The Engineering Staff of
TEXAS INSTRUMENTS INCORPORATED
Semiconductor Group

TM 990/514 AND TM 990/515 EPROM PERSONALITY PROGRAMMING CARDS

NOVEMBER 1978

TEXAS INSTRUMENTS
INCORPORATED

IMPORTANT NOTICES

Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

TI cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.

TM 990/514 AND TM 990/515 EPROM PERSONALITY PROGRAMMING CARDS

1. GENERAL

The TM 990/514 and TM 990/515 EPROM Personality Programming Cards (EPPC) are designed to be used with the TM 990/302 Software Development Board. The Personality Programming Cards are used to configure the EPROM programming socket for the EPROM that is selected to be programmed or read. Figure 1 shows the EPROM Personality Programming Cards.

The TM 990/514 or TM 990/515 EPPCs are pushed on board edge connector P3 on the TM 990/302 Software Development Board. All the signals and voltages that are required for programming and reading the various EPROMs are brought out to edge connector P3. A description of each of these signals and voltages is given in the TM 990/302 Hardware User's Guide.

Software that can be used to program or read an EPROM is available as TM 990/302 development software (resident on the TM 990/302 as shipped from the factory), Enhancement BASIC, and the example program given in this manual.

2. TM 990/514

The TM 990/514 accommodates the following EPROMs:

- TMS 2708 1K × 8 bits each
- TMS 2716 2K × 8 bits each.

The schematic diagram for the TM 990/514 is shown in Figure 2.

3. TM 990/515

The TM 990/515 accommodates the following EPROMs:

- TMS 2508 1K × 8 bits each
- TMS 2516 2K × 8 bits each
- TMS 2532 4K × 8 bits each.

The schematic diagram for the TM 990/515 is shown in Figure 3.

4. CARD AND EPROM INSERTION

To use these cards, do the following steps:

(a) Three jumpers on the right side of the card designate which EPROM type is to be used. Set these jumpers to that EPROM type. See Figure 4.

(b) Press the card onto connector P3 (left side facing) of the TM 990/302 board. When correctly installed with EPROM voltage applied, the LED marked GO will illuminate.

(c) Install the EPROM to be programmed into the socket with pin 1 of the EPROM in the upper right corner.

(d) While programming, the LED marked PROG will glow.

