MARCH 10, 1983

GEAR FOR BYPASSING LOCAL PHONE SERVICES LURES BIG USERS/93

Software verifies computer maintenance program/144

Work station boasts high-level interactive graphics/129

SIX DOLLARS A Mc GRAW-HILL PUBLICATION

Electronics®

SPECIAL REPORT

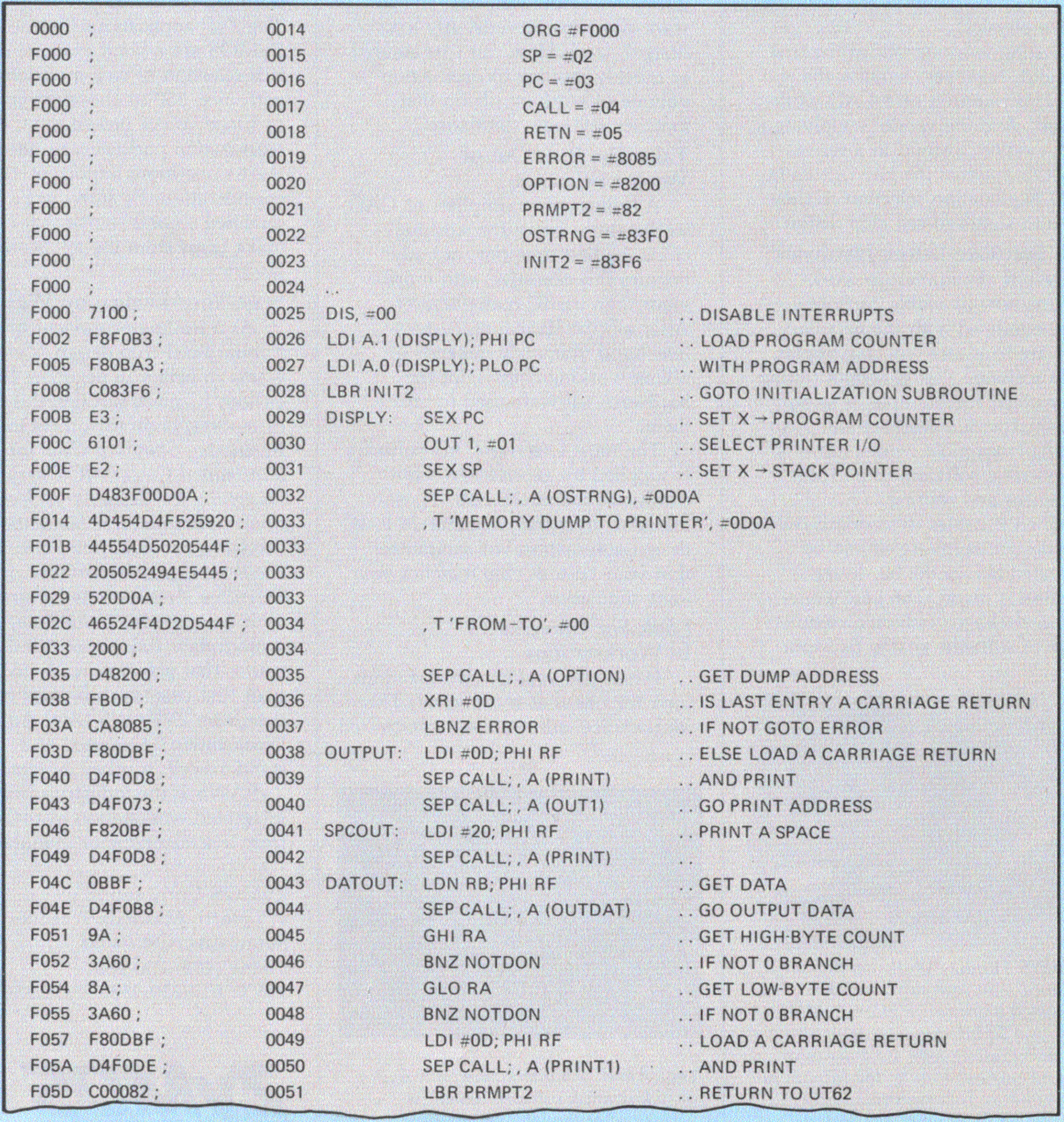
Self-testing the untestable

Program displays contents of memory on printer

by David V. Fansler
Biomedical Reference Laboratories Inc., Burlington, N. C.

