SPC-16
fundamental operations manual

88A00424A-A

88A00424A-A

SPC-16
fundamental operations manual

GENERAL AUTOMATION, INC.
1055 South East Street
Anaheim, California 92805
(714) 778-4800

REVISION

Symbol	Description	Manager Publications	Engineering Approval	Date
A	Original Issue			April 74

This manual provides information that should be treated as appendix material, to be used in conjunction with any of the SPC-16 Operating System manuals.

CONTENTS

Contents (continued)

SECTION 1
SPC-16 CONSOLE

The controls and indicators, which are located on the front panel of the computer, allow manual data entry, examination of register and memory contents and initiation of automatic operation.

A key-operated console lock switch on the front panel provides a safety feature to prevent inadvertent operation of console controls when the system is on-line and operating.

The console is illustrated in Figure 1-1. The console panel contains indicators (lighted/unlighted) and switches (up = 0 or not set, down = 1 or set). The ENTER (8), STEP (10), SYSTEMS RESET (12), and CONSOLE INTERRUPT (14) switches return to the up (off) position after being manually depressed. All other switches may be manually set either up or down. Switches and indicators are discussed individually in Section 1.1.

Step-by-step instructions for performing the following operations are given in Section 1.2.

a. Display/modify the contents of registers and indicators.
b. Display/modify the contents of memory locations.
c. Start automatic program execution.
d. Manual step-execute through a program.

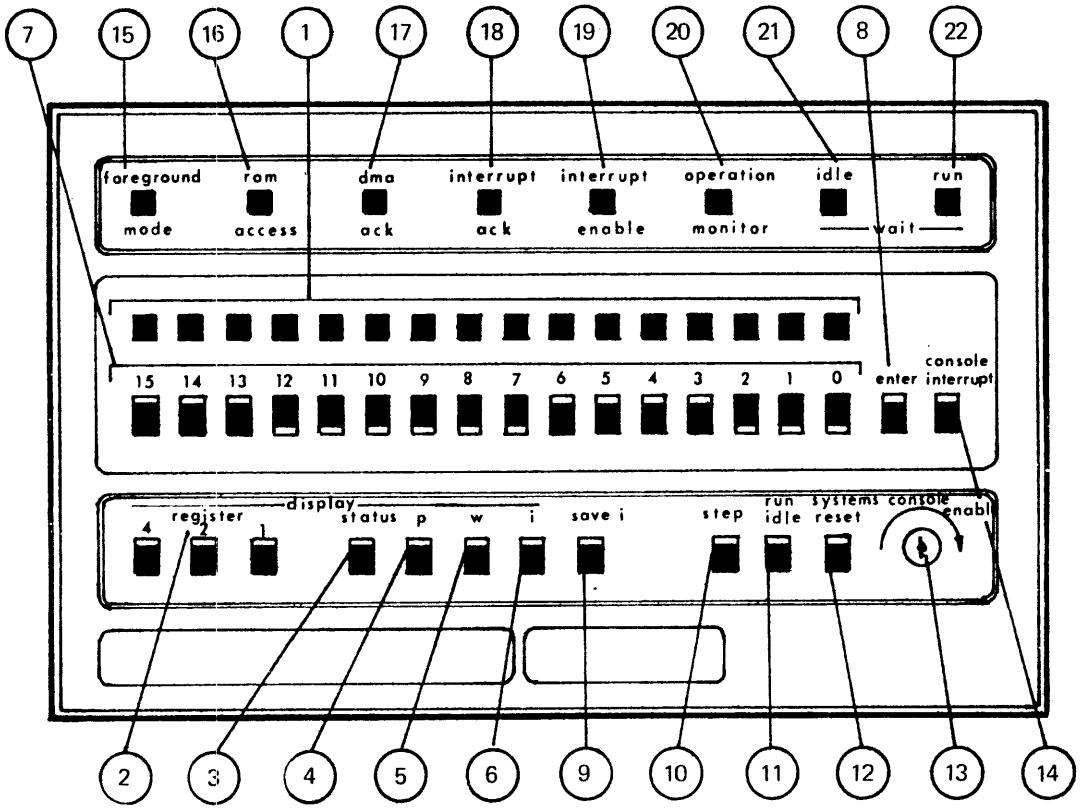
1.1 SWITCHES AND INDICATORS

(1) REGISTER DISPLAY indicators (0 through 15) indicate contents of the System Console Display register (K register). The K register is loaded under program control by the DSPL instruction in the RUN mode; in the IDLE mode, it is loaded with the contents of the selected register (see Register Select switches below).

REGISTER SELECT SWITCHES (2 through 6)

These switches are used to select a register for display or data entry. They have a left-to-right priority; for example, if switch (6) is on (down), the other Register Select switches are disabled and the I register is selected.

(2) GENERAL PURPOSE REGISTER select switches are used to select one of the eight operational registers to receive data subsequently keyed into the Data Entry switches (7). The other Register Select switches ((3), (4), (5), and (6)) must be in the up (off) position in order to select a register with switches (2). After keying in a register code in the IDLE mode, the register contents are displayed on the Register Display indicator lights (1).



1. 16 REGISTER DISPLAY indicators
2. 3 GENERAL PURPOSE REGISTER select switches
3. STATUS Register select switch
4. P Register select switch
5. W Register select switch
6. I Register select switch
7. 16 DATA Entry switches
8. ENTER switch
9. SAVE I switch
10. STEP switch
11. RUN/IDLE switch
12. SYSTEMS RESET switch
13. CONSOLE KEY LOCK
14. CONSOLE INTERRUPT switch
15. FOREGROUND MODE indicator
16. READ ONLY MEMORY ACCESS indicator
17. DIRECT MEMORY ACCESS/AUTOMATIC DATA CHANNEL ACKNOWLEDGE indicator
18. INTERRUPT ACKNOWLEDGE indicator
19. INTERRUPT ENABLE indicator
20. OPERATIONS MONITOR ALARM indicator
21. IDLE indicator
22. RUN indicator

Figure 1-1. The SPC-16 Computer Console

The register codes are the same as those used in the SPC-16 instruction set.

Register	Select Switches
A	000
X	001
Y	010
Z	011
B	100
C	101
D	110
E	111

3 STATUS REGISTER select switch places the contents of the indicators and the shift counter (S register) into the Console Display register. The Register Display indicators (1) will show the status as follows:

Shift Counter → $DSPL_{0-3}$

Link → $DSPL_4$

Overflow → $DSPL_5$

Plus → $DSPL_6$

Zero → $DSPL_7$

Foreground → $DSPL_8$ (same as (15))

0 → $DSPL_{9-15}$

The indicators may be changed by the Data Entry switches (7).

(4) 'P' REGISTER select switch displays the contents of the Program Counter register on the Register Display indicators (1), bits 0-14. The interrupt status (same as (19)) is displayed in bit 15. When manually stopping an executing program, the current instruction is completed before execution stops. Since the program counter is incremented by the current instruction, the P register will contain the address of the next instruction to be executed.

The program counter may be changed via the Data Entry switches (7) and the P register select switch (4). The interrupt status enable (bit 15) is not affected when the program counter is changed.

(5) 'W' REGISTER select switch displays the contents of the Working register. The W Register contains the effective operand address for the last instruction executed. The W Register may be changed by the Data Entry switches (7).

(6) 'I' REGISTER select switch displays the contents of the Instruction register. When manually stopping an executing program, execution of the current instruction is completed and then stops, so that the Instruction register will contain the instruction just completed. The Instruction register may be changed by the Data Entry switches (7).

(7) DATA ENTRY switches (0 through 15) are used to enter data or instructions into the selected register in the IDLE mode. These switches may be tested as sense switches or used as data input switches during program execution. Data switches in down position = 1, in up position = 0.

(8) ENTER switch. In the IDLE mode, this switch replaces the contents of the selected register with the contents of the Data Entry switches.

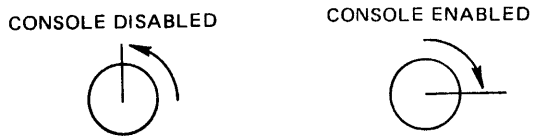
(9) SAVE I switch preserves the contents of the Instruction register for repeated execution (e.g., referencing memory locations using the same load/store instruction) in the IDLE mode. SAVE I is effective only in the IDLE mode.

(10) STEP switch. When this switch is depressed, the computer fetches an instruction from the location specified by the P register, executes the instruction and increments the P register. In the IDLE mode, only a single instruction cycle occurs. (If the SAVE I switch is down, the current contents of the I register are executed and preserved; the P register is still incremented, however.) In the RUN mode automatic program execution is initiated by pressing STEP. A WAIT condition may be terminated by pressing the STEP switch.

(11) RUN/IDLE switch is used to place the computer in either the RUN mode (switch up) or the IDLE mode (switch down).

(12) SYSTEMS RESET switch is used to initialize the computer and peripheral controllers. This switch sets foreground mode, if the foreground option is installed (shown by FOREGROUND MODE indicator (15)), and disables all interrupts.

(13) CONSOLE ENABLE Key Lock. This is a two position lock, as follows:



Turning the key counterclockwise disables the console (i.e., changing any of the console switches has no effect). Turning the key clockwise will enable the console switches. The console may be enabled/disabled in either the RUN or IDLE mode.

(14) CONSOLE INTERRUPT switch. Pressing this switch causes a console-interrupt request when the console is enabled.

(15) FOREGROUND MODE indicator indicates which set of eight General Purpose registers is being used. The indicator is on for foreground, off for background. If the foreground/background option is not installed, this indicator is always off.

(16) READ ONLY MEMORY ACCESS indicator indicates when Read Only Memory is being accessed for instructions or data.

(17) DIRECT MEMORY ACCESS/AUTOMATIC DATA CHANNEL ACKNOWLEDGE indicator indicates that a DMA cycle 'steal' is taking place.

(18) INTERRUPT ACKNOWLEDGE indicator indicates when an interrupt is being serviced. Manually stepping through an interrupt will result in the address of the branch vector being displayed in the indicators. Exit must be made by pressing STEP (to execute the interrupt) or SYSTEMS RESET (to reinitialize the system).

(19) INTERRUPT ENABLE indicator indicates the system interrupt status: on = enabled, off = disabled.

(20) OPERATIONS MONITOR ALARM indicator indicates when the alarm has been activated.

(21) IDLE indicator indicates when the computer is in the Idle mode.

(22) RUN indicator indicates when the computer is in the RUN mode.

NOTE

When executing a program in the Run mode and a WAIT instruction is encountered, both the RUN and IDLE indicators come on.

1.2.0 FUNDAMENTAL OPERATIONS

1.2.1 DISPLAYING/MODIFYING REGISTERS

1. Turn key lock (13) clockwise to enable console switches.

2. Set the RUN/IDLE switch (11) to IDLE (Indicator (21) will be illuminated).

3. Select register from Register Select switches (2) through (6). The contents of the selected register will be displayed on the Register Display indicators (1).

4. To enter data into the selected register, key the data into the Data Entry switches (7) and press the ENTER switch (8).

1.2.2 DISPLAYING/MODIFYING MEMORY LOCATIONS

1. Set RUN/IDLE switch (11) to IDLE.

2. Load the value (ADDRESS-1) into the P register, where ADDRESS is the address of the first memory location to be displayed/modified. (To select location 0, enter $FFFF_{16}$ ($7FFF_{16}$ will have the same effect.))

3. Load the 'LDA 0,0' instruction (X'4000') or 'STA 0,0 instruction (X'5000') into the I register for displaying or modifying the memory location of ADDRESS, respectively.

4. Preserve the contents of the I register by pressing the SAVE I switch (9) down.

5. Select the A register for display.

6a. To display the contents of the memory location at ADDRESS, press STEP switch (10).

6b. To enter data into the memory location at ADDRESS, key the data into the Data Entry switches (7), press ENTER switch (8) and then press STEP.

7a. The contents of sequential memory locations are displayed each time STEP is pressed.

7b. To enter data into sequential memory locations, repeat step 6b.

1.2.3 EXECUTING A PROGRAM

Once a program has been loaded into memory, it can be executed via the console as follows:

1. Set the RUN/IDLE switch (11) to IDLE.
2. Load beginning address of program into the P register.
3. Press SYSTEMS RESET switch (12) .
4. Set the RUN/IDLE switch (11) to RUN.
5. Press STEP switch (10) . The program will execute.

To stop the execution of a program, set the RUN/IDLE switch to IDLE.

1.2.4 STEPPED PROGRAM EXECUTION VIA THE COMPUTER CONSOLE

While debugging programs it is frequently helpful to execute the program one step at a time; this allows the programmer to determine program path and intermediate values of data registers.

1. Set the RUN/IDLE switch (11) to IDLE.
2. Load beginning address of program into the P register.
3. Press SYSTEMS RESET switch (12) .
4. Press STEP switch (10) . The program will execute one instruction. Relevant registers may then be selected for display.
5. Repeat step 4 to execute the program one instruction at a time.

NOTE

When stepping through an instruction and an interrupt is acknowledged (INTERRUPT ACK indicator (18) is lighted), the interrupt vector location is displayed on the Register Display indicators and all console enter/display switches are disabled. Pressing STEP will execute the interrupt cycle and re-enable the console.

