usual.

4.1.9 Store Half-Word (5E)

The content of the half of the A accumulator specified in the M field is stored in the half-word whose leftmost byte is specified by the effective address.

The M field is set as follows:

01 — left half
10 — right half

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the accumulator (half-word) is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

4.1.10 Store Byte (4E)

The content of the byte of the A accumulator specified in the M field is stored in the byte specified by the effective address.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator (byte) is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

Note

If the literal and indirect modes of this instruction are both specified, the effective operand is stored in the address field of the first word found having the indirect bit equal to zero. If the literal mode is not specified, the operand address is obtained from this location, as usual.

4.1.11 Store Selective (69)

For each bit in the B register set to one, the corresponding bit in the memory word is set to the value of the corresponding bit in the A register. All other bits in the memory word are undisturbed. The leftmost byte of the memory word is specified by the effective address. Thus, the content of the B register forms a mask that controls the storage operation.

The M field is ignored for this instruction.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified storage location (word) is examined, and the condition indicator is set as follows:

1 — all bits are one in the memory word
2 — all bits are zero in the memory word
3 — neither of the above; the result is mixed zeros and ones

The direct and indirect addressing modes are applicable.

4.1.12 Store Magnitude (55)

The absolute value of the accumulator specified in the M field is stored in memory beginning at the word(s) whose leftmost byte is specified by the effective address.

If the content of the accumulator is a negative operand, the two's complement is formed before the storage operation is performed.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

If the two's complement operation is performed on the maximum 32-bit or 64-bit negative number before insertion into memory, overflow occurs and the overflow indicator is set.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero (overflow)
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

4.1.13 Store Magnitude Half-Word (5F)

The absolute value of the half-word of the A accumulator specified in the M field is stored in memory beginning at the half-word whose leftmost byte is specified by the effective address.

If the content of the A accumulator is a negative number, the two's complement is formed before the storage operation is performed.

The M field is set as follows:

01 — left half
10 — right half

If the two's complement operation is performed on the maximum 16-bit negative number before insertion into memory, overflow occurs and the overflow indicator is set.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator (half-word) is compared to zero, and the condition indicator is set as follows:

1 — less than zero (overflow)
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

4.1.14 Store Index Register (48)

The content of the index register specified in the M field is stored in the address field of the word whose leftmost byte is specified by the effective address.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified index register is compared to zero, and the condition indicator is set as follows:

1 — leftmost bit is one
2 — index register content equal to zero
3 — index register content nonzero and leftmost bit is zero

It should be noted that the index register content is always treated as an unsigned integer.

All addressing modes are applicable.

Note

If the literal and indirect modes of this instruction are both specified, the effective operand is stored in the address field of the first word found having the indirect bit equal to zero. If the literal mode is not specified, the operand address is obtained from this location, as usual.

4.1.15 Exchange Storage With Accumulator Register (40)

The content of the accumulator register specified in the M field is exchanged with the content of the word specified by the effective address.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero after the exchange and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

The direct and indirect addressing modes are applicable.

4.1.16 Exchange Storage With Accumulator Register and Negate (41)

The content of the accumulator register specified in the M field is exchanged with the content of the word specified by the effective address. The two's complement of the accumulator content is formed during the exchange operation.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

If the maximum 32-bit or 64-bit negative number is negated, overflow occurs and the overflow indicator is set.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero after the exchange and the condition indicator is set as follows:

1 — less than zero (overflow)
2 — equal to zero
2 — greater than zero

The direct and indirect addressing modes are applicable.

4.1.17 Exchange Storage With Index Register (43)

The content of the index register specified in the M field is exchanged with bits 14 through 31 of the word specified by the effective address. The remaining bits of the memory word are undisturbed.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero after the exchange and the condition indicator is set as follows:

1 — leftmost bit is one
2 — index register content equal to zero
3 — index register content nonzero and leftmost bit is zero

The direct and indirect addressing modes are applicable.

4.1.18 Exchange Storage With Index and Negate (53)

The content of the index register specified in the M field is exchanged with bits 14 through 31 of the word specified by the effective address. The remaining bits of the memory word are undisturbed. The two's complement of the memory word is formed during the exchange operation.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

If the maximum 18-bit negative operand is negated, overflow occurs and the overflow indicator is set. It should be noted that the final index register content is treated as an unsigned integer.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero after the exchange and the condition indicator is set as follows:

1 — leftmost bit is one
2 — index register content equal to zero
3 — index register content nonzero, and leftmost bit is zero

The direct and indirect addressing modes are applicable.

4.1.19 Transfer Register to Register (56)

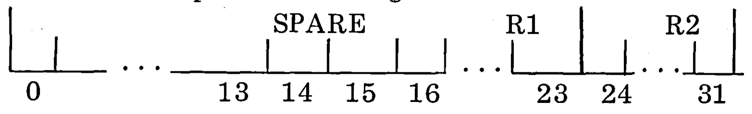
This single operation code is modified by the M field to specify four individual operations. The meanings of the bits in the M field are as follows:

Most Significant Bit of M	Meaning
0	No complement
1	Two's complement

Least Significant Bit of M	Meaning
0	Register transfer (R1 to R2)
1	Register exchange

The effective address specifies two registers (R_1 and R_2) that are involved in the operation. Any pair of the following registers may be specified: A, B, X1, X2 or X3. The D register may only be specified if $R_1 = R_2$.

The format of the word which specifies these registers is as follows:



Where	$A = 1_{16}$	and	$X1 = 81_{16}$
	$B = 2_{16}$		$X2 = 82_{16}$
	$D = 3_{16}$		$X3 = 83_{16}$

If an index register is transferred to A or B, the leftmost 14 bits of A or B are undisturbed. If an index register is a destination register for the contents of A or B, only the rightmost 18 bits are transferred.

If two's complement is specified, R_2 contains the two's complement of the content of R_1 after the operation.

If the two's complement option is taken when A or B is R_1 and R_1 is the maximum 32-bit negative number, overflow occurs. If the two's complement option is taken when X1, X2, or X3 is R_1 and R_1 is the maximum 18-bit negative number, overflow occurs. The overflow indicator is set if overflow occurs.

The following is a list and description of the combinations that the M field can select.

00 — register to register transfer
01 — exchange registers
10 — register to register transfer and negate
11 — exchange registers and negate

Transfer register to register: The registers involved are R_1, the source register, and R_2 the destination register. The content of R_1 is transferred to R_2. The content of R_1 is undisturbed.

Transfer register to register and negate: This instruction performs the same function as the transfer register to register instruction with an additional feature. R_2 contains the two's complement of the content of R_1 after the transfer operation is performed. The content of R_1 is undisturbed.

Exchange registers: The contents of the specified registers (R_1 and R_2) are exchanged.

Exchange registers and negate: This instruction performs the same function as the exchange registers instruction with an additional feature. R_2 contains the two's complement of the content of R_1 after the exchange operation is performed.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the destination register (R_2) is compared to zero, and the condition indicator is set according to the specified destination register.

Accumulator:

1 — less than zero
2 — equal to zero
3 — greater than zero

Index register:

1 — leftmost bit is one
2 — equal to zero
3 — nonzero; leftmost bit is zero

Note that index registers are always considered unsigned.

The effective address provides the register pair designation for this instruction regardless of the state of the L field. The register pair designation can be indirectly specified and modified by indexing.

Note

> The exchange of a register with itself without negation is illegal and the register content is unpredictable for such an exchange.

4.2 SHIFT INSTRUCTIONS

4.2.1 Logical Rotate Left (2E)

The content of the accumulator specified in the M field is shifted left by the value of the rightmost bits of the effective address; the number of bits that determine the shift depends upon which accumulator is specified in the M field. If A or B is specified, the rightmost five bits are examined; if D is specified, the rightmost six bits are examined. During the execution of the rotate instruction, each bit shifted out of the left end of the accumulator is entered into the right end of the accumulator. The sign (leftmost) bit is treated as any magnitude bit.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is examined, and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

Literal mode is always assumed; that is, the rightmost bits of the address field determine the number of bits shifted. However, the address can be specified as indirect and modified by indexing.

4.2.2 Logical Right Shift (2F)

The content of the accumulator specified in the M field is shifted right by the value of the rightmost bits of the effective address; the number of bits that determine the shift depends upon which accumulator is specified in the M field. If A or B is specified, the rightmost five bits are examined; if D is specified, the rightmost six bits are examined. All bits shifted out of the right end of the accumulator are lost, and zeros appear in all bit positions vacated by the shift instruction. The sign (leftmost) bit is treated as any magnitude bit.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is examined, and the condition indicator is set as follows:

1 — all bits are ones
2 — all bits are zeros
3 — mixed zeros and ones

Literal mode is always assumed; that is, the rightmost bits of the address field determine the number of bits shifted. However, the address can be specified as indirect and modified by indexing.

4.2.3 Logical Left Shift (39)

The content of the accumulator specified in the M field is shifted left by the value of the rightmost bits of the effective address; the number of bits that determine the shift depends upon which accumulator is specified in the M field. If A or B is specified, the rightmost five bits are examined; if D is specified, the rightmost six bits are examined. All bits shifted out of the left end of the accumulator are lost. Zeros appear in all bit positions vacated by the shift instruction. The sign (leftmost) bit is treated as any magnitude bit.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is examined, and the condition indicator is set as follows:

1 — all bits are ones
2 — all bits are zeros
3 — mixed zeros and ones

Literal mode is always assumed; that is, the rightmost bits of the address field determine the number of bits shifted. However, the address can be specified as indirect and modified by indexing.

4.2.4 Arithmetic Right Shift (0E)

The content of the accumulator specified in the M field is shifted right by the value of the rightmost bits of the effective address; the number of bits which determine the shift depends upon which accumulator is specified in the M field. If A or B is specified, the rightmost five bits are examined; if D is specified, the rightmost six bits are examined. All bits shifted out of the right end of the accumulator are lost. The value of the sign bit appears in all bit positions vacated by the shift instruction.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

Literal mode is always assumed; that is, the rightmost bits of the address field determine the number of bits shifted. However, the address can be specified as indirect and modified by indexing.