RCA's CDP18S694 microcomputer development system—a low-cost solution when programs in Basic or assembly language need to be created—may be linked with the proper interface to Centronics' compatible parallel interface printer for hard-copy output. With this program, an operator can use the printer to print out the contents of the development system's memory in the format employed by the system's built-in monitor UT-62.

MEMORY DISPLAY PROGRAM FOR RCA's CDP18S694 DEVELOPMENT SYSTEM



```
0000  ;                                     ORG =F000
F000  ;                                     SP = =02
F000  ;                                     PC = =03
F000  ;                                     CALL = =04
F000  ;                                     RETN = =05
F000  ;                                     ERROR = =8085
F000  ;                                     OPTION = =8200
F000  ;                                     PRMPT2 = =82
F000  ;                                     OSTRNG = =83F0
F000  ;                                     INIT2 = =83F6
F000  ;                 ..
F000  7100 ;            DIS, =00                                  .. DISABLE INTERRUPTS
F002  F8F0B3 ;          LDI A.1 (DISPLY); PHI PC                  .. LOAD PROGRAM COUNTER
F005  F80BA3 ;          LDI A.0 (DISPLY); PLO PC                  .. WITH PROGRAM ADDRESS
F008  C083F6 ;          LBR INIT2                                 .. GOTO INITIALIZATION SUBROUTINE
F00B  E3 ;              DISPLY:   SEX PC                          .. SET X → PROGRAM COUNTER
F00C  6101 ;                      OUT 1, =01                      .. SELECT PRINTER I/O
F00E  E2 ;                        SEX SP                          .. SET X → STACK POINTER
F00F  D483F00D0A ;                SEP CALL; , A (OSTRNG), =0D0A
F014  4D454D4F525920 ;            , T 'MEMORY DUMP TO PRINTER', =0D0A
F01B  44554D5020544F ;
F022  205052494E5445 ;
F029  520D0A ;
F02C  46524F4D2D544F ;            , T 'FROM-TO', =00
F033  2000 ;
F035  D48200 ;                    SEP CALL; , A (OPTION)          .. GET DUMP ADDRESS
F038  FB0D ;                      XRI #0D                         .. IS LAST ENTRY A CARRIAGE RETURN
F03A  CA8085 ;                    LBNZ ERROR                      .. IF NOT GOTO ERROR
F03D  F80DBF ;          OUTPUT:   LDI #0D; PHI RF                 .. ELSE LOAD A CARRIAGE RETURN
F040  D4F0D8 ;                    SEP CALL; , A (PRINT)           .. AND PRINT
F043  D4F073 ;                    SEP CALL; , A (OUT1)            .. GO PRINT ADDRESS
F046  F820BF ;          SPCOUT:   LDI #20; PHI RF                 .. PRINT A SPACE
F049  D4F0D8 ;                    SEP CALL; , A (PRINT)           ..
F04C  0BBF ;            DATOUT:   LDN RB; PHI RF                  .. GET DATA
F04E  D4F0B8 ;                    SEP CALL; , A (OUTDAT)          .. GO OUTPUT DATA
F051  9A ;                        GHI RA                          .. GET HIGH-BYTE COUNT
F052  3A60 ;                      BNZ NOTDON                      .. IF NOT 0 BRANCH
F054  8A ;                        GLO RA                          .. GET LOW-BYTE COUNT
F055  3A60 ;                      BNZ NOTDON                      .. IF NOT 0 BRANCH
F057  F80DBF ;                    LDI #0D; PHI RF                 .. LOAD A CARRIAGE RETURN
F05A  D4F0DE ;                    SEP CALL; , A (PRINT1)          .. AND PRINT
F05D  C00082 ;                    LBR PRMPT2                      .. RETURN TO UT62
```

```
F060  2A ;             0052  NOTDON:  DEC RA                    .. ELSE BYTE COUNT-1
F061  8BFA0F ;         0053           GLO RB; ANI #0F           ... GET LOW-BYTE COUNT
F064  3A6E ;           0054           BNZ SAMELN                .. IF NOT 0 BRANCH
F066  F83BBF ;         0055           LDI #3B; PHI RF           .. ELSE LOAD ';'
F069  D4F0D8 ;         0056           SEP CALL; , A (PRINT)     .. AND PRINT
F06C  303D ;           0057           BR OUTPUT                 .. GOTO OUTPUT
F06E  F6 ;             0058  SAMELN:  SHR                       .. SHIFT BYTE COUNT
F06F  334C ;           0059           BDF DATOUT                .. IF DF = 1 BRANCH
F071  3046 ;           0060           BR SPCOUT                 .. ELSE PRINT A SPACE
F073  9B ;             0061  OUT1:    GHI RB                    .. GET HIGH ADDRESS
F074  F6F6F6F6 ;       0062           SHR; SHR; SHR; SHR        .. MASK LOWER FOUR BITS
F078  FCF6 ;           0063           ADI #F6                   .. AND CONVERT TO