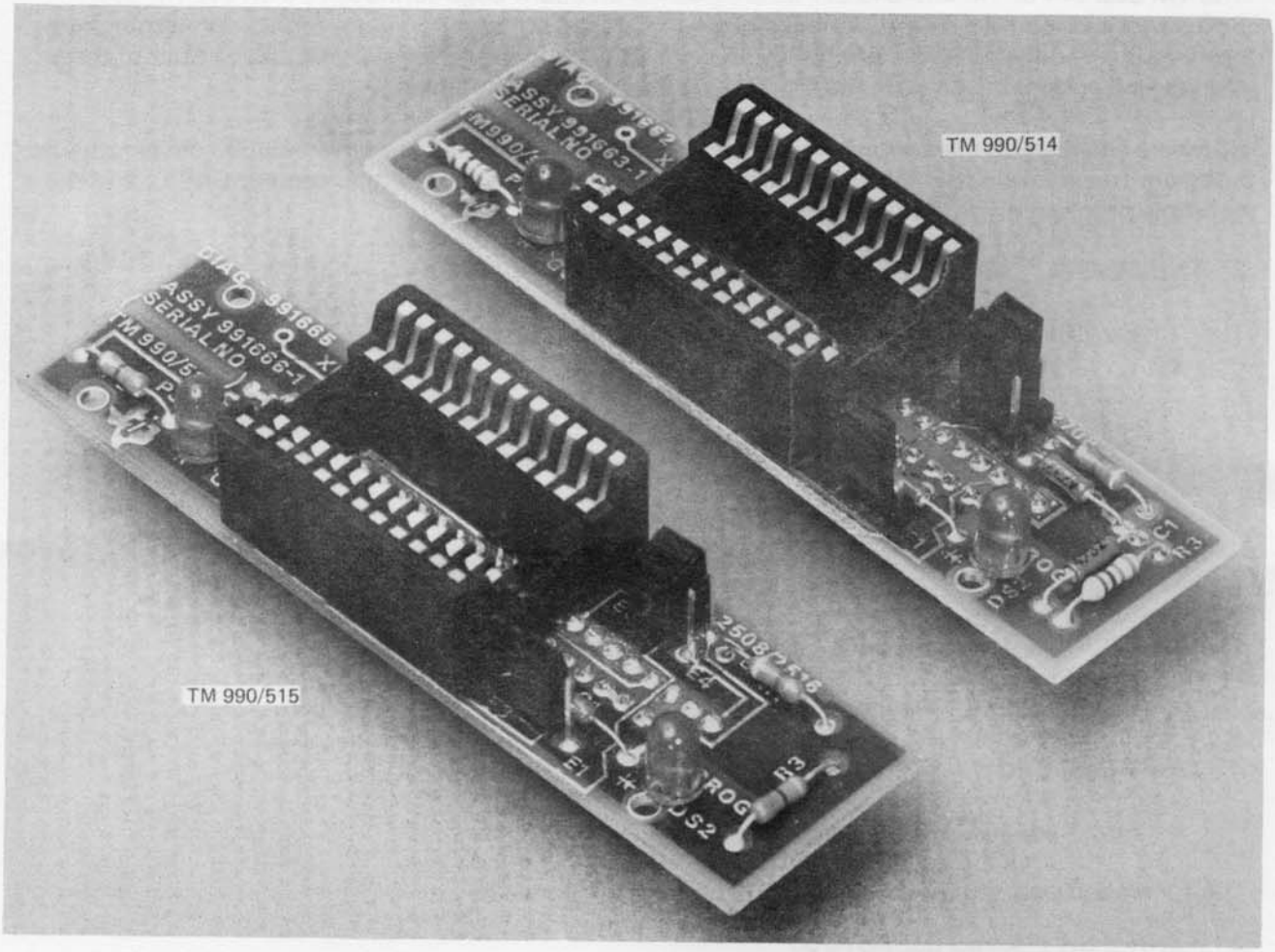

Figure 1. TM 990/514 and TM 990/515 EPROM Personality Programming Modules

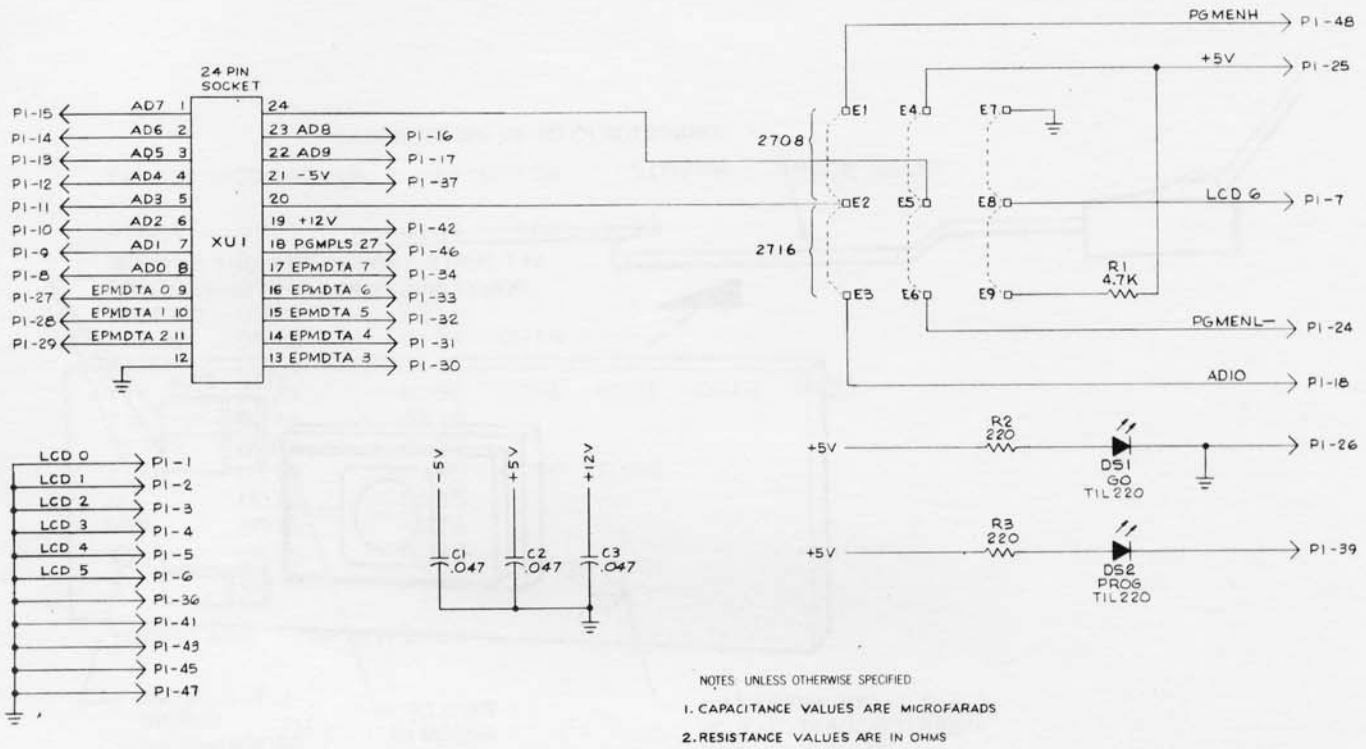

Figure 2. TM 990/514 Schematic Diagram

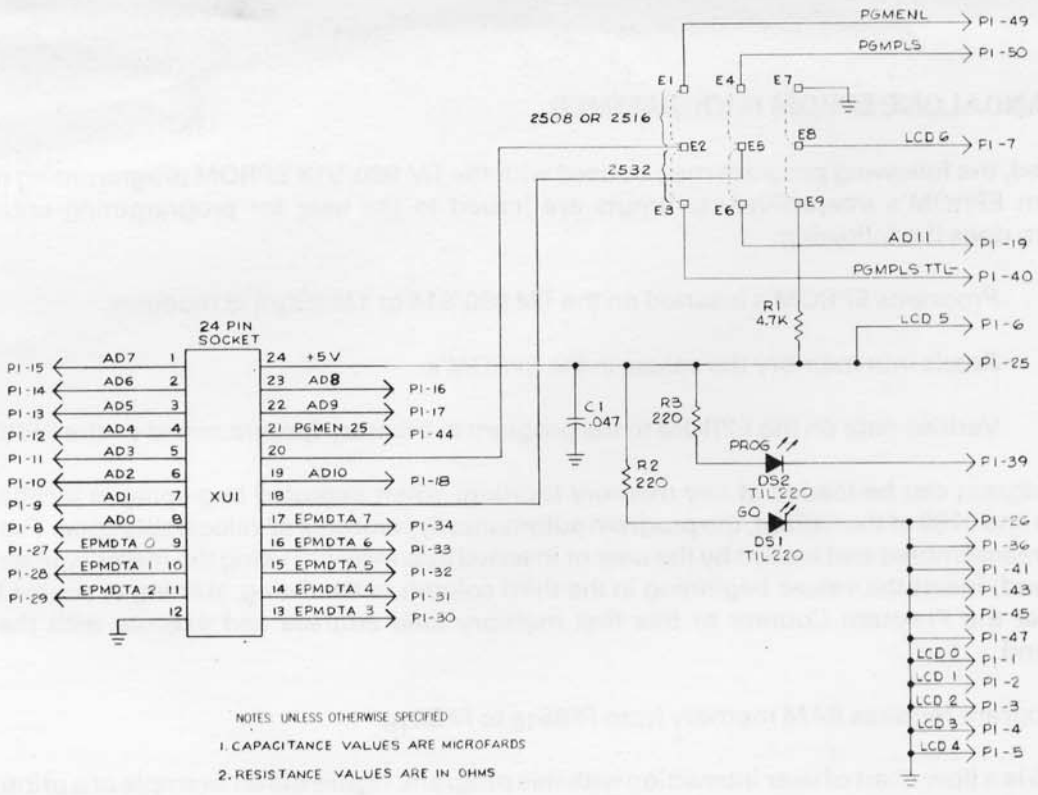

Figure 3. TM 990/515 Schematic Diagram

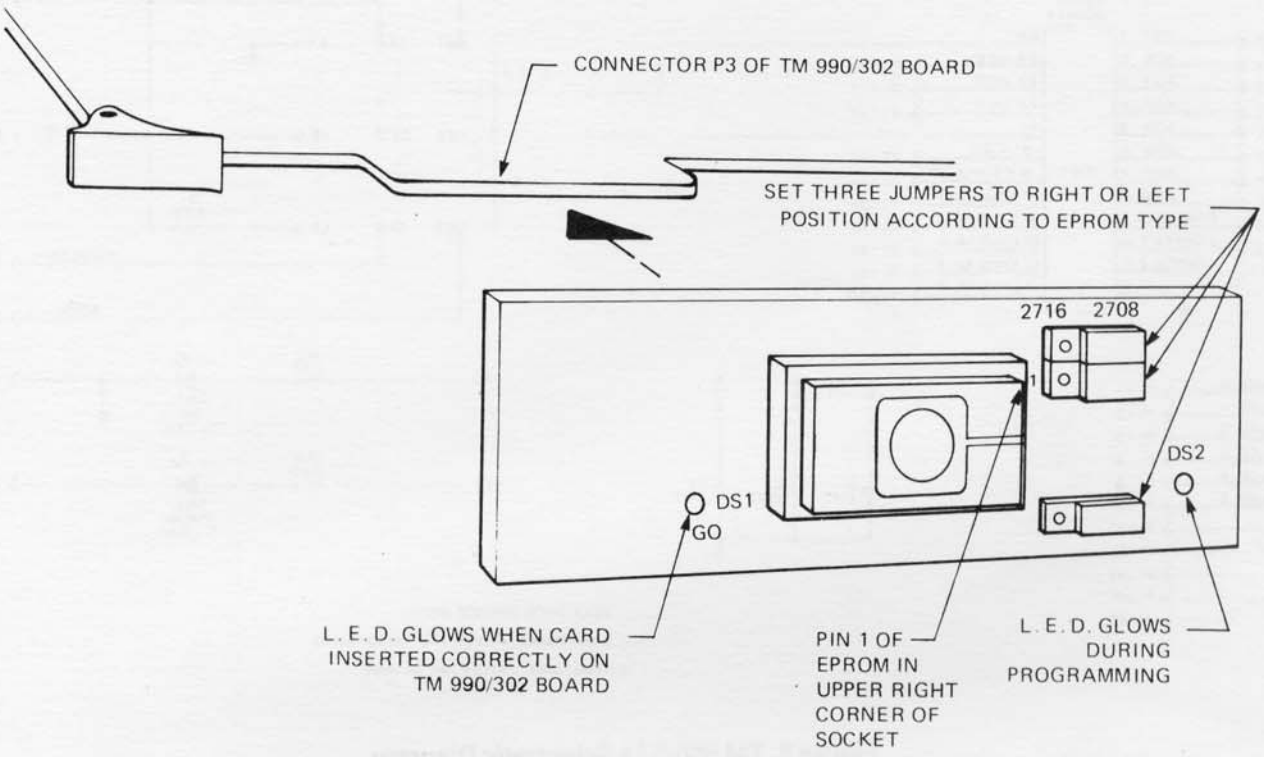

Figure 4. Inserting Personality Card and EPROM

5. STANDALONE EPROM PROGRAMMER

If desired, the following program may be used with the TM 990/51X EPROM programming module to program EPROM's interactively (prompts are issued to the user for programming criteria). This program does the following:

- Programs EPROM's inserted on the TM 990/514 or TM 990/515 modules.
- Reads into memory the values in the EPROM's.
- Verifies data on the EPROM to the program in memory (programmed on the EPROM).

This program can be loaded at any memory location; when executed beginning at location START (source line 0160 of the listing), the program automatically relocates all relocatable code. The program can be reassembled and loaded by the user or inserted into memory using the memory insert/change command; insert the values beginning in the third column of the listing, starting at source line 0160. Then set the Program Counter to this first memory load address and execute with the Execute command.

