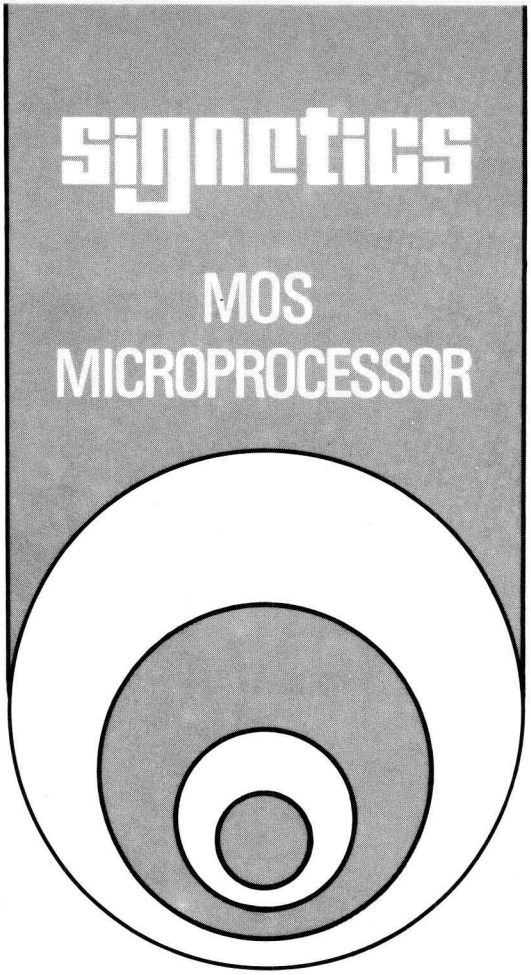

INTRODUCTION

Conversion routines like binary to BCD, BCD to binary, and BCD to ASCII are often used in microprocessor based systems. This applications memo describes routines for converting:

- Eight-bit unsigned binary to BCD.
- Sixteen-bit signed binary to BCD.
- Signed BCD to binary conversion 1 (using an addition method).
- Signed BCD to binary conversion 2 (using a multiplication method).
- Signed BCD to ASCII
- ASCII to BCD
- Hexadecimal to ASCII
- ASCII to Hexadecimal

1. EIGHT-BIT UNSIGNED BINARY-TO-BCD CONVERSION

FUNCTION:

Converts an unsigned binary number to a BCD number (3 digits).
(BINN) —Conversion→ R0, R1
A multiplication method is used.

PARAMETERS:

Input: BINN contains the binary number (8 bits unsigned.

Output: Registers R0, R1 contain the BCD result (3 BCD digits).
R0 is the most-significant byte.
The maximum BCD result is 256 decimal.

Refer to figures 1.1 and 1.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0 X	R1 X	R2	R3	R1'	R2'	R3'
PSU	F	II	SP				
PSL	CC X	IDC X	RS	WC X	OVF X	COM X	C X

RAM REQUIRED (BYTES): 1

ROM REQUIRED (BYTES): 28

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

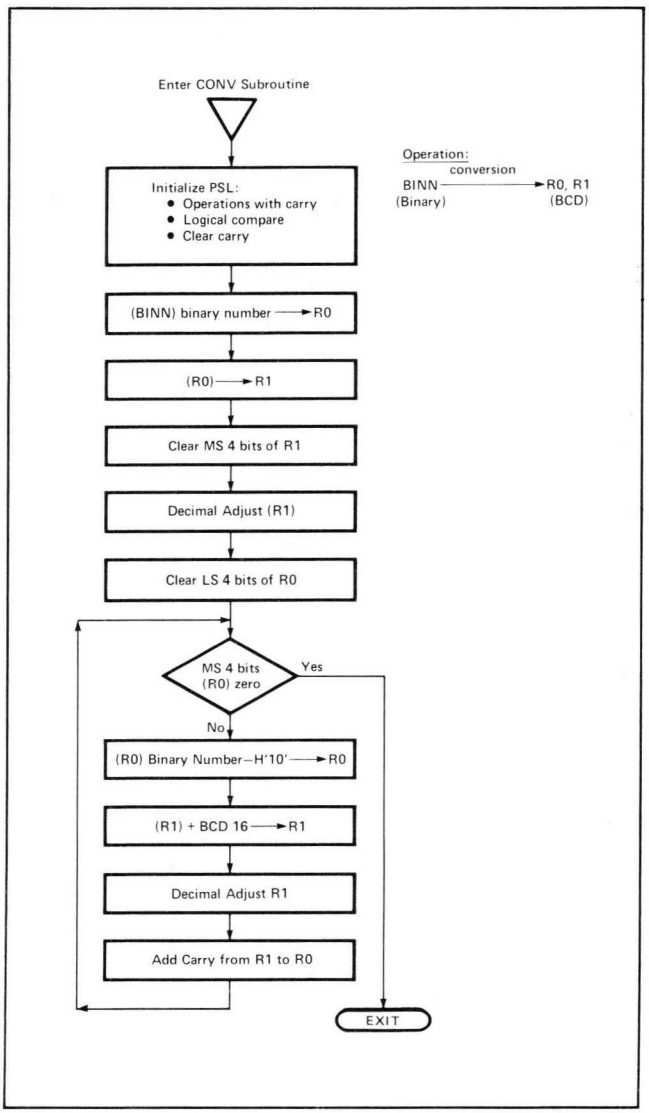

FIGURE 1-1 Flowchart for Eight-Bit Unsigned Binary-to-BCD Conversion (Multiplication Method)

```
 1                                      *     PD76ØØ5Ø
 2                                      *++++++++++++++++++++++++++++++++++++++++++++++
 3                                      *    8 BIT UNSIGNED BINARY TO BCD CONVERSION
 4                                      *++++++++++++++++++++++++++++++++++++++++++++++
 5                                      *
 6                                      *THIS ROUTINE CONVERTS AN 8 BIT UNSIGNED BINARY
 7                                      *NUMBER INTO AN UNSIGNED BCD NUMBER.
 8                                      *
 9                                      *BINARY NUMBER IS IN BINN.
10                                      *BCD NUMBER (AFTER CONVERSION) IS IN RØ,R1.
11                                      *    HUNDREDS IN RØ
12                                      *    TENS,UNITS IN R1.
13                                      *
14                                      *DEFINITIONS OF SYMBOLS:
15                                      *
16        ØØØØ                          R0   EQU     Ø          PROCESSOR-REGISTERS
17        ØØØ1                          R1   EQU     1
18        ØØØ8                          WC   EQU     H'Ø8'      PSL: 1=WITH, Ø=WITHOUT CARRY
19        ØØØ2                          COM  EQU     H'Ø2'           1=LOGIC, Ø=ARITH.COMPARE
20        ØØØ1                          C    EQU     H'Ø1'           CARRY:BORROW
21        ØØØ3                          UN   EQU     3          BRANCH COND.: UNCONDITIONAL
22        ØØØ2                          LT   EQU     2                        LESS THAN
23                                      *
24                                      *
25                                           ORG     H'6ØØ'
26                                      *
27        Ø6ØØ                          BINN RES     1          BINARY NUMBER.
28                                      *
29                                           ORG     H'5ØØ'     START ADDRESS OF ROUTINE.
30                                      *
31                                      *                       INITIALISATION:
32  Ø5ØØ  Ø5ØØ  77 ØA                   CONV PPSL    WC+COM     WITH CARRY,LOGICAL COMPARE
33  Ø5Ø2        75 Ø1                        CPSL    C          CLEAR CARRY FLAG IN PSL.
34  Ø5Ø4        ØC Ø6 ØØ                     LODA,RØ BINN       8 BIT BIN.NUMBER -> RØ.
35  Ø5Ø7        C1                           STRZ    R1         (RØ) -> R1.
36  Ø5Ø8        45 ØF                        ANDI,R1 H'ØF'      CLEAR MS 4 BITS BIN. NUMBER
37  Ø5ØA        85 66                        ADDI,R1 H'66'      PREPARE R1 FOR DECIMAL ADJUST.
38  Ø5ØC        95                           DAR,R1
39                                      *
40  Ø5ØD        44 FØ                        ANDI,RØ H'FØ'      CLEAR LS 4 BITS.
41                                      *
42  Ø5ØF  Ø5ØF  E4 1Ø                   LOOP COMI,RØ H'1Ø'
43  Ø511        1A Ø9                        BCTR,LT EXIT       IF MS 4 BITS ZERO THEN RETRUN.
44  Ø513        A4 ØF                        SUBI,RØ H'1Ø'-1    SUBTRACT 1 FROM MS 4 BITS
45  Ø515        85 7B                        ADDI,R1 H'16'+H'66'-1 ADD BCD 16 AND PREPARE
46  Ø517        95                           DAR,R1             FOR DECIMAL ADJUST.
47  Ø518        84 ØØ                        ADDI,RØ Ø          ADD CARRY TO MS BCD DIGIT
48  Ø51A        1B 73                        BCTR,UN LOOP       BRANCH AGAIN
49                                      *
50  Ø51C  Ø51C  4Ø                      EXIT HALT               END OF CONVERSION.
51                                           END
```

FIGURE 1-2 Program Listing for Eight-Bit Unsigned Binary-to-BCD Conversion

2. SIXTEEN-BIT SIGNED BINARY-TO-BCD CONVERSION

FUNCTION:

Converts a signed 16-bit binary number to a signed BCD number.
Subtraction of base numbers is used.