F07A  3B7E ;           0064           BNF TY4                   .. HEXADECIMAL
F07C  FC07 ;           0065           ADI #07                   ..
F07E  FFC6BF ;         0066  TY4:     SMI #C6; PHI RF           .. CONVERT TO ASCII
F081  D4F0D8 ;         0067           SEP CALL; , A (PRINT)     .. AND PRINT
F084  9BFA0F ;         0068           GHI RB; ANI #0F           .. GET HIGH ADDRESS
F087  FCF6C7 ;         0069           ADI #F6; LSNF             .. MASK FOUR HIGH BITS
F08A  FC07 ;           0070           ADI #07                   ..
F08C  FFC6BF ;         0071           SMI #C6; PHI RF           .. CONVERT TO ASCII
F08F  D4F0D8 ;         0072           SEP CALL; , A (PRINT)     .. AND PRINT
F092  8B ;             0073           GLO RB                    .. GET LOW ADDRESS
F093  F6F6F6F6 ;       0074           SHR; SHR; SHR; SHR        .. MASK FOUR LOW BITS
F097  FCF6 ;           0075           ADI #F6                   .. THEN CONVERT TO
F099  3B9D ;           0076           BNF TY5                   .. HEXADECIMAL
F09B  FC07 ;           0077           ADI #07                   ..
F09D  FFC6BF ;         0078  TY5:     SMI #C6; PHI RF           .. CONVERT TO ASCII
F0A0  D4F0D8 ;         0079           SEP CALL; , A (PRINT)     .. AND PRINT
F0A3  8BFA0F ;         0080           GLO RB; ANI #0F           .. GET LOW ADDRESS
F0A6  FCF6C7 ;         0081           ADI #F6; LSNF             .. MASK FOUR HIGH BITS
F0A9  FC07 ;           0082           ADI #07                   .. THEN
F0AB  FFC6BF ;         0083           SMI #C6; PHI RF           .. CONVERT TO ASCII
F0AE  D4F0D8 ;         0084           SEP CALL; , A (PRINT)     .. AND PRINT
F0B1  F820BF ;         0085           LDI #20; PHI RF           .. LOAD A SPACE
F0B4  D4F0D8 ;         0086           SEP CALL; , A (PRINT)     .. AND PRINT
F0B7  D5 ;             0087           SEP RETN                  .. THEN RETURN
F0B8  9F ;             0088  OUTDAT:  GHI RF                    .. GET DATA AND MASK
F0B9  F6F6F6F6 ;       0089           SHR; SHR; SHR; SHR        .. THE FOUR LOW BITS
F0BD  FCF6 ;           0090           ADI #F6                   .. CONVERT TO HEXADECIMAL
F0BF  3BC3 ;           0091           BNF TY6                   .. THEN
F0C1  FC07 ;           0092           ADI #07                   ..
F0C3  FFC6BF ;         0093  TY6:     SMI #C6; PHI RF           .. CONVERT TO ASCII
F0C6  D4F0D8 ;         0094           SEP CALL; , A (PRINT)     .. AND PRINT
F0C9  4BFA0F ;         0095           LDA RB; ANI #0F           .. GET DATA AND MASK
F0CC  FCF6C7 ;         0096           ADI #F6; LSNF             .. THE FOUR HIGH BITS
F0CF  FC07 ;           0097           ADI #07                   .. CONVERT TO HEXADECIMAL
F0D1  FFC6BF ;         0098           SMI #C6; PHI RF           .. CONVERT TO ASCII
F0D4  D4F0D8 ;         0099           SEP CALL; , A (PRINT)     .. AND PRINT
F0D7  D5 ;             0100           SEP RETN                  .. THEN RETURN
F0D8  9F ;             0101  PRINT:   GHI RF                    .. GET DATA
F0D9  FB0A ;           0102           XRI #0A                   .. CHECK FOR LINE FEED
F0DB  32F0 ;           0103           BZ EXITDF                 .. IF SO GOTO EXITDF
F0DD  9F ;             0104           GHI RF                    .. ELSE GET DATA
F0DE  9F ;             0105  PRINT1:  GHI RF                    .. GET DATA AGAIN
F0DF  FBFF ;           0106           XRI #FF                   .. AND INVERT
F0E1  52 ;             0107           STR SP                    .. PLACE ON THE STACK
F0E2  34E2 ;           0108           B1 *                      .. LOOP IF PRINTER BUSY
F0E4  66 ;             0109           OUT 6                     .. OUTPUT DATA
F0E5  22 ;             0110           DEC SP                    .. REPOSITION STACK POINTER
F0E6  9F ;             0111           GHI RF                    .. GET DATA
F0E7  FB0D ;           0112           XRI #0D                   .. CHECK FOR CARRIAGE RETURN
F0E9  3AF0 ;           0113           BNZ EXITDF                .. IF NOT GOTO EXITDF
F0EB  F80ABF ;         0114           LDI #0A; PHI RF           .. IF SO OUTPUT A LINE FEED
F0EE  30DE ;           0115           BR PRINT1                 .. AND PRINT
F0F0  D5 ;             0116  EXITDF:  SEP RETN                  .. AND RETURN
F0F1  ;                0117
F0F1  ;                0118
F0F1  ;                0119  END
0000
```