4.2.5 Arithmetic Left Shift (19)

The content of the accumulator specified in the M field is shifted left by the value of the rightmost bits of the effective address; the number of bits that determine the shift depends upon which accumulator is specified in the M field. If A or B is specified, the rightmost five bits are examined; if D is specified, the rightmost six bits are examined. The sign bit is unaffected by this instruction. All magnitude bits shifted out of the left end of the accumulator are lost, and zeros appear in all bit positions vacated by the shift instruction.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

If, during the instruction execution, any magnitude bit unequal to the sign bit is shifted out of the register, the overflow indicator is set. The shift operation continues, however.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is compared to zero, and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

Literal mode is always assumed; that is, the rightmost bits of the address field determine the number of bits shifted. However, the address can be specified as indirect and modified by indexing.

4.3 LOGICAL INSTRUCTIONS

4.3.1 AND Accumulator (24)

The logical product (AND) of the content of the effective address and the content of the specified accumulator is formed and replaces the content of the accumulator. If A or B is specified as the accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64-bits of the double-word specified by the effective address participate in the operation.

The following defines the logical AND operation.

m	n	m AND n
0	0	0
0	1	0
1	0	0
1	1	1

The operation is performed on corresponding bits of the accumulator and memory location.

The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire specified accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the 18 bits of the address field, prefixed with bits equal to the leftmost bit of the effective address, form the 32-bit or 64-bit operand. The result of the operation replaces the content of the specified accumulator, and the condition code reflects the status of the entire accumulator.

4.3.2 AND Half-Word (30)

The logical product (AND) of the half-word specified by the effective address and the specified half (16 bits) of accumulator A is formed and the result replaces the specified half of accumulator A. The remaining half of accumulator A is undisturbed.

The M field specifies which half of accumulator A is involved in the operation. The M field may also specify to AND the half-word into the right half of accumulator A with the sign bit of the half-word propagated to the left to form a 32-bit operand before the AND operation is performed on the entire A accumulator.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the address field participate in the operation with the specified half of the A accumulator.

4.3.3 AND Byte (20)

The logical product (AND) of the byte specified by the effective address and the specified byte in accumulator A is formed and replaces the specified byte in accumulator A. the remaining bytes of accumulator A are undisturbed.

The M field specifies the position in accumulator A of the byte involved in the operation.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the address field participate, as a byte, in the operation.

4.3.4 AND to Storage (64)

The logical product (AND) of the content of the effective address and the content of the accumulator specified is formed and replaces the content of the effective address. The content of the specified accumulator remains undisturbed.

If A or B is specified, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64 bits of the double-word specified by the effective address participate in the operation.

The M field denotes the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified word or double-word is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones.

The direct and indirect addressing modes are applicable.

4.3.5 Inclusive OR Accumulator (25)

The logical sum (OR) of the content of the effective address and the content of the specified accumulator is formed and replaces the content of the accumulator.

If A or B is specified as the accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64 bits of the double-word specified by the effective address participate in the operation.

The following defines the logical OR operation.

m	n	m OR n
0	0	0
0	1	1
1	0	1
1	1	1

The operation is performed on corresponding bits of the register and the memory location.

The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire specified accumulator is examined, and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the 18 bits of the address field, prefixed with bits equal to the leftmost bit of the address field, form the 32-bit or 64-bit operand. The result of the operation replaces the content of the specified accumulator.

4.3.6 Inclusive OR Half-Word (31)

The logical sum (OR) of the half-word specified by the effective address and the specified half (16 bits) of accumulator A is formed and replaces the specified half of accumulator A. The remaining half of accumulator A is undisturbed.

The M field specifies which half of accumulator A is involved in the operation. The M field may also indicate to OR the half-word into the right half of accumulator A with the sign bit of the half-word propagated to the left to form a 32-bit operand before the OR operation is performed on the entire A accumulator.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire specified accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones.

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the address field participate in the operation with the specified half of accumulator A.

4.3.7 Inclusive OR Byte (21)

The logical sum (OR) of the byte specified by the effective address and the specified byte in accumulator A is formed and replaces the specified byte in accumulator A. The remaining bytes of accumulator A are undisturbed.

The M field specifies the position in accumulator A of the byte involved in the operation. The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field should contain a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

The entire accumulator is considered to determine the condition indication.

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the address field participate in the operation.

4.3.8 Inclusive OR to Storage (65)

The logical sum (OR) of the content of the effective address and the content of the specified accumulator is formed and replaces the content of the effective address. The content of the specified accumulator remains undisturbed.

If A or B is specified as the accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64 bits of the double-word specified by the effective address participate in the operation. The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified word or double-word is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 —neither of the above; the result is mixed zeros and ones

The direct and indirect addressing modes are applicable.

4.3.9 Exclusive OR (EOR) Accumulator (27)

The exclusive OR of the content of the effective address and the content of the specified accumulator is formed and replaces the content of the accumulator.

If A or B is specified as the accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64 bits of the double-word specified by the effective address participate in the operation.

The exclusive OR operation is defined by the following table.

m	n	m EOR n
0	0	0
0	1	1
1	0	1
1	1	0

The operation is performed on corresponding bits of the register and the memory location.

The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire specified accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the 18 bits of the address field, prefixed with bits equal to the leftmost bit of the address field, form the 32-bit or 64-bit operand.

4.3.10 Exclusive OR Half-Word (33)

The exclusive OR of the half-word specified by the effective address and of the specified half (16 bits) of accumulator A is formed and replaces the specified half of accumulator A. The remaining half of accumulator A is undisturbed.

The M field specifies which half of accumulator A is involved in the operation. The M field may also specify to exclusive OR the half-word into the right half of accumulator A with the sign bit of the half-word propagated to the left to form a 32-bit operand before the exclusive OR operation, which is then performed on the entire A accumulator.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The C field contains either zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire specified accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the address field participate in the operation with the specified half of accumulator A.

4.3.11 Exclusive OR Byte (23)

The exclusive OR of the byte specified by the effective address and the specified byte in accumulator A is formed and replaces the specified byte in accumulator A. The remaining bytes of accumulator A are undisturbed.

The M field specifies the position in accumulator A of the byte involved in the operation.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the entire A accumulator is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the address field participate in the operation.

4.3.12 Exclusive OR to Storage (65)

The exclusive OR of the content of the effective address and the content of the specified accumulator is formed and replaces the content of the effective address. The content of the specified accumulator remains undisturbed.

If A or B is specified as the accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is specified as the accumulator, all 64 bits of the double-word specified by the effective address participate in the operation.

The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word).

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of the specified word or double-word is examined and the condition indicator is set as follows:

1 — all bits are one
2 — all bits are zero
3 — neither of the above; the result is mixed zeros and ones

The direct and indirect addressing modes are applicable.

4.3.13 Complement (3B)

The one's complement of the content of the specified accumulator or of the effective address is formed and replaces the original content.

The M field specifies either the accumulator to be complemented (A, B, or D) or a memory word to be complemented.

The M field is set as follows:

00 — memory
01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals one, the condition indicator is set after the operation to reflect the content of the accumulator specified or memory word as follows:

1 — all bits are ones
2 — all bits are zeros
3 — mixed zeros and ones

If the M field specifies that a memory word is to be complemented, all combinations of values of the I and X fields are permitted. The L field is ignored. If the M field specifies the A, B, or D, the I, X, and address fields are ignored.

4.4 COMPARE INSTRUCTIONS

4.4.1 Comparative AND (34)

The logical product (AND) of the content of the effective address (m) and the content of the accumulator specified in the M field is formed. If A or B is the specified accumulator, all 32 bits of the word specified by the effective address participate in the operation. If D is the specified accumulator, all 64 bits of the double-word specified by the effective address participate in the operation. The result of the AND operation (r) is compared to the content of the effective address. If equality is not found, r is compared to zero. The results of these comparisons are reflected in the condition indicator settings as described below:

RESULT OF AND OPERATION	RELATION	CONDITION INDICATOR SETTING
r	= m	1
r	= 0 but ≠ m	2
r	= any other value	3

The content of the specified accumulator is not altered by the instruction.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field must be set to one; otherwise the instruction is treated as a no operation.

All addressing modes are applicable. All combinations of values of the I, L, and X fields are permitted. In the instance where L equals one, the leftmost bit of the address field is propagated to the left to form a 32-bit or 64-bit operand.

4.4.2 Compare Logical Accumulator (17)

The content of the specified accumulator is compared to the content of the memory location specified by the effective address. The condition indicator reflects the result of the comparison. The logical compare operation treats both operands as unsigned binary integers; that is, the leftmost bit is not treated as a sign bit. The specified accumulator and memory remain unaltered after the comparison.

The M field specifies the accumulator involved in the operation (A, B, or D) and thus determines the size of the operand (word or double-word). The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

For comparison operations, the C field must be set to one; otherwise the instruction is treated as a no operation.

The possible condition indicator values for this instruction are as follows:

1 — accumulator content less than memory
2 — accumulator content equal to memory
3 — accumulator content greater than memory

All addressing modes are applicable. In the instance where that L equals one, the 18 bits of the address field, prefixed with bits equal to the leftmost bit of the address field, form the 32-bit or 64-bit operand.

4.4.3 Compare Logical Half-Word (10)

The half of accumulator A specified in the M field is compared to the half-word specified by the effective address. The condition indicators reflect the result of the operation. The logical comparison operation treats both operands as unsigned binary integers. The specified accumulator and memory remain unaltered by the comparison.

The M field is set as follows:

01 — left half
10 — right half

For comparison operations, the C field must be set to one; otherwise the instruction is treated as a no operation.

The possible condition indicator values for this instruction are as follows:

1 — accumulator half-word less than memory
2 — accumulator half-word equal to memory
3 — accumulator half-word greater than memory

Note that only the specified half-word of the accumulator is considered in the comparison.

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the address field form the operand.

4.4.4 Compare Logical Byte (0B)

The byte in accumulator A specified in the M field is compared to the byte specified by the effective address. The condition indicator reflects the result of the operation. The logical comparison operation treats both operands as unsigned binary integers. The specified accumulator and memory remain unaltered by the comparison.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

For comparison operations, the C field must be set to one; otherwise the instruction is treated as a no operation.

The possible condition indicator values for the instruction are as follows:

1 — accumulator byte less than memory
2 — accumulator byte equal to memory
3 — accumulator byte greater than memory

Note that only the specified byte in the accumulator is considered in the comparison.

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the address field form the operand.

4.4.5 Compare to Zero (15)

The content of either the specified accumulator or the effective address is algebraically compared to zero. The condition indicator reflects the result. Algebraic comparison considers the operands as signed integers. The specified accumulator and memory remain unaltered by the comparison.