PARAMETERS:

Input: BINN, BINN+1 contain the signed binary number.
BINN is the most-significant byte.
Binary number is destroyed after conversion.

Output: BCDD, BCDD+1, BCDD+2 contain the BCD result.
BCDD contains the sign and the most-significant BCD digit.
The minimum BCD result is –32768 decimal.
The maximum BCD result is +32767 decimal.

Refer to figures 2.1 and 2.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0	R1	R2	R3	R1'	R2'	R3'
	X	X	X	X			
PSU	F	II	SP				
PSL	CC	IDC	RS	WC	OVF	COM	C
	X	X		X	X		X

RAM REQUIRED (BYTES): 5

ROM REQUIRED (BYTES): 106

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

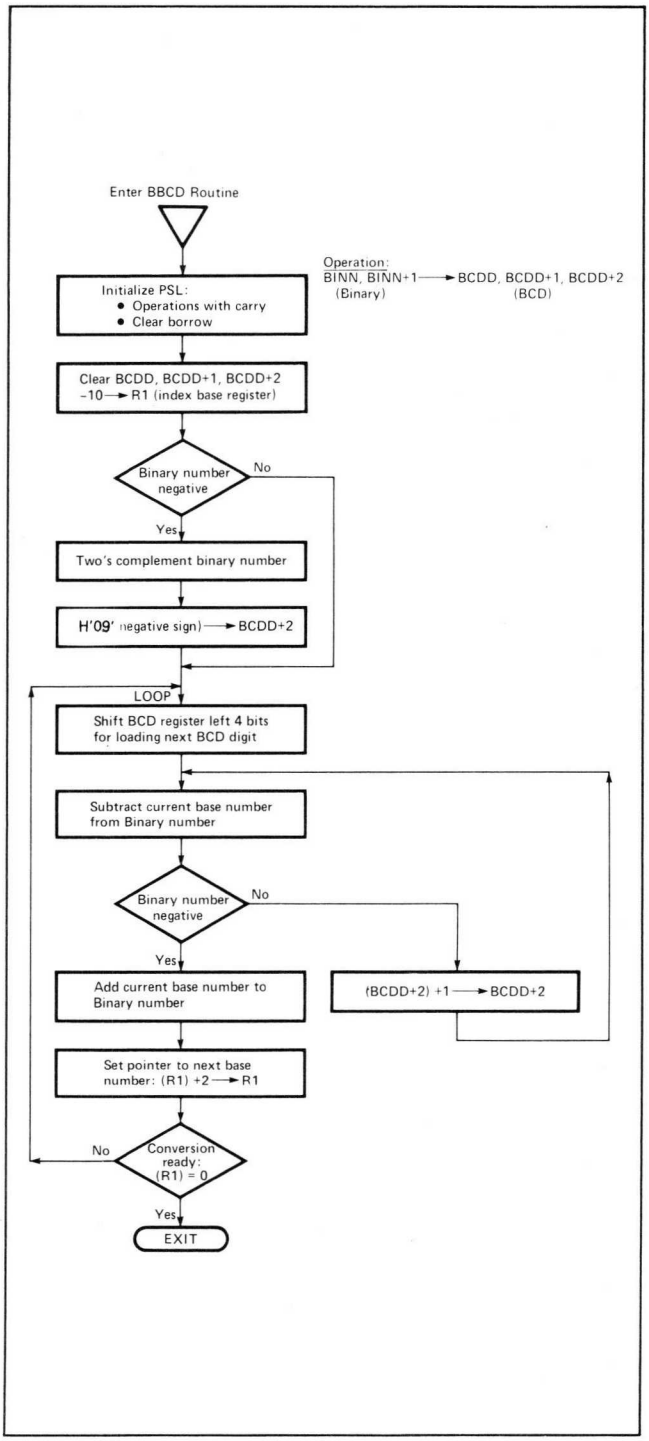

FIGURE 2-1 Flowchart for Signed Binary-to-BCD Conversion

```
1                                       *    PD7600S1
2                                       ++++++++++++++++++++++++++++++++++++++++++++++++++
3                                       *      BINARY TO BCD CONVERSION                  +
4                                       ++++++++++++++++++++++++++++++++++++++++++++++++++
5                                       *
6                                       *THIS ROUTINE CONVERTS A SIGNED BINARY NUMBER
7                                       *(16 BITS) INTO A SIGNED BCD NUMBER
8                                       *(24 BITS: SIGN + 5 BCD DIGITS).
9                                       *
10                                      *THE BINARY NUMBER IS IN BINN,BINN+1.
11                                      *THE BCD NUMBER IS IN BCDD,BCDD+1,BCDD+2.
12                                      *BINN AND BCDD ARE MOST SIGNIFICANT BYTES.
13                                      *MS NIBBLE OF BCDD=0 FOR POSITIVE BINARY NUMBERS.
14                                      *MS NIBBLE OF BCDD=9 FOR NEGATIVE BINARY NUMBERS.
15                                      *
16                                      *SUBTRAHENDS ARE PLACED IN REGISTER BASE (10 BYTES)
17                                      *
18                                      *DEFINITION OF SYMBOLS:
19                                      *
20      0000                            R0   EQU    0          PROCESSOR-REGISTERS
21      0001                            R1   EQU    1
22      0002                            R2   EQU    2
23      0003                            R3   EQU    3
24      0008                            WC   EQU    H'08'      PSL: 1=WITH, 0=WITHOUT CARRY
25      0001                            C    EQU    H'01'           CARRY+BORROW
26      0002                            N    EQU    2          BRANCH COND.: NEGATIVE
27      0003                            UN   EQU    3                        UNCONDITIONALY
28                                      *
29                                      *
30                                          ORG    H'6000'    START ADDRESS
31                                      *
32      0600                            BINN RES    2          BINARY NUMBER MEMORY LOCATION
33      0602                            BCDD RES    3          BCD REGISTER
34  0605 0605  27 10                    BASE DATA   H'27,10'   10000
35  0607       03 E8                        DATA   H'03,E8'   1000
36  0609       00 64                        DATA   H'00,64'   100
37  060B       00 0A                        DATA   H'00,0A'   10
38  060D       00 01                        DATA   H'00,01'   1
39      000A                            LEN  EQU    $-BASE     LENGTH BASE REGISTER
41                                          ORG    H'5000'    START ADDR. OF PROGRAM
42                                      *
43  0500 0500  77 09                    BBCD PPSL   WC+C        ARITHMETIC+ROTATE WITH CARRY:
44                                      *                      CLEAR BORROW.
45  0502       20                           EORZ   R0         INITIALISATION: CLEAR R0.
46  0503       07 03                        LODI,R3 3
47  0505 0505  CF 46 02                 LOC0 STRA,R0 BCDD,R3,- CLEAR 3 BYTES OF BCD REGISTER.
48  0508       5B 7B                        BRNR,R3 LOC0
49  050A       05 F6                        LODI,R1 -LEN       LENGTH OF BASE REGISTER.
50                                      *
51  050C       0E 06 00                     LODA,R2 BINN       MS 4 BITS BINARY NUMBER.
52  050F       9A 10                        BCFR,N  LOOP       IF POS. GO TO LOOP
53  0511 0511  06 02                    COMP LODI,R2 2          LOAD INDEX REGISTER.
54  0513 0513  20                       LOC1 EORZ   R0         TWO'S COMPLEMENT BY
55  0514       AE 46 00                     SUBA,R0 BINN,R2,-  SUBTRACTING FROM ZERO.
56  0517       CE 66 00                     STRA,R0 BINN,R2
57  051A       5A 77                        BRNR,R2 LOC1       RETURN IF NOT READY.
58  051C       04 09                        LODI,R0 H'09'      NEGATIVE SIGN INDICATION.
59  051E       CC 06 04                     STRA,R0 BCDD+2     SIGN IN LSB OF BCD REGISTER.
60                                      *                      SHIFT BCD REG. LEFT 4 TIMES.
61  0521 0521  75 01                    LOOP CPSL   C          CLEAR CARRY FOR ROTATE.
62  0523       06 04                        LODI,R2 4          BIT COUNT.
63  0525 0525  07 03                    LP2  LODI,R3 3          INDEX BYTE SHIFT.
64  0527 0527  0F 46 02                 LP1  LODA,R0 BCDD,R3,- BCD DIGIT INTO R0.
65  052A       D0                           RRL,R0             CARRY (PREVIOUS MS BIT)-> LSB
66  052B       CF 66 02                     STRA,R0 BCDD,R3    AND MS BIT -> CARRY.
67  052E       5B 77                        BRNR,R3 LP1
68  0530       FA 73                        BDRR,R2 LP2
69                                      *
70  0532 0532  85 02                    SUBL ADDI,R1 2          RESTORE BASE INDEX.
71  0534       06 02                        LODI,R2 2          INDEX REGISTER
72  0536       77 01                        PPSL   C           CLEAR BORROW
73  0538 0538  0E 46 00                 LOC2 LODA,R0 BINN,R2,-  LOAD BINN AND SUBTRACT
74  053B       AD 45 0F                     SUBA,R0 BASE-256+LEN,R1,- CORRESPONDING
75  053E       CE 66 00                     STRA,R0 BINN,R2    BASE DIGIT
76  0541       5A 75                        BRNR,R2 LOC2
77  0543       1A 09                        BCTR,N  CORR       IF BINN NEG. THEN CORRECTION.
78  0545       0C 06 04                     LODA,R0 BCDD+2
79  0548       82                           ADDZ   R2          ADD 1 TO LSB OF BCD NUMBER
80  0549       CC 06 04                     STRA,R0 BCDD+2     C=1 IN PSL AND (R2)=0
81  054C       1B 64                        BCTR,UN SUBL
82                                      *
83  054E 054E  06 02                    CORR LODI,R2 2          INDEX COUNT
84  0550 0550  0E 46 00                 LOC3 LODA,R0 BINN,R2,- ADD CORRESPONDING BASE BYTE TO
85  0553       8D 45 11                     ADDA,R0 BASE-256+LEN+2,R1,- BINARY NUMBER.
86  0556       CE 66 00                     STRA,R0 BINN,R2
87  0559       5A 75                        BRNR,R2 LOC3       RETURN IF NOT READY
88  055B       85 03                        ADDI,R1 3          UPDATE BASE POINTER:C=1IN PSL
89  055D       59 42                        BRNR,R1 LOOP       RETURN IF CONVERSION NOT READY
90  055F 055F  40                       EXIT HALT   END OF CONVERSION
91                                          END
```