SECTION 2
BOOTSTRAPPING

Programs may always be loaded into the computer's memory manually by entering the individual instruction codes and data via the console. However, if a program that reads data from a peripheral and places it in memory in a sequential fashion is already in memory, it may be executed and used to load another program automatically.

Two data formats are used for object programs on paper tape. Most system software (the CAP-16 Assembler, the FORTRAN compiler, operating systems, etc.) as well as the object tapes of user programs that have been assembled or compiled are in PGS format. In addition to the actual program instruction codes and data, this format includes linking and relocation information that can be used by the loading program to build an executable program unit from separately assembled or compiled program modules. The smallest loading program that will accept programs in PGS format is the PGS loader; it consists of more than 250 instructions and it is, therefore, unpractical to load it manually. (PGS data format is described in Section 3.)

A few basic system programs, including the PGS loader program, are supplied in simple binary format on paper tape. A binary tape consists of a sequence of data bytes (each representing half an instruction or data word). These bytes may be input via an appropriate peripheral (teletype paper tape reader or high speed paper tape reader) and stored in sequential words (two data bytes per word) in memory. The program that performs this operation is known as a bootstrap loader.

The bootstrap loader that loads binary tape format from the teletype (Teletype Bootstrap) consists of only eight instructions; thus, it may be easily keyed in via the computer console. Another bootstrap loader, which loads binary tape format from the high speed paper tape reader (High Speed Paper Tape Reader Bootstrap) consists of 20 or less instructions; it may be keyed in manually (see listing in Section 2.2.1) or loaded from the teletype using the Teletype Bootstrap (the High-Speed Paper Tape Reader Bootstrap program is supplied on paper tape in binary format). Either bootstrap loader may be used to load the more complex PGS loader, which is also supplied in binary format.

READ-ONLY-MEMORY BOOTSTRAPS

Read-Only-Memory (ROM) bootstrap loaders are hard-wired and need not be manually entered.

A section of computer memory (usually 32 words) is installed as read-only-memory, and the instructions which make up the bootstrap loader are imprinted on this read-only-memory. To load a bootstrap-loadable program into memory, the ROM bootstrap need only be executed at its predetermined location.

Usually the ROM containing the bootstrap loader(s) occupies the highest 32 memory locations. Execution address(es) of the ROM bootstrap loader(s) will be a given number of locations backwards from the highest core address of the system. For example, if a ROM bootstrap loader has a starting

address of top-of-core minus X'1F', then for a 4K system the ROM bootstrap is executed at X'FE0'; for an 8K system, the ROM bootstrap is executed at X'1FE0'; for a 12K system, the ROM bootstrap is executed at X'2FE0'; etc. This variable execution address will be indicated in the instructions below as: execution address X'xFE0'. Special documentation is provided with any system where the ROM is not in the highest 32 memory locations.

The coding of ROM bootstraps may vary slightly from the listings given in this section. Refer to the supplied ROM bootstrap Program Manual for listings of ROM bootstraps.

2.1.0 TELETYPE BOOTSTRAP

The Teletype Bootstrap may be used to load binary programs or data input on paper tape from the teletype tape reader.

2.1.1 LOADING THE TELETYPE BOOTSTRAP

If the system does not include a high speed paper tape reader, then the Teletype Bootstrap will normally be loaded into the eight highest memory locations (e.g., for a 4K machine, the beginning address would be X'0FF8').

If the Teletype Bootstrap is used for loading the High Speed Paper Tape Bootstrap, the Teletype Bootstrap may be loaded into the eight lowest memory locations (i.e., beginning at X'0000'); the High Speed Paper Tape Bootstrap may then be loaded into the highest locations using the Teletype Bootstrap.

1. Load the Teletype Bootstrap into sequential memory locations as described in Section 1.2.2 for modifying memory locations:

```
        HEX.
        CODE
0000                0001  * TELETYPE BOOTSTRAP
0000                0002          PSECT
0000    0020        0003          ZERO    X          CLEAR MEMORY COUNTER
0001    0524        0004  LOOP    DSPL    X          DISPLAY COUNTER INDEX
0002    10FF        0005          TEST    0,X'5F'    TEST READY
0003    73FE        0006          JMP     $-1
0004    18BF        0007          DTIR    A,X'5F'    INPUT BYTE
0005    9900        0008          STBY    A,0,X,1    STORE IN MEMORY
0006    072E        0009          INCR    X          INCREMENT COUNTER
0007    73F9        0010          JMP     LOOP
                    0011          END
```

2. Verify that the bootstrap was keyed in correctly by displaying the eight memory locations, following the procedure described in Section 1.2.2 for displaying memory locations.

ROM TELETYPE BOOTSTRAPS

The standard ROM Teletype Bootstrap is coded as shown in step above. For the SPC-16/40 Series computers, an alternate ROM Teletype Bootstrap is available that displays an accumulated checksum (total of all bytes input) instead of the memory byte counter.

2.1.2 EXECUTING THE TELETYPE BOOTSTRAP

With the bootstrap in memory, the following steps are used to load programs or data using the bootstrap:

1. Set the RUN/IDLE switch to IDLE.
2. Turn the teletype on (non-local).
3. Place the program paper tape into the teletype paper tape reader. The first binary frame must be over the read station.
4. Set the D register to the starting address at which the program or data is to be loaded.
5. Set the P register to the starting address of the bootstrap.
6. Press SYSTEMS RESET.
7. Set the RUN/IDLE switch to RUN.
8. Press STEP.
9. Turn on the reader. The program will load. The REGISTER DISPLAY indicators will display the memory byte counter.
10. When the program/data has been loaded, turn the reader off. The loading process does not stop after the last data byte has been read but continues until the physical end of tape is reached or the reader is turned off.

Refer to the appropriate system loader description for subsequent execution procedures for the loaded program (e.g., the Stand Alone Utilities Manual for the PGS Loader or BUS).

2.2.0 HIGH SPEED PAPER TAPE BOOTSTRAP

The High Speed Paper Tape (HSPT) Bootstrap can be used to load binary programs or data input on paper tape from the high speed paper tape reader.

2.2.1 LOADING THE HSPT BOOTSTRAP

A. VIA THE COMPUTER CONSOLE

1. Load the HSPT Bootstrap into sequential memory locations as described in Section 1.2.2 for modifying memory locations:

HSPT Bootstrap for SPC-16/30 Series

The SPC-16/30 Series HSPT Bootstrap may be loaded into the 24 highest memory locations (high core address minus X'17'). For example, in a 12K system the bootstrap would occupy locations X'2FE8' through X'2FFF'.

```
        HEX.
        CODE
                 0001   * 304 HSPT BOOTSTRAP
0000             0002          PSECT
0000    0020     0003   PTB    ZERO    X          CLEAR MEMORY COUNTER
0001    6005     0004   PTB1   JSR     IBY        INPUT BYTE
0002    21FE     0005          SKZ     PTB1       IGNORE LEADER
0003    0900     0006   PTB2   STBY    A,0,X,1    STORE BYTE IN MEMORY
0004    072E     0007          INCR    X          INCREMENT COUNTER
0005    6001     0008          JSR     IBY        INPUT BYTE
0006    75FC     0009          JMP     PTB2
0007    0115     0010   IBY    LDV     A,X'1C00'  SET CONTROL
0008    1C00
0009    1640     0011          DTOR    A,0
000A    1641     0012          DTOR    A,1        STEP READER
000B    0115     0013   IBY1   LDV     A,X'1F01'  SET SENSE
000C    1F01
000D    1640     0014          DTOR    A,0
000E    1681     0015          DTIR    A,1        INPUT STATUS
000F    0117     0016          ANDV    A,X'40'
0010    0040
0011    20F9     0017          SKZ     IBY1       IF NOT READY
0012    0115     0018          LDV     A,X'1A00'  SET READ
0013    1A00
0014    1640     0019          DTOR    A,0
0015    1681     0020          DTIR    A,1        INPUT BYTE
0016    0015     0021          RTR     A,A        SET CONDITION CODES
0017    05E3     0022          RTRN    E
                 0023          END
```

HSPT Bootstrap for SPC-16/40 Series

The SPC-16/40 Series HSPT Bootstrap may be loaded into the 20 highest memory locations (high core address minus X'13'). For example, in a 12K system the bootstrap would occupy locations X'2FEC' through X'2FFF'.

```
       HEX.
       CODE
0000                 0001  * 40+ HSPT BOOTSTRAP
0000                 0002          PSECT
0000   0C20          0003  PTB     ZERO    X          CLEAR MEMORY COUNTER
0001   0640          0004          ZERO    Y          CLEAR CHECKSUM
0002   6007          0005  PTB1    JSR     IBY        INPUT A BYTE
0003   2FFE          0006          SKZ     PTB1       IGNORE LEADER
0004   9900          0007  PTB2    STBY    A,0,X,1    STORE BYTE IN MEMORY
0005   072E          0008          INCR    X          INCREMENT COUNTER
0006   0659          0009          ADD     Y,A        ACCUMULATE CHECKSUM
0007   6002          0010          JSR     IBY        INPUT A BYTE
0008   0594          0011          DSPL    Y          DISPLAY CHECKSUM
0009   7FFA          0012          JMP     PTB2
000A   1100          0013  IBY     TEST    1,0        TEST READY
000B   7FF1          0014          JMP     $-1
000C   1500          0015          CTRL    5,0        STEP READER
000D   1000          0016          TEST    0,0        TEST UP COMPLETE
000E   7FFD          0017          JMP     $-1
000F   1500          0018          TEST    5,0        TEST COMMAND REJECT
0010   7FFD          0019          JMP     $-1
0011   1000          0020          DTIR    A,0        INPUT BYTE
0012   0615          0021          RTR     A,A        SET CONDITION CODES
0013   0BE3          0022          RTRN    L
                     0023          END
```

For SPC-16/40 Series computers, an alternate ROM HSPT Bootstrap is available that displays a memory byte counter instead of the accumulated checksum.

2. Verify that the bootstrap was keyed in correctly by displaying the HSPT Bootstrap locations, following the procedure described in Section 1.2.2 for displaying memory locations.

B. VIA THE TELETYPE BOOTSTRAP

The HSPT Bootstrap is supplied in binary format on paper tape and may be loaded from the teletype tape reader as follows:

1. Key in the Teletype Bootstrap as described in Section 2.1.1 (any block of locations other than the HSPT Bootstrap locations may be used).

2. Execute the Teletype Bootstrap as described in Section 2.1.2, setting the D register to the beginning load address of the HSPT Bootstrap. The HSPT Bootstrap will be loaded into core.

2.2.2 EXECUTING THE HSPT BOOTSTRAP

With the bootstrap in memory, the following steps are used to load programs or data:

1. Set the RUN/IDLE switch to IDLE.
2. Turn the high speed paper tape reader on.
3. Load the binary formatted paper tape into the high speed paper tape reader. Leader will be ignored.
4. Set the D register to the starting address at which the program or data is to be loaded.
5. Set the P register to the starting address of the bootstrap. For a ROM bootstrap this is standardly the high core address minus X'1F' (X'xFE0').
6. Press SYSTEMS RESET.
7. Set the RUN/IDLE switch to RUN.
8. Press STEP. The program will load.
9. When the program has loaded, set the RUN/IDLE switch to IDLE. Press SYSTEMS RESET.

For SPC-16/40 Series, either the memory byte counter or the accumulated checksum will be displayed at the console.

2.3.0 CARD READER BOOTSTRAP

The Card Reader Bootstrap can be used to load specially formatted binary programs or data input on cards from the card reader.

2.3.1 BOOTSTRAP CARD FORMAT

Cards processed by the Card Reader Bootstrap must be formatted as shown in Figure 2-1. Columns 1 through 72 are punched and contain one binary byte per column in the first eight rows (positions 12, 11, 0, 1, 2, 3, 4, and 5). The last card in the deck must have a row 9 punch in column 72.

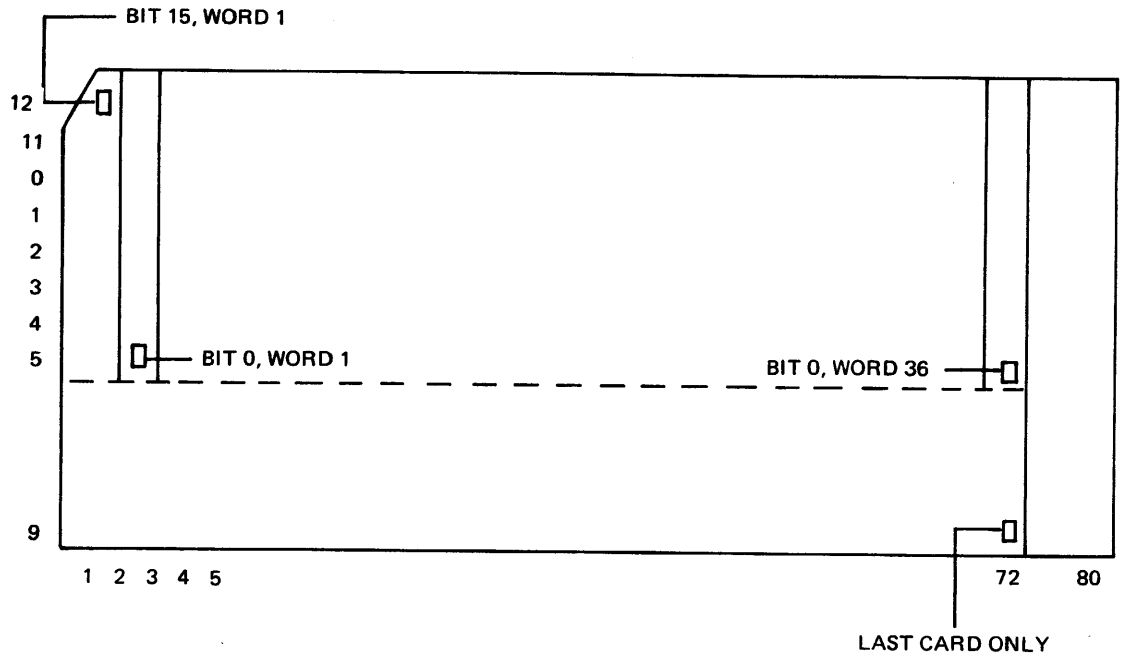


Figure 2-1. Punched Card Bootstrap Format

The PGS Card Loader is supplied in Card Reader Bootstrap format.