The M field indicates that either the accumulator (A, B, or D) or memory is to be compared to zero. In the latter instance, the content of the effective address (a word) is algebraically compared to zero.

The M field is set as follows:

00 — memory
01 — accumulator A
10 — accumulator B
11 — accumulator D

For comparison operations, the C field must be set to one; otherwise the instruction is treated as a no operation.

The possible condition indicator values for this instruction are as follows:

1 — accumulator (or memory) content less than zero
2 — accumulator (or memory) content equal to zero
3 — accumulator (or memory) content greater than zero.

If the M field specifies that a memory word is to be compared, direct addressing, indirect addressing, and indexing are applicable. The L field is ignored. If the M field specifies A, B, or D, the I, X and address fields are ignored.

4.4.6 Compare Index (16)

The content of the index register specified in the M field is logically compared to the memory word specified by the effective address. The condition code reflects the result of the operation. Logical comparison treats both operands as unsigned binary integers. The index register and memory remain unaltered by the comparison.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

For comparison operations, the C field must be set to one; otherwise the instruction will be treated as a no operation.

The possible condition indicator values for this instruction are as follows:

1 — index register content less than memory
2 — index register content equal to memory
3 — index register content greater than memory

All addressing modes are permitted. In the instance where L equals one, the entire address field forms the operand.

4.4.7 Compare Selective (35)

Each bit of accumulator A is compared with the corresponding bit of the word specified by the effective address, provided the corresponding bit of accumulator B is one. That is, if the nth bit of accumulator B is one, then the nth bit of the memory word and the nth bit of accumulator A are compared. Bits of accumulator A and the memory word corresponding to zero bits of accumulator B are not compared. Thus, the content of accumulator B form a mask, which controls the comparison.

The M field is ignored for this operation.

For comparison operations, the C field must be set to one. Otherwise the instruction is treated as a no operation.

The possible condition indicator values for this instruction are as follows:

1 — none of the bits compared are equal
2 — all the bits compared are equal
3 — some but not all of the bits compared are equal

All addressing modes are applicable. In the instance where L equals one, the leftmost bit of the 18-bit address field is propagated to the left to form a 32-bit operand.

4.4.8 Compare Algebraic (14)

The content of the accumulator specified in the M field is algebraically compared to the content of the memory location specified by the effective address. The condition indicator reflects the result of the comparison. Algebraic comparison treats both the accumulator content and the memory content as signed binary integers. The specified accumulator and memory remain unaltered by the comparison.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

For comparison operations, the C field must be set to one. Otherwise the instruction will be considered a no operation.

The possible condition indicator values for this instruction are as follows:

1 — accumulator content less than memory
2 — accumulator content equal to memory
3 — accumulator content greater than memory

All addressing modes are applicable. In the instance where L equals one, the sign (leftmost) bit of the 18-bit address field is propagated to the left to form a 32-bit or 64-bit operand.

4.5 ARITHMETIC INSTRUCTIONS

4.5.1 Add Accumulator (06)

The single or double-word specified by the effective address is added to the accumulator specified in the M field. The sum replaces the content of the accumulator involved in the operation.

Addition is performed by adding the 32 or 64 bits of the single- or double-word operands. If the carries out of the sign bit position and the high-order numeric bit position of the result agree, the sum is satisfactory; if they disagree, an overflow condition exists and the overflow indicator is set. The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either 0 or 1. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the address field is expanded into a signed word of appropriate length by propagating the high-order bit to the left.

4.5.2 Add Half-Word (12)

The half-word specified by the effective address is added to the left or right half-word of accumulator A as specified in the M field.

The sum is placed in the specified half-word of the A accumulator. The M field can also specify that the half-word is to be added to the right half of accumulator A with the sign bit propagated to the left to form a 32-bit operand. The sign bit is extended before the addition operation is performed.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The left half-word of the A accumulator is altered whenever a carry occurs out of the high-order bit of the sum when addition to the right half-word of the A accumulator with extended sign is specified.

Overflow occurs when, as a result of the addition, the carry out of the A accumulator sign bit disagrees with the carry out of the accumulator high-order numeric bit (position 1).

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are permitted. In the instance where L equals one, the rightmost 16 bits of the address field form the signed operand.

4.5.3 Add Byte (02)

The byte specified by the effective address is added to the A accumulator byte specified in the M field. The sum is placed in the specified A accumulator byte.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

One or more accumulator bytes to the left of the byte specified by M, will be altered if a carry occurs out of the high-order bit of the sum (except where the leftmost byte is specified).

Overflow occurs when, as a result of the addition, the carry out of the accumulator sign bit disagrees with the carry out of the accumulator high-order numeric bit (position 1). Overflow may also occur as a direct result of the addition when adding to the leftmost byte, but it is an indirect result when adding to any of the other bytes.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the address field participate in the operation.

4.5.4 Replace Add to Memory (42)

The accumulator specified in the M field is added to the single- or double-word specified by the effective address. The sum replaces the data at the specified effective address, and the accumulator remains unaltered by the instruction.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

Addition is performed by adding the 32 or 64 bits of the single-or double-word operands. If the carries out of the sign bit position and the high-order numeric bit position of the result agree, the sum is satisfactory; if they disagree, an overflow condition exists and the overflow indicator is set.

The C field can be set as desired. If C equals one, the condition indicator is set prior to the next instruction execution and reflects the new content of the memory location specified by the effective address.

1 — memory content is less than zero
2 — memory content is equal to zero
3 — memory content is greater than zero

The direct and indirect addressing modes are applicable.

4.5.5 Replace Add M to Memory (52)

The positive value specified by the M field is added to the memory word specified by the effective address, and the sum replaces the memory word. The M field specifies the increment that is added to the word contained in the specified memory location.

The M field is set as follows:

00 — increment by 1
01 — increment by 2
10 — increment by 3
11 — increment by 4

If the carries out of the sign bit position and the high-order numeric bit position of the result agree, the sum is satisfactory; if they disagree, an overflow condition exists and the overflow indicator is set.

The C field contains either a zero or one. If C equals one, the condition indicator is set piror to the next instruction execution and reflects the new content of the memory location specified by the effective address.

1 — memory content is less than zero
2 — memory content is equal to zero
3 — memory content is greater than zero

Only the direct and indirect addressing modes are applicable.

4.5.6 Repeat Add (57)

The contents of a series of words starting at the location specified by the effective address are added to or subtracted from the A accumulator, depending upon the content of the B accumulator. A one in the B accumulator causes the content of the corresponding memory word to be subtracted from accumulator A; whereas, a zero causes the content to be added to accumulator A. Bits in the B accumulator are examined from left to right with bit 0 to corresponding to the first iteration, bit 1 to the second and so on until the operation is terminated.

The number of words to be added to or subtracted from accumulator A must be in bits 13-17 of index register 1 before the repeat add instruction is executed. The maximum allowable count is 32. If index register 1 contains zero initially, 32 iterations are performed. If index register 1 contains one, a single iteration is performed.

Index register 1 is decremented by one and the least significant 5 bits are examined for zeros after each iteration. If bits 13-17 of index register 1 are zero, the operation is terminated. The operation is terminated after 32 iterations regardless of the state of index register 1.

The M field is ignored for this instruction.

If overflow occurs on any iteration, the overflow indicator is set and the operation continues. The overflow indicator is not cleared if overflow does not occur.

The C field contains either a zero or one. If C equals zero, no condition indicator is set. If C equals one, the content of accumulator A is compared to zero and the condition indicator is set as follows:

1 — less than zero
2 — equal to zero
3 — greater than zero

The effective address provides the starting location of the series of words to be added to or subtracted from the A accumulator, regardless of the state of L. Direct addressing, indirect addressing, and indexing are applicable.

4.5.7 Subtract Accumulator (07)

The single- or double-word specified by the effective address is subtracted from the accumulator specified in the M field, and the difference is placed in the specified accumulator.

If the borrow required out of the high-order bit and the sign bit of the result agree, the difference is satisfactory; if they disagree, an overflow condition exists and the overflow indicator is set.

The M field is set as follows:

01 — accumulator A
10 — accumulator B
11 — accumulator D

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the address field is expanded into a signed word of appropriate length by propagating the high-order bit to the left.

4.5.8 Subtract Half-Word (13)

The half-word specified by the effective address is subtracted from the left or right half-word of the A accumulator. The difference is placed in the A accumulator.

The M field specifies the A accumulator half-word, which participates as the minuend of the operation. The M field can also indicate that the memory half-word with the sign bit propagated left to form a 32-bit operand is to be subtracted from the right half of accumulator A. The memory half-word sign bit is extended before the subtraction operation is performed.

The M field is set as follows:

01 — left half
10 — right half
11 — right half with extended sign

The left half-word of the A accumulator is altered whenever a borrow occurs out of the high-order bit position of the difference and the right half-word of the A accumulator participates in the operation.

Overflow occurs when, as a result of the subtraction, the borrow required out of the high-order bit and the sign bit of the result disagree. Overflow can occur as a direct result of the subtraction operation when subtracting from the left half-word, but it is an indirect result when subtracting from the right half-word.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the address field form the signed operand.

4.5.9 Subtract Byte (03)

The byte specified by the effective address is subtracted from the A accumulator byte specified in the M field. The difference is placed in the specified A accumulator byte.

The M field is set as follows:

00 — byte 1 (leftmost byte)
01 — byte 2
10 — byte 3
11 — byte 4

One or more accumulator bytes to the left of the byte specified by M will be altered if a borrow occurs out of the high-order bit of the difference.

Overflow occurs when, as a result of the subtraction, the borrow required out of the high-order bit and the sign bit of the result disagree. Overflow may also occur, as a direct result when subtracting from the leftmost byte, but it is an indirect result when subtracting from one of the other bytes.

The C field contains either a zero or one. If C equals one, the condition indicator is as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the rightmost 8 bits of the effective address participate, in the operation.

4.5.10 Multiply (OD)

The signed product of the multiplier, which is specified by the effective address, and the multiplicand, which is the content of the A accumulator, are placed in the 64-bit D accumulator. Both multiplier and multiplicand are treated as 1-sign bit and 31 magnitude bits, except where L equals 1. In this instance, the multiplier has a sign bit and 17 magnitude bits. Since the product is placed in the 64-bit D accumulator, overflow cannot occur.

The sign of the product is determined algebraically except that a zero product always has a positive sign.

The M field is not applicable to this instruction.