This program requires RAM memory from $FF88_{16}$ to $FFB0_{16}$.

Figure 5 is a flow chart of user interaction with this program. Figure 6 is an example of a printout of this program in use. The program menu explains allowable inputs to the prompts.

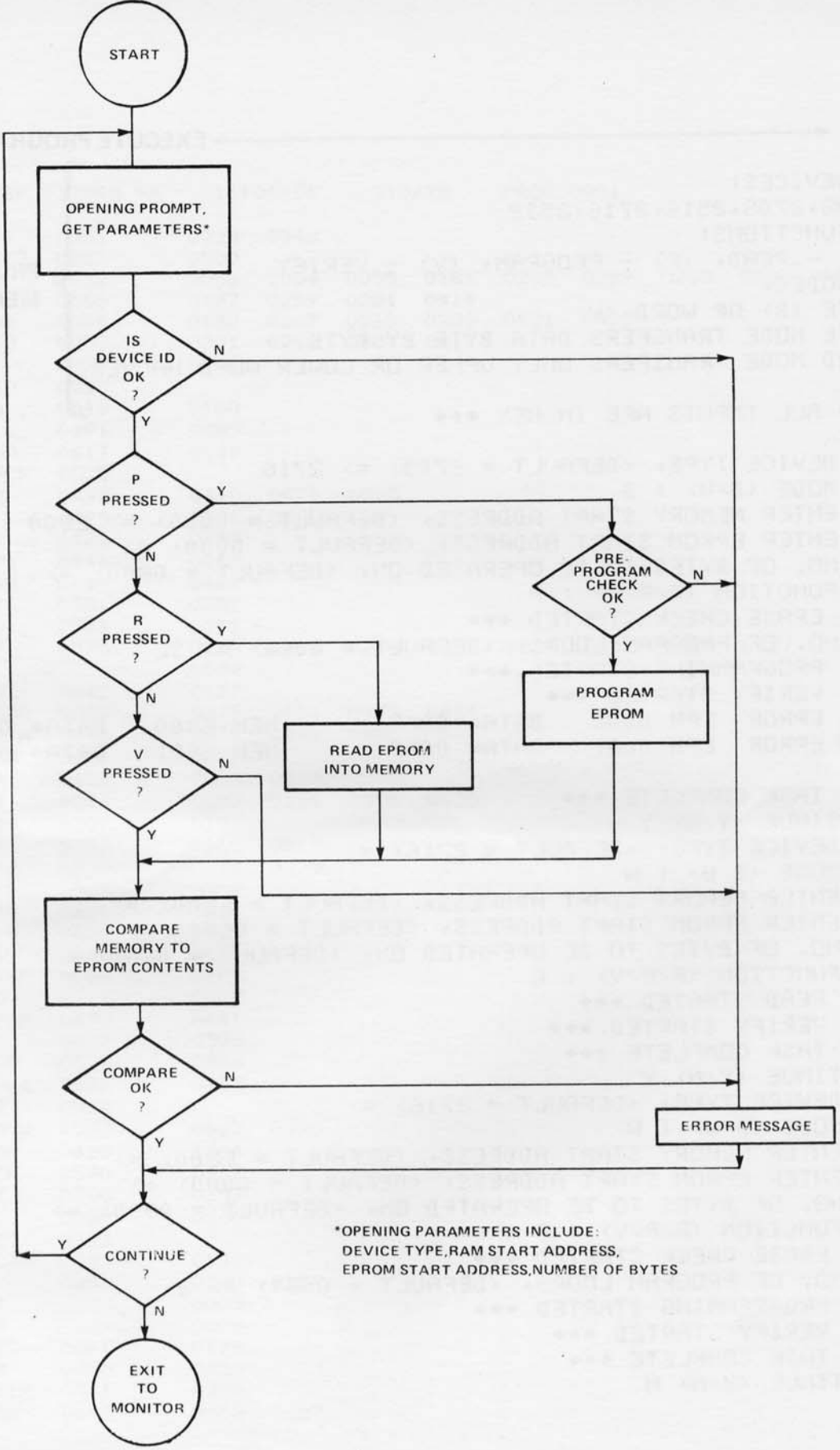

Figure 5. Flowchart of Interaction with Standalone EPROM Programmer

```
E
```

EXECUTE PROGRAM

```
  DEVICES:
2508,2708,2516,2716,2532
  FUNCTIONS:
(R) - READ, (P) - PROGRAM, (V) - VERIFY
  MODES:
BYTE (B) OR WORD (W)
BYTE MODE TRANSFERS DATA BYTE BY BYTE.
WORD MODE TRANSFERS ONLY UPPER OR LOWER WORD HALVES.

♦♦♦ ALL INPUTS ARE IN HEX ♦♦♦

1. DEVICE TYPE, (DEFAULT = 2708) => 2716
2. MODE (B/W) : B
3. ENTER MEMORY START ADDRESS, (DEFAULT = 0000) => E800
4. ENTER EPROM START ADDRESS, (DEFAULT = 0000) =>
5. NO. OF BYTES TO BE OPERATED ON, (DEFAULT = 0800) =>
6. FUNCTION (R/P/V) : P
♦♦♦ ERASE CHECK STARTED ♦♦♦
7. NO. OF PROGRAM LOOPS, (DEFAULT = 0064) => 32
♦♦♦ PROGRAMMING STARTED ♦♦♦
♦♦♦ VERIFY STARTED ♦♦♦
♦♦♦ ERROR  EPM 0000   DATA= 00FF        MEM E800   DATA= 0031
♦♦♦ ERROR  EPM 0001   DATA= 00FF        MEM E801   DATA= 0033

♦♦♦ TASK COMPLETE ♦♦♦
CONTINUE (Y/N) Y
1. DEVICE TYPE, (DEFAULT = 2716) =>
2. MODE (B/W) : W
3. ENTER MEMORY START ADDRESS, (DEFAULT = E800) =>
4. ENTER EPROM START ADDRESS, (DEFAULT = 0000) =>
5. NO. OF BYTES TO BE OPERATED ON, (DEFAULT = 0800) =>
6. FUNCTION (R/P/V) : R
♦♦♦ READ STARTED ♦♦♦
♦♦♦ VERIFY STARTED ♦♦♦
♦♦♦ TASK COMPLETE ♦♦♦
CONTINUE (Y/N) Y
1. DEVICE TYPE, (DEFAULT = 2716) =>
2. MODE (B/W) : W
3. ENTER MEMORY START ADDRESS, (DEFAULT = E800) =>
4. ENTER EPROM START ADDRESS, (DEFAULT = 0000) =>
5. NO. OF BYTES TO BE OPERATED ON, (DEFAULT = 0800) =>
6. FUNCTION (R/P/V) : P
♦♦♦ ERASE CHECK STARTED ♦♦♦
7. NO. OF PROGRAM LOOPS, (DEFAULT = 0064) => 2
♦♦♦ PROGRAMMING STARTED ♦♦♦
♦♦♦ VERIFY STARTED ♦♦♦
♦♦♦ TASK COMPLETE ♦♦♦
CONTINUE (Y/N) N

?
```