2.3.2 LOADING THE CARD READER BOOTSTRAP

The Card Reader Bootstrap is normally not keyed into memory. The Card Reader Bootstrap is available on paper tape in binary format for systems in which the bootstrap is not implemented in Read-Only-Memory.

The Card Reader Bootstrap may be loaded from the teletype tape reader as follows:

1. Key in the Teletype Bootstrap as described in Section 2.1.1 (any block of locations other than the Card Reader Bootstrap locations may be used).

2. Execute the Teletype Bootstrap as described in Section 2.1.2, setting the D register to the beginning load address of the Card Reader Bootstrap. The Card Reader Bootstrap will be loaded into core.

Card Reader Bootstrap for SPC-16/30 Series

The SPC-16/30 Series Card Reader Bootstrap may be loaded into the 32 highest memory locations (high core address minus X'1F'). For example, in a 12K system the bootstrap would occupy locations X'2FE0' through X'2FFF'.

```
                        0001  * 30+ CARD READER BOOSTRAP
0000                    0002          PSECT
0000  0EB5              0003  START   RTR     C,D          SAVE DSECT START
0001  0E05              0004          RTR     D,E          START OF BOOTSTRAP LOAD
0002  D00E              0005  L1      STR     D,X'27'      SET CAR
0003  0027
0004  0115              0006          LDV     A,X'8E00'    INITIATE READ
0005  8E00
0006  1340              0007          DTOR    A,0
0007  0115              0008  L3      LDV     A,X'8F01'    SENSE AND RESET
0008  8F01
0009  1340              0009          DTOR    A,0
000A  1341              0010          DT1R    A,1
000B  0509              0011          SRC     A,12
000C  2FFA              0012          SKP     L3           IF NOT OP COMPLETE
000D  0501              0013          SRC     A,2
000E  2E02              0014          SKP     $+3          IF NO ERROR
000F  0000              0015          WAIT
0010  73F1              0016          JMP     L1
0011  D020              0017          ZERO    X            CLEAR MEMORY COUNTER
0012  9801              0018  L5      LDBY    A,1,0,1      LOAD A BYTE
0013  9900              0019          STBY    A,0,X,1      STORE BYTE IN MEMORY
0014  07CE              0020          INCR    D            NEXT CARD COLUMN
0015  4000              0021          LDA     D,1          LOAD CURRENT COLUMN
0016  0722              0022          DECR    X
0017  0126              0023          SUBVC   X,-72
0018  FF88
0019  2E18              0024          SKN     L5           IF NOT END OF CARD
001A  0106              0025          SUBV    D,36         SET NEXT CAR ADDRESS
001B  1024
001C  1504              0026          SRC     A,5
001D  2FE4              0027          SKP     L1           IF NOT LAST CARD
001E  0E05              0028          RTR     D,C          RESTORE DSECT START
001F  05F2              0029          EXIT    E            EXIT TO LOADED PROGRAM
                        0030          END
```

Card Reader Bootstrap for SPC-16/40 Series

The SPC-16/40 Series Card Reader Bootstrap may be loaded into the 24 highest memory locations (high core address minus X'17'). For example, in a 12K system the bootstrap would occupy locations X'2FE8' through X'2FFF'.

```
                        0001  * 40+ CARD READER BOOTSTRAP
0000                    0002          PSECT
0000  0FD5              0003  CRBA    RTR     D,L          START OF BOOTSTRAP LOAD
0001  11CB              0004  CRB1    TEST    1,X'B'       TEST READY
0002  73FE              0005          JMP     $-1
0003  0D0F              0006          STR     D,X'27'      SET CAR
0004  0027
0005  100B              0007          CTRL    D,X'B'       INITIATE READ
0006  10CB              0008          TEST    D,X'B'       TEST OP COMPLETE
0007  73FE              0009          JMP     $-1
0008  12CB              0010          TEST    2,X'B'       TEST ERROR
0009  0003              0011          WAIT                 WAIT IF ERROR
000A  0660              0012          ZERO    Z            CLEAR MEMORY COUNTER
000B  30A1              0013  CRB3    LDBY    C,1,0,1      LOAD A BYTE
000C  3EA0              0014          STBY    C,0,Z,1      STORE BYTE IN MEMORY
000D  C0A1              0015          LDR     C,1,0,1      LOAD CURRENT COLUMN
000E  07C1              0016          INCR    D            NEXT CARD COLUMN
000F  0762              0017          DECR    Z            DECREMENT COLUMN COUNT
0010  0166              0018          SUBVC   Z,-72
0011  FFB8
0012  25F8              0019          SKN     CRB3         IF NOT END OF CARD
0013  0106              0020          SUBV    D,36         SET NEXT CAR ADDRESS
0014  0024
0015  03A4              0021          SRC     C,5
0016  2FFA              0022          SKP     CRB1         IF NOT LAST CARD
0017  05E2              0023          EXIT    E            EXIT TO LOADED PROGRAM
                        0024          END
```

2.3.3 EXECUTING THE CARD READER BOOTSTRAP

When executed, the Card Reader Bootstrap loads a program into memory starting at the address contained in the E register and then passes control to the loaded program at the starting (E register) address. The following register contents are preserved and passed to the loaded program:

SPC-16/30 Series	Registers Y, Z, B, D and E are preserved.
SPC-16/40 Series	Registers A, X, Y, B and E are preserved.

The Card Reader Bootstrap is executed as follows:

1. Set the RUN/IDLE switch to IDLE.
2. Place a program deck in Card Reader Bootstrap format into the card reader.
3. Ready the card reader.
4. Set the E register to the starting address of the program to be loaded.
5. Set the P register to the starting address of the bootstrap. For a ROM bootstrap, this is standardly the high core address minus X'1F' (X'xFE0').
6. Press SYSTEMS RESET.
7. Set the RUN/IDLE switch to RUN.
8. Press STEP. The program will load and execute.

If the computer comes to a WAIT state before all cards have been read, a load error has occurred and the above procedure must be repeated.

The program normally loaded by the Card Reader Bootstrap is the PGS Card Loader. Refer to the Stand Alone Utilities Manual for a complete description of PGS Card Loader loading and execution sequences.

2.4.0 DISK/DRUM BOOTSTRAP

The Disk/Drum ('Bulk') Bootstrap is used to load and execute a binary program stored on bulk in sector 0.

Each disk bootstrap reads a specified number of words, varying from 255 words to a full sector (320 or 400 words). Each drum (including head/track disk) bootstrap performs the same function, reading in a specified number of words beginning at sector 0. The program is loaded starting at a

number of words below the Bulk Bootstrap. The loaded program is executed by the bootstrap at the beginning program location (starting load address plus one when the first word contains the input sector number).

The program loaded and executed by the Bulk Bootstrap is normally the system's Sector Loader, which in turn loads and executes the bulk-resident monitor.

2.4.1 LOADING THE BULK BOOTSTRAP

The Bulk Bootstrap will normally not require manual entry. The Bulk Bootstrap is written to the top of core automatically via the Write Monitor (WMON) command during system generation; if the Bulk Bootstrap is not implemented in Read-Only-Memory, a copy of the system's Bulk Bootstrap can be obtained on paper tape in binary format as part of the WMON operation (refer to the applicable operating system manual). If the Bulk Bootstrap is subsequently overlayed (non-ROM), it can be reloaded from the paper tape copy via the Teletype Bootstrap (see Section 2.1).

If manually loaded, the Bulk Bootstrap should be placed at the top of core. To locate the Bulk Bootstrap in the same memory block written to by the system, the beginning load address will be the highest non-ROM location minus X'1F'. For example, if the system has no ROM in upper core, the Bulk Bootstrap will be loaded beginning at X'xFE0'; if the system has a 32-word ROM in upper core, the Bulk Bootstrap will be loaded beginning at X'xFC1'.

Individual Disk/Drum Bootstraps are listed on the following pages.

Model 1341/43 Disk Bootstrap

The Model 1341/43 Disk Bootstrap reads 321 words into memory beginning 900 locations below the start of the bootstrap

```
                  0001  * 1341/43 DISK BOOTSTRAP
0000              0002        PSECT
0000  0402        0003  B1    INH
0001  0195        0004        LDV     B,X'2409'
0002  2409
0003  1C40        0005        DTOR    B,0         DESELECT UNIT 0.
0004  1841        0006        DTOR    A,1
0005  0195        0007        LDV     B,X'2480'   RESTORE UNIT 0
0006  2480
0007  1C40        0008        DTOR    B,0
0008  1841        0009        DTOR    A,1
0009  600D        0010        JSR     TRDY        SENSE OP COMPLETE
000A  0115        0011        LDV     A,321       WORD COUNT
000B  0141
000C  0135        0012        LDV     X,X'22'
000D  0022
000E  0100        0013        STR     A,0,X       INITIALIZE SCR
000F  01F6        0014        SUBV    E,910       STARTING LOAD ADDRESS-1
0010  038E
0011  01E1        0015        STR     E,1,X       INITIALIZE CAR
0012  0135        0016        LDV     X,X'2600'   READ OP.
0013  2600
0014  1940        0017        DTOR    X,0         INITIATE READ
0015  01F9        0018        ADDV    E,2         EXECUTE AT START+1
0016  0002
0017  0195        0019  TRDY  LDV     B,X'2701'
0018  2701
0019  1C40        0020        DTOR    B,0
001A  1881        0021        DTIR    A,1         SENSE AND RESET
001B  021D        0022        SRA     A,14
001C  2BFA        0023        SKS     TRDY        IF NOT OP COMPLETE
001D  05E2        0024        EXIT    E           RETURN/EXIT TO SECTOR LOADER
                  0025        END
```

Model 1342 Drum Bootstrap

The Model 1342 Drum Bootstrap reads 320 words into memory directly below the bootstrap.

```
                  0001  * 1342 DRUM BOOTSTRAP
0000              0002         PSECT
0000  6000        0003         JSR     $+1          SET E=START OF BOOT+1
0001  0155        0004         LDV     Y,X'22'      LOAD SCR ADDRESS
0002  0022
0003  0115        0005         LDV     A,320        WORD COUNT
0004  0140
0005  07E2        0006         DECR    E
0006  0EF0        0007         SUB     E,A
0007  02E1        0008         STR     E,1,Y        INITIALIZE CAR
0008  F241        0009         DECM    1,Y
0009  0200        0010         STR     A,0,Y        INITIALIZE SCR
000A  0115        0011         LDV     A,X'3C00'
000B  3C00
000C  1040        0012         DTOR    A,0
000D  0E00        0013         ZERO    A
000E  1041        0014         DTOR    A,1          SELECT TRACK ZERO
000F  0115        0015         LDV     A,X'3E00'    LOAD READ COMMAND
0010  3E00
0011  1040        0016         DTOR    A,0          INITIATE READ
0012  0115        0017  DBT2   LDV     A,X'3F01'
0013  3F01
0014  1040        0018         DTOR    A,0          SENSE AND RESET
0015  1081        0019         DTIR    A,1
0016  0210        0020         SRA     A,13
0017  28FA        0021         SKS     DBT2         IF BUSY
0018  05E2        0022         EXIT    E            EXIT TO SECTOR LOADER
                  0023         END
```