In addition, UT-62 communicates with the user's terminal, cassette-storage and memory-handling functions and produces a hexadecimal memory display on the terminal.

The development system selects the printer port by issuing a 61_{16} output instruction with group number 01_{16} while output instruction 66_{16} feeds data into the printer. In addition, external flag EF_1 monitors the busy and acknowledge lines of the printer. The stack pointer, program counter, subroutine-call program, and subroutine-return program are assigned to registers R_2, R_3, R_4, and R_5. ERROR, OPTION, PROMPT2, OSTRNG, and INIT2, which are subroutines of the UT-62, are also incorporated in the program.

An initialization procedure ends with the beginning address in register RB and the byte count in register RA. The American Standard Code Information-Interchange code for a carriage return is loaded in register RF.1, as is all other data that is to be printed. As a result, when the print subroutine is called, it initiates a carriage return followed by an automatic line feed. Thereupon a subroutine jump to OUT1 via TY5 converts the address of the first byte on the line into ASCII and prints it.

DATOUT loads the data from the address in RB into RF.1 and then calls subroutine OUTDAT. This subroutine converts the data into a pair of ASCII hexadecimal characters for printing. Upon return from OUTDAT, DATOUT checks RA for a zero byte count. If the byte count is zero, a final CR is printed and the program returns to the monitor. Otherwise the byte count is decremented by 1 and a check is made to see if the 4 least significant bits of the address are F_{16}. When this condition is met, a semicolon is printed and the program branches back to OUTPUT; if the condition is not met, the LSB is checked at SAMELN.

When the LSB bit is 1, the program branches to DATOUT for the next byte of data, but when it is 0, goes to SPCOUT for printing a space between printed data. □

Software notebook is a regular feature in *Electronics*. We invite readers to submit short, original, unpublished programs and software solutions to engineering problems. Explain briefly and thoroughly the program's operation. We'll pay $75 for each item published.