PROGRAM MENU

Figure 6. Interactive Execution of Standalone EPROM Programmer

302 EPROM CHECK ROUTINE

```
0002                    IDT  'EPROM'
0003                ********DEFINITIONS********EQUATES********
0004        0080    TIBUG   EQU  >0080
0005                        DXOP HEXI,9
0006                        DXOP HEXO,10
0007                        DXOP ECHO,11
0008                        DXOP PRNT,14
0009                ************************
0010                *  REGISTER EQUATES  *
0011                ************************
0012        0000    R0      EQU  0
0013        0001    R1      EQU  1
0014        0002    R2      EQU  2
0015        0003    R3      EQU  3
0016        0004    R4      EQU  4
0017        0005    R5      EQU  5
0018        0006    R6      EQU  6
0019        0007    R7      EQU  7
0020        0008    R8      EQU  8
0021        0009    R9      EQU  9
0022        000A    R10     EQU  10
0023        000B    R11     EQU  11
0024        000C    R12     EQU  12
0025        000D    R13     EQU  13
0026        000E    R14     EQU  14
0027        000F    R15     EQU  15
0028                ****************************
0029                * ASCII CHARACTER EQUATES *
0030                ****************************
0031        4200    B       EQU  >4200
0032        5000    P       EQU  >5000
0033        5200    R       EQU  >5200
0034        5600    V       EQU  >5600
0035        5700    W       EQU  >5700
0036        5900    Y       EQU  >5900
0037                *
0038                ********************
0039                * ADDRESS EQUATES *
0040                ********************
0041                *
0042        0080    CRU02   EQU  >0080          CRU BASE FOR 9902
0043        1710    CRUDTA  EQU  >1710          CRU BASE FOR TM990/514
0044                *                           EPROM DATA
0045        1720    CRUADD  EQU  >1720          CRU BASE FOR TM990/514
0046                *                           EPROM ADD.
0047        1702    CRUID   EQU  >1702          CRU BASE FOR DEVICE I.D.
0048        0460    BCMD    EQU  >0460          'B' COMMAND
0049        045B    RTCMD   EQU  >045B          'RT' COMMAND
0050        FF90    TSTWS   EQU  >FF90          WORKSPACE FOR THIS MODULE
0051                *
0052        FF88    AREA    EQU  >FF88          THIS AREA IS USED IN
0053        FF8A    AREA1   EQU  AREA+2         CONJUNCTION WITH THE 'HEXI'
0054        FF8C    AREA2   EQU  AREA+4         XOP. PROGRAM EXECUTION IS
0055        FF8E    AREA3   EQU  AREA+6         TRANSFERRED TO THIS AREA WHICH
```

```
0056              *                                CAN BE PRELOADED MUCH LIKE
0057              *                                PROGRAMMING INTERRUPTS.
0058              *
0059              *      'HEXI' REQUIRES TWO ADDRESSES TO FOLLOW IT
0060              *      THE FIRST ONE IS THE ADDRESS IT BRANCHES TO IF
0061              *      ONLY A VALID TERMINATION CHARACTER IS ENTERED.
0062              * THE SECOND ADDRESS IS THE ADDRESS THAT IS BRANCHED
0063              *      TO IF AN INVALID ENTRY IS MADE.
0064              *
0065              *           VALID TERMINATION CHARACTERS ARE:
0066              *                                CARRIAGE RETURN
0067              *                                SPACE
0068              *                                COMMA
0069              *                                MINUS SIGN
0070              *
0071              *      THE ONLY VALID ENTRIES ACCEPTED ARE:
0072              *           0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
0073              *           E.G. FFB0
0074              *
0075              *
0076              *      FOR MORE INFORMATION ON THE 'HEXI' XOP REFER
0077              *      TO THE TM990/100M MICROCOMPUTER USER'S GUIDE
0078              *      OR THE TM990/101M MICROCOMPUTER USER'S GUIDE.
0079              *
0080              ***************************************************************
0081              *
0082              *           THIS PROGRAM IS FOR USE WITH A VERSION OF
0083              *      TIBUG THAT RESIDES IN LOW MEMORY.
0084              *           IT REQUIRES RAM MEMORY FROM >FF88 TO >FFB0.
0085              *
0086              *           THIS PROGRAM SERVES THREE PURPOSES.
0087              *      THE FIRST IS TO READ THE CONTENTS OF THE
0088              *      EPROM IN THE PROGRAMMING SOCKET OF THE
0089              *      TM990/514 EPROM PERSONALITY PROGRAMMING CARD (EPPC),
0090              *      OR THE TM990/515 EPPC. THE CONTENTS ARE READ
0091              *      INTO MEMORY STARTING AT THE MEMORY LOCATION
0092              *      RECIEVED FROM THE USER BY PROMPT.
0093              *           THE SECOND FUNCTION PERFORMED IS TO
0094              *      PROGRAM THE EPROM IN THE TM990/514 OR /515 EPM.
0095              *      THIS IS ACCOMPLISHED BY TRANSFERRING THE
0096              *      CONTENTS OF THE MEMORY SPECIFIED TO THE
0097              *      EPROM AND OUTPUTING THE CORRECT PROGRAMMING
0098              *      SIGNALS.
0099              *           THE THIRD FUNCTION IS VERIFICATION. THIS
0100             *      FUNCTION IS EMPLOYED AFTER A READ OR
0101             *      PROGRAMMING FUNCTION OR UNDER USER REQUEST.
0102             *      IT IS ACCOMPLISHED BY COMPARRING THE CONTENTS
0103             *      OF THE MEMORY SPECIFIED AND THE CONTENTS
0104             *      OF THE EPROM IN THE PROGRAMMING SOCKET ON THE
0105             *      TM990/514  OR /515 EPPC.
0106             *           ALL TRANSFERS ARE DONE BYTE BY BYTE. IN WORD
0107             *      MODE IF THE START ADDRESS GIVEN IS ODD, DATA BITS
0108             *      8-15 ARE TRANSFERRED.....IF THE START MEMORY ADDRESS
0109             *      IS EVEN, DATA BITS 0-7 ARE TRANSFERRED.
```

```
0110                *
0111                *
0112                ***************************************************************
0113                *
0114                *       EPPC - EPROM PERSONALITY PROGRAMMING MODULE
0115                *
0116                ***************
0117                * PROCEDURE *
0118                ***************
0119                *
0120                *            TO PROGRAM A 'TMS27' SERIES DEVICE, 100 LOOPS
0121                *       (PROGRAMMING CYCLES) ARE EXECUTED. EACH PROGRAMMING
0122                *       CYCLE CONSISTS OF ADDRESSING EACH BYTE ONCE AT WHICH
0123                *       A PROGRAMMING VOLTAGE IS APPLIED AND HELD FOR 1 MS.
0124                *            TO PROGRAM A 'TMS25' SERIES DEVICE, 50 LOOPS
0125                *       ARE EXECUTED. IN THIS CASE THE HIGH VOLTAGE NEEDED
0126                *       FOR PROGRAMMING IS APPLIED AND A TTL LEVEL SIGNAL
0127                *       IS TOGGLED TO EFFECT PROGRAMMING.
0128                *            THE DIFFERENCE BETWEEN THE TWO IS TAKEN CARE
0129                *       OF BY THE EPPC.
0130                *
0131                ***************
0132                * SPECIFICS *
0133                ***************
0134                *
0135                *            ANY NUMBER OF BYTES CAN BE PROGRAMMED STARTING
0136                *       AT THE START ADDRESS AND CONTINUING FOR THE NUMBER
0137                *       OF BYTES SPECIFIED.
0138                *
0139                *            TO TRUNCATE THE ERROR LISTINGS, TYPE ANY
0140                *       CHARACTER.
0141                *
0142                *
0143                *            A TMS EPROM ERASES TO ALL ONE'S. IF A ONE IS
0144                *       NOT PRESENT AFTER ERASE WHERE A ONE IS TO BE
0145                *       PROGRAMMED, PROGRAMMING CANNOT BE DONE. THIS
0146                *       PROGRAM FEATURES A PREPROGRAMMING CHECK IN WHICH
0147                *       ALL LOCATIONS THAT REQUIRE ONE'S ARE CHECKED FOR
0148                *       ONE'S. ERROR LOCATIONS ARE PRINTED.
0149                *
0150                *            FOR EXAMPLE IF >AB WAS TO BE PROGRAMMED
0151                *       INTO A EPROM THAT DID NOT ERASE FULLY AND HAD
0152                *       THE DATA >FE, THE PROGRAM WOULD PRINT AN ERROR.
0153                *       BUT IF THAT LOCATION HAD >BF, NO ERROR WOULD
0154                *       BE PRINTED.
0155                *
0156                *            THE '>' SIGN DENOTES A HEXIDECIMAL VALUE.
0157                *
```