Model 1344/PERTEC/DIABLO Disk Bootstrap

The Model 1344/PERTEC/DIABLO Disk Bootstrap reads 321 words into memory beginning 900 locations below the start of the bootstrap.

```
                  0001  * 1344 /PERTEC/DIABLO DISK BOOTSTRAP
0000              0002         PSECT
0000  0115        0003  B44S   LDV     A,X'2600'   READ COMMAND FOR REM/1344
0001  2600
0002  7002        0004         JMP     B44S1
0003  0115        0005         LDV     A,X'2610'   READ COMMAND FOR FIXED
0004  2610
0005  0620        0006  B44S1  ZERO    X
0006  0195        0007         LDV     B,X'2400'
0007  2400
0008  1040        0008         DTOR    B,0         SEEK CYLINDER ZERO
0009  1941        0009         DTOR    X,1
000A  6008        0010         JSR     B44SS       WAIT FOR OP COMPLETE
000B  0135        0011         LDV     X,X'02'
000C  0022
000D  01F0        0012         SUBV    E,911       STARTING ADDRESS-1
000E  038F
000F  01F1        0013         STR     E,1,X       INITIALIZE CAR
0010  0195        0014         LDV     B,321       WORD COUNT
0011  0141
0012  0180        0015         STR     B,0,X       INITIALIZE SCR
0013  1040        0016         DTOR    A,0         INITIATE READ
0014  01F9        0017         ADDV    E,2         EXECUTE AT START+1
0015  0002
0016  0195        0018  B44SS  LDV     B,X'2701'
0017  2701
0018  1040        0019         DTOR    B,0         SENSE AND RESET
0019  1041        0020         DTIR    C,1
001A  0240        0021         SRA     C,14
001B  2BFA        0022         SKS     B44SS       IF BUSY
001C  00E2        0023         EXIT    E           RETURN/EXIT TO SECTOR LOADER
                  0024         END
```

Model 1345 Disk Bootstrap

The Model 1345 Disk Bootstrap reads 321 words into memory beginning 900 locations below the start of the bootstrap.

```
                  0001  * 1345 DISK BOOTSTRAP
0000              0002         PSECT
0000  0402        0003  B45S   INH
0001  6014        0004         JSR     B45SE       TEST READY
0002  0135        0005         LDV     X,207
0003  00CF
0004  0115        0006         LDV     A,X'2404'
0005  2404
0006  1840        0007         DTOR    A,0         HOME DISK
0007  1941        0008         DTOR    X,1
0008  600D        0009         JSR     B45SE       WAIT FOR OP COMPLETE
0009  0115        0010         LDV     A,321       WORD COUNT
000A  0141
000B  0135        0011         LDV     X,X'22'
000C  0022
000D  0100        0012         STR     A,0,X       INITIALIZE SCR
000E  01F6        0013         SUBV    E,909       STARTING LOAD ADDRESS-1
000F  038D
0010  01E1        0014         STR     E,1,X       INITIALIZE CAR
0011  0195        0015         LDV     B,X'2600'
0012  2600
0013  1C40        0016         DTOR    B,0         INITIATE READ
0014  01F9        0017         ADDV    E,2         EXECUTE AT START+1
0015  0002
0016  0195        0018  B45SE  LDV     B,X'2701'
0017  2701
0018  1C40        0019         DTOR    B,0         SENSE AND RESET
0019  1881        0020         DTIR    A,1
001A  021D        0021         SRA     A,14
001B  25FA        0022         SKS     B45SE       IF BUSY
001C  05E2        0023         EXIT    E
                  0024         END
```

Model 3341/43/46/47 Disk Bootstrap

The Model 3341/43/46/47 Disk Bootstrap reads 255 words into memory beginning 900 locations below the start of the bootstrap.

```
                  * 3341/43/46/47 DISK BOOTSTRAP
0000                        PSECT
0000  0115        ROMR      LDV     A,X'E000'   REMOVABLE ENTRY
0001  E000
0002  7002                  JMP     ROME
0003  0115        ROMF      LDV     A,X'E004'   FIXED ENTRY
0004  E004
0005  1641        ROME DTOR         A,X'E'      SELECT DRIVE
0006  0175                  LDV     Z,X'8000'
0007  8000
0008  8C0A                  JSR     XEC         RESTORE DRIVE
0009  01F6                  SUBV    E,909       STARTING LOAD ADDRESS-1
000A  038D
000B  00FF                  STR     E,X'23'     INITIALIZE CAR
000C  0023
000D  056E                  SRC     Z,7         WORD COUNT=255
000E  007F                  STR     Z,X'22'     INITIALIZE SCR
000F  0022
0010  01F9                  ADDV    E,2         EXECUTE AT START+1
0011  0002
0012  0369                  SRC     Z,10        READ DRIVE
0013  164E        XEC       DTOR    Z,X'E'      OUTPUT FUNCTION
0014  1A8E                  DTIR    Y,X'E'      INPUT STATUS
0015  0256                  SRA     Y,7
0016  2BFD                  SKS     XEC+1       IF NOT READY
0017  05E2                  EXIT    E           RETURN/EXIT TO SECTOR LOADER
                            END
```

Model 3342 Head/Track Disk Bootstrap

The Model 3342 Head/Track Disk Bootstrap reads 400 words into memory directly below the bootstrap.

```
                  0001  * 3342 HEAD/TRACK DISK BOOTSTRAP
0000              0002         PSECT
0000  600E        0003  B42S   JSR     RDY          TEST READY
0001  01F6        0004         SUBV    E,403        STARTING LOAD ADDRESS-1
0002  0193
0003  0135        0005         LDV     A,X'20'
0004  0020
0005  0195        0006         LDV     B,X'4000'+400  400=WORD COUNT
0006  4190
0007  0183        0007         STR     B,0,X        INITIALIZE SCR
0008  01E1        0008         STR     E,1,X        INITIALIZE CAR
0009  0600        0009         ZERO    A
000A  1852        0010         DTOR    A,X'12'      SELECT TRACK ZERO
000B  0115        0011         LDV     A,X'1000'
000C  1000
000D  1852        0012         DTOR    A,X'12'      SELECT AND READ FROM SECTOR 0
000E  07EE        0013         INCR    E            EXECUTE AT LOAD ADDRESS
000F  1892        0014  RDY    DTIR    A,X'12'      SENSE STATUS
0010  1212        0015         CTRL    2,X'12'      RESET STATUS
0011  0201        0016         SRLC    A,2
0012  2EFC        0017         SKS     RDY          IF BUSY/NOT READY
0013  0202        0018         SRLC    A,3
0014  2BEB        0019         SKS     B42S         IF ANY ERROR
0015  05E2        0020         EXIT    E            EXIT TO SECTOR LOADER
                  0021         END
```

Model 3349 Disk Bootstrap

The Model 3349 Disk Bootstrap reads 260 words into memory beginning 298 locations below the start of the bootstrap.

```
                  0001  * 3349 DISK BOOTSTRAP
0000              0002         PSECT
0000  0115        0003  FDR    LDV     A,X'6000'   DRIVE 0 ENTRY
0001  3000
0002  600F        0004         JSR     TRD         TEST READY
0003  184A        0005         DTOR    A,X'A'      SELECT UNIT
0004  0175        0006         LDV     Z,X'1040'   RESTORE (MAX SEEK REVERSE)
0005  1040
0006  600A        0007         JSR     XEC         INITIATE/WAIT FOR RESTORE
0007  01F6        0008         SUBV    E,305       STARTING LOAD ADDRESS-1
0008  0131
0009  001F        0009         STR     E,X'29'     INITIALIZE CAR
000A  0029
000B  0275        0010         SRA     Z,4         WORD COUNT = 260
000C  007F        0011         STR     Z,X'28'     INITIALIZE SCR
000D  0028
000E  01F9        0012         ADDV    E,2         EXECUTE AT START+1
000F  0002
0010  0369        0013         SRC     Z,10        READ COMMAND
0011  1E4A        0014  XEC    DTOR    Z,X'A'      INITIATE FUNCTION
0012  1A8A        0015  TRD    DTIR    Y,X'A'      SENSE STATUS
0013  0357        0016         SRCL    Y,8
0014  2BFD        0017         SKS     TRD         IF BUSY
0015  27FC        0018         SKM     TRD         IF NOT READY
0016  120A        0019         CTRL    Z,X'A'      RESET STATUS
0017  05E2        0020         EXIT    E           RETURN/EXIT TO SECTOR LOADER
                  0021         END
```

2.4.2 EXECUTING THE BULK BOOTSTRAP

For multiple disk systems with selectable drives, the Bulk Bootstrap has standard entry points for selecting a primary drive when the bootstrap is being used to load from a primary drive. (For fixed/removable drives, drive 0 (D0) is the primary removable drive and drive 4 (D4) is the primary fixed drive; for other systems drive 0 (D0) is the primary drive.) When loading from secondary drives (SPC-16/40 Series only), the operator must specify the drive by setting the A register as shown in step 4 below.

The Bulk Bootstrap is executed as follows:

1. Set the RUN/IDLE switch to IDLE.

2. For removable disks, place the disk from which loading is to occur on the selected (or only) disk drive.

3. Turn on the disk unit.

4. If applicable, set the A register to the selected disk drive, as follows:

	A Register
Model 3341/43/46/47	X'E00v'
Model 3349	X'800v'

where: v = 0 [primary drive (Model 3346 removable)]
1 2nd drive (Model 3346 removable)
2 3rd drive (Model 3346 removable)
3 4th drive (Model 3346 removable)
4 [primary fixed drive (Model 3346 only)]
5 2nd fixed drive (Model 3346 only)
6 3rd fixed drive (Model 3346 only)
7 4th fixed drive (Model 3346 only)

5. Set the P register to the execution address of the Bulk Bootstrap. For any bootstrap starting at X'xFE0', the execution addresses are as follows:

	P Register
Primary or only [removable] drive All models	X'xFE0'
Primary or only fixed drive Models 3346 and Pertec	X'xFE3'
Secondary drives (must set A reg) Models 3341/43/46/47	X'xFE4'
Secondary drives (must set A reg) Model 3349	X'xFE2'

6. Press SYSTEMS RESET.

7. Set the RUN/IDLE switch to RUN.

8. Press STEP. The sector 0 program will load and execute. (If the bulk drive is not yet ready, the bootstrap loops until the unit becomes ready.)

SECTION 3
PROGRAM AND DATA FORMATS

Object programs for the SPC-16 can exist in either of two formats, binary or PGS. Either format is valid for either cards or paper tape. Data and source language programs are in ASCII format.

3.1 BINARY FORMAT

Binary format is simply a stream of bytes just as they appear in memory. There is no control or check-sum information. See Figures 3-1 and 3-3 for the paper tape and card representation of the Binary format.

3.2.0 PGS FORMAT

PGS format is the universal SPC-16 object program representation. Object programs may be either absolute or relocatable. All program loaders read PGS format (except HSPTL) and all object output is punched in PGS format (BUS-16 can punch Binary). The major division of PGS data is the 54 word (108 byte) record. Within a record are data items. A data item consists of a control byte and its associated data bytes. See Figures 3-2 and 3-4.

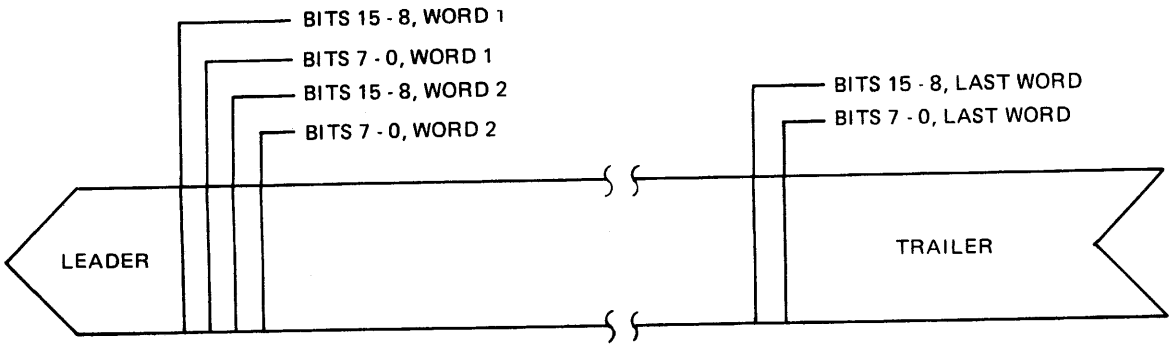
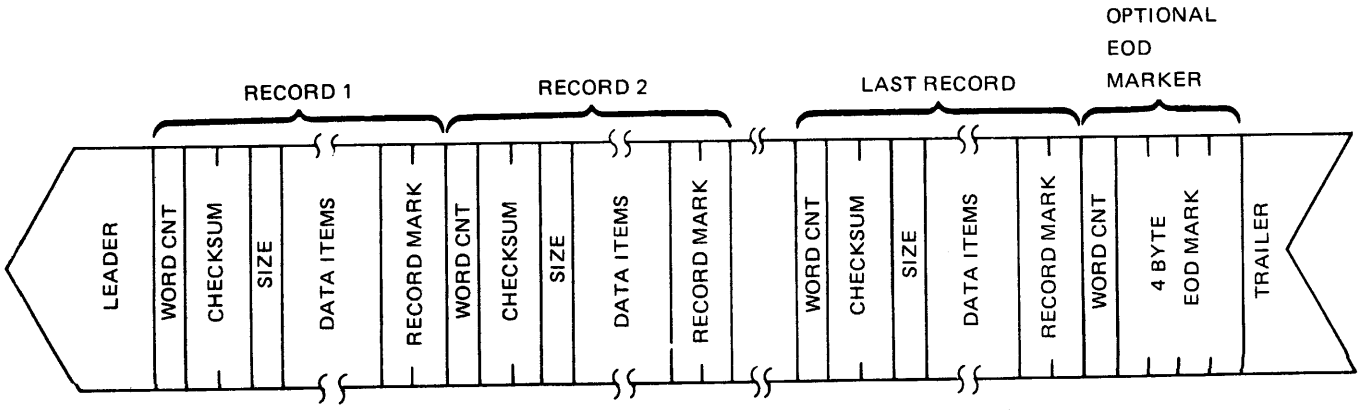