FIGURE 2-2 Program Listing for Signed Binary-to-BCD Conversion

3. SIGNED BCD-TO-BINARY CONVERSION 1

FUNCTION:

Converts a five-digit signed BCD number to a sixteen-bit signed binary number.

Addition of base numbers is used.

PARAMETERS:

Input: BCDD, BCDD+1, BCDD+2 contain the BCD number.
BCDD contains the sign plus the most-significant BCD digit.
The range of BCD numbers is: $-32768 < \text{BCD Number} < +32767$.
BCDD is destroyed after the conversion.

Output: BINN, BINN+1 contain the signed binary number.
BINN is the most-significant byte.

Refer to figures 3.1 and 3.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0	R1	R2	R3	R1'	R2'	R3'
	X	X	X	X			
PSU	F	II	SP				
PSL	CC	IDC	RS	WC	OVF	COM	C
	X	X		X	X		X

RAM REQUIRED (BYTES): 5

ROM REQUIRED (BYTES): 86

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

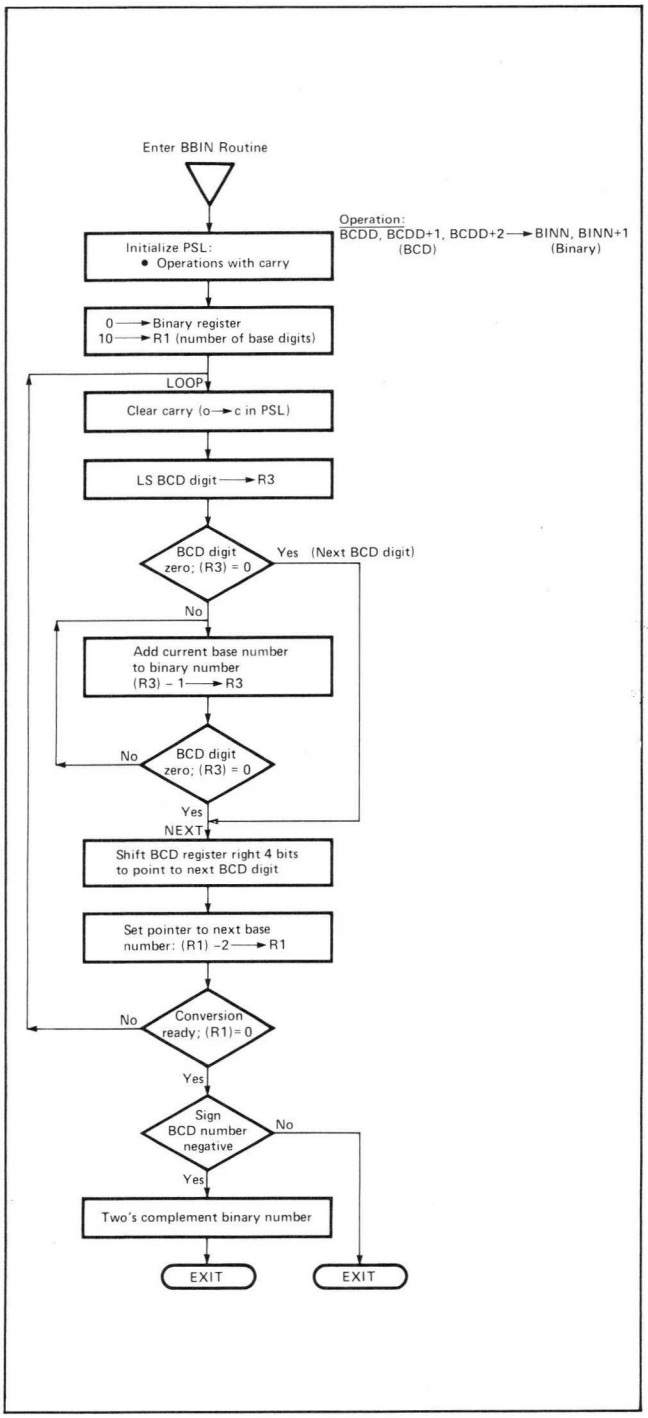

FIGURE 3-1: Flowchart for signed BCD-to-Binary Conversion

```
1                                       * PD76##52
2                                       ++++++++++++++++++++++++++++++++++++++++++++++
3                                       * BCD TO BINARY CONVERSION
4                                       ++++++++++++++++++++++++++++++++++++++++++++++
5                                       *
6                                       * THIS ROUTINE CONVERTS A SIGNED BCD NUMBER
7                                       * (24 BITS: SIGN+5 BCD DIGITS) INTO A SIGNED
8                                       * BINARY NUMBER (16 BITS).
9                                       * -32768 <BCD NUMBER <+32767
10                                      *    BCD NUMBER IS LOST AFTER CONVERSION.
11                                      *
12                                      * THE BINARY NUMBER IS IN BINN,BINN+1.
13                                      * THE BCD NUMBER IS IN BCDD,BCDD+1,BCDD+2 (A#-A4).
14                                      * THE BASE NUMBERS ARE IN BASE,- -,BASE+9 (R#,R4).
15                                      * BINN AND BCDD ARE MOST SIGNIFICANT BYTES.
16                                      *
17                                      * PRINCIPLE OF CONVERSION IS:
18                                      * BINN = A#.R#+ A1.R1+ A2.R2+ A3.R3+ A4.R4
19                                      *    A# -A4 = NUMBER OF DIGITS OF BCD NUMBER.
20                                      *    R# -R4 = BASE NUMBERS FOR CONVERSION.
21                                      *
22                                      * DEFINITIONS OF SYMBOLS:
23      0000                            R#   EQU  #          PROCESSOR-REGISTERS
24      0001                            R1   EQU  1
25      0002                            R2   EQU  2
26      0003                            R3   EQU  3
27      0008                            WC   EQU  H'08'      PSL: 1=WITH, #=WITHOUT CARRY
28      0001                            C    EQU  H'01'           CARRY:BORROW
29      0000                            Z    EQU  #          BRANCH COND: ZERO
30      0000                            ON    EQU  #                      ALL BITS ARE 1
31      0008                            SIGN EQU  H'08'      TO TEST BCD. NUMBER
32      000A                            LEN  EQU  10         INDEX NUMBER (LENGTH BASE REG)
33                                      *
34                                        ORG    H'6##'
35                                      *
36      0600                            BINN RES  2          BINARY NUMBER
37      0602                            BCDD RES  3          BCD NUMBER
38  0605 0605  27 10                    BASE DATA H'27,10'   10000
39  0607        03 E8                        DATA H'03,E8'   1000
40  0609        00 64                        DATA H'00,64'   100
41  060B        00 0A                        DATA H'00,0A'   10
42  060D        00 01                        DATA H'00,01'   1
44                                          ORG    H'450#'   START OF PROGRAM
45  0450 0450  77 08                    BBIN PPSL   WC        ARITHMETIC+ROTATE WITH CARRY
46  0452        20                           EORZ   R#        CLEAR R#
47  0453        CC 06 00                     STRA,R# BINN      CLEAR BINARY REGISTERS
48  0456        CC 06 01                     STRA,R# BINN+1
49  0459        05 0A                        LODI,R1 LEN       INDEX FOR BASE DIGITS
50  045B 045B  75 01                    LOOP CPSL   C         CLEAR CARRY
51  045D        0F 06 04                     LODA,R3 BCDD+2    LOAD LS BCD DIGIT IN R3
52  0460        47 0F                        ANDI,R3 H'0F'     CLEAR MS 4 BITS
53  0462        18 11                        BCTR,Z  NEXT      IF ZERO GO TO NEXT
54  0464 0464  06 02                    LOC1 LODI,R2 2         LOAD INDEX
55  0466 0466  0E 46 00                 LOC2 LODA,R# BINN,R2,-
56  0469        8D 46 05                     ADDA,R# BASE,R1,-  ADD BASE DIGIT TO BIN. NUMBER
57  046C        CE 66 00                     STRA,R# BINN,R2
58  046F        5A 75                        BRNR,R2 LOC2
59  0471        85 02                        ADDI,R1 2         RESTORE BASE POINTER
60  0473        FB 6F                        BDRR,R3 LOC1      IF NOT READY RETURN TO LOC1
61                                      *
62  0475 0475  06 04                    NEXT LODI,R2 4         BIT COUNT
63  0477 0477  07 FD                    LP2  LODI,R3 -3        INDEX FOR BYTE COUNT
64  0479 0479  0F 65 05                 LP1  LODA,R# BCDD-256+3,R3 BCD DIGIT INTO R#
65  047C        50                           RRR,R#            CARRY (PREVIOUS LS BIT) -) MSB
66  047D        CF 65 05                     STRA,R# BCDD-256+3,R3     AND LS BIT -) CARRY.
67  0480        DB 77                        BIRR,R3 LP1       NEXT BCDD BYTE
68  0482        75 01                        CPSL   C
69  0484        FA 71                        BDRR,R2 LP2       NEXT SHIFT OF BCD REG. BIT
70  0486        F9 00                        BDRR,R1 $+2       UPDATE BASE POINTER WITHOUT
71  0488        F9 51                        BDRR,R1 LOOP        AFFECTING C FLAG IN PSL AND
72                                      *                             GO TO LOOP IF NOT READY
73  048A        F4 08                        TMI,R#  SIGN
74  048C        98 0D                        BCFR,ON EXIT      IF SIGN POS. THEN READY.
75  048E 048E  77 01                    COMP PPSL   C         CLEAR BORROW
76  0490        06 02                        LODI,R2 2         NUMBER OF DIGITS
77  0492 0492  20                       LP3  EORZ   R#        TWO'S COMPLEMENT BY
78  0493        AE 46 00                     SUBA,R# BINN,R2,- SUBTRACTION FROM ZERO
79  0496        CE 66 00                     STRA,R# BINN,R2
80  0499        5A 77                        BRNR,R2 LP3
81                                      *
82  049B 049B  40                       EXIT HALT             END OF CONVERSION
83                                          END
```