The C field contains either a zero or one. If C equals one, the condition indicator is set prior to the next instruction execution to reflect the new content of the D accumulator as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the 18 bits of the address field with the leftmost bit propagated to the left forms the signed 32-bit operand.

Note

This instruction is available only with the optional algorithm unit.

4.5.11 Multiply Half-Word (OC)

The product of the half-word multiplier specified by the effective address, and the multiplicand contained in the A accumulator is placed in the A accumulator. Both the product and multiplicand are 32-bit signed integers. The multiplier is a signed 16-bit integer.

The sign of the product is determined algebraically, except that a zero product is always positive.

If the carries out of the sign bit position and the high-order numeric bit position of the product disagree, an overflow condition exists and the overflow indicator is set. Note that the product is restricted to 31 magnitude bits.

The M field is not applicable to this instruction.

The C field contains either a zero or one. If C equals one, the condition indicator is set prior to the next instruction execution and reflects the new content of the A accumulator as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. In the instance where L equals one, the rightmost 16 bits of the effective address are treated as a signed 16-bit multiplier.

Note

This instruction is available only with the optional algorithm unit.

4.5.12 Divide (OA)

The signed content of the 64-bit D accumulator is divided by a signed 32-bit word contained in the memory location specified by the effective address. The B accumulator content is replaced by the signed 32-bit integral quotient and the A accumulator content is replaced by the signed 32-bit integral remainder resulting from the division.

The sign of the quotient is determined algebraically, except that a zero quotient and a zero remainder are positive. The remainder has the sign of the dividend.

Division by zero is not permitted and results in an overflow condition; no division will occur. If the quotient exceeds 32 bits, overflow occurs and the content of the accumulator is unpredictable.

The M field is not applicable to this instruction.

The C field contains either a zero or one. If C equals one, the condition indicator is set prior to the next instruction execution and reflects the new content of the B accumulator (quotient) as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

All addressing modes are applicable. When the literal addressing mode is used, the 18-bit address field is treated as a signed divisor containing 17 magnitude bits, and the sign bit of the divisor is propagated to the left to form the 32-bit signed operand.

Note

The content of the entire D accumulator is treated as the dividend. Thus if the dividend is entirely in B, A must be cleared prior to the divide operation. Also, if the divisor is greater in magnitude than the dividend, a quotient of zero results with the remainder equal to the dividend.

Note

This instruction is available only with the optional algorithm unit.

4.5.13 Modify Index (08)

The content of the index register (always considered unsigned) specified by the M field is added to the signed word specified by the effective address. The sum becomes the new value of the specified index register.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field contains either a zero or one. If C equals one, the condition indicator is set prior to the next instruction execution and reflects the new content of the index register as follows:

1 — leftmost bit of index register content is one
2 — index register content is zero
3 — index register content is nonzero; leftmost bit is zero

If the sum exceeds 32 bits, the overflow indicator is set.

All addressing modes are applicable.

4.6 FLOATING-POINT INSTRUCTIONS

4.6.1 Floating Add (4C)

The floating-point number contained in the specified accumulator is added to the floating-point number (of equivalent length) specified by the effective address. The normalized sum is entered into the specified accumulator. The result will be normalized.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The overflow indicator is set whenever exponent overflow or underflow of the final result occurs. If exponent overflow occurs, the result is unreliable. If exponent underflow occurs, the result is set to zeros.

Addition is executed by performing an exponent comparison and a fraction addition. If the exponents are unequal, the fraction with the smaller exponent is right-shifted one hexadecimal digit. For each right shift, one is added to the smaller exponent and a zero (hexadecimal) is entered in the high-order digit position of the fraction. This process continues until the two exponents are equal.

In single-word precision, if a right shift is required to cause the exponents to agree, the last digit shifted out of the A accumulator is retained to increase the precision of the intermediate fraction. When the two exponents agree, the fractions are added algebraically to form an intermediate fraction. If a carry occurs out of the high-order digit, the intermediate fraction is right-shifted by one digit such that the carry occupies the high-order digit of the intermediate fraction, and one is added to the exponent value. The occurrence of exponent overflow as a result of the above right shift terminates the operation.

For each left shift required to normalize the fraction, the exponent value is decremented by one. The operation is terminated at this point if exponent underflow occurs.

If the sum of the floating-point fractions is zero, the entire floating-point number (single-word or double-word precision) is set to zero.

The floating-point sum is entered into either the A or D accumulator as specified by the M field. In short precision the B accumulator is not modified.

Only the direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.2 Floating Add Unnormalized (4D)

The floating-point number contained in the specified accumulator is added to the floating-point number (of equivalent precision) specified by the effective address. The sum is entered in the accumulator involved in the operation. The result is not normalized following the add operation.

The M field specifies the accumulator (A or D) involved in the operation and implies a single-word or double-word precision operand located at the effective address.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The overflow indicator is set whenever exponent overflow of the final result occurs, and the resulting accumulator content is unreliable.

Addition is executed by performing an exponent comparison and a fraction addition. If the exponents are unequal, the fraction with the smaller exponent is right-shifted one hexadecimal digit.

For each right shift, one is added to the smaller exponent and a zero (hexadecimal) is entered in the high-order digit position of the fraction. This process continues until the two exponents are equal.

In single-word precision, if a right shift is required to cause the exponents to agree, the last digit shifted out of the A accumulator is retained to increase the precision of the intermediate fraction. When the two exponents agree, the fractions are added algebraically to form an intermediate fraction. If a carry occurs out of the high-order digit, the intermediate fraction is right-shifted by one digit such that the carry occupies the high-order digit of the intermediate fraction, and one is added to the exponent value. The occurrence of exponent overflow resulting from the above right shift terminates the operation.

If the sum of the floating-point fractions is zero, the entire floating-point number (single-word or double-word precision) is set to zero.

The floating-point sum is entered into either the A or D accumulator as specified by the M field. In short precision the B accumulator is not modified.

Only the direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.3 Floating Compare (6A)

The floating-point number contained in the specified accumulator is compared to the floating-point number (of equivalent precision) specified by the effective address. The accumulator and memory contents remain unaltered by this operation.

The M field specifies the accumulator (A or D) that is involved in the comparision and implies a single-word or double-word precision operand located at the effective address.

The floating compare instruction does not normalize either operand before the comparison operation is performed. The comparison is accomplished by performing a floating-point subtraction. The result of this subtraction, if zero, indicates an equality.

For comparison operations, the C field must be set to one; otherwise the instruction is treated as a no operation. (Software . . . always sets condition code.)

The possible condition indicator values for this instruction are as follows:

1 — accumulator content less than memory
2 — accumulator content equal to memory
3 — accumulator content greater than memory

The direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.4 Floating Divide (6B)

The floating-point number contained in the specified accumulator (dividend) is divided by the floating-point number of equivalent precision specified by the effective address. The quotient replaces the content of the specified accumulator.

The M field specifies the accumulator (A or D) involved in the operation and implies a single word or double-word precision operand located at the effective address.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

If either floating-point number is zero, the operation is terminated and the accumulator remains unchanged. If the divisor is zero, the overflow indicator is set.

The floating-divide operation assumes that the divisor and dividend are normalized. The exponent of the divisor is subtracted from the exponent of the dividend and the difference is increased by 64 to form an excess 64 exponent. The fraction of the floating-point quotient is then calculated. Although this fraction does not require normalization, a right shift may be required. If a right shift is necessary, the intermediate exponent is incremented by one.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The overflow indicator is set whenever exponent overflow or exponent underflow of the final result occurs. If exponent overflows occurs, the result is unreliable, and if exponent underflow occurs, the result is set to zero.

The direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.5 Floating Multiply (6F)

The floating-point number contained in the specified accumulator is multiplied by the floating-point number (of equivalent precision) contained in the effective address. The normalized floating-point product replaces the original content of the specified accumulator.

The M field specifies the accumulator (A or D) involved in the operation and implies a single-word or double-word precision operand located at the effective address.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

The floating multiply operation assumes that the multiplier and multiplicand are normalized. The exponents are added and reduced by 64 to form an intermediate exponent value. The product of the fractions (if nonzero) is normalized, and the intermediate exponent value is decremented by 1 for each hexadecimal digit left-shifted. If single word precision is indicated, the fractional product represents only the six high-order hexadecimal digits calculated. Should the product be zero, the entire floating-point number (single- or double-precision) entered in the accumulator is set to zero.

Exponent overflow or exponent underflow of the final product causes the overflow indicator to be set and the operation is terminated. If exponent overflows occurs, the final result is unreliable. If exponent underflow occurs, the result is set to zeros.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.6 Floating Negate (6E)

The floating-point number contained in either the specified accumulator or the effective address is negated by sign-bit inversion.

The M field specifies either the accumulator (A or D) or the memory word to be negated. The M field is set as follows:

00 — memory
01 — accumulator A
11 — accumulator D

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator (or memory) content is less than zero
2 — accumulator (or memory) content is equal to zero
3 — accumulator (or memory) content is greater than zero

If the M field specifies that a memory word is to be negated, direct addressing, indirect addressing and indexing are applicable. If the M field specifies either A or D, the I, X, and address fields are ignored.

Note

This instruction is available only with the optional algorithm unit.

4.6.7 Floating Normalize (6D)

If the floating-point number contained in either the specified accumulator or the effective address is nonzero, the number is normalized.

The M field specifies either the accumulator (A or D) or the memory word to be normalized. The M field is set as follows:

00 — memory
01 — accumulator A
11 — accumulator D

If the leading or high-order hexadecimal digit of the floating-point fraction is zero, the fraction is shifted one digit to the left. For each left shift the exponent value is decremented by one. This process continues either until exponent underflow occurs or until the high-order digit is nonzero. The normalized result of this operation replaces the original content of the specified accumulator or the effective address.

If exponent underflow occurs, the overflow indicator is set, the operation is terminated, and the final result is unreliable.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator (or memory) content is less than zero
2 — accumulator (or memory) content is equal to zero
3 — accumulator (or memory) content is greater than zero

If the M field specifies that a memory word is to be normalized, direct addressing, indirect addressing, and indexing are applicable. If the M field specifies either A or D, the I, X, and address fields are ignored.

Note

This instruction is available only with the optional algorithm unit.

4.6.8 Floating Subtract (5C)

The floating-point number contained in the effective address is subtracted from the floating-point number contained in the specified accumulator. The difference replaces the content of the accumulator involved in the operation, and the result is normalized.