Figure 3-1. Binary Paper Tape Format



NOTE: OPTIONAL EOD MARKER IS A FOUR-BYTE ENTRY CONTAINING 44281040_{16}

Figure 3-2. PGS Paper Tape Format

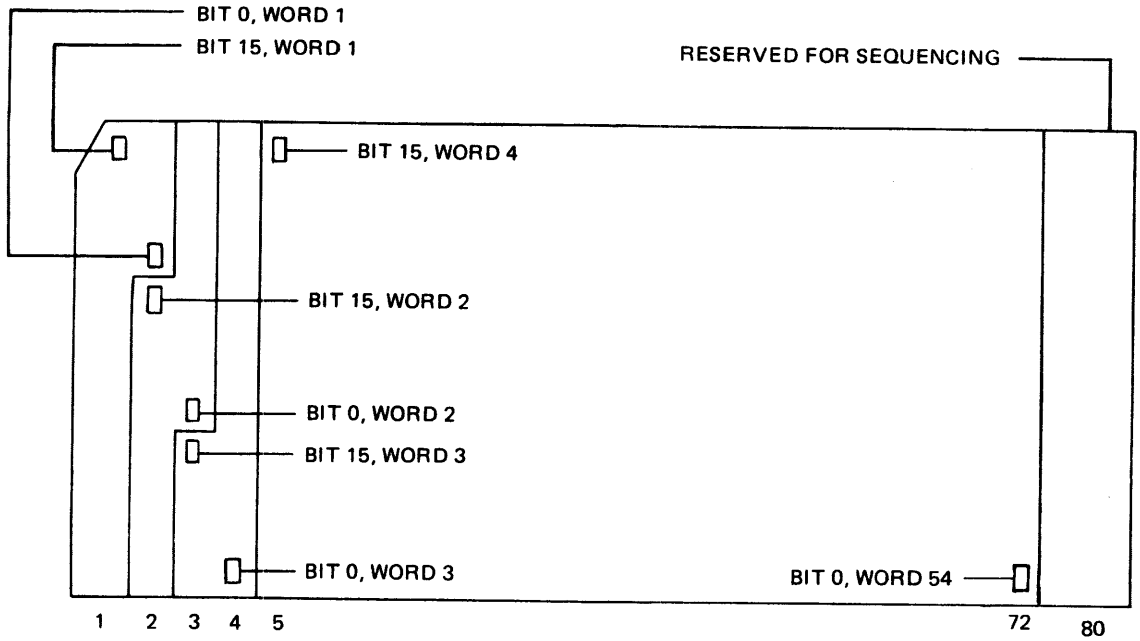
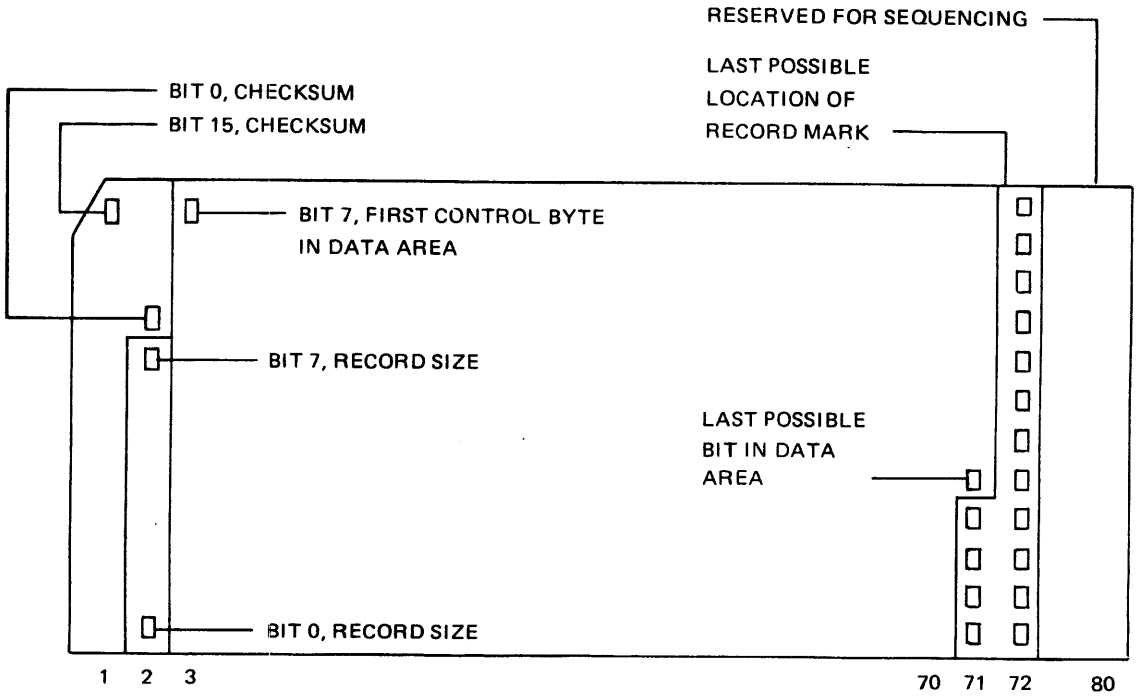


Figure 3-3. Binary Card Format



NOTE: ONE RECORD PER CARD REGARDLESS OF LENGTH

Figure 3-4. PGS Card Format

3.2.1 PGS RECORD

The PGS record consists of up to 54 words (108 bytes) of information and appears as follows:

2-BYTE CHECKSUM	1-BYTE DATA SIZE	DATA AREA (UP TO 103 BYTES)	2-BYTE RECORD MARK

where:

Checksum	is the one-word result of a summation of the remainder of the record up to the record mark.
Data size	is a one-byte count of the number of bytes in the data area. This may be a maximum of 103 (67_{16}).
Data area	contains the data items and may contain up to 103 (67_{16}) bytes.
Record mark	is a 2-byte record terminator containing $FFFF_{16}$, or on newer tapes $E7E7_{16}$.

On physical paper tape only, each record is preceded by a one-byte word count which indicates the number of words in the record. The maximum is $54(36_{16})$.

3.2.2 PGS DATA ITEMS

The Data Area in a PGS record is made up of individual data items. Each data item consists of a control byte followed by one or more data bytes.

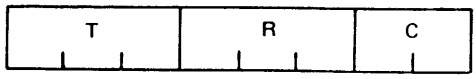
A control byte defines the type of data item and appears as:

T	RC

where:

T	is a 3-bit value indicating the type of item.
RC	is a 5-bit field used to further define the item.

When T = 0, 1, 2, 5 or 7, the RC field is divided into a 3-bit R field and a 2-bit C field as follows:

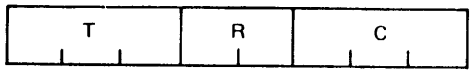


where:

R =	0	Absolute
	1	COMMON
	2	Program (PSECT)
	3	GLOBAL
	4	Data (DSECT)
	5	Unused
	6	External

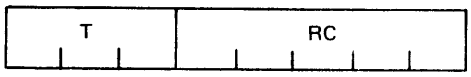
C = number of data bytes-1 in the item.

When T = 3 or 4, the RC field is divided into a 2-bit R field and a 3-bit C field as follows:



where R and C have meaning as described above.

When T = 6, the RC field is a 5-bit field:



where:

RC = 0	A secondary control byte follows
RC ≠ 0	This field contains a double word count of absolute data items that follow the control byte.

3.2.3 CONTROL BYTE TYPES

T = 0 Data - from one to four data bytes follow depending on field C. They are loaded according to the relocation specified in field R.

T = 1 Define Origin - two data bytes follow which replace the contents of the loading location counter.

T = 2 Define PSECT = two data bytes follow which replace the contents of the PSECT relocation base.

T = 3 External Reference - data bytes which follow are an external reference by name whose length in bytes is equal to C+1. R defines the type of reference as follows:

R = 0 Primary reference (REF)
R = 1 Secondary reference (SREF)
R = 2 Labeled COMMON

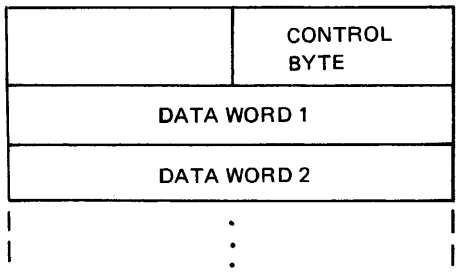
T = 4 External Definition - the data bytes which follow contain the name of an entry point in the program being loaded followed by the 2-byte address of the entry point. C-1 equals number of bytes in the name. C+1 equals the total number of data bytes in the item. The R field signifies the relocation base of the address as follows:

R = 0 Absolute
R = 1 Program (PSECT) Relocatable
R = 2 Data (DSECT) Relocatable

T = 5 Define DSECT - Two data bytes follow which replace the contents of the DSECT relocation base. If R = 6, this value must be zero.

T = 6 a. RC = 0 Special - see following section.

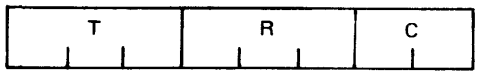
b. RC ≠ 0
n absolute data words follow, where n = (RC)*2. All data words in the count are contained in the same record. This type of control byte always appears as the righthand byte of the preceding word:



T = 7 End item - signifies the end of object text. If C = 3, the 2-byte value following is the execution address of the program.

3.2.4 SECONDARY CONTROL BYTE TYPES

When the item type is T = 6, RC = 0, the primary control byte is followed by a secondary control byte which specifies one of six functions. This secondary control byte also has a 3-bit type field, and except as indicated, R and C fields are defined as for primary control bytes.



Secondary Control Byte types:

1. $T_S = 0$

   R is not significant
   - C = 0 Byte addressing mode
   - C = 1 Word addressing mode
   - C = 2 End of core resident (main) module
   - C = 3 The current sector number for writing overlays is stored at the current location counter value.

2. $T_S = 1$

   Address Chain - The value following this control byte is the starting address in a chain. The core location at the starting address contains the next address in the chain. This chain is followed until some location in the chain has contents zero, signalling the end of the chain. Into bits 14-0 of each location in the chain is placed the current value of the loading location counter. (This value may be set by an ORG item before the address chain.)

   R is the appropriate relocation specification as for primary control bytes.

   C is the length of the following item (=1).

3. $T_S = 2$

   Deflection Chain - The value following this control byte is the starting address in a chain. Bits 7-0 of the starting location contain a displacement back to the next item in the chain. The chain continues until bits 7-0 of some location in the chain have contents zero, signalling the end of the chain. Each link of the chain is resolved by placing the difference between the current location counter value and chain address into bits 7-0 of the chain location.

   R is the appropriate relocation specification as for primary control bytes.

   C is the length of the following item (=1).

4. $T_S = 3$ — Define COMMON, dynamic storage, or GLOBAL size. The value following gives the absolute size of the COMMON, dynamic, or GLOBAL block specified by R (R=1 COMMON, R=2 dynamic storage, R=3 GLOBAL). The program with the largest COMMON requirement must be loaded first. When R=1 or 3, this item type is always preceded by an item type T=3 (External Reference).

   C is length of the following item.

5. $T_S = 4$ — Memory Module Select - The value following is a data word for extended memory mapping. This data word is output via an XIO instruction specifying the Extended Memory device select code (X'39').

   R is not significant.

   C is length of the following item (=1).

6. $T_S = 5$ — Not used.

7. $T_S = 6$ — Write Overlay to Bulk. The two addresses following are the beginning and ending addresses of the overlay module. The overlay module is written to bulk beginning at the current (first available) sector in the scratch file. The current sector number is then updated to the next available sector.

   R is the appropriate relocation specification as for primary control bytes.

   C is length of the following item (=3).

8. $T_S = 7$ — Not used.

3.3 ASCII PAPER TAPE FORMAT

ASCII records contained on paper tape (or input from the teletype keyboard) consist of a stream of ASCII characters with special form and record control characters. An ASCII record has the following format:

Line Feed	Start of record
ASCII Characters	Characters within record
Carriage Return	End of record indicator

Any amount of leader may exist prior to the line feed and after the Carriage Return characters. Within a record the following characters have special meaning:

Line Feed	Ignored
Rubout	Delete the entire record and start a new record
←(Back arrow)	Delete the previous character. Multiple ← will delete as many characters as the number of ←

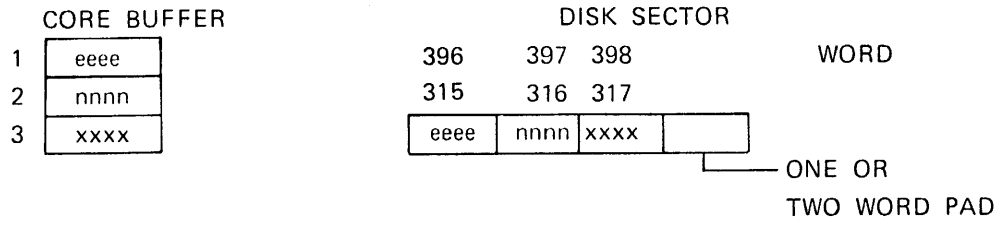
SECTION 4
DIRECTORY FORMATS FOR BULK UNITS AND MAGNETIC TAPE

4.1.0 BULK UNIT

In a directoried bulk file, the last sector(s) in the file contains the directory. A bulk unit directory consists of a 3-word directory header, directory entries and a 3-word directory trailer or end directory record.