FIGURE 3-2 Program Listing for Signed BCD-to-Binary Conversion

4. SIGNED BCD-TO-BINARY CONVERSION 2

FUNCTION:

Converts a five-digit signed BCD number to a sixteen-bit signed binary number.

A multiplication method is used.

PARAMETERS:

Input: BCDD, BCDD+1 contain the BCD number.
BCDD contains the sign plus the most-significant BCD digit.
The range of BCD numbers is: -32768<BCD Number<+32767

Output: BINN, BINN+1 contain the signed binary number.
BINN is the most-significant byte.

Refer to figures 4.1 and 4.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0 X	R1 X	R2 X	R3 X	R1'	R2'	R3'
PSU	F	II	SP				
PSL	CC X	IDC X	RS	WC X	OVF X	COM	C X

RAM REQUIRED (BYTES): 6

ROM REQUIRED (BYTES): 87

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

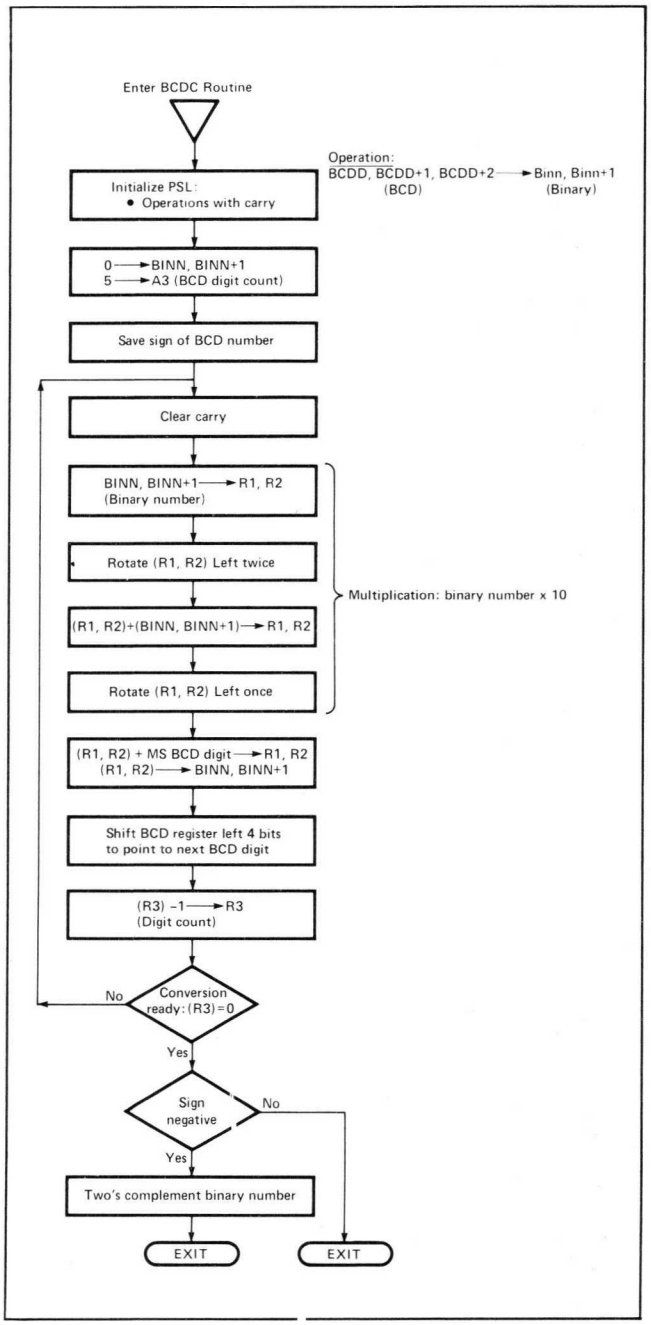

FIGURE 4-1: Flowchart for signed BCD-to-Binary Conversion (Multiplication Method).

```
1                                       *   PD76@@53
2                                       ++++++++++++++++++++++++++++++++++++++++++++
3                                       *  BCD TO BINARY CONVERSION              +
4                                       ++++++++++++++++++++++++++++++++++++++++++++
5                                       *
6                                       * THIS ROUTINE CONVERTS A SIGNED BCD NUMBER
7                                       * (24 BITS: SIGN  + 5 BCD DIGITS) INTO A SIGNED
8                                       * BINARY NUMBER (16 BITS).
9                                       *   -32768 < BCD NUMBER < +32767
10                                      *     BCD NUMBER IS LOST AFTER CONVERSION
11                                      *
12                                      * PRINCIPLE:
13                                      * BIN.=(((((((A*1@) +B) *1@) +C) *1@) +D) *1@) +E
14                                      *  ABCDE= BCD NUMBER
15                                      *
16                                      * MULTIPLICATION BY 1@ IS DONE BY:
17                                      *  LOAD R2,R1 WITH BIN. NUMBER, SHIFT LEFT TWICE,
18                                      *   ADD BIN. NUMBER TO R2,R1, SHIFT LEFT ONCE,
19                                      *    STORE R2,R1 IN BINN,BINN+1 AS RESULT
20                                      *
21                                      * DEFINITION OF SYMBOLS:
22      @@@@                            R@   EQU    @          PROCESSOR-REGISTERS
23      @@@1                            R1   EQU    1
24      @@@2                            R2   EQU    2
25      @@@3                            R3   EQU    3
26      @@@8                            WC   EQU    H'@8'      PSL: 1=WITH, @=WITHOUT CARRY
27      @@@1                            C    EQU    H'@1'           CARRY:BORROW
28      @@@2                            N    EQU    2          COND: NEGATIVE
29      @@@5                            NUM  EQU    5          INDEX FOR NUMBER OF BCD DIGITS
30                                      *
31                                          ORG    H'6@@'
32                                      *
33      @6@@                            BINN RES    2          BINARY NUMBER
34      @6@2                            BCDD RES    3          BCD NUMBER AND SIGN
35      @6@5                            SIGN RES    1          SAVE SIGN DIGIT
36                                      *
38                                      *
39                                          ORG    H'45@'     START OF PROGRAM
40                                      *
41  @45@  @45@  77 @8                   BCDC PPSL   WC         ARITH.:ROTATE WITH CARRY
42  @452        2@                          EORZ   R@         CLEAR R@
43  @453        CC @6 @@                    STRA,R@ BINN      CLEAR BINARY NUMBERS
44  @456        CC @6 @1                    STRA,R@ BINN+1
45  @459        @7 @5                       LODI,R3 NUM       BCD INDEX REGISTER
46                                      *
47  @45B        @C @6 @2                    LODA,R@ BCDD      SAVE SIGN OF BCD NUMBER IN
48  @45E        CC @6 @5                    STRA,R@ SIGN      MEMORY LOC. SIGN
49                                      *
50                                      *                     MULTIPLY BINARY NUMBER BY 1@
51  @461  @461  75 @1                   LOOP CPSL   C         CLEAR CARRY
52  @463        @D @6 @@                    LODA,R1 BINN      LOAD BIN. NUMBER IN R1,R2
53  @466        @E @6 @1                    LODA,R2 BINN+1
54  @469        D2                          RRL,R2            ROTATE REGISTERS R1,R2 LEFT 2
55  @46A        D1                          RRL,R1
56  @46B        D2                          RRL,R2
57  @46C        D1                          RRL,R1
58  @46D        8E @6 @1                    ADDA,R2 BINN+1    ADD BIN. NUMBER TO R1,R2
59  @47@        8D @6 @@                    ADDA,R1 BINN
60  @473        D2                          RRL,R2            SHIFT R1,R2 LEFT ONCE
61  @474        D1                          RRL,R1
62                                      *
63                                      *
64  @475        @C @6 @2                    LODA,R@ BCDD      LOAD MS BCD DIGIT IN R@
65  @478        44 @F                       ANDI,R@ H'@F'     CLEAR MS 4 BITS
66  @47A        82                          ADDZ   R2         ADD BCD TO BINARY NUMBER
67  @47B        85 @@                       ADDI,R1 @         ADD CARRY TO MS BYTE
68  @47D        CD @6 @@                    STRA,R1 BINN      STORE RESULT IN BINN,BINN+1
69  @48@        CC @6 @1                    STRA,R@ BINN+1
70                                      *                     ROTATE BCD NUMBER 4 TIMES LEFT
71                                      *                      TO POINT TO NEXT BCD DIGIT
72  @483        @5 @4                       LODI,R1 4         BIT COUNT
73  @485  @485  @6 @3                   LP2  LODI,R2 3         INDEX FOR BYTE COUNT
74  @487  @487  @E 46 @2                LP1  LODA,R@ BCDD,R2,-
75  @48A        D@                          RRL,R@            SHIFT BCD BYTE LEFT
76  @48B        CE 66 @2                    STRA,R@ BCDD,R2
77  @48E        5A 77                       BRNR,R2 LP1       NEXT BYTE OF BCD REGISTER
78  @49@        F9 73                       BDRR,R1 LP2       NEXT BIT SHIFT
79  @492        FB 4D                       BDRR,R3 LOOP      TO LOOP IF MULTIPLY NOT READY
80                                      *
81  @494        @C @6 @5                    LODA,R@ SIGN
82  @497        9A @D                       BCFR,N  EXIT      IF SIGN POS. THEN READY
83  @499        77 @1                       PPSL   C          CLEAR CARRY
84  @49B        @6 @2                       LODI,R2 2         INDEX LOADING
85  @49D  @49D  2@                      LP3  EORZ   R@         TWO'S COMPLEMENT BY
86  @49E        AE 66 @@                    SUBA,R@ BINN,R2   SUBTRACTING FROM ZERO
87  @4A1        CC @6 @@                    STRA,R@ BINN
88  @4A4        5A 77                       BRNR,R2 LP3
89                                      *
90  @4A6  @4A6  4@                      EXIT HALT   END OF CONVERSION
91                                          END
```

FIGURE 4-2 Program Listing for Signed BCD-to-Binary Conversion

5. SIGNED BCD-TO-ASCII CONVERSION

FUNCTION:

Converts n BCD digits plus sign to n + 1 ASCII characters (sign included).

PARAMETERS:

Input: BCDD, BCDD+1, ------ BCDD+ (numb – 1)
BCDD contains the sign plus the most-significant digit (2 BCD digits/byte).
Numb is the number of BCD bytes.

Output: ASCI, ASCI+1, ------, ASCI + (num – 1) contains the signed result.
ASCI contains the sign.
ASCI+1 contains the most-significant byte.

Refer to figures 5.1 and 5.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0 X	R1 X	R2 X	R3 X	R1'	R2'	R3'
PSU	F	II	SP				
PSL	CC X	IDC X	RS	WC X	OVF X	COM	C X