The M field specifies the accumulator (A or D) involved in the operation and implies a single-word or double-word precision operand located at the effective address.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

Floating-point subtraction is accomplished by inverting the sign bit of the number specified by the effective address and then performing a floating-point addition.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The overflow indicator is set whenever exponent overflow or exponent underflow of the final result occurs. If exponent overflow occurs, the resulting accumulator content is unreliable. If exponent underflow occurs, the accumulator content is set to zero.

The direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.6.9 Floating Subtract Unnormalized (5D)

The floating-point number contained in the effective address is subtracted from the floating-point number in the specified accumulator. The unnormalized difference replaces the content of the specified accumulator.

The M field specifies the accumulator (A or D) involved in the operation and implies a single-word or double-word precision operand located at the effective address.

The M field is set as follows:

01 — accumulator A
11 — accumulator D

Floating-point subtraction is accomplished by inverting the sign bit of the number specified by the effective address and then performing a floating-point addition.

The C field contains either a zero or one. If C equals one, the condition indicator is set as follows:

1 — accumulator content is less than zero
2 — accumulator content is equal to zero
3 — accumulator content is greater than zero

The overflow indicator is set whenever exponent overflow of the final result occurs. If exponent overflow occurs, the result is unreliable.

The direct and indirect addressing modes are applicable.

Note

This instruction is available only with the optional algorithm unit.

4.7 BRANCH INSTRUCTIONS

4.7.1 Branch on A Accumulator (58)

The branch on A instruction causes the A accumulator to be inspected for the condition (zero, nonzero, positive, negative) specified in the M field. If the A accumulator is set to the value specified in the M field, the branch address is transferred to the instruction address counter. If not, then the instruction specified by the present content of the instruction address counter is executed. In either case, the instruction address counter specifies the location of the next instruction.

The M field values and related branch conditions are as follows:

00 — accumulator A zero
01 — accumulator A nonzero
10 — accumulator A plus
11 — accumulator A minus

The sign bit is tested to determine if the accumulator is plus or minus and the entire content of the accumulator is checked to determine either zero or nonzero equality.

The C field is not applicable to this instruction.

The direct and indirect addressing modes are applicable. The L field is ignored.

4.7.2 Branch (59)

The operation of the branch instruction is specified in the M field.

The M field is set as follows:

00 — Branch and set return link
01 — Branch unconditional
10 — Branch on overflow
11 — Branch on bits even (even number of bits in A accumulator)

The resulting operations performed by the branch instructions are as follows:

Branch and set return link: The content of the instruction address counter is stored in the right most 18 bits of the word specified by the effective address. The leftmost 14 bits of the word are undisturbed. The content of the instruction address counter is replaced by the sum of the effective address plus four. An unconditional branch to the instruction specified by the resultant content of the instruction address counter then occurs.

Branch unconditional: The effective address unconditionally replaces the content of the instruction address counter and thereby specifies the location of the next instruction.

Branch on overflow: The content of the instruction address counter is replaced with the effective address whenever the overflow indicator is set. After the overflow indicator is tested, it is reset. If the overflow indicator is initially reset, the instruction specified by the current content of the instruction address counter is executed. In either case, the instruction address counter specifies the location of the next instruction. Once the overflow indicator is set, it remains set until it has been tested.

Branch on bits even: If the total number of bits set to one in the A accumulator is even, the branch instruction replaces the content of the instruction address counter with the effective address. If the sum is an odd number, the instruction address counter retains the original content. In any case, the final content of the instruction address counter specifies the address of the next instruction.

The C field is not applicable to the branch instruction.

The direct and indirect addressing modes are applicable. The L field is ignored.

Note

The branch and set return link instruction is particularly suited to providing an effective address for the final (exit) instruction of a frequently used closed subroutine. As an example, consider the closed subroutine RVD to exist in memory from location x through y; consider the branch and return link instruction to be located in memory at location m. The effective address specified by the branch and return link instruction would be x, the instruction address counter content when executed the branch and return link instruction would be (m + 4) and the start of subroutine return address would be at location x. This arrangement would cause subroutine RVD to be executed with a final return of control (normally through indirect addressing) to location (m + 4).

4.7.3 Branch on Indicator (78)

The M and C fields are combined to form a binary number with values of 0, 1, 2, 3, 4, 5, 6 or 7. Each bit of the binary number corresponds to a specific state of the condition indicator as indicated by the following:

BIT	CONDITION INDICATOR VALUE
First bit of M	3
Second bit of M	2
C	1

Whenever a bit in the field is set, the instruction inspects the state of the condition indicator. If the specified state exists, the instruction address counter content is replaced with the effective address. If the specified state does not exist, the instruction address counter retains its content. In any case, the final content of the instruction address counter specifies the location of the next instruction. Thus, a value of 5 (101 binary) specified by the M and C field causes a branch if the condition indicator has value 0 (first state) or value 2 (third state). The meaning of the condition indicator state depends on the operation that caused it to be set.

When M = 7, the content of the instruction address counter is always replaced by the branch address. If M = 0, the instruction address counter always retains its content. Thus, an M value of 7 is equivalent to an unconditional branch, and a value of 0 is effectively a no operation.

The direct and indirect addressing modes are applicable. The L field is ignored.

To clarify the use of the branch on indicator instruction, the following chart correlates the value of the condition indicator and the values the programmer places in the M and C fields. The relationship shown in the chart exists after a compare instruction is executed.

CONDITION CODE VALUE	RELATION OF COMPARISON	COMBINED M AND C FIELD CONTENT
0	Results in no operation	0
1	Less than	1
2	Equal to	2
2	Greater than	4
1 or 2	Less than or equal to	3
1 or 3	Greater than or less than	5
2 or 3	Greater than or equal to	6
1, 2, or 3	Results in unconditional branch	7

The instruction executed dictates the actual meaning of the condition code setting, but the relationship between the condition indicator value and the M field value as shown previously remains constant.

4.7.4 Branch and Set Index (5B)

The content of the index register specified in the M field is replaced by the content of the instruction address counter and a branch to the location specified by the effective address occurs by replacing the content of the instruction address counter with the branch address.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field is not applicable to this instruction.

The direct and indirect addressing modes are applicable. The L field is ignored.

4.7.5 Branch on Index Zero (5A)

If the content of the index register specified by the M field is zero, the content of the instruction address counter is replaced by the effective address and a branch to that location occurs. If the content of the index register specified by the M field is not zero, then the instruction specified by the current content of the instruction address counter is executed.

The M field is set as follows:

01 — index register 1
10 — index register 2
11 — index register 3

The C field is not applicable to this instruction.

The direct and indirect addressing modes are applicable. The L field is ignored.

4.7.6 Branch and Set Return Link Protected (18)

The branch and set return link protected instruction is used to transfer control from the applications program to a control program in protected memory and to point to parameters that define a service requested by the applications program.

The instruction stores the instruction address counter in bits 14-31, the overflow indicator in bit 8, and the condition code in bits 6 and 7 at a fixed location within protected main core storage. (See appendix D for a description of fixed locations in protected memory.) This instruction also sets the privileged mode to enable the execution of privileged communication instructions and inhibit memory protection for computer unit access.

The M field has no meaning with this instruction.

The C field is not applicable to the branch instruction.

The various addressing modes have no meaning with this instruction.

4.7.7 Branch and Enable Protection (79)

The branch and enable protection instruction returns control to the applications program upon completion of a service request or service of a clock interrupt.

The instruction causes an unconditional branch to the location specified by the effective address. It also sets the processor mode, which inhibits execution of privileged communication instructions and enables memory protection.

The M field has no meaning for this instruction.

The C field is not applicable to the branch instruction.

The direct and indirect addressing modes are applicable. The L field is not used in determining the branch address.

4.8 INPUT-OUTPUT INSTRUCTIONS

The C-8561A-2 processor requires no input-output instructions because all communication equipment is initiated by their own (simultaneous) access of fixed memory locations.

4.9 MISCELLANEOUS INSTRUCTIONS

4.9.1 Execute (4A)

The execute instruction causes an instruction located at the effective address to be executed.

The content of the instruction address counter is not modified by this instruction, and unless the instruction specified by the effective address directly affects the content of the instruction address counter (such as a branch instruction), program control is returned to the next sequential instruction. The M and C fields are not applicable to this instruction.

The direct and indirect addressing modes are applicable.

Note

> The instruction executed may be itself an execute instruction; any number of execute instructions may be performed in this manner. If the first nonexecute instruction following the original execute instruction does not alter the instruction address counter, program control is returned to the instruction following the original execute instruction.

4.9.2 Direct Control (7E)

The direct control instruction is used to initiate actions by processor modules other than the alcu. If an attempt is made to execute the direct control instruction while the computer is in processor mode, the program is interrupted. The M field together with the 8 low order bits of the effective address specify the function to be performed as follows:

M FIELD	EFFECTIVE ADDRESS	FUNCTION
01 or 10	0000 1000	initiate algorithm unit
01	0000 0000	stop timer 1
01	0001 0000	stop timer 0
10	0000 0000	reset and start timer 0
10	0001 0000	reset and start timer 1

The algorithm unit will expect to find all hardware registers stored in memory locations hexadecimal DO-DC as they would normally be stored following a trapped op code interrupt. In addition the algorithm unit will use memory location hexadecimal C4 as the A accumulator. These locations will be updated as required upon completion of the operation. The condition code returned by the algorithm unit will be the correct code for the operation performed if the C field of the command (stored in memory location hexadecimal D8) is set. If the C field is not set the condition code will be set equal to the value at bits 6 and 7 of memory location hexadecimal DC. If an attempt is made to initiate an algorithm unit and none is present, the condition code will be set to 3 and no operation will be performed. The timers which are controlled by the four direct control timer instructions are the 24 millisecond timer zero and the 2.4 millisecond timer one in the processor service unit (refer to section 1.1.5). These timers will also be started (but not reset) every time that the reset machine failure monitor instruction is executed as explained in the next paragraph. The condition code will always be set to 3 following any of the four direct control timer instructions.

4.9.3 Reset Machine Failure Monitor (3F)

The reset machine failure monitor instruction resets an independent counter. If the counter is allowed to decrement to 0, an external alarm is generated, indicating detection of a hardware error condition. The machine failure monitor does not detect program errors such as continual looping. The monitor console decrements to zero in 213 milliseconds, and resetting the timeout of the counter indicates that the program is running in a normal manner. The C field and the various addressing modes are not applicable. The applications program normally need not be concerned with this instruction since the various control programs in protected memory reset the machine failure monitor when they gain control at timer interrupt frequency.