4.1.1 DIRECTORY HEADER

The first entry in a bulk unit directory is the header record. It is three words in length and appears as follows:



where:

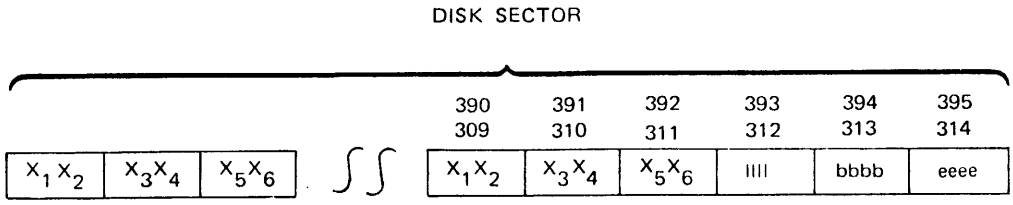
eeee — is the binary value of the first sector in the directory (last sector of directoried file)

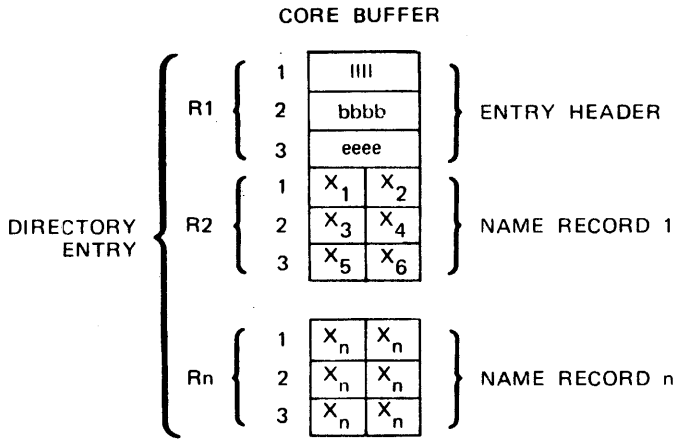
nnnn — is the next available sector in the directoried file.

xxxx — is reserved for future expansion

4.1.2 DIRECTORY ENTRY

The directory header is followed by one or more directory entries. Directory entries are variable length but are at least six words. The format of a directory entry is as follows when read as six byte records.





where:

llll	is the length of this entry -3
bbbb	is the beginning sector of this data space
eeee	is the ending sector of this data space
x	represents an ASCII character in the name(s) assigned to this data space.

4.1.3 DIRECTORY TRAILER

The final directory record is the directory trailer or end directory record. It signifies that no more directory entries follow. It is three words of binary zeroes.

4.1.4 SECTOR USAGE

Directories are built from the ending sector toward the beginning sector in sector multiples. (When directory and file data spaces meet, the file is full.) In a like manner directory entries are built from the last word in the sector toward the first. Whenever a sector boundary is encountered, the directory entry is continued in the next sector beginning at the end of the sector.

Since all directory entries (including header and trailer) are a multiple of 3 words in length, the last two words of each 320 word sector or last word of a 400 word sector is used for padding. This padding prevents a name or entry header from being split. Figure 4-1 illustrates a typical 2 sector directory.

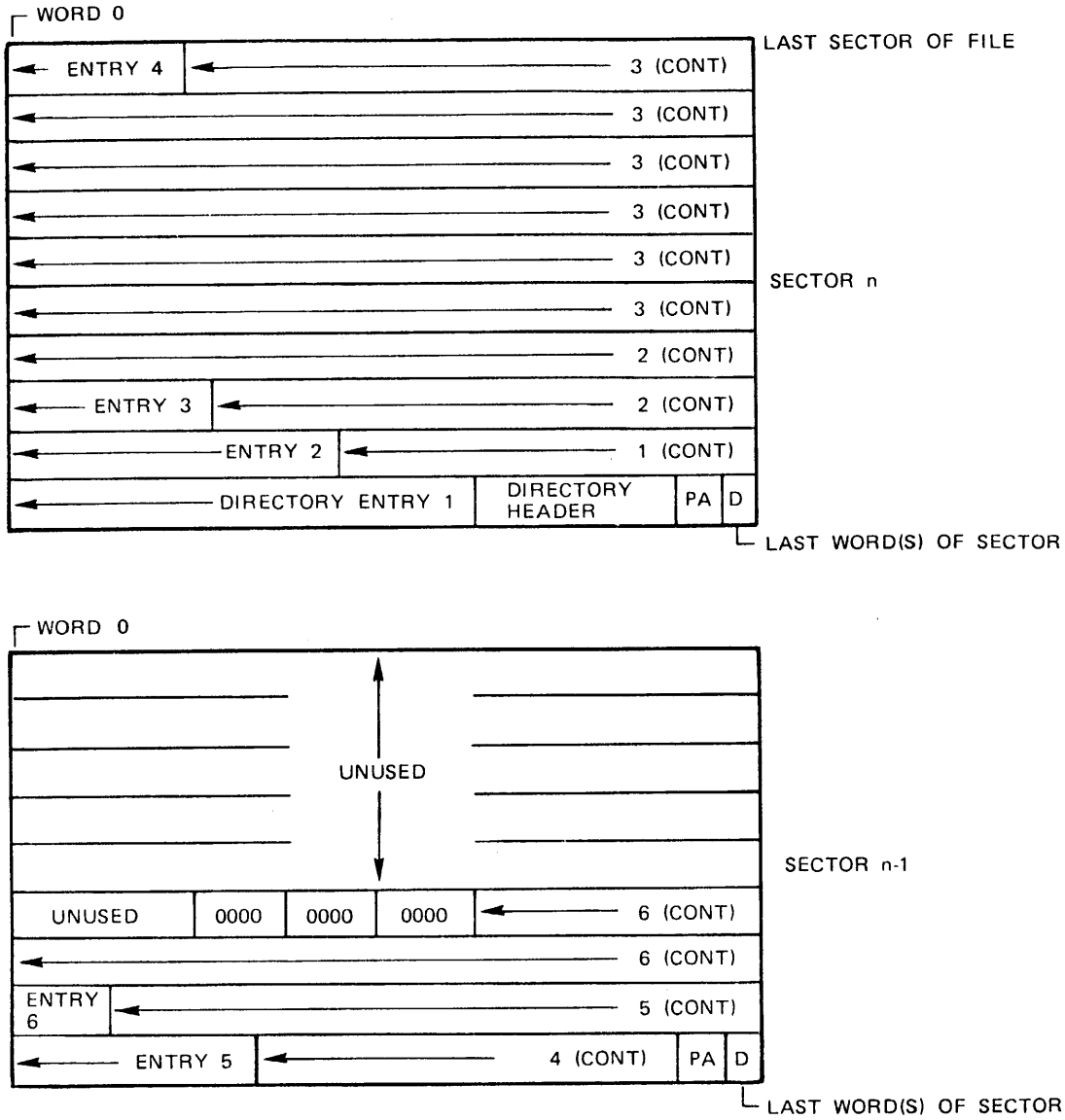


Figure 4-1. Typical 2 Sector Directory Layout

4.2.0 MAGNETIC TAPE

On a directoried magnetic tape file each data space contains, as its first record, the directory entry. No directory header appears. A directory trailer follows the last data space. Data spaces are separated by tape marks. A directoried tape appears as follows:

where:

‡ is a tape mark
IRG is an inter-record gap

Data spaces may contain subsequent inter-record gaps.

4.2.1 DIRECTORY ENTRY

Directory entries on magnetic tape are variable length and have the following format:

1	0003_2		DIRECTORY IDENTIFIER
2	X_1	X_1	NAME RECORD 1
3	X_1	X_1	
4	X_1	X_1	
n+1	X_n	X_n	NAME RECORD Q
n+2	X_n	X_n	
n+3	X_n	X_n	

where:

X_1 represents the ASCII characters comprising the name(s) allocated to this data space.

4.2.2 DIRECTORY TRAILER

The directory trailer for magnetic tape is the same as that for disk.

SECTION 5
CONVERSION TABLES

This section contains the following reference tables:

HEXADECIMAL ARITHMETIC

ADDITION TABLE

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

MULTIPLICATION TABLE

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

TABLE OF POWERS OF SIXTEEN$_{10}$

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

TABLE OF POWERS OF 10_{16}

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION

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

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

Hexadecimal fractions may be converted to decimal fractions as follows:

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

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

2. Find the decimal equivalent of the hexadecimal integer

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

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

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

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

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

$$\begin{array}{r} 0.895 \\ \underline{16} \\ 14.320 \\ \underline{16} \\ 5.120 \\ \underline{16} \\ 1.920 \\ \underline{16} \\ 14.720 \end{array}$$

$0.E51\ E_{16}$

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont.)

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont.)

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont.)

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont.)

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

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont.)

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

HEXADECIMAL-DECIMAL FRACTION CONVERSION

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

HEXADECIMAL-DECIMAL FRACTION CONVERSION (Cont.)

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 07225
.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 (Cont.)