RAM REQUIRED (BYTES): N Numb, N Num+1

ROM REQUIRED (BYTES): 53

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

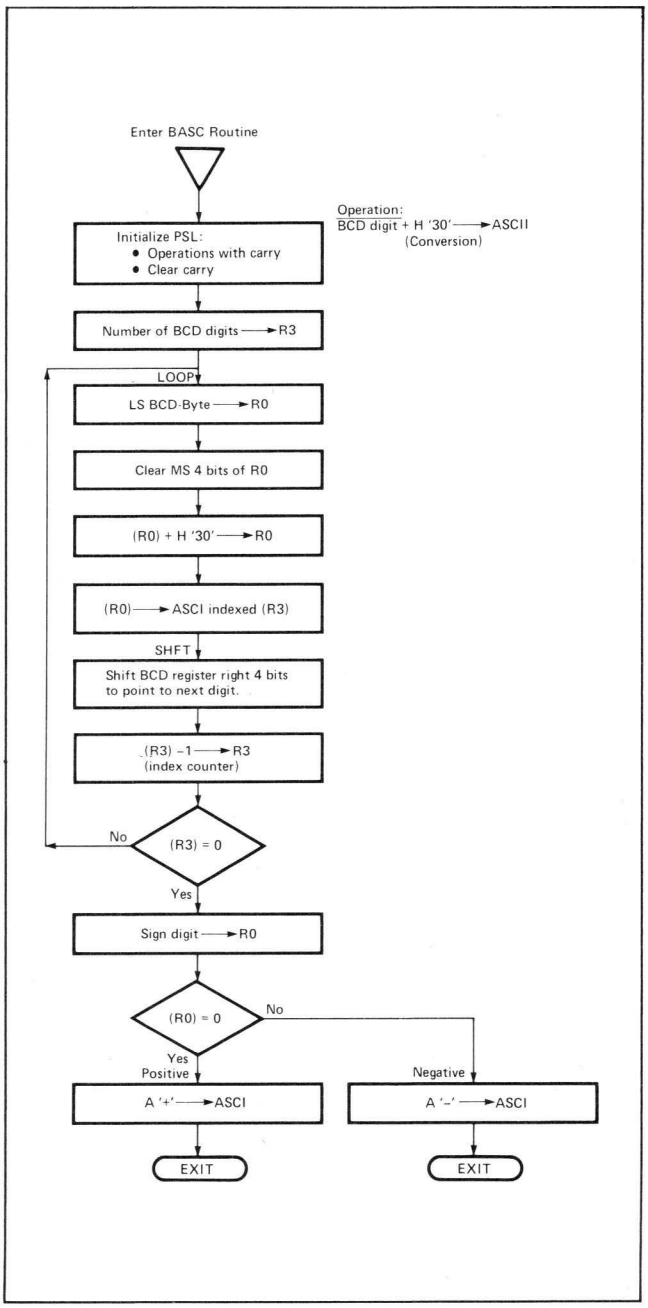

FIGURE 5-1 Flowchart for BCD-to-ASCII Conversion (signed)

```
1                                       *
2                                       *    PD7600S4
3                                       *++++++++++++++++++++++++++++++++++++++++++++
4                                       *  BCD TO ASCII CONVERSION                  +
5                                       *++++++++++++++++++++++++++++++++++++++++++++
6                                       *
7                                       * THIS ROUTINE CONVERTS A SIGNED BCD NUMBER
8                                       * INTO ASCII CHARACTERS (SIGN INCLUDED).
9                                       * BCD FORMAT: SIGN + BCD DIGITS (TWO DIGITS:BYTES)
10                                      * THE NUMBER OF BCD DIGITS -> R3 = NUM
11                                      * THE NUMBER OF BCD BYTES -> R2 = NUMB
12                                      * BCD NUMBER IS IN BCDD,BCDD+1,---,BCDD+(N-1)
13                                      * ASCII CHARACTERS ARE IN ASCII,ASCII+1,---,ASCII+NUM
14                                      *         (SIGN) (BCD DIGITS)
15                                      *
16                                      * DEFINITIONS OF SYMBOLS:
17                                      *
18      0000                            R0  EQU     0
19      0001                            R1   EQU     1
20      0002                            R2   EQU     2
21      0003                            R3   EQU     3
22      0008                            WC   EQU     H'08'      PSL: 1=WITH, 0=WITHOUT CARRY
23      0001                            C    EQU     H'01'           CARRY:BORROW
24      0003                            UN   EQU     3          COND: UNCONDITIONAL
25      0000                            Z    EQU     0               ZERO
26                                      *
27                                      * IN THIS EXAMPLE THE CONVERSION OF 5 BCD DIGITS
28                                      *   IS PERFORMED.
29                                      *
30      0003                            NUMB  EQU     3          NUMBER OF BCD BYTES
31      0005                            NUM  EQU     5          NUMBER OF BCD DIGITS
32                                      *
34                                      *
35                                          ORG     H'4E0'
36                                      *
37      04E0                            BCDD RES     NUMB       RESERVE FOR BCD NUMBER
38      04E3                            ASCI RES     NUM+1      RESERVE FOR SIGN,ASCII DIGITS
39                                      *
40                                          ORG     H'500'     PROGRAM START HERE
41                                      *
42  0500  0500  77 08                  BASC PPSL    WC         ARITHMETIC:ROTATE WITH CARRY
43  0502        75 01                       CPSL    C          CLEAR CARRY
44  0504        07 05                       LODI,R3 NUM        INDEX REGISTER
45                                      *
46  0506  0506  0C 04 E2               LOOP LODA,R0 BCDD+NUMB-1 LOAD LS BCD DIGIT IN R0
47  0509        44 0F                       ANDI,R0 H'0F'      CLEAR MS 4 BITS
48  050B        84 30                       ADDI,R0 H'30'      ASCII CHARACTER
49  050D        CF 64 E3                    STRA,R0 ASCI,R3    STORE ASCII CHARACTER
50                                      *
51  0510  0510  05 04                   SHFT LODI,R1 4          BIT COUNT
52  0512  0512  06 FD                   LP2  LODI,R2 -NUMB      INDEX FOR BYTE SHIFT
53  0514  0514  0E 63 E3                LP1  LODA,R0 BCDD-256+NUMB,R2
54  0517        50                          RRR,R0             CARRY (PREVIOUS LS BIT) -> MSB
55  0518        CE 63 E3                    STRA,R0 BCDD-256+NUMB,R2   AND LS BIT ->CARRY
56  051B        DA 77                       BIRR,R2 LP1
57  051D        75 01                       CPSL    C          CLEAR CARRY
58  051F        F9 71                       BDRR,R1 LP2
59  0521        FB 63                       BDRR,R3 LOOP       IF NOT READY GO TO LOOP
60                                      *
61  0523  0523  0C 04 E2               SIGN LODA,R0 BCDD+NUMB-1   SIGN -> R0
62  0526        18 07                       BCTR,Z  POS