302 EPROM CHECK ROUTINE

```
0159                    *
0160                    START
0161 0000 0360          RSET                    RESET
0162                    *---------- PROVIDES RELOCATION ----------
0163 0002 02E0          LWPI TSTWS              GET TEST WORKSPACE
     0004 FF90
0164 0006 020D          LI   R13,START          PICK UP LOAD BIAS
     0008 0000'
0165 000A 050D          NEG  R13                AND GET NEGATIVE OF IT
0166 000C 0201          LI   R1,RTCMD           R1 = RT INSTRUCTION
     000E 045B
0167 0010 0681          BL   R1                 LOAD R11 WITH ADD. OF NEXT INS
0168                    RELOC
0169 0012 022B          AI   R11,START-RELOC    ADJUST TO GET THE ACTUAL
     0014 FFEE
0170                    *                       START ADDRESS IN MEMORY
0171 0016 A34B          A    R11,R13            R13 = BASE ADDRESS
0172                    *-------------------------------------------------
0173                    *       THE BASE ADDRESS IS THE ACTUAL LOCATION IN
0174                    *       MEMORY MINUS ANY LOAD BIAS. THIS ALLOWS THE
0175                    *       EXECUTION OF THIS MODULE ANYWHERE IN MEMORY.
0176                    *         +EXECUTION STARTS AT THE LABEL START+
0177                    *-------------------------------------------------
0178                    STARTX
0179 0018 2FAD          PRNT @HLIST(13)         OUTPUT HELP LIST
     001A 0479'
0180 001C 0200          LI   R0,BCMD            LOAD R0 WITH 'B' COMMAND
     001E 0460
0181 0020 C800          MOV  R0,@AREA           STORE FOR BRANCH
     0022 FF88
0182 0024 C800          MOV  R0,@AREA2          *
     0026 FF8C
0183 0028 04C3          CLR  R3                 SET RAM DEFAULT ADDRESS
0184 002A 0207          LI   R7,TABLE           GET INTIAL TABLE ADDRESS
     002C 02CC'
0185 002E A1CD          A    R13,R7             ADD BASE ADDRESS
```

302 EPROM CHECK ROUTINE

```
0187                    *
0188                    *       INPUT DEVICE AND CONTROL PARAMETERS
0189                    *
0190                    *            ***********************
0191                    *            * DEVICE TYPE AND ID *
0192                    *            ***********************
0193                    TYPE
0194 0030 C057                  MOV   *R7,R1              GET DEFAULT DEVICE TYPE
0195 0032 020F                  LI    R15,TYPE            GET 'BRANCH TO' ADD.
     0034 0030'
0196 0036 A3CD                  A     R13,R15             ADD BASE ADDRESS
0197 0038 C80F                  MOV   R15,@AREA1          STORE IT
     003A FF8A
0198 003C 020F                  LI    R15,DFLTO           GET ADDRESS TO BRANCH TO
     003E 005A'
0199                    *                                 IF DEFAULT TAKEN
0200 0040 A3CD                  A     R13,R15             ADD BASE ADDRESS
0201 0042 C80F                  MOV   R15,@AREA3          AND STORE IT
     0044 FF8E
0202 0046 2FAD                  PRNT  @DVCTYP(13)         PROMPT FOR DEVICE TYPE
     0048 0329'
0203 004A 2FAD                  PRNT  @DFLT(13)           OUTPUT 'DEFAULT ='
     004C 03D9'
0204 004E 2E81                  HEXO  R1                  OUTPUT DEFAULT TYPE.
0205 0050 2FAD                  PRNT  @TIC(13)            AND PROMPT
     0052 059A'
0206 0054 2E41                  HEXI  R1                  ACCEPT DEVICE TYPE INPUT
0207 0056 FF8C                  DATA  AREA2
0208 0058 FF88                  DATA  AREA
0209                    DFLTO
0210 005A 2FA0                  PRNT  @CRLF               CLEAR PRINT LINE
     005C 03B9'
0211 005E 04C0                  CLR   R0                  CLEAR DEVICE ID.
0212 0060 020C                  LI    R12,CRUID           SET UP TO READ DEVICE ID.
     0062 1702
0213 0064 35C0                  STCR  R0,7                READ AND STORE DEVICE ID.
0214 0066 0207                  LI    R7,TABLE            LOAD ADDRESS OF FIRST TABLE
     0068 02CC'
0215 006A A1CD                  A     R13,R7              ADD BASE ADDRESS
0216 006C 0202                  LI    R2,TABEND           GET TABLE END ADDRESS
     006E 02FE'
0217 0070 A08D                  A     R13,R2              ADD BASE ADDRESS
0218                    SEARCH
0219 0072 85C1                  C     R1,*R7              IS ENTRY IN THIS TABLE
0220 0074 1305                  JEQ   IDCK                IF SO, GO CHECK DEVICE I.D.
0221 0076 0227                  AI    R7,10               OTHERWISE TRY NEXT TABLE
     0078 000A
0222 007A 8087                  C     R7,R2               IF ALL TABLES HAVE BEEN
0223 007C 1472                  JHE   ABORT               CHECKED, ABORT !.OTHERWISE
0224 007E 10F9                  JMP   SEARCH              CHECK NEXT TABLE.
0225                    IDCK
0226 0080 89C0                  C     R0,@2(R7)           CHECK DEVICE I.D.
     0082 0002
0227 0084 1303                  JEQ   MERGE               IF CORRECT, GO GET MODE
```

302 EPROM CHECK ROUTINE

```
0228                    BADID
0229 0086 2FAD                  PRNT @JUMPER(13)        PRINT ERROR MESSAGE
     0088 02FE'
0230 008A 10D2                  JMP  TYPE               AND TRY AGAIN
0231                    *
0232                    *
0233                    *           **********************
0234                    *           * BYTE OR WORD MODE *
0235                    *           **********************
0236                    *
0237                    *
0238                    MERGE
0239 008C 2FAD                  PRNT @BYTWD(13)         PROMT FOR BYTE OR WORD MODE
     008E 05A0'
0240                    *                       BYTE MODE TRANSFERS BYTE BY BYTE
0241                    *                       WORD MODE TRANSFERS ONLY UPPER
0242                    *                          OR LOWER WORD HALVES
0243 0090 2EC0                  ECHO R0                 GET MODE
0244 0092 0280                  CI   R0,W               IS IT WORD MODE
     0094 5700
0245 0096 1304                  JEQ  GET                IF SO GO TO GET
0246 0098 0280                  CI   R0,B               IS IT BYTE MODE ?
     009A 4200
0247 009C 16F7                  JNE  MERGE              IF NOT TRY AGAIN
0248 009E 04C0                  CLR  R0 .               SET BYTE FLAG
0249                    *
0250                    *
0251                    *           **********************
0252                    *           * RAM START ADDRESS *
0253                    *           **********************
0254                    *
0255                    *
0256                    GET
0257 00A0 020F                  LI   R15,GET            GET 'BRANCH TO' ADDRESS
     00A2 00A0'
0258 00A4 A3CD                  A    R13,R15            ADD BASE ADDRESS
0259 00A6 C80F                  MOV  R15,@AREA1         STORE IT
     00A8 FF8A
0260 00AA 020F                  LI   R15,DFLT1          GET 'BRANCH TO' ADD.
     00AC 00C8'
0261 00AE A3CD                  A    R13,R15            ADD BASE ADDRESS
0262 00B0 C80F                  MOV  R15,@AREA3         STORE IT
     00B2 FF8E
0263 00B4 2FAD                  PRNT @CRLF(13)          CLEAR PRINT LINE
     00B6 03B9'
0264 00B8 2FAD                  PRNT @STRTAD(13)        OUTPUT PROMPT FOR ADDRESS
     00BA 03BC'
0265 00BC 2E83                  HEXO R3                 OUTPUT DEFAULT VALUE
0266 00BE 2FAD                  PRNT @TIC(13)           *
     00C0 059A'
0267 00C2 2E43                  HEXI R3                 GET VALUE
0268 00C4 FF8C                  DATA AREA2
0269 00C6 FF88                  DATA AREA
0270                    *
```