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 00 40 00	.00000 38146	.00 00 80 00	.00000 76293	.00 00 C0 00	.00001 14440
.00 00 01 00	.00000 00596	.00 00 41 00	.00000 38743	.00 00 81 00	.00000 76889	.00 00 C1 00	.00001 15036
.00 00 02 00	.00000 01192	.00 00 42 00	.00000 39339	.00 00 82 00	.00000 77486	.00 00 C2 00	.00001 15633
.00 00 03 00	.00000 01788	.00 00 43 00	.00000 39935	.00 00 83 00	.00000 78082	.00 00 C3 00	.00001 16229
.00 00 04 00	.00000 02384	.00 00 44 00	.00000 40531	.00 00 84 00	.00000 78678	.00 00 C4 00	.00001 16825
.00 00 05 00	.00000 02980	.00 00 45 00	.00000 41127	.00 00 85 00	.00000 79274	.00 00 C5 00	.00001 17421
.00 00 06 00	.00000 03576	.00 00 46 00	.00000 41723	.00 00 86 00	.00000 79870	.00 00 C6 00	.00001 18017
.00 00 07 00	.00000 04172	.00 00 47 00	.00000 42319	.00 00 87 00	.00000 80466	.00 00 C7 00	.00001 18613
.00 00 08 00	.00000 04768	.00 00 48 00	.00000 42915	.00 00 88 00	.00000 81062	.00 00 C8 00	.00001 19209
.00 00 09 00	.00000 05364	.00 00 49 00	.00000 43511	.00 00 89 00	.00000 81658	.00 00 C9 00	.00001 19805
.00 00 0A 00	.00000 05960	.00 00 4A 00	.00000 44107	.00 00 8A 00	.00000 82254	.00 00 CA 00	.00001 20401
.00 00 0B 00	.00000 06556	.00 00 4B 00	.00000 44703	.00 00 8B 00	.00000 82850	.00 00 CB 00	.00001 20997
.00 00 0C 00	.00000 07152	.00 00 4C 00	.00000 45299	.00 00 8C 00	.00000 83446	.00 00 CC 00	.00001 21593
.00 00 0D 00	.00000 07748	.00 00 4D 00	.00000 45895	.00 00 8D 00	.00000 84042	.00 00 CD 00	.00001 22189
.00 00 0E 00	.00000 08344	.00 00 4E 00	.00000 46491	.00 00 8E 00	.00000 84638	.00 00 CE 00	.00001 22785
.00 00 0F 00	.00000 08940	.00 00 4F 00	.00000 47087	.00 00 8F 00	.00000 85234	.00 00 CF 00	.00001 23381
.00 00 10 00	.00000 09536	.00 00 50 00	.00000 47683	.00 00 90 00	.00000 85830	.00 00 D0 00	.00001 23977
.00 00 11 00	.00000 10132	.00 00 51 00	.00000 48279	.00 00 91 00	.00000 86426	.00 00 D1 00	.00001 24573
.00 00 12 00	.00000 10728	.00 00 52 00	.00000 48875	.00 00 92 00	.00000 87022	.00 00 D2 00	.00001 25169
.00 00 13 00	.00000 11324	.00 00 53 00	.00000 49471	.00 00 93 00	.00000 87618	.00 00 D3 00	.00001 25765
.00 00 14 00	.00000 11920	.00 00 54 00	.00000 50067	.00 00 94 00	.00000 88214	.00 00 D4 00	.00001 26361
.00 00 15 00	.00000 12516	.00 00 55 00	.00000 50663	.00 00 95 00	.00000 88810	.00 00 D5 00	.00001 26957
.00 00 16 00	.00000 13113	.00 00 56 00	.00000 51259	.00 00 96 00	.00000 89406	.00 00 D6 00	.00001 27553
.00 00 17 00	.00000 13709	.00 00 57 00	.00000 51856	.00 00 97 00	.00000 90003	.00 00 D7 00	.00001 28149
.00 00 18 00	.00000 14305	.00 00 58 00	.00000 52452	.00 00 98 00	.00000 90599	.00 00 D8 00	.00001 28746
.00 00 19 00	.00000 14901	.00 00 59 00	.00000 53048	.00 00 99 00	.00000 91195	.00 00 D9 00	.00001 29342
.00 00 1A 00	.00000 15497	.00 00 5A 00	.00000 53644	.00 00 9A 00	.00000 91791	.00 00 DA 00	.00001 29938
.00 00 1B 00	.00000 16093	.00 00 5B 00	.00000 54240	.00 00 9B 00	.00000 92387	.00 00 DB 00	.00001 30534
.00 00 1C 00	.00000 16689	.00 00 5C 00	.00000 54836	.00 00 9C 00	.00000 92983	.00 00 DC 00	.00001 3113[illegible]
.00 00 1D 00	.00000 17285	.00 00 5D 00	.00000 55432	.00 00 9D 00	.00000 93579	.00 00 DD 00	.00001 3172[illegible]
.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 370[illegible]
.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 3828[illegible]
.00 00 29 00	.00000 24437	.00 00 69 00	.00000 62584	.00 00 A9 00	.00001 00731	.00 00 E9 00	.00001 38878
.00 00 2A 00	.00000 25033	.00 00 6A 00	.00000 63180	.00 00 AA 00	.00001 01327	.00 00 EA 00	.00001 39474
.00 00 2B 00	.00000 25629	.00 00 6B 00	.00000 63776	.00 00 AB 00	.00001 01923	.00 00 EB 00	.00001 40070
.00 00 2C 00	.00000 26226	.00 00 6C 00	.00000 64373	.00 00 AC 00	.00001 02519	.00 00 EC 00	.00001 40666
.00 00 2D 00	.00000 26822	.00 00 6D 00	.00000 64969	.00 00 AD 00	.00001 03116	.00 00 ED 00	.00001 41263
.00 00 2E 00	.00000 27418	.00 00 6E 00	.00000 65565	.00 00 AE 00	.00001 03712	.00 00 EE 00	.00001 41859
.00 00 2F 00	.00000 28014	.00 00 6F 00	.00000 66161	.00 00 AF 00	.00001 04308	.00 00 EF 00	.00001 42455
.00 00 30 00	.00000 28610	.00 00 70 00	.00000 66757	.00 00 B0 00	.00001 04904	.00 00 F0 00	.00001 43051
.00 00 31 00	.00000 29206	.00 00 71 00	.00000 67353	.00 00 B1 00	.00001 05500	.00 00 F1 00	.00001 43647
.00 00 32 00	.00000 29802	.00 00 72 00	.00000 67949	.00 00 B2 00	.00001 06096	.00 00 F2 00	.00001 44243
.00 00 33 00	.00000 30398	.00 00 73 00	.00000 68545	.00 00 B3 00	.00001 06692	.00 00 F3 00	.00001 44839
.00 00 34 00	.00000 30994	.00 00 74 00	.00000 69141	.00 00 B4 00	.00001 07288	.00 00 F4 00	.00001 45435
.00 00 35 00	.00000 31590	.00 00 75 00	.00000 69737	.00 00 B5 00	.00001 07884	.00 00 F5 00	.00001 46031
.00 00 36 00	.00000 32186	.00 00 76 00	.00000 70333	.00 00 B6 00	.00001 08480	.00 00 F6 00	.00001 46627
.00 00 37 00	.00000 32782	.00 00 77 00	.00000 70929	.00 00 B7 00	.00001 09076	.00 00 F7 00	.00001 47223
.00 00 38 00	.00000 33378	.00 00 78 00	.00000 71525	.00 00 B8 00	.00001 09672	.00 00 F8 00	.00001 47819
.00 00 39 00	.00000 33974	.00 00 79 00	.00000 72121	.00 00 B9 00	.00001 10268	.00 00 F9 00	.00001 48415
.00 00 3A 00	.00000 34570	.00 00 7A 00	.00000 72717	.00 00 BA 00	.00001 10864	.00 00 FA 00	.00001 49011
.00 00 3B 00	.00000 35166	.00 00 7B 00	.00000 73313	.00 00 BB 00	.00001 11460	.00 00 FB 00	.00001 4960[illegible]
.00 00 3C 00	.00000 35762	.00 00 7C 00	.00000 73909	.00 00 BC 00	.00001 12056	.00 00 FC 00	.00001 502[illegible]
.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 (Cont.)

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 00455
.00 00 00 05	.00000 00011	.00 00 00 45	.00000 00160	.00 00 00 85	.00000 00309	.00 00 00 C5	.00000 00458
.00 00 00 06	.00000 00013	.00 00 00 46	.00000 00162	.00 00 00 86	.00000 00311	.00 00 00 C6	.00000 00461
.00 00 00 07	.00000 00016	.00 00 00 47	.00000 00165	.00 00 00 87	.00000 00314	.00 00 00 C7	.00000 00463
.00 00 00 08	.00000 00018	.00 00 00 48	.00000 00167	.00 00 00 88	.00000 00316	.00 00 00 C8	.00000 00465
.00 00 00 09	.00000 00020	.00 00 00 49	.00000 00169	.00 00 00 89	.00000 00318	.00 00 00 C9	.00000 00467
.00 00 00 0A	.00000 00023	.00 00 00 4A	.00000 00172	.00 00 00 8A	.00000 00321	.00 00 00 CA	.00000 00470
.00 00 00 0B	.00000 00025	.00 00 00 4B	.00000 00174	.00 00 00 8B	.00000 00323	.00 00 00 CB	.00000 00472
.00 00 00 0C	.00000 00027	.00 00 00 4C	.00000 00176	.00 00 00 8C	.00000 00325	.00 00 00 CC	.00000 00474
.00 00 00 0D	.00000 00030	.00 00 00 4D	.00000 00179	.00 00 00 8D	.00000 00328	.00 00 00 CD	.00000 00477
.00 00 00 0E	.00000 00032	.00 00 00 4E	.00000 00181	.00 00 00 8E	.00000 00330	.00 00 00 CE	.00000 00479
.00 00 00 0F	.00000 00034	.00 00 00 4F	.00000 00183	.00 00 00 8F	.00000 00332	.00 00 00 CF	.00000 00481
.00 00 00 10	.00000 00037	.00 00 00 50	.00000 00186	.00 00 00 90	.00000 00335	.00 00 00 D0	.00000 00484
.00 00 00 11	.00000 00039	.00 00 00 51	.00000 00188	.00 00 00 91	.00000 00337	.00 00 00 D1	.00000 00486
.00 00 00 12	.00000 00041	.00 00 00 52	.00000 00190	.00 00 00 92	.00000 00339	.00 00 00 D2	.00000 00488
.00 00 00 13	.00000 00044	.00 00 00 53	.00000 00193	.00 00 00 93	.00000 00342	.00 00 00 D3	.00000 00491
.00 00 00 14	.00000 00046	.00 00 00 54	.00000 00195	.00 00 00 94	.00000 00344	.00 00 00 D4	.00000 00493
.00 00 00 15	.00000 00048	.00 00 00 55	.00000 00197	.00 00 00 95	.00000 00346	.00 00 00 D5	.00000 00495
.00 00 00 16	.00000 00051	.00 00 00 56	.00000 00200	.00 00 00 96	.00000 00349	.00 00 00 D6	.00000 00498
.00 00 00 17	.00000 00053	.00 00 00 57	.00000 00202	.00 00 00 97	.00000 00351	.00 00 00 D7	.00000 00500
.00 00 00 18	.00000 00055	.00 00 00 58	.00000 00204	.00 00 00 98	.00000 00353	.00 00 00 D8	.00000 00502
.00 00 00 19	.00000 00058	.00 00 00 59	.00000 00207	.00 00 00 99	.00000 00356	.00 00 00 D9	.00000 00505
.00 00 00 1A	.00000 00060	.00 00 00 5A	.00000 00209	.00 00 00 9A	.00000 00358	.00 00 00 DA	.00000 00507
.00 00 00 1B	.00000 00062	.00 00 00 5B	.00000 00211	.00 00 00 9B	.00000 00360	.00 00 00 DB	.00000 00509
.00 00 00 1C	.00000 00065	.00 00 00 5C	.00000 00214	.00 00 00 9C	.00000 00363	.00 00 00 DC	.00000 00512
.00 00 00 1D	.00000 00067	.00 00 00 5D	.00000 00216	.00 00 00 9D	.00000 00365	.00 00 00 DD	.00000 00514
.00 00 00 1E	.00000 00069	.00 00 00 5E	.00000 00218	.00 00 00 9E	.00000 00367	.00 00 00 DE	.00000 00516
.00 00 00 1F	.00000 00072	.00 00 00 5F	.00000 00221	.00 00 00 9F	.00000 00370	.00 00 00 DF	.00000 00519
.00 00 00 20	.00000 00074	.00 00 00 60	.00000 00223	.00 00 00 A0	.00000 00372	.00 00 00 E0	.00000 00521
.00 00 00 21	.00000 00076	.00 00 00 61	.00000 00225	.00 00 00 A1	.00000 00374	.00 00 00 E1	.00000 00523
.00 00 00 22	.00000 00079	.00 00 00 62	.00000 00228	.00 00 00 A2	.00000 00377	.00 00 00 E2	.00000 00526
.00 00 00 23	.00000 00081	.00 00 00 63	.00000 00230	.00 00 00 A3	.00000 00379	.00 00 00 E3	.00000 00528
.00 00 00 24	.00000 00083	.00 00 00 64	.00000 00232	.00 00 00 A4	.00000 00381	.00 00 00 E4	.00000 00530
.00 00 00 25	.00000 00086	.00 00 00 65	.00000 00235	.00 00 00 A5	.00000 00384	.00 00 00 E5	.00000 00533
.00 00 00 26	.00000 00088	.00 00 00 66	.00000 00237	.00 00 00 A6	.00000 00386	.00 00 00 E6	.00000 00535
.00 00 00 27	.00000 00090	.00 00 00 67	.00000 00239	.00 00 00 A7	.00000 00388	.00 00 00 E7	.00000 00537
.00 00 00 28	.00000 00093	.00 00 00 68	.00000 00242	.00 00 00 A8	.00000 00391	.00 00 00 E8	.00000 00540
.00 00 00 29	.00000 00095	.00 00 00 69	.00000 00244	.00 00 00 A9	.00000 00393	.00 00 00 E9	.00000 00542
.00 00 00 2A	.00000 00097	.00 00 00 6A	.00000 00246	.00 00 00 AA	.00000 00395	.00 00 00 EA	.00000 00544
.00 00 00 2B	.00000 00100	.00 00 00 6B	.00000 00249	.00 00 00 AB	.00000 00398	.00 00 00 EB	.00000 00547
.00 00 00 2C	.00000 00102	.00 00 00 6C	.00000 00251	.00 00 00 AC	.00000 00400	.00 00 00 EC	.00000 00549
.00 00 00 2D	.00000 00104	.00 00 00 6D	.00000 00253	.00 00 00 AD	.00000 00402	.00 00 00 ED	.00000 00551
.00 00 00 2E	.00000 00107	.00 00 00 6E	.00000 00256	.00 00 00 AE	.00000 00405	.00 00 00 EE	.00000 00554
.00 00 00 2F	.00000 00109	.00 00 00 6F	.00000 00258	.00 00 00 AF	.00000 00407	.00 00 00 EF	.00000 00556
.00 00 00 30	.00000 00111	.00 00 00 70	.00000 00260	.00 00 00 B0	.00000 00409	.00 00 00 F0	.00000 00558
.00 00 00 31	.00000 00114	.00 00 00 71	.00000 00263	.00 00 00 B1	.00000 00412	.00 00 00 F1	.00000 00561
.00 00 00 32	.00000 00116	.00 00 00 72	.00000 00265	.00 00 00 B2	.00000 00414	.00 00 00 F2	.00000 00563
.00 00 00 33	.00000 00118	.00 00 00 73	.00000 00267	.00 00 00 B3	.00000 00416	.00 00 00 F3	.00000 00565
.00 00 00 34	.00000 00121	.00 00 00 74	.00000 00270	.00 00 00 B4	.00000 00419	.00 00 00 F4	.00000 00568
.00 00 00 35	.00000 00123	.00 00 00 75	.00000 00272	.00 00 00 B5	.00000 00421	.00 00 00 F5	.00000 00570
.00 00 00 36	.00000 00125	.00 00 00 76	.00000 00274	.00 00 00 B6	.00000 00423	.00 00 00 F6	.00000 00572
.00 00 00 37	.00000 00128	.00 00 00 77	.00000 00277	.00 00 00 B7	.00000 00426	.00 00 00 F7	.00000 00575
.00 00 00 38	.00000 00130	.00 00 00 78	.00000 00279	.00 00 00 B8	.00000 00428	.00 00 00 F8	.00000 00577
.00 00 00 39	.00000 00132	.00 00 00 79	.00000 00281	.00 00 00 B9	.00000 00430	.00 00 00 F9	.00000 00579
.00 00 00 3A	.00000 00135	.00 00 00 7A	.00000 00284	.00 00 00 BA	.00000 00433	.00 00 00 FA	.00000 00582
.00 00 00 3B	.00000 00137	.00 00 00 7B	.00000 00286	.00 00 00 BB	.00000 00435	.00 00 00 FB	.00000 00584
.00 00 00 3C	.00000 00139	.00 00 00 7C	.00000 00288	.00 00 00 BC	.00000 00437	.00 00 00 FC	.00000 00586
.00 00 00 3D	.00000 00142	.00 00 00 7D	.00000 00291	.00 00 00 BD	.00000 00440	.00 00 00 FD	.00000 00589
.00 00 00 3E	.00000 00144	.00 00 00 7E	.00000 00293	.00 00 00 BE	.00000 00442	.00 00 00 FE	.00000 00591
.00 00 00 3F	.00000 00146	.00 00 00 7F	.00000 00295	.00 00 00 BF	.00000 00444	.00 00 00 FF	.00000 00593

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 66[illegible] 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 83[illegible] 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125

MATHEMATICAL CONSTANTS

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

SECTION 6
STANDARD CHARACTER CODES

Graphic or Control	System ASCII See Note 1 (Hexadecimal)	Card Code See Note 2		Printer See Note 3
		IBM 029	IBM 026	
NULL	80	No Equivalent Codes		No Equivalent Codes
SOM	81			
EOA	82			
EOM	83			
EOT	84			
WRU	85			
RU	86			
BELL	87			
FE	88			
H. Tab	89			
Line Feed	8A			
V. Tab	8B			
Form	8C			
Return†	8D			
SO	8E			
SI	8F			
DCO	90			
X-On	91			
Tape Aux. On	92			
X-Off	93			
Tape Aux. Off	94			
Error	95			
Sync	96			
LEM	97			
S0	98			
S1	99			
S2	9A			
S3	9B			
S4	9C			
S5	9D			
S6	9E			
S7	9F			

† When output is to a Teletype the Return code should precede the Line Feed to prevent overprinting.