63  0528  0528  04 2D                  NEG  LODI,R0 A'-'
64  052A        CC 04 E3                    STRA,R0 ASCI
65  052D        1B 05                       BCTR,UN EXIT
66  052F  052F  04 2B                  POS  LODI,R0 A'+'
67  0531        CC 04 E3                    STRA,R0 ASCI
68                                      *
69  0534  0534  40                      EXIT HALT    END OF CONVERSION
70                                          END
```

FIGURE 5-2 Program Listing for BCD-to-ASCII Conversion (Signed)

6. ASCII-TO-BCD CONVERSION

FUNCTION:

Converts n ASCII digits to n BCD digits.
ASCII ⟶ BCD

PARAMETERS:

Input: ADIG, ADIG+1, ----, ADIG+(n − 1) contain ASCII digits.
The most-significant digit is in ADIG (byte/digit).

Output: BCDD, BCDD+1, ----, BCDD + (n−1) contains BCD digits.
The most-significant digit is in BCDD (2 digits/byte).

Refer to figures 6.1 and 6.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0	R1	R2	R3	R1'	R2'	R3'
	X	X	X	X			
PSU	F	II	SP				
PSL	CC	IDC	RS	WC	OVF	COM	C
	X	X		X	X		X

RAM REQUIRED (BYTES): nADIG + nBCDD

ROM REQUIRED (BYTES): 37

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

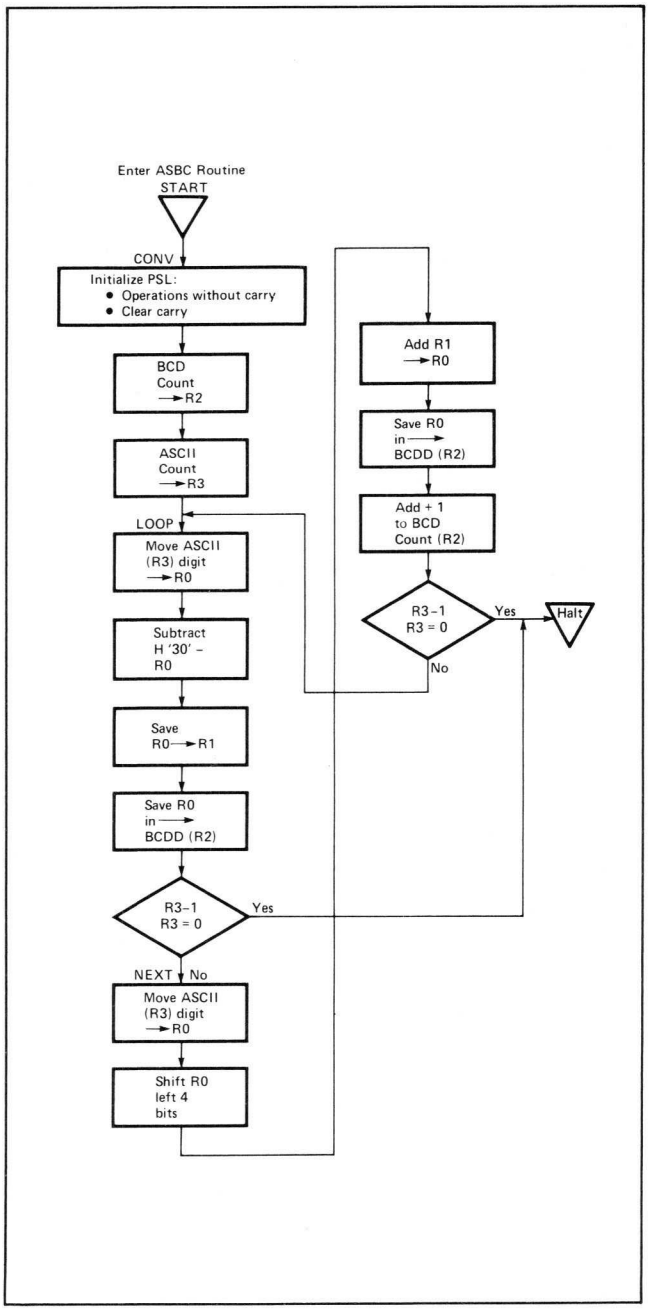

FIGURE 6-1 Flowchart for ASCII-to-BCD Conversion

```
1                       *    PD76ØØ55
2                       ++++++++++++++++++++++++++++++++++++++++++++
3                       *   ASCII TO BCD CONVERSION                +
4                       ++++++++++++++++++++++++++++++++++++++++++++
5                       *
6                       * THIS ROUTINE CONVERTS A STRING OF ASCII
7                       * DIGITS TO A STRING OF BCD DIGITS.
8                       *
9                       * ADIG IS MS DIGIT ASCII
10                      * BCDD IS MS DIGIT BCD
11                      *
12                      * DEFINITIONS OF SYMBOLS:
13 ØØØØ                 R0    EQU     Ø          PROCESSOR-REGISTERS
14 ØØØ1                 R1    EQU     1
15 ØØØ2                 R2    EQU     2
16 ØØØ3                 R3    EQU     3
17 ØØØ8                 WC    EQU     H'Ø8'      PSL: 1-WITH, Ø-WITHOUT
18 ØØØ1                 C     EQU     H'Ø1'           CARRY:BORROW
19 ØØØ3                 UN    EQU     3          BR.COND: ALWAYS
20                      *
21                      * IN THIS EXAMPLE THE CONVERSION OF 5
22                      * ASCII CHARACTERS IS PERFORMED.
23                      *
24 ØØØ5                 NUM   EQU     5
25 ØØØ3                 NUM1  EQU     3
26                      *
28                            ORG     H'75Ø'     RAM DEFINITIONS
29 Ø75Ø                 ADIG  RES     NUM        ASCII BYTES RESERVED
30 ØØØ5                 ACNT  EQU     $-ADIG     ASCII DIGIT COUNT
31 Ø755                 BCDD  RES     NUM1       BCD BYTES RESERVED
32 ØØØ3                 BCNT  EQU     $-BCDD     BCD BYTE COUNT
33                      *
34                            ORG     H'5ØØ'     START OF SUBROUTINE
35 Ø5ØØ 75 Ø9           CONV  CPSL    WC+C       ARITH.WITHOUT:NO CARRY
36 Ø5Ø2 Ø6 Ø3                 LODI,R2 BCNT       BCD COUNT -> R2
37 Ø5Ø4 Ø7 Ø5                 LODI,R3 ACNT       ASCII COUNT ->R3
38 Ø5Ø6 ØF 67 4F        LOOP  LODA,RØ ADIG-1,R3  RØ HAS ASCII DIGIT
39 Ø5Ø9 A4 3Ø                 SUBI,RØ H'3Ø'      MAKE IT BCD
40 Ø5ØB C1                    STRZ    R1         RØ ->R1
41 Ø5ØC CE 67 54              STRA,RØ BCDD-1,R2  SAVE 1 BCD DIGIT
42 Ø5ØF FB Ø3                 BDRR,R3 NEXT       DECREMENT -NON ZERO BR
43 Ø511 1F Ø5 25              BCTA,UN BYE        CONVERSION COMPLETE
44 Ø514 ØF 67 4F        NEXT  LODA,RØ ADIG-1,R3  NEXT ASCII DIGIT
45 Ø517 A4 3Ø                 SUBI,RØ H'3Ø'      MAKE IT BCD
46 Ø519 DØ                    RRL,RØ             SHIFT LEFT 4 BITS
47 Ø51A DØ                    RRL,RØ
48 Ø51B DØ                    RRL,RØ
49 Ø51C DØ                    RRL,RØ
50 Ø51D 61                    IORZ    R1         INCLUSIVE OR LOW ORDER
51 Ø51E CE 67 54              STRA,RØ BCDD-1,R2  STORE 2 BCD DIGITS
52 Ø521 FA ØØ                 BDRR,R2 $+2        DECREMENT BCD COUNT
53 Ø523 FB 61                 BDRR,R3 LOOP       DECREMENT-NON ZERO BR.
54 Ø525 4Ø              BYE   HALT               END OF ASCII -> BCD
55                            END
```

FIGURE 6-2 Program Listing for ASCII-to-BCD Conversion

7. HEXADECIMAL-TO-ASCII CONVERSION

FUNCTION:

Converts a string of hexadecimal digits to a string of ASCII digits.