This instruction will also instruct the processor service unit to update the operational status of all processor hardware to agree with the processor control word in memory location hexadecimal 40. (Refer to section 1.1.5). In addition, the reset machine failure monitor will start (but not reset) both timer zero and timer one in the processor service unit.

appendix a
numbering systems

INTRODUCTION

A numbering system is an orderly system of symbols controlled by a basic set of rules. The first thing necessary to know when working with a specific numbering system is its base radix. This indicates the number of digit marks used in the particular system. The decimal system, for example, operates on a base of 10. Therefore, there are ten different marks in this system: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. In contrast, the binary system has a base of 2 which permits only two marks: 0 and 1. The hexadecimal system has a base of 16.

THE DECIMAL NUMBER SYSTEM

Numbers in the decimal system are formed by a summation of products. Each of these products is composed of a mark multiplied by an integral power of the base 10. (The marks are 0, 1, 2 through 9.) The first mark to the left of the decimal point is multiplied by 10^0 (one), so this mark is counted at face value. The second mark to the left is multiplied by 10^1 (ten), and so on. The complete number is the sum of all of the products as shown below:

$$\begin{array}{lr}
6214.389 = & \\
6 \times 10^3 = & 6000. \\
2 \times 10^2 = & 200. \\
1 \times 10^1 = & 10. \\
4 \times 10^0 = & 4. \\
3 \times 10^{-1} = & .3 \\
8 \times 10^{-2} = & .08 \\
9 \times 10^{-3} = & .009 \\
\hline
 & 6214.389
\end{array}$$

THE BINARY NUMBER SYSTEM

Numbers in the binary system consist of the sum of products of integral powers of 2 (1, 2, 4, 8, etc.). In a binary number, if there is a 1 in a given position, then the power of 2 associated with that position is included in the sum; if there is a 0 in the position, then that power of 2 is omitted. The binary digit position is called a *bit* position; the digits are called bits.

COUNTING IN THE BINARY SYSTEM

Decimal: Binary	Decimal: Binary	Decimal: Binary
1 = 1	10 = 1010	19 = 10011
2 = 10	11 = 1011	20 = 10100
3 = 11	12 = 1100	21 = 10101
4 = 100	13 = 1101	22 = 10110
5 = 101	14 = 1110	23 = 10111
6 = 110	15 = 1111	24 = 11000
7 = 111	16 = 10000	25 = 11001
8 = 1000	17 = 10001	
9 = 1001	18 = 10010	

The following conversion of the binary equivalent for number 23_{10} from binary-to-decimal indicates the positional values in the binary system:

BINARY NUMBER:	1	0	1	1	1
POSITION VALUE:	16	8	4	2	1

DECIMAL SUMMATION: $16 + 0 + 4 + 2 + 1 = 23$

DECIMAL-TO-BINARY CONVERSION

To convert, divide the decimal number by the radix of the binary number system (2) and then divide the quotient again by 2 following the rules below.

Rules

1. If there is no remainder from any division, record a "0".
2. If there is a remainder, it will always be a "1". Record it.
3. When the point is reached that the dividend is a "1", record it as the remainder (at this point the conversion is complete).

Example

Conversion of the Decimal 125 to its Binary Equivalent.

(Remainders are enclosed in rectangle for clarification)

$125 \div 2 = 62 +$	$\boxed{1}$
$62 \div 2 = 31 +$	$\boxed{0}$
$31 \div 2 = 15 +$	$\boxed{1}$
$15 \div 2 = 7 +$	$\boxed{1}$
$7 \div 2 = 3 +$	$\boxed{1}$
$3 \div 2 = 1 +$	$\boxed{1}$
$1 \div 2 = 0 +$	$\boxed{1}$

$\uparrow$ $125 = 1111101 \longrightarrow$

Note

To write the binary number, read the remainders from the bottom up, and list them from left to right.

THE HEXADECIMAL NUMBER SYSTEM

The hexadecimal system provides a convenient method for writing the equivalent of binary numbers. The base of the system is 16. The 16 conventional digit symbols are 0 through 9, and A, B, C, D, E, and F. Because 16 is a power of 2, the conversion of a binary number to a hexadecimal number (or vice versa) is straightforward: the count in four bits of a binary number can be expressed as one hexadecimal digit. This is shown on the following example:

BINARY:	1 1 0 0	1 1 1 0	0 0 1 0	1 1 0 1
	(12_{10})	(14_{10})	(2_{10})	(13_{10})
HEXADECIMAL:	C	E	2	D

As shown by the previous example, hexadecimal notation is a convenient short-hand for expressing binary numbers. It is also very useful for floating-point arithmetic operations in a computer that is basically a binary machine. (Hexadecimal floating-point operations are used in the C-8500 system.) The table below gives the hexadecimal digit symbols with their binary and decimal equivalents:

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

To convert from hexadecimal to decimal and back, use the conversion tables given in appendix F.

Appendix F gives conversion tables for converting from hexadecimal to decimal and back. The two following tables can be used to perform manual hexadecimal arithmetic operations.

HEXADECIMAL ARITHMETIC

ADDITION TABLE

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

MULTIPLICATION TABLE

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

appendix b
instruction execution times

The following instruction execution times are applicable to all C-8561A-2 memory units.

INSTRUCTION AND OPERATION CODE (hexadecimal)		M-FIELD	EXECUTION TIME IN MICROSECONDS (1)			
			NONINDEXED LITERAL	INDEXED LITERAL	NONINDEXED NONLITERAL	INDEXED NONLITERAL
DATA TRANSFER INSTRUCTIONS						
Load accumulator (1C)		A	2.9	5.3	5.6	5.6
		B	5.7	8.3	8.5	8.5
		D	5.7	8.3	11.2	11.2
Load half-word (11)		L, R, E	3.0	5.4	5.7	5.7
Load byte (00)		1, 2, 3, 4	3.2	5.6	5.9	5.9
Load byte and clear (01)		1, 2, 3, 4	3.2	5.7	5.9	5.9
Load selective (37)		Unused	5.5	8.1	8.3	8.3
Load magnitude accumulator (1D)		A	2.9	5.8	5.7	5.7
		B	5.7	8.7	8.7	8.7
		D	5.7	8.7	11.3	11.3
Load index register (09)		1, 2, 3	2.9	5.3	5.6	5.6
Store accumulator or zero (54)		Zero			5.7	5.7
		A			5.7	5.7
		B			8.5	8.5
		D			11.4	11.4
Store half-word (5E)		L, R			8.5	8.5
Store byte (4E)		1, 2, 3, 4			8.5	8.5
Store selective (69)		Unused			11.2	11.2
Store magnitude (55)		A			5.7	5.7
		B			8.6	8.6
		D			11.4	11.4
Store magnitude half-word (5F)		L, R			8.3	8.3
Store index register (48)		1, 2, 3	8.5	8.5	8.5	8.5
Exchange storage with accumulator register (40)		A			8.5	8.5
		B			14.1	14.1
		D			19.7	19.7
Exchange storage with accumulator register and negate (41)		A			8.5	8.5
		B			14.1	14.1
		D			19.7	19.7
Exchange storage with index register (43)		1, 2, 3			8.5	8.5
Exchange storage with index register and negate (53)		1, 2, 3			8.5	8.5
Transfer register to register (56)						
R1	R2	Register transfer				
A, 1, 2, 3	A, 1, 2, 3	00	7.9	7.9	7.9	7.9
A, 1, 2, 3	B	00	11.2	11.2	11.2	11.2
B	A, 1, 2, 3	00	7.9	7.9	7.9	7.9
		Register exchange				
A, 1, 2, 3	A, 1, 2, 3	01	7.9	7.9	7.9	7.9
A, 1, 2, 3	B	01	11.2	11.2	11.2	11.2
B	A, 1, 2, 3	01	8.5	8.5	8.5	8.5
A, 1, 2, 3	A, 1, 2, 3	10	7.9	7.9	7.9	7.9
A, 1, 2, 3	B	10	11.2	11.2	11.2	11.2
B	A, 1, 2, 3	10	7.9	7.9	7.9	7.9
B	B	10	11.2	11.2	11.2	11.2
D	D	10	8.5	8.5	8.5	8.5
		Register exchange and negate				
A, 1, 2, 3	A, 1, 2, 3	11	7.9	7.9	7.9	7.9
A, 1, 2, 3	B	11	11.2	11.2	11.2	11.2
B	A, 1, 2, 3	11	8.5	8.5	8.5	8.5
B	B	11	14.1	14.1	14.1	14.1
D	D	11	8.5	8.5	8.5	8.5

INSTRUCTION AND OPERATION CODE (hexadecimal)	M-FIELD	EXECUTION TIME IN MICROSECONDS (1)			
		NONINDEXED LITERAL	INDEXED LITERAL	NONINDEXED NONLITERAL	INDEXED NONLITERAL
SHIFT INSTRUCTIONS					
Logical rotate left (2E)					
Even number of bits	A	7.8	7.8	7.8	7.8
	B	8.5	8.5	8.5	8.5
	D	13.2	13.2	13.2	13.2
Odd number of bits	A	10.0	10.0	10.0	10.0
	B	10.8	10.8	10.8	10.8
	D	18.0	18.0	18.0	18.0
Logical right shift (2F)					
Even number of bits	A	7.8	7.8	7.8	7.8
	B	8.5	8.5	8.5	8.5
	D	13.2	13.2	13.2	13.2
Even number of bits	A	10.0	10.0	10.0	10.0
	B	10.8	10.8	10.8	10.8
	D	18.0	18.0	18.0	18.0
Logical left shift (39)					
Even number of bits	A	10.0	10.0	10.0	10.0
	B	10.8	10.8	10.8	10.8
	D	18.4	18.4	18.4	18.4
Odd number of bits	A	12.4	12.4	12.4	12.4
	B	13.2	13.2	13.2	13.2
	D	23.2	23.2	23.2	23.2
Arithmetic right shift (0E)					
Even number of bits	A	7.8	7.8	7.8	7.8
	B	8.5	8.5	8.5	8.5
	D	13.2	13.2	13.2	13.2
	A	10.0	10.0	10.0	10.0
	B	10.8	10.8	10.8	10.8
	D	18.0	18.0	18.0	18.0
Arithmetic left shift (19)					
Even number of bits	A	10.0	10.0	10.0	10.0
	B	10.8	10.8	10.8	10.8
	D	18.4	18.4	18.4	18.4
Odd number of bits	A	12.4	12.4	12.4	12.4
	B	13.2	13.2	13.2	13.2
	D	23.2	23.2	23.2	23.2
LOGICAL INSTRUCTIONS					
And accumulator (24)	A	2.9	5.3	5.6	5.6
	B	8.5	11.0	11.8	11.8
	D	11.8	13.7	13.9	13.9
And half-word (30)	L, R, E	3.0	5.4	5.7	5.7
And byte (20)	1, 2, 3, 4	3.2	5.7	5.9	5.9
And to storage (64)	A			8.5	8.5
	B			11.8	11.8
	D			16.8	16.8
Inclusive OR accumulator (25)	A	2.9	5.3	5.6	5.6
	B	8.5	11.0	11.2	11.2
	D	11.2	13.7	13.9	13.9
Inclusive OR half-word (31)	L, R, E	3.0	5.4	5.7	5.7
Inclusive OR byte (21)	1, 2, 3, 4	3.2	5.7	5.9	5.9
Inclusive OR to storage (65)	A	2.9	5.3	5.6	5.6
Exclusive OR accumulator (27)	B	8.5	11.0	11.2	11.2
	D	11.2	13.5	13.7	13.7
Exclusive OR half-word (33)	L, R, E	3.0	5.4	5.7	5.7
Exclusive OR byte (23)	1, 2, 3, 4	3.2	5.7	5.9	5.9
Exclusive OR to storage (67)	A			8.5	8.5
	B			11.2	11.2
	D			16.8	16.8
Complement (3B)	A	2.4	5.3 (4)	2.9	5.3 (4)
	B, D	8.5	8.5	8.5	8.5
Effective address	00	8.5	8.5	8.5	8.5