302 EPROM CHECK ROUTINE

```
0271                    *
0272                    *          **********************
0273                    *          * EPROM START ADDRESS *
0274                    *          **********************
0275                    *
0276                    *
0277                    DFLT1
0278 00C8 2FAD          PRNT @CRLF(13)          CLEAR PRINT LINE
     00CA 03B9'
0279 00CC 020F          LI   R15,DFLT2          GET 'BRANCH TO' ADD.
     00CE 00EC'
0280 00D0 A3CD          A    R13,R15            ADD BASE ADDRESS
0281 00D2 C80F          MOV  R15,@AREA1         STORE IT
     00D4 FF8A
0282 00D6 04C4          CLR  R4                 INITIALIZE EPROM ADDRESS
0283 00D8 2FAD          PRNT @EPMAD(13)         PRINT EPROM START ADD. PROMPT
     00DA 03E7'
0284 00DC 2FAD          PRNT @DFLT(13)          OUTPUT DEFAULT MESSAGE
     00DE 03D9'
0285 00E0 2E84          HEXO R4                 OUTPUT DEFAULT
0286 00E2 2FAD          PRNT @TIC(13)           *
     00E4 059A'
0287 00E6 2E44          HEXI R4                 ACCEPT NEW EPROM ADD.
0288 00E8 FF88          DATA AREA
0289 00EA FF8C          DATA AREA2
0290                    *
0291                    *
0292                    *
0293                    *
0294                    *          ********************
0295                    *          * NUMBER OF BYTES *
0296                    *          ********************
0297                    *
0298                    *     NOTE THAT THE DEFAULT IS DEPENDENT ON # OF BYTES
0299                    *          IN EPROM FROM START ADDRESS TO ITS NORMAL DEFAULT
0300                    DFLT2
0301 00EC 2FAD          PRNT @CRLF(13)          CLEAR PRINT LINE
     00EE 03B9'
0302 00F0 020F          LI   R15,DFLT3          GET 'BRANCH TO' ADD.
     00F2 0126'
0303 00F4 A3CD          A    R13,R15            ADD BASE ADDRESS
0304 00F6 C80F          MOV  R15,@AREA3         STORE IT
     00F8 FF8E
0305 00FA C167          MOV  @4(R7),R5          GET DEFAULT # OF BYTES
     00FC 0004
0306 00FE 8144          C    R4,R5              IF START ADD. > # OF BYTES
0307 0100 1A06          JL   OK1                INDICATE, OTHERWISE GO
0308 0102 2FAD          PRNT @LARGE(13)         ON.
     0104 057A'
0309 0106 2FAD          PRNT @MAX(13)           *
     0108 0591'
0310 010A 2E85          HEXO R5                 OUTPUT MAXIMUM VALUE
0311 010C 10DD          JMP  DFLT1              AND TRY AGAIN
0312                    OK1
```

302 EPROM CHECK ROUTINE

```
0313 010E 6144          S    R4,R5              ADJUST DEFAULT FOR #
0314                *                           OF BYTES LEFT
0315 0110 2FAD          PRNT @LENGTH(13)        PROMPT FOR # OF BYTES
     0112 0440'
0316 0114 2FAD          PRNT @DFLT(13)          PRINT DEFAULT MESSAGE
     0116 03D9'
0317 0118 2E85          HEXO R5                 OUTPUT DEFAULT VALUE
0318 011A 2FAD          PRNT @TIC(13)           *
     011C 059A'
0319 011E C245          MOV  R5,R9              STORE FOR COMPARE
0320 0120 2E49          HEXI R9                 GET # OF BYTES
0321 0122 FF8C          DATA AREA2
0322 0124 FF88          DATA AREA
0323              DFLT3
0324 0126 2FAD          PRNT @CRLF(13)          CLEAR PRINT LINE
     0128 03B9'
0325 012A 8149          C    R9,R5              IS ENTRY GREATER THAN DEFLT.
0326 012C 1206          JLE  FUNCTO             IF NOT GO ON !
0327 012E 2FAD          PRNT @LARGE(13)         OTHERWISE, INDICATE.
     0130 057A'
0328 0132 2FAD          PRNT @MAX(13)           *
     0134 0591'
0329 0136 2E85          HEXO R5                 *
0330 0138 10D9          JMP  DFLT2              AND TRY AGAIN
```

302 EPROM CHECK ROUTINE

```
0332                    *               ******************
0333                    *               * FUNCTION FETCH *
0334                    *               ******************
0335                    FUNCTO
0336 013A C149                  MOV   R9,R5             IF NOT USE THIS NEW VALUE
0337                    FUNCT
0338 013C 070E                  SETO  R14               CLEAR PREPGM CHECK FLAG
0339 013E 04CF                  CLR   R15               CLEAR ERASE ERROR FLAG
0340 0140 2FAD                  PRNT  @FCN(13)          GET FUNCTION TO BE PERFORMED
     0142 0462'
0341 0144 2EC1                  ECHO  R1                *
0342 0146 2FAD                  PRNT  @CRLF(13)         CLEAR PRINT LINE
     0148 03B9'
0343 014A 0281                  CI    R1,P              CHECK FOR P
     014C 5000
0344 014E 1317                  JEQ   PREPGM            GO EXECUTE PROGRAMMING ROUTINE
0345 0150 0281                  CI    R1,V              CHECK FOR V
     0152 5600
0346 0154 1379                  JEQ   CMP               GO EXECUTE VERIFY ROUTINE
0347 0156 0281                  CI    R1,R              CHECK FOR R
     0158 5200
0348 015A 1603                  JNE   ABORT             IF NOT R, ABORT !
0349 015C 2FAD                  PRNT  @RD(13)           OTHERWISE, PRINT READ
     015E 0356'
0350 0160 1011                  JMP READ                BANNER AND GO READ
0351                    ABORT
0352 0162 2FAD                  PRNT  @CONT(13)         PROMPT FOR CONTINUE Y/N
     0164 0404'
0353 0166 2FAD          ABORT2  PRNT  @CONT1(13)        *
     0168 0418'
0354 016A 2EC1                  ECHO  R1                GET INPUT
0355 016C 2FAD                  PRNT  @CRLF(13)         CLEAR PRINT LINE
     016E 03B9'
0356 0170 0281                  CI    R1,Y              IS IT YES ?
     0172 5900
0357 0174 1602                  JNE   EXIT              IF SO, EXIT
0358 0176 046D                  B     @TYPE(13)         OTHERWISE GO AGAIN !
     0178 0030'
0359                    *
0360                    EXIT
0361 017A 0460                  B     @TIBUG            IF NOT, EXIT
     017C 0080
```

302 EPROM CHECK ROUTINE

```
0363                    *
0364                    **********ERASE CHECK SET UP**********
0365                    *
0366 017E 04CE  PREPGM  CLR  R14                SET PREPGM CHECK FLAG
0367 0180 2FAD          PRNT @ERSCK(13)         PRINT BANNER
     0182 036D'
0368                    *
0369                    **********READ ROUTINE**********
0370                    *
0371                    READ
0372 0184 C203          MOV  R3,R8              STORE START ADDRESS
0373 0186 C244          MOV  R4,R9              BUFFER EPROM ADDRESS
0374 0188 C285          MOV  R5,R10             BUFFER BYTE COUNT
0375                    READ1
0376 018A 04C2          CLR  R2                 CLEAR DATA BUFFER
0377 018C 020C          LI   R12,CRUADD         SET CRU FOR EPM ADDRESS
     018E 1720
0378 0190 3009          LDCR R9,0               OUTPUT ADR TO HARDWARE
0379 0192 020C          LI   R12,CRUDTA         SET CRU ADR FOR READ DATA
     0194 1710
0380 0196 3602          STCR R2,8               GET DATA
0381 0198 C38E          MOV  R14,R14            IF ERASE DONT USE ANY MEMORY
0382 019A 130C          JEQ  ERASE1             BY GOING TO ERASE1
0383 019C D602          MOVB R2,*R8             XFER READ DATA TO MEMORY
0384                    MERGE1
0385 019E 0588          INC  R8                 UPDATE MEMORY ADDRESS
0386 01A0 C000          MOV  R0,R0              IF IN BYTE MODE
0387 01A2 1301          JEQ  ROUND1             INCERMENT BY ONE
0388 01A4 0588          INC  R8                 OTHERWISE INC. BY TWO
0389                    ROUND1
0390 01A6 0589          INC  R9                 INCREMENT EPROM ADR
0391 01A8 060A          DEC  R10                CHECK FOR LAST BYTE
0392 01AA 16EF          JNE  READ1              GO BACK IF NOT
0393 01AC C38E          MOV  R14,R14            CHECK ERASE FLAG
0394 01AE 1309          JEQ  PGM                IF NOT SET, GO COMPARE
0395 01B0 046D          B    @CMP(13)           OTHERWISE GO COMPARE
     01B2 0248'
0396                    ERASE1
0397 01B4 C198          MOV  *R8,R6             BUFFER MEMORY DATA
0398 01B6 5182          SZCB R2,R6              CHECK THAT BYTE CAN
0399                    *                       BE PROGRAMMED
0400 01B8 13F2          JEQ  MERGE1             IF SO, KEEP GOING
0401 01BA 070F          SETO R15                SET ERROR FLAG
0402 01BC 06AD          BL   @ERRMSG(13)        OTHERWISE PRINT ERROR
     01BE 0280'
0403 01C0 10EE          JMP  MERGE1             AND CONTINUE
0404                    *
0405                    **********PROGRAM ROUTINE**********
0406                    *
0407                    PGM
0408 01C2 C3CF          MOV  R15,R15            ERASE ERROR ?
0409 01C4 1302          JEQ  PGM1               IF NOT GO PROGRAM !!
0410 01C6 046D          B    @CKOK(13)          OTHERWISE END
     01C8 0278'
```