PARAMETERS:

Input: HEX, HEX+1, ----, HEX + (n – 1)
HEX is the most-significant digit (2 Hex. digit/byte).

Output: ASCI, ASCI+1, ----, ASCI + (n – 1)
ASCI is the most-significant digit.

Refer to figures 7.1 and 7.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0 X	R1 X	R2 X	R3 X	R1′	R2′	R3′
PSU	F	II	SP				
PSL	CC X	IDC X	RS	WC X	OVF X	COM	C X

RAM REQUIRED (BYTES): nHEX + nASCI

ROM REQUIRED (BYTES): 59

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 0

ASSEMBLER/COMPILER USED: PIPHASM

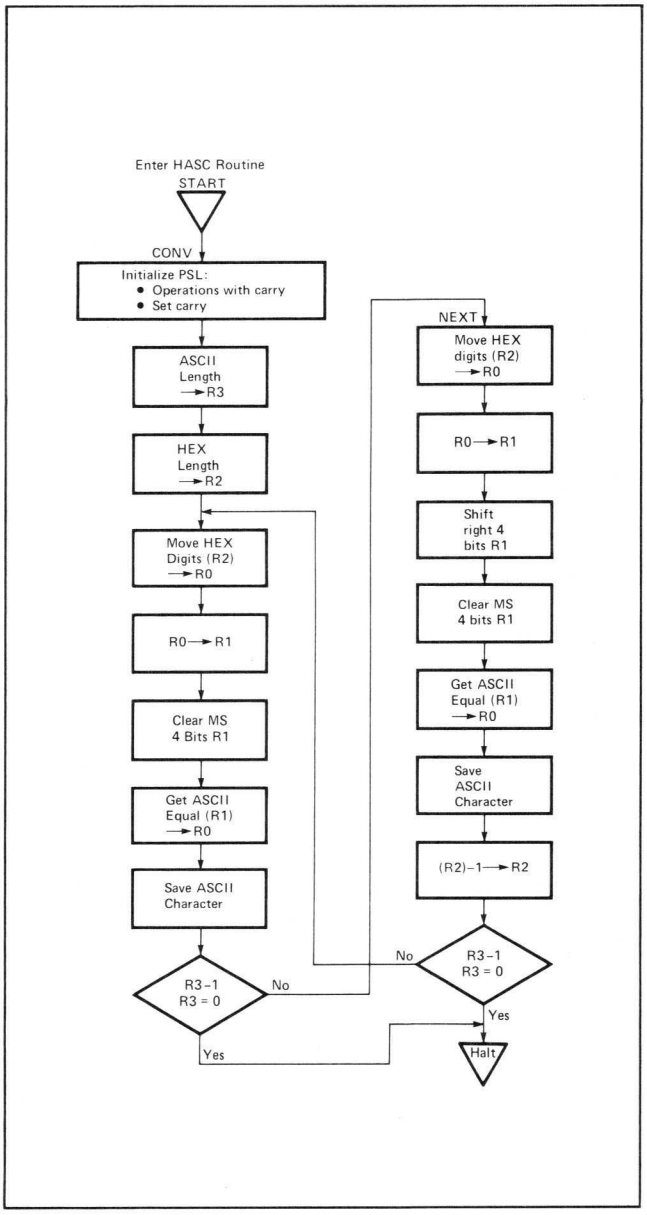

FIGURE 7-1 Flowchart for Hexadecimal-to-ASCII Conversion

```
1                       *    PD76Ø056
2                       ++++++++++++++++++++++++++++++++++++++++
3                       *  HEXIDECIMAL TO ASCII CONVERSION      +
4                       ++++++++++++++++++++++++++++++++++++++++
5                       *
6                       * THIS ROUTINE CONVERTS A STRING OF ASCII
7                       * DIGITS TO A STRING OF HEX DIGITS.
8                       *
9                       * ASCI IS MS DIGIT ASCII.
10                      * HEX IS MS DIGIT HEXIDECIMAL.
11                      *
12                      * DEFINITION OF SYMBOLS:
13 ØØØØ                 RØ   EQU     Ø          PROCESSOR-REGISTER
14 ØØØ1                 R1   EQU     1
15 ØØØ2                 R2   EQU     2
16 ØØØ3                 R3   EQU     3
17 ØØØ8                 WC   EQU     H'Ø8'      ARITHMETIC CARRY
18 ØØØ1                 C    EQU     H'Ø1'      CARRY:BORROW
19 ØØØ3                 UN   EQU     3          UNCOND. BRANCH
20                      *
21                      * IN THIS EXAMPLE 3 HEXIDECIMAL
22                      * CHARACTERS ARE CONVERTED.
23 ØØØ2                 NUM  EQU     2          HEX BYTE COUNT
24 ØØØ3                 NUM1 EQU     3          ASCII BYTE COUNT
25                      *
27                           ORG     H'6ØØ'     RAM DEFINITIONS
28 Ø6ØØ                 HEX  RES     NUM        RESERVES HEX BYTES
29 ØØØ2                 HLEN EQU     $-HEX      LENGTH OF HEX
30 Ø6Ø2                 ASCI RES     NUM1       RESERVES ASCII BYTES
31 ØØØ3                 ALEN EQU     $-ASCI     LENGTH OF ASCII
32                      *
33                           ORG     H'5ØØ'     START OF ROUTINE
34 Ø5ØØ 77 Ø9           CONV PPSL    WC+C       ARITH.WITH, SET CARRY
35 Ø5Ø2 Ø7 Ø3                LODI,R3 ALEN       R3= ASCII LENGTH
36 Ø5Ø4 Ø6 Ø2                LODI,R2 HLEN       R2= HEX LENGTH
37 Ø5Ø6 ØE 65 FF        CHEX LODA,RØ HEX-1,R2   GET HEX DIGITS
38 Ø5Ø9 C1                   STRZ    R1         RØ ->R1
39 Ø5ØA 45 ØF                ANDI,R1 H'ØF'      CLEAR MS 4 BITS
40 Ø5ØC ØD 65 2C             LODA,RØ ANSI,R1    LOAD ASCII CORRESPONDI
41 Ø5ØF CF 66 Ø1             STRA,RØ ASCI-1,R3  SAVE IT
42 Ø512 FB Ø3                BDRR,R3 NEXT       R3-1, R3() BRANCH
43 Ø514 1F Ø5 2B             BCTA,UN BYE        END OF CONVERSION
44 Ø517 ØE 65 FF        NEXT LODA,RØ HEX-1,R2   GET HEX DIGITS
45 Ø51A C1                   STRZ    R1         RØ -) R1
46 Ø51B 51                   RRR,R1             SHIFT RIGHT 4 BITS
47 Ø51C 51                   RRR,R1
48 Ø51D 51                   RRR,R1
49 Ø51E 51                   RRR,R1
50 Ø51F 45 ØF                ANDI,R1 H'ØF'      CLEAR MS 4 BITS
51 Ø521 ØD 65 2C             LODA,RØ ANSI,R1    LOAD ASCII CORRESPONDI
52 Ø524 CF 66 Ø1             STRA,RØ ASCI-1,R3  SAVE IT
53 Ø527 FA ØØ                BDRR,R2 $+2        R2 - 1 CONT.
54 Ø529 FB 5B                BDRR,R3 CHEX       R3-1, R3() BRANCH
55                      *
56 Ø52B 4Ø              BYE  HALT               END OF CONVERSION
57                      *
58 Ø52C 3Ø 31 32 33     ANSI DATA    A'Ø123456789ABCDEF'
       34 35 36 37
       38 39 41 42
       43 44 45 46
59                      *
60                           END
```

FIGURE 7-2 Program Listing for Hexadecimal-to-ASCII Conversion

8. ASCII-TO-HEXADECIMAL CONVERSION

FUNCTION:

Converts a string of ASCII digits to a string of hexadecimal digits. The conversion is done by table look-up. Non-numeric ASCII halts this routine. It may be changed to report non-numeric.