INSTRUCTION AND OPERATION CODE (hexadecimal)	M-FIELD	EXECUTION TIME IN MICROSECONDS (1)			
		NONINDEXED LITERAL	INDEXED LITERAL	NONINDEXED NONLITERAL	INDEXED NONLITERAL
COMPARE INSTRUCTIONS					
Comparative AND (34)	A	2.9	5.3	5.5	5.5
	B	5.6	8.1	8.3	8.3
	D	8.3	10.8	11.0	11.0
Compare logical accumulator (17)	A	2.9	5.3	5.6	5.6
	B	5.6	8.1	8.3	8.3
	D	8.3	10.8	11.0	11.0
Compare logical half-word (10)	L, R	3.0	5.4	5.7	5.7
Compare logical byte (0B)	1, 2, 3, 4	3.2	5.7	5.9	5.9
Compare to zero (15)	A	2.9	5.3 (4)	2.9	5.3 (4)
	B	5.6	5.6	5.6	5.6
	D	8.3	8.3	8.3	8.3
Storage	00	5.6	5.6	5.6	5.6
Compare index (16)	1, 2, 3	2.9	5.3	5.6	5.6
Compare selective (35)	Any	5.6	8.1	8.3	8.3
Compare algebraic (14)	A	2.9	5.3	5.6	5.6
	B	5.6	8.1	8.3	8.3
	D	8.3	10.8	11.0	11.0
ARITHMETIC INSTRUCTIONS					
Add accumulator (D6)	A	2.9	5.3	5.6	5.6
	B	8.5	11.0	11.8	11.8
	D	11.8	13.7	13.9	13.9
Add half-word (12)	L, R, E	3.0	5.4	5.7	5.7
Add byte (02)	1, 2, 3, 4	3.2	5.7	5.9	5.9
Replace add to memory (42)	A			8.5	8.5
	B			11.2	11.2
	D			16.8	16.8
Replace add M to memory (52)	1, 2, 3, 4			8.6	8.6
Repeat add (57)	Unused	11.2 (2)	11.2 (2)	11.2 (2)	11.2 (2)
Subtract accumulator (07)	A	2.9	5.3	5.6	5.6
	B	8.5	11.0	11.2	11.2
	D	11.2	13.7	13.9	13.9
Subtract half-word (13)	L, R, E	3.0	5.4	5.9	5.9
Subtract byte (03)	1, 2, 3, 4	3.2	5.7	5.9	5.9
Modify index (08)	1, 2, 3	2.9	5.3	5.6	5.6
BRANCH INSTRUCTIONS					
Branch on accumulator (58), (7C), (7D)	00	5.7	5.7	5.7	5.7
	01	5.7	5.7	5.7	5.7
	10	2.9	2.9	2.9	2.9
	11	2.9	2.9	2.9	2.9
Branch (59)	00	8.5	8.5	8.5	8.5
	01	2.9	2.9	2.9	2.9
	10	2.9	2.9	2.9	2.9
	11	5.7	5.7	5.7	5.7
Branch on indicator (78)	0-7	2.9	2.9	2.9	2.9
Branch and set index (5B)	1, 2, 3	5.7	5.7	5.7	5.7
Branch on index zero (5A)	1, 2, 3	5.7	5.7	5.7	5.7
Branch and set return link protected (18)	Any	8.5	11.0	8.5	11.0
Branch and enable protection (79)	Any	2.9	2.9	2.9	2.9
MISCELLANEOUS INSTRUCTIONS					
Execute (4A)	Any	2.9	2.9	2.9	2.9
Direct Control (7E)	01, 10	4.79	7.06	4.79	7.06
Reset Machine Failure Monitor (3F)	Unused	2.9	2.9	2.9	2.9
Operation Code Trap	Unused	13.00 (3)	15.17 (3)	15.27 (3)	15.27 (3)

Notes:
1. Add 2.9 microseconds for each level of indirect addressing.
2. Add 2.9 n microseconds where n is the number of counts specified by X1; $1 \leq n \leq 31$.
3. Add execution of trapped routine for the appropriate instruction.
4. X Field is not ignored resulting in longer execution time if $X \neq 0$.
5. M and C combined.

TRAPPED INSTRUCTIONS

INSTRUCTION AND OPERATION CODE (HEXADECIMAL)

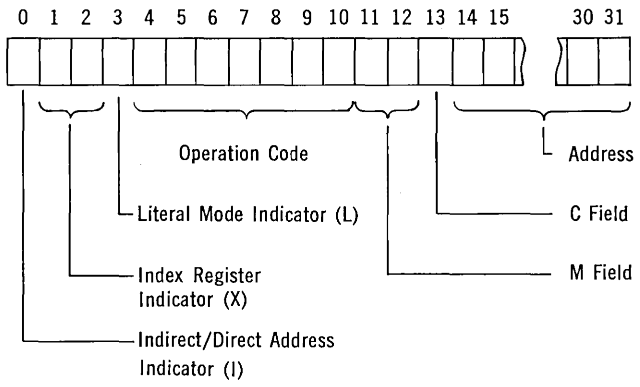
Load unaligned accumulator (1A)
Store unaligned (62)
Shift left and decrement (63)
Compare logical unaligned (1E)
Replace subtract M from memory (4B)
Negate (1F)
  Storage
Branch on D accumulator (7D)
Move field (71)
Compare field (70)
Translate field (61)
Scan field (60)
Convert to character (75)
Convert to binary (74)

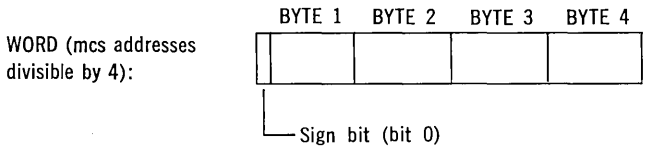
ALGORITHM UNIT INSTRUCTIONS

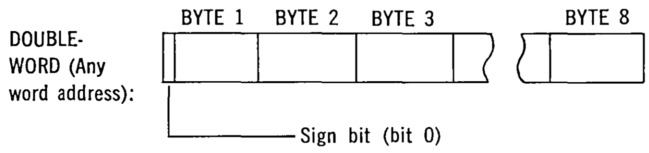
INSTRUCTION AND OPERATION CODE (hexadecimal)	M-FIELD	BASIC EXECUTION TIME IN MICROSECONDS	
		SINGLE PRECISION	DOUBLE PRECISION
Multiply (0D)	Unused	12.1 to 20.4	
Multiply Half-Word (0C)	Unused	6.8 to 11.4	
Divide (0A)	Unused	26.3 to 28.1	
Floating Add (4C)	A, D	1.6 to 11.8	1.6 to 14.8
Floating Add Unnormalized (4D)	A, D	1.6 to 11.8	1.6 to 14.8
Floating Compare (6A)	A, D	1.3 to 5.2	1.3 to 8.2
Floating Divide (6B)	A, D	23.6 to 29.9	53.1 to 59.4
Floating Multiply (6F)	A, D	10.9 to 27.5	24.4 to 73.3
Floating Negate (6E)	A, D	0.4	0.4
Storage	00	–	–
Floating Normalize (6D)	A, D	0.6 to 2.8	0.6 to 6.9
Floating Subtract (5C)	A, D	1.6 to 11.8	1.6 to 14.8
Floating Subtract Unnormalized (5D)	A, D	1.6 to 11.8	1.6 to 14.8

appendix C
data formats

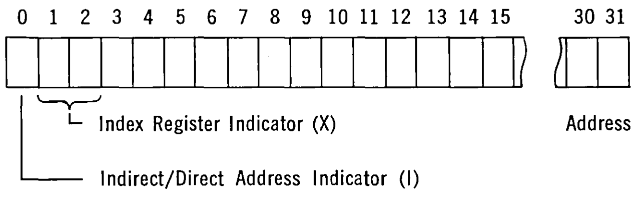
INSTRUCTION WORD


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 30 31
Operation Code
Address
Literal Mode Indicator (L)
C Field
Index Register Indicator (X)
M Field
Indirect/Direct Address Indicator (I)



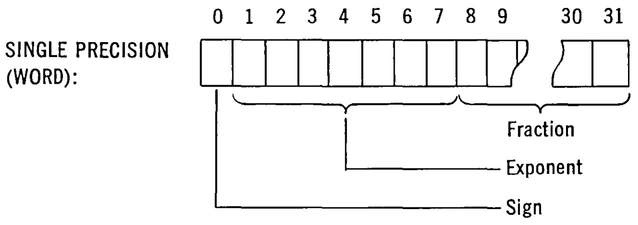


ADDRESS WORD USED IN INDIRECT ADDRESSING

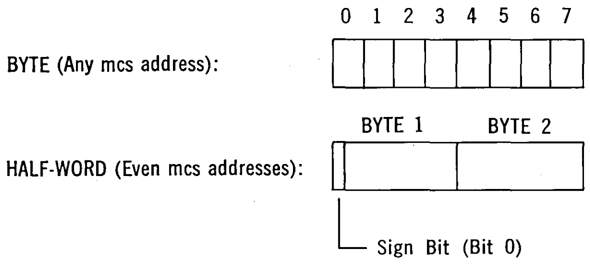

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 30 31
Index Register Indicator (X)
Address
Indirect/Direct Address Indicator (I)

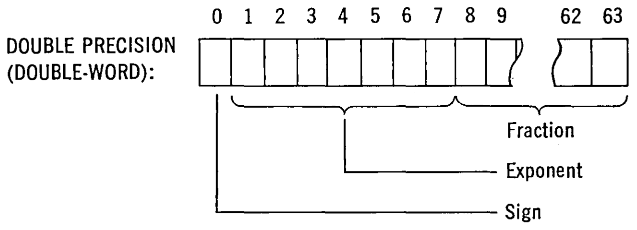
Note: An instruction can be used as an address word in indirect addressing.