302 EPROM CHECK ROUTINE

```
0411                PGM1
0412 01CA 020F              LI   R15,DFLT4          GET 'BRANCH TO' ADD.
     01CC 01F6'
0413 01CE A3CD              A    R13,R15            ADD BASE ADDRESS
0414 01D0 C80F              MOV  R15,@AREA1         STORE IT
     01D2 FF8A
0415 01D4 020F              LI   R15,PGM1           GET 'BRANCH TO' ADD.
     01D6 01CA'
0416 01D8 A3CD              A    R13,R15            ADD BASE ADDRESS
0417 01DA C80F              MOV  R15,@AREA3         STORE IT
     01DC FF8E
0418 01DE 2FAD              PRNT @LPCNT(13)         PROMPT FOR LOOP COUNT
     01E0 0428'
0419 01E2 C067              MOV  @6(R7),R1          GET DEFAULT LOOP COUNT
     01E4 0006
0420 01E6 2FAD              PRNT @DFLT(13)          PRINT DEFAULT MESSAGE
     01E8 03D9'
0421 01EA 2E81              HEXO R1                 OUTPUT DEFAULT VALUE
0422 01EC 2FAD              PRNT @TIC(13)           *
     01EE 059A'
0423 01F0 2E41              HEXI R1                 GET VALUE
0424 01F2 FF88              DATA AREA
0425 01F4 FF8C              DATA AREA2
0426                DFLT4
0427 01F6 2FAD              PRNT @CRLF(13)          CLEAR PRINT LINE
     01F8 03B9'
0428 01FA C1A7              MOV  @4(R7),R6          GET DEFAULT BYTE COUNT
     01FC 0004
0429 01FE A184              A    R4,R6              R6 HAS EPROM END ADD.
0430                *                               EPM START+ # OF BYTES=END ADD.
0431 0200 020C              LI   R12,CRUADD         SET CRU BASE FOR EPM. ADD.
     0202 1720
0432                *                               SET PROGRAM ENABLE
0433 0204 1DF3              SBO  -13                BIT -13 IS PGMEN
0434 0206 C244              MOV  R4,R9              EPROM START ADD. +
0435 0208 A245              A    R5,R9              # OF BYTES TO BE MOVED =
0436                *                               LAST EPROM ADD.
0437 020A 2FAD              PRNT @PROG(13)          OUTPUT BANNER
     020C 0338'
0438 020E 04CA              CLR  R10                CLEAR LOOP COUNT
0439                CNLP2
0440 0210 804A              C    R10,R1             DONE YET ?
0441 0212 1419              JHE  ENDPGM             IF SO, QUIT !
0442 0214 C084              MOV  R4,R2              OTHERWISE, INIT.EPROM ADD.
0443 0216 C203              MOV  R3,R8              GET START ADDRESS
0444                CNLP1
0445 0218 3002              LDCR R2,0               OUTPUT 16 ADDRESS BITS
0446 021A 020C              LI   R12,CRUDTA         SET CRU BASE FOR EPM DATA
     021C 1710
0447 021E 3218              LDCR *R8,8              OUTPUT DATA
0448 0220 0588              INC  R8                 UPDATE ADDRESS
0449 0222 C000              MOV  R0,R0              IF IN BYTE MODE....
0450 0224 1301              JEQ  ROUND2             INC. BY ONE
0451 0226 0588              INC  R8                 OTHERWISE INC. BY TWO
```

302 EPROM CHECK ROUTINE

```
0452                    ROUND2
0453 0228 020C                  LI   12,CRUADD          SET CRU BASE FOR EPM ADD.
     022A 1720
0454 022C 8242                  C    R2,R9              IF TOO HIGH JUST DELAY
0455 022E 1401                  JHE  DLY                *
0456 0230 1DF4                  SBO  -12                SET PROGRAM PULSE
0457                    DLY
0458 0232 C2E7                  MOV  @8(R7),R11         LOAD PROG. TIME VALUE
     0234 0008
0459                    *                               PROVIDES A 1MS OR 50MS
0460                    *                               PULSE AT A SYSTEM CLOCK
0461                    *                               OF THREE MEGAHERTZ.
0462 0236 060B          DLY1    DEC  R11                DELAY LOOP
0463 0238 16FE                  JNE  DLY1
0464 023A 1EF4                  SBZ  -12                RESET PROGRAM PULSE
0465 023C 0582                  INC  R2                 UPDATE EPROM ADDRESS
0466 023E 8182                  C    R2,R6              DONE YET ?
0467 0240 1AEB                  JL   CNLP1              IF NOT, FINISH LOOP !
0468 0242 058A                  INC  R10                INCREMENT LOOP COUNT 2
0469 0244 10E5                  JMP  CNLP2              GO CHECK IF DONE !
0470                    ENDPGM
0471 0246 1EF3                  SBZ  -13                RESET PGMEN
0472                    *
0473                    **********COMPARE ROUTINE**********
0474                    *
0475                    CMP
0476 0248 2FAD                  PRNT @VER(13)           OUTPUT VERIFY MESSAGE
     024A 038B
0477 024C C203                  MOV  R3,R8              BUFFER START ADDRESS
0478 024E C285                  MOV  R5,R10             BUFFER BYTE COUNT
0479 0250 1313                  JEQ  CKOK               IF BYTE COUNT = 0
0480                    *                               DON'T CHECK ANYTHING
0481 0252 C244                  MOV  R4,R9              BUFFER EPROM ADD.
0482                    CNCK
0483 0254 04C2                  CLR  R2                 CLEAR DATA STORE REG.
0484 0256 020C                  LI   R12,CRUADD         SET CRU BASE FOR EPM ADD.
     0258 1720
0485 025A 3009                  LDCR R9,0               OUTPUT 16 BITS
0486 025C 020C                  LI   R12,CRUDTA         SET CRU BASE TO DATA
     025E 1710
0487 0260 3602                  STCR R2,8               READ EPROM DATA
0488 0262 9602                  CB   R2,*R8             COMPARE DATA
0489 0264 1302                  JEQ  BOK
0490 0266 06AD                  BL   @ERRMSG(13)        OUTPUT ERROR MESSAGE
     0268 0280
0491 026A 0588          BOK     INC  R8                 INC MEMORY ADR
0492 026C C000                  MOV  R0,R0              IF IN BYTE MODE...
0493 026E 1301                  JEQ  ROUND3             INC. BY ONE
0494 0270 0588                  INC R8                  OTHERWISE INC. BY TWO
0495                    ROUND3
0496 0272 0589                  INC  R9                 UPDATE EPROM ADDRESS
0497 0274 060A                  DEC  R10                DONE YET ?
0498 0276 16EE                  JNE  CNCK               GO BACK IF NOT COMPLETE
0499 0278 2FAD          CKOK    PRNT @TSKCMP(13)        OUTPUT END OF TASK MSG.
```

```
302 EPROM CHECK ROUTINE

      027A 03A4'
0500  027C 046D          B     @ABORT2(13)
      027E 0166'
0501                *
0502                *
0503                ERRMSG
0504  0280 2FAD          PRNT @ERROR(13)          PRINT ERROR MSG
      0282 05C9'
0505  0284 2FAD          PRNT @EPM(13)            PRINT 'EPM'
      0286 05B6'
0506  0288 2E89          HEXO R9                  OUTPUT EPROM ADDRESS
0507  028A 2FAD          PRNT @SPACE(13)          SPACE
      028C 05B2'
0508  028E 2FAD          PRNT @DATA(13)           PRINT 'DATA'
      0290 05C2'
0509  0292 0982          SRL  R2,8                SHIFT DATA FOR PRINT
0510  0294 2E82          HEXO R2                  OUTPUT DATA VALUE
0511  0296 2FAD          PRNT @SPACE(13)          SPACE
      0298 05B2'
0512                ERR2
0513  029A 2FAD          PRNT @SPACE(13)          SPACE
      029C 05B2'
0514  029E 2FAD          PRNT @MEM(13)            PRINT 'MEM'
      02A0 05BC'
0515  02A2 D098          MOVB *R8,R2              GET MEMORY DATA
0516  02A4 0982          SRL  R2,8                SHIFT FOR PRINT
0517  02A6 2E88          HEXO R8                  OUTPUT MEMORY ADDRESS
0518  02A8 2FAD          PRNT @SPACE(13)          SPACE
      02AA 05B2'
0519  02AC 2FAD          PRNT @DATA(13)           PRINT 'DATA'
      02AE 05C2'
0520  02B0 2E82          HEXO R2                  OUTPUT DATA
0521  02B2 2FAD          PRNT @CRLF(13)           CLEAR PRINT LINE
      02B4 03B9'
0522                *
0523                ******TEST FOR CHARACTER ENTRY, TO TERMINATE ERROR PRINT****
0524                *
0525                CHARCK
0526  02B6 C38C          MOV  R12,R14             SAVE R12 CONTENTS
0527  02B8 020C          LI   R12,CRU02           LOAD CRU BASE ADD. FOR 9902
      02BA 0080
0528  02BC 1F15          TB   21                  CHECK BUFFER FULL FLAG
0529  02BE 1302          JEQ  ROUND4              IF SET, END TEST
0530  02C0 C30E          MOV R14,R12              OTHERWISE RESTORE R12
0531  02C2 045B          RT                       AND RETURN
0532                ROUND4
0533  02C4 1E12          SBZ  18                  RESET 9902
0534  02C6 2FAD          PRNT @CRLF(13)           CLEAR PRINT LINE
      02C8 03B9'
0535  02CA 10D6          JMP CKOK
0536                *
```