PARAMETERS:

Input: ASCI, ASCI+1, ----, ASCI + (n – 1)
ASCI is the most-significant digit.

Output: HEX, HEX+1, ----, HEX + (n – 1)
HEX is the most-significant digit (2 Hex. digits/byte)

Refer to figures 8.1 and 8.2 for flowchart and program listing.

	HARDWARE AFFECTED						
REGISTERS	R0	R1	R2	R3	R1'	R2'	R3'
	X	X	X	X			
PSU	F	II	SP				
PSL	CC	IDC	RS	WC	OVF	COM	C
	X	X		X	X		X

RAM REQUIRED (BYTES): nASCI + nHEX

ROM REQUIRED (BYTES): 68

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 1

ASSEMBLER/COMPILER USED: PIPHASM

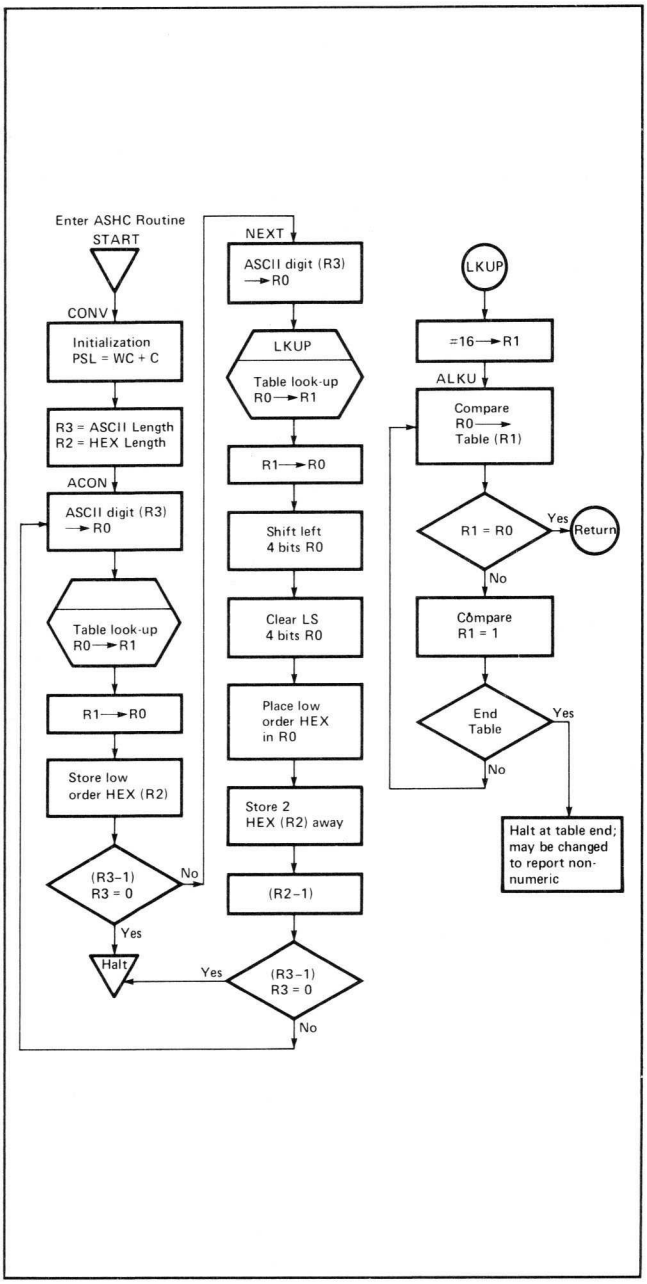

FIGURE 8-1 Flowchart for ASCII-to-Hexadecimal Conversion

```
1                       *    PD7600S7
2                       ++++++++++++++++++++++++++++++++++++++++
3                       *   ASCII TO HEX CONVERSION           +
4                       ++++++++++++++++++++++++++++++++++++++++
5                       *
6                       * THIS ROUTINE CONVERTS A STRING OF ASCII
7                       * DIGITS TO A STRING OF HEXIDECIMAL DIGITS
8                       * ASCI IS MS DIGIT ASCII
9                       * HEX IS MS DIGIT HEXIDECIMAL
10                      * CONVERSION DONE BY TABLE LOOKUP
11                      * NON NUMERIC ASCII HALT ROUTINE
12                      *
13                      * DEFINITION OF SYMBOLS:
14 0000                 R0   EQU     0          REGISTER-PROCESSOR
15 0001                 R1   EQU     1
16 0002                 R2   EQU     2
17 0003                 R3   EQU     3
18 0008                 WC   EQU     H'08'      ARITHMETIC CARRY
19 0001                 C    EQU     H'01'      CARRY:BORROW
20 0003                 UN   EQU     3          BRANCH UNCOND.
21 0002                 LT   EQU     2                LESS THAN
22 0000                 EQ   EQU     0                EQUAL
23                      *
24                      * IN THIS EXAMPLE 3 ASCII DIGITS
25                      * ARE CONVERTED TO HEXIDECIMAL
26                      *
27 0003                 NUM  EQU     3          ASCII BYTE COUNT
28 0002                 NUM1 EQU     2          HEX BYTE COUNT
29                      *
31                      *
32                           ORG     H'6000'    RAM DEFINITIONS
33 0600                 ASCI RES     NUM        RESERVED ASCII BYTES
34 0003                 ALEN EQU     $-ASCI     LENGTH OF ASCII
35 0603                 HEX  RES     NUM1       RESERVED HEX BYTES
36 0002                 HLEN EQU     $-HEX      LENGTH OF HEX
37                      *
38                           ORG     H'5000'    START OF ROUTINE
39 0500 77 09           CONV PPSL    WC+C       ARITH.WITH + CARRY SET
40 0502 07 03                LODI,R3 ALEN       R3 = ASCII LENGTH
41 0504 06 02                LODI,R2 HLEN       R2 = HEX LENGTH
42 0506 0F 65 FF        ACON LODA,R0 ASCI-1,R3  GET ASCII DIGIT
43 0509 3B 08                BSTR,UN LKUP       LOOKUP SUBROUTINE
44 050B 01                   LODZ    R1         R1 -> R0
45 050C CE 66 02             STRA,R0 HEX-1,R2   SAVE HEX CORRESPONDING
46 050F FB 1D                BDRR,R3 NEXT       (R3-1), R3 <> BRANCH
47 0511 1B 31                BCTR,UN BYE        END OF CONVERSION
48                      *
49 0513 05 10           LKUP LODI,R1 16         LOOP CONSTANT
50 0515 ED 45 1E        ALKU COMA,R0 ANSI,R1,-  COMPARE TO TABLE
51 0518 14                   RETC,EQ            RETURN - MATCH FOUND
52 0519 E5 01                COMI,R1 1          TEST END OF TABLE
53 051B 9A 78                BCFR,LT ALKU       NO- LOOK AGAIN
54 051D 40                   HALT               ERROR - NON NUMERIC HE
55                      *
56 051E 30 31 32 33     ANSI DATA    A'0123456789ABCDEF'
          34 35 36 37
          38 39 41 42
          43 44 45 46
57                      *
58 052E 0F 65 FF        NEXT LODA,R0 ASCI-1,R3  GET NEXT ASCII DIGIT
59 0531 3B 60                BSTR,UN LKUP       LOOK UP SUBROUTINE
60 0533 01                   LODZ    R1         R1 -> R0
61 0534 D0                   RRL,R0             SHIFT LEFT 4 BITS
62 0535 D0                   RRL,R0
63 0536 D0                   RRL,R0
64 0537 D0                   RRL,R0
65 0538 44 F0                ANDI,R0 H'F0'      CLEAR LS 4 BITS
66 053A 6E 66 02             IORA,R0 HEX-1,R2   COMBINE LOW ORDER
67 053D CE 66 02             STRA,R0 HEX-1,R2   SAVE 2 HEX DIGITS
68 0540 FA 00                BDRR,R2 $+2        (R2-1), CONTIUNE
69 0542 FB 42                BDRR,R3 ACON       (R3-1), R3 <> BRANCH
70                      *
71 0544 40              BYE  HALT               END OF CONVERSION
72                           END
```

FIGURE 8-2 Program Listing for ASCII-to-Hexadecimal Conversion

Signetics 2650 Microprocessor application memos currently available:

AS50	Serial Input/Output
AS51	Bit and Byte Testing Procedures
AS52	General Delay Routines
AS53	Binary Arithmetic Routines
AS54	Conversion Routines
SP50	2650 Evaluation Printed Circuit Board Level System (PC1001)
SP51	2650 Demo Systems
SP52	Support Software for use with NCSS Timesharing System
SP53	Simulator, Version 1.2
SP54	Support Software for use with the General Electric Mark III Timesharing System
SS50	PIPBUG
SS51	Absolute Object Format (Revision 1)
MP51	2650 Initialization
MP52	Low Cost Clock Generator Circuits
MP53	Address and Data Bus Interfacing Techniques