FLOATING-POINT FORMATS



FIXED-POINT FORMATS





appendix d
assigned locations in protected memory

MCS BYTE LOCATION IN HEXADECIMAL	CONTENT OF MEMORY LOCATION
00-3F	Spares
40	Processor control word
44	Processor status word
48	*Absolute time clock
4C-7F	*Working channel status
80-9F	Spares
A0	New IAC for program interrupt
A4	condition code, overflow indicator and current IAC for program interrupt
A8	New IAC for MCS parity interrupt
AC	Condition code, overflow indicator, and current IAC for MCS parity interrupt
B0	New IAC for timer interrupts, IPL, and INIT
B4	Condition code, overflow indicator, and current IAC for timer interrupts, IPL, and INIT
B8	New IAC for branch return link to protected area entry
BC	Condition code, overflow indicator, and current IAC for branch return link to protected area entry
C0	Accumulator B
C4	Accumulator A (For use by the Algorithm Unit)
C8-CF	Spares
D0	Trapping mechanism new IAC
D4	Trapping mechanism effective address
D8	Trapping mechanism function word
DC	Trapping mechanism condition code, overflow indicator, and current IAC
FF8	Device status word upon completion of IPL
FFC	Channel status word upon completion of IPL
1000	Time division exchange address used for IPL

*These locations can be changed by altering hardware straps.

appendix e
powers of 2 and 16

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

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 10$_{16}$

10^n	n	10^{-n}
1	0	1.0000 0000 0000 0000
A	1	0.1999 9999 9999 999A
64	2	0.28F5 C28F 5C28 F5C3 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 8652 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}

appendix f
hexadecimal-decimal conversion tables

HEXADECIMAL-DECIMAL CONVERSION TABLE

The following table provides for direct conversion of decimal and hexadecimal numbers within the following ranges.

Hexadecimal	Decimal
000 to FFF	0000 to 4095

For numbers outside these ranges, the following values should be added to the table figures:

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
1000	4096	6000	24576	B000	45056
2000	8192	7000	28672	C000	49152
3000	12288	8000	32768	D000	53248
4000	16384	9000	36864	E000	57344
5000	20484	A000	40960	F000	61440

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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
100	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
110	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
120	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
130	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
140	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
150	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
160	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
170	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
180	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
190	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A0	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B0	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C0	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D0	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E0	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F0	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
200	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
210	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
220	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
230	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
240	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
250	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
260	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
270	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
280	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
290	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A0	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B0	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C0	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D0	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E0	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F0	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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
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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	1471	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
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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3030	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	3688	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

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	.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	.08934 37500	.57 00 00 00	.33984 37500	.97 00 00 00	.58984 37500	.D7 00 00 00	.83984 37500
.18 00 00 00	.09375 00000	.58 00 00 00	.34375 00000	.98 00 00 00	.59375 00000	.D8 00 00 00	.84375 00000
.19 00 00 00	.09765 62500	.59 00 00 00	.34765 62500	.99 00 00 00	.59765 62500	.D9 00 00 00	.84765 62500
.1A 00 00 00	.10156 25000	.5A 00 00 00	.35156 25000	.9A 00 00 00	.60156 25000	.DA 00 00 00	.85156 25000
.1B 00 00 00	.10546 87500	.5B 00 00 00	.35546 87500	.9B 00 00 00	.60546 87500	.DB 00 00 00	.85546 87500
.1C 00 00 00	.10937 50000	.5C 00 00 00	.35937 50000	.9C 00 00 00	.60937 50000	.DC 00 00 00	.85937 50000
.1D 00 00 00	.11328 12500	.5D 00 00 00	.36328 12500	.9D 00 00 00	.61328 12500	.DD 00 00 00	.86328 12500
.1E 00 00 00	.11718 75000	.5E 00 00 00	.36718 75000	.9E 00 00 00	.61718 75000	.DE 00 00 00	.86718 75000
.1F 00 00 00	.12109 37500	.5F 00 00 00	.37109 37500	.9F 00 00 00	.62109 37500	.DF 00 00 00	.87109 37500
.20 00 00 00	.12500 00000	.60 00 00 00	.37500 00000	.A0 00 00 00	.62500 00000	.E0 00 00 00	.87500 00000
.21 00 00 00	.12890 62500	.61 00 00 00	.37890 62500	.A1 00 00 00	.62890 62500	.E1 00 00 00	.87890 62500
.22 00 00 00	.13281 25000	.62 00 00 00	.38281 25000	.A2 00 00 00	.63281 25000	.E2 00 00 00	.88281 25000
.23 00 00 00	.13671 87500	.63 00 00 00	.38671 87500	.A3 00 00 00	.63671 87500	.E3 00 00 00	.88671 87500
.24 00 00 00	.14062 50000	.64 00 00 00	.39062 50000	.A4 00 00 00	.64062 50000	.E4 00 00 00	.89062 50000
.25 00 00 00	.14453 12500	.65 00 00 00	.39453 12500	.A5 00 00 00	.64453 12500	.E5 00 00 00	.89453 12500
.26 00 00 00	.14843 75000	.66 00 00 00	.39843 75000	.A6 00 00 00	.64843 75000	.E6 00 00 00	.89843 75000
.27 00 00 00	.15234 37500	.67 00 00 00	.40234 37500	.A7 00 00 00	.65234 37500	.E7 00 00 00	.90234 37500
.28 00 00 00	.15625 00000	.68 00 00 00	.40625 00000	.A8 00 00 00	.65625 00000	.E8 00 00 00	.90625 00000
.29 00 00 00	.16015 62500	.69 00 00 00	.41015 62500	.A9 00 00 00	.66015 62500	.E9 00 00 00	.91015 62500
.2A 00 00 00	.16406 25000	.6A 00 00 00	.41406 25000	.AA 00 00 00	.66406 25000	.EA 00 00 00	.91406 25000
.2B 00 00 00	.16796 87500	.6B 00 00 00	.41796 87500	.AB 00 00 00	.66796 87500	.EB 00 00 00	.91796 87500
.2C 00 00 00	.17187 50000	.6C 00 00 00	.42187 50000	.AC 00 00 00	.67187 50000	.EC 00 00 00	.92187 50000
.2D 00 00 00	.17578 12500	.6D 00 00 00	.42578 12500	.AD 00 00 00	.67578 12500	.ED 00 00 00	.92578 12500
.2E 00 00 00	.17968 75000	.6E 00 00 00	.42968 75000	.AE 00 00 00	.67968 75000	.EE 00 00 00	.92968 75000
.2F 00 00 00	.18359 37500	.6F 00 00 00	.43359 37500	.AF 00 00 00	.68359 37500	.EF 00 00 00	.93359 37500
.30 00 00 00	.18750 00000	.70 00 00 00	.43750 00000	.B0 00 00 00	.68750 00000	.F0 00 00 00	.93750 00000
.31 00 00 00	.19140 62500	.71 00 00 00	.44140 62500	.B1 00 00 00	.69140 62500	.F1 00 00 00	.94140 62500
.32 00 00 00	.19531 25000	.72 00 00 00	.44531 25000	.B2 00 00 00	.69531 25000	.F2 00 00 00	.94531 25000
.33 00 00 00	.19921 87500	.73 00 00 00	.44921 87500	.B3 00 00 00	.69921 87500	.F3 00 00 00	.94921 87500
.34 00 00 00	.20312 50000	.74 00 00 00	.45312 50000	.B4 00 00 00	.70312 50000	.F4 00 00 00	.95312 50000
.35 00 00 00	.20703 12500	.75 00 00 00	.45703 12500	.B5 00 00 00	.70703 12500	.F5 00 00 00	.95703 12500
.36 00 00 00	.21093 75000	.76 00 00 00	.46093 75000	.B6 00 00 00	.71093 75000	.F6 00 00 00	.96093 75000
.37 00 00 00	.21484 37500	.77 00 00 00	.46484 37500	.B7 00 00 00	.71484 37500	.F7 00 00 00	.96484 37500
.38 00 00 00	.21875 00000	.78 00 00 00	.46875 00000	.B8 00 00 00	.71875 00000	.F8 00 00 00	.96875 00000
.39 00 00 00	.22265 62500	.79 00 00 00	.47265 62500	.B9 00 00 00	.72265 62500	.F9 00 00 00	.97265 62500
.3A 00 00 00	.22656 25000	.7A 00 00 00	.47656 25000	.BA 00 00 00	.72656 25000	.FA 00 00 00	.97656 25000
.3B 00 00 00	.23046 87500	.7B 00 00 00	.48046 87500	.BB 00 00 00	.73046 87500	.FB 00 00 00	.98046 87500
.3C 00 00 00	.23437 50000	.7C 00 00 00	.48437 50000	.BC 00 00 00	.73437 50000	.FC 00 00 00	.98437 50000
.3D 00 00 00	.23828 12500	.7D 00 00 00	.48828 12500	.BD 00 00 00	.73828 12500	.FD 00 00 00	.98828 12500
.3E 00 00 00	.24218 75000	.7E 00 00 00	.49218 75000	.BE 00 00 00	.74218 75000	.FE 00 00 00	.99218 75000
.3F 00 00 00	.24609 37500	.7F 00 00 00	.49609 37500	.BF 00 00 00	.74609 37500	.FF 00 00 00	.99609 37500

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE (Continued)

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

HEXADECIMAL-DECIMAL FRACTION CONVERSION TABLE (Continued)

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	.000 0 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 (Continued)

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