Graphic or Control	System ASCII See Note 1 (Hexadecimal)	Card Code See Note 2		Printer See Note 3
		IBM 029	IBM 026	
blank	A0	No Punch		
! exclamation	A1	11-2-8	11-2-8	
" quote	A2	7-8	0-5-8	
#	A3	3-8	0-7-8	
$	A4	11-3-8		
%	A5	0-4-8	11-7-8	
&	A6	12	12-7-8	
' apostrophe	A7	5-8	4-8	
(open paren.	A8	12-5-8	0-4-8	
) close paren.	A9	11-5-8	12-4-8	
*	AA	11-4-8		
+	AB	12-6-8	12	
, comma	AC	0-3-8		
- minus	AD	11		
. period	AE	12-3-8		
/ slash	AF	0-1		
0 zero	B0	0		All characters are valid
1	B1	1		
2	B2	2		
3	B3	3		
4	B4	4		
5	B5	5		
6	B6	6		
7	B7	7		
8	B8	8		
9	B9	9		
: colon	BA	2-8	5-8	
; semicolon	BB	11-6-8		
< less	BC	12-4-8	12-6-8	
= equal	BD	6-8	3-8	
> greater	BE	0-6-8	6-8	
?	BF	0-7-8	12-2-8	
@	C0	4-8	0-2-8	

Graphic or Control	System ASCII See Note 1 (Hexadecimal)	Card Code See Note 2 IBM 029	Card Code See Note 2 IBM 026	Printer See Note 3
A	C1	12-1		All characters valid
B	C2	12-2		
C	C3	12-3		
D	C4	12-4		
E	C5	12-5		
F	C6	12-6		
G	C7	12-7		
H	C8	12-8		
I	C9	12-9		
J	CA	11-1		
K	CB	11-2		
L	CC	11-3		
M	CD	11-4		
N	CE	11-5		
O	CF	11-6		
P	D0	11-7		
Q	D1	11-8		
R	D2	11-9		
S	D3	0-2		
T	D4	0-3		
U	D5	0-4		
V	D6	0-5		
W	D7	0-6		
X	D8	0-7		
Y	D9	0-8		
Z	DA	0-9		
[open bracket	DB	12-8-2	12-5-3	
\ back slash	DC	0-8-2	0-6-8	
] close bracket	DD	0-8-5	11-5-8	
↑ Λ ††	DE	11-7-8	7-8	
← ___ ††	DF	12-7-8	2-8	
ACK	FC	No equivalent code		No equivalent code
Alt. Mode	FD			
Rubout	FF			

†† ASCII codes DE and DF produce up arrow and left arrow for Teletype; circumplex and underscore for the Line Printer. See note 3.

NOTES FOR STANDARD CHARACTER CODES

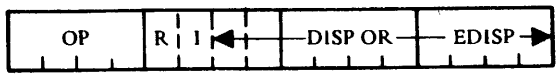
Note 1. ASCII is a seven bit data code represented by the seven least significant bits of an eight bit data byte. The eighth bit of an ASCII data byte may be used for a parity bit. The parity scheme used may be odd or even or not used at all, depending on the particular device and model. GA I/O systems ignore this parity bit when received. When the data byte is stored in memory the eighth bit will be set to a one. The data is then in System ASCII form. On output the I/O systems always set the eighth bit to a one.

Note 2. All GA I/O systems are set up to recognize IBM 029 keypunch codes. IBM 026 codes are given only where they are different from the IBM 029 codes.

Note 3. The line printers use a 64-character subset of the ASCII code. The seventh and eighth bit are completely ignored and may be in any state. The System ASCII codes given in column one are fully usable as printer codes for all legal characters. Codes DE_{16} and DF_{16} produce unique characters on the printer which differ from the Teletype. See graphic column.

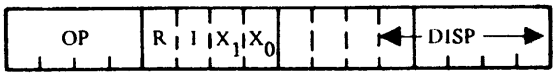
SECTION 7
SPC-16 INSTRUCTION SUMMARY HEXADECIMAL CODING

MEMORY REFERENCE



LDA	4	DISP for R=1 (Base)
STA	5	
JSR	6	EDISP for R=0 (Progr)
JMP	7	

MEMORY REFERENCE WITH INDEX



LDR	C	REG	
STR	D	REG	
CMR	E	REG	R=1 (Base)
LDBY	8	REG	R=0 (Direct)
STBY	9	REG	
SBIT	B	Bit ID	
RBIT	3	Bit ID	
TBIT	A	Bit ID	
LARS	F	1 0 0	
SARS	F	1 1 0	
INCM	F	0 0 0	
DECM	F	0 1 0	

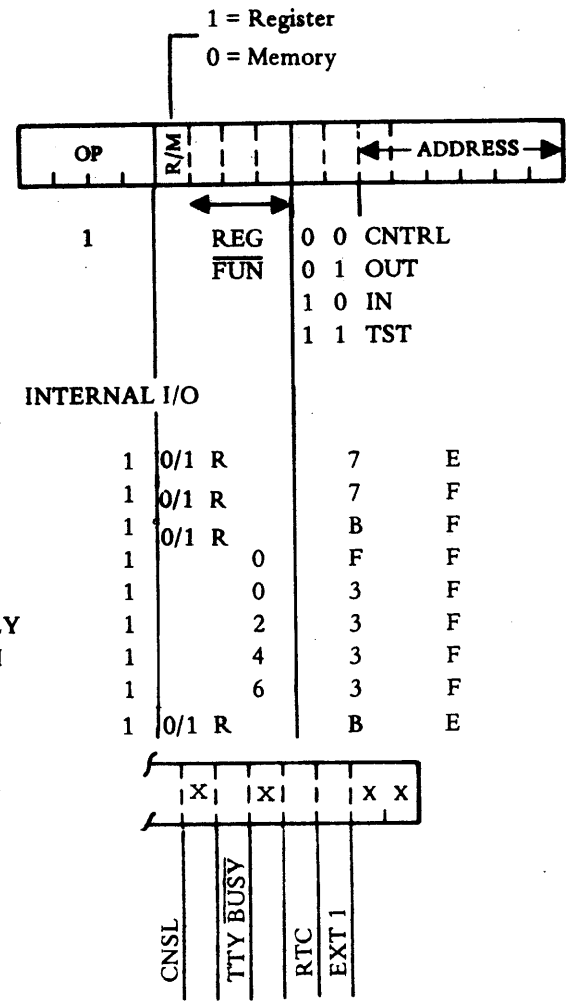
SKIP

0 0	OVFL
0 1	LINK
1 0	ZERO
1 1	PLUS

	SKIP FWD			SKIP REV	
SKOT	20	XX	OVFL	21	XX
SKOF	28	XX	$\overline{\text{OVFL}}$	29	XX
SKS	2A	XX	LINK	2B	XX
SKR	22	XX	$\overline{\text{LINK}}$	23	XX
SKZ	2C	XX	ZERO	2D	XX
SKN	24	XX	NONZERO	25	XX
SKP	2E	XX	PLUS	2F	XX
SKM	26	XX	MINUS	27	XX

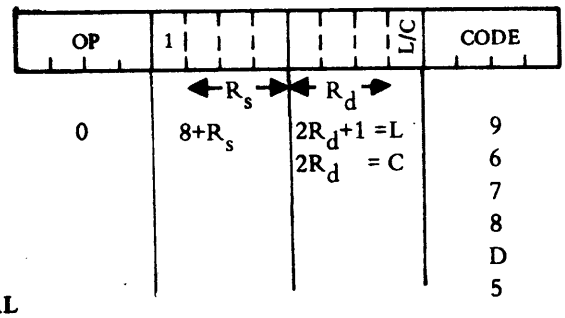
NOTE: SKIP REV XX is in 2's complement form.

XIO

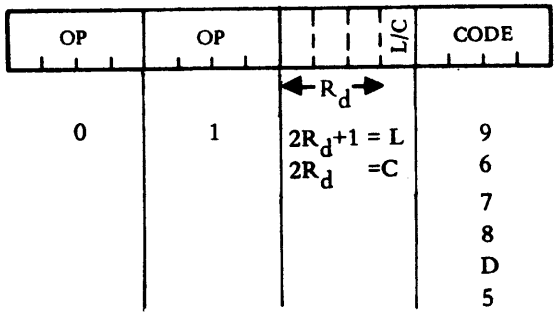


MASK BITS

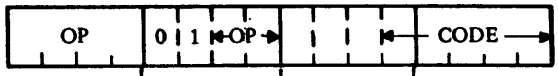
REGISTER OPERATE

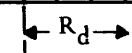


REGISTER OPERATE LITERAL



REGISTER CHANGE





XEC	0	5	$2R_d+1$	0
RISE	0	5	$2R_d$	1
RTRN	0	5	$2R_d$	3
DSPL	0	5	$2R_d$	4
TRS	0	5	$2R_d$	8
ZERO	0	6	$2R_d$	0
ZRBY	0	6	$2R_d$	1
ZLBY	0	6	$2R_d$	2
EXBY	0	6	$2R_d$	4
TSR	0	6	$2R_d$	8
RCSW	0	6	$2R_d+1$	0
CMPL	0	7	$2R_d$	0
RLK	0	7	$2R_d$	1
DECR	0	7	$2R_d$	2
ADDS	0	7	$2R_d$	B
INCR	0	7	$2R_d$	E

SHIFTS

SRLC	0	2	$2R_d$	CNT
SRA	0	2	$2R_d+1$	CNT
SRC	0	3	$2R_d$	CNT
SRCL	0	3	$2R_d+1$	CNT

CONTROL

INE	0	4	0	3
INH	0	4	0	2
FMS	0	4	0	C
BMS	0	4	0	8
LKS	0	4	3	0
LKR	0	4	2	0
PMA	0	4	4	0
SYNC	0	4	8	0
WAIT	0	0	0	X
MPY	0	0	8	n
DIV	0	0	A	n

COMMENT SHEET

FROM:

NAME: _____

BUSINESS ADDRESS: _____

_____

_____

Does this publication meet your requirements? Yes ☐ No ☐

If no, please explain. _____

_____

_____

_____

_____

Do you wish a reply? Yes ☐ No ☐

COMMENTS: (Describe any errors, suggested ideas, additions, or deletions etc. Please include page number.)
All comments and suggestions become the property of General Automation, Inc.

_____

_____

_____

_____

_____

_____

_____

_____

No Postage Stamp Necessary If Mailed In the U.S.A.
(See Other Side)
Fold On Dotted Lines And Staple

YOUR COMMENTS, PLEASE . . .

This publication serves as a reference for systems analysts, programmers and operators of General Automation systems. Your answers to the questions on the back of this form help us produce better publications for your use.

FOLD

FIRST CLASS
PERMIT NO. 423
ANAHEIM, CALIF.

BUSINESS REPLY MAIL
No Postage Necessary if Mailed in the United States

Postage Will Be Paid By . . .

GENERAL AUTOMATION, INC.
1055 East Street
Anaheim, California 92803

Attention: Technical Publications

CUT ALONG LINE

FOLD

General Automation, Inc.
1055 East Street
Anaheim, California 92803

ADDITIONAL COMMENTS:

GENERAL AUTOMATION, INC.
1055 South East Street, Anaheim, California 92805 (714) 778-4800