302 EPROM CHECK ROUTINE

```
0538                    ************************
0539                    * TABLE OF PARAMETERS *
0540                    ************************
0541                    *
0542                    TABLE
0543                    *
0544                    *       DATA DEVICE, I.D. ,BYTE   ,PGM    ,TIME
0545                    *                         COUNT   LOOP    VALUE
0546                    *                                 COUNT
0547                    *
0548 02CC 2708                  DATA >2708,>0000,>0400,>0064,>0086
     02CE 0000
     02D0 0400
     02D2 0064
     02D4 0086
0549 02D6 2716                  DATA >2716,>4000,>0800,>0064,>0086
     02D8 4000
     02DA 0800
     02DC 0064
     02DE 0086
0550 02E0 2508                  DATA >2508,>2000,>0400,>0032,>0086
     02E2 2000
     02E4 0400
     02E6 0032
     02E8 0086
0551 02EA 2516                  DATA >2516,>2000,>0800,>0032,>0086
     02EC 2000
     02EE 0800
     02F0 0032
     02F2 0086
0552 02F4 2532                  DATA >2532,>6000,>1000,>0032,>0086
     02F6 6000
     02F8 1000
     02FA 0032
     02FC 0086
0553                    TABEND
0554                    ************************
0555                    *
```

302 EPROM CHECK ROUTINE

```
0557                    ****************************************************************
0558                    *SAGES  ***  MESSAGES  ***  MESSAGES  ***  MESSAGES  ***  ME
0559                    ****************************************************************
0560                    *
0561 02FE   43  JUMPER TEXT 'CHECK THE JUMPER POSITIONS ON'
0562 031B   20         TEXT ' THE EPPC.'
0563 0325   0D         BYTE >D,>A,>A,>0
     0326   0A
     0327   0A
     0328   00
0564 0329   31  DVCTYP TEXT '1. DEVICE TYPE'
0565 0337   00         BYTE 0
0566 0338   2A  PROG   TEXT '*** PROGRAMMING STARTED ***'
0567 0353   0D         BYTE >D,>A,>0
     0354   0A
     0355   00
0568 0356   2A  RD     TEXT '*** READ STARTED ***'
0569 036A   0D         BYTE >D,>A,>0
     036B   0A
     036C   00
0570 036D   2A  ERSCK  TEXT '*** ERASE CHECK STARTED ***'
0571 0388   0D         BYTE >D,>A,>0
     0389   0A
     038A   00
0572 038B   2A  VER    TEXT '*** VERIFY STARTED ***'
0573 03A1   0D         BYTE >D,>A,>0
     03A2   0A
     03A3   00
0574 03A4   2A  TSKCMP TEXT '*** TASK COMPLETE ***'
0575 03B9   0D  CRLF   BYTE >D,>A,>0
     03BA   0A
     03BB   00
0576 03BC   33  STRTAD TEXT '3. ENTER MEMORY START ADDRESS'
0577 03D9   2C  DFLT   TEXT ', (DEFAULT = '
0578 03E6   00         BYTE >0
0579 03E7   34  EPMAD  TEXT '4. ENTER EPROM START ADDRESS'
0580 0403   00         BYTE 0
0581 0404   2A  CONT   TEXT '*** ILLEGAL ENTRY, '
0582 0417   00         BYTE 0
0583 0418   43  CONT1  TEXT 'CONTINUE (Y/N) '
0584 0427   00         BYTE >0
0585 0428   37  LPCNT  TEXT '7. NO. OF PROGRAM LOOPS'
0586 043F   00         BYTE >0
0587 0440   35  LENGTH TEXT '5. NO. OF BYTES TO BE OPERATED ON'
0588 0461   00         BYTE >0
0589 0462   36  FCN    TEXT '6. FUNCTION (R/P/V) : '
0590 0478   00         BYTE >0
0591 0479   0D  HLIST  BYTE >D,>A,>A
     047A   0A
     047B   0A
0592 047C   20         TEXT ' DEVICES:'
0593 0486   0D         BYTE >D,>A
     0487   0A
0594 0488   32         TEXT '2508,2708,2516,2716,2532'
```

```
302 EPROM CHECK ROUTINE

 0595 04A0   0D          BYTE >D,>A
      04A1   0A
 0596 04A2   20          TEXT '  FUNCTIONS:'
 0597 04AE   0D          BYTE >D,>A
      04AF   0A
 0598 04B0   28          TEXT '(R) - READ, (P) - PROGRAM, (V) - VERIFY'
 0599 04D7   0D          BYTE >D,>A
      04D8   0A
 0600 04D9   20          TEXT '  MODES:'
 0601 04E1   0D          BYTE >D,>A
      04E2   0A
 0602 04E3   42          TEXT 'BYTE (B) OR WORD (W) '
 0603 04F8   0D          BYTE >D,>A
      04F9   0A
 0604 04FA   42          TEXT 'BYTE MODE TRANSFERS DATA BYTE BY BYTE.'
 0605 0520   0D          BYTE >D,>A
      0521   0A
 0606 0522   57          TEXT 'WORD MODE TRANSFERS ONLY UPPER OR LOWER'
 0607 0549   20          TEXT ' WORD HALVES.'
 0608 0556   0D          BYTE >D,>A,>A
      0557   0A
      0558   0A
 0609 0559   2A          TEXT '*** ALL INPUTS ARE IN HEX ***'
 0610 0576   0D          BYTE >D,>A,>A,>0
      0577   0A
      0578   0A
      0579   00
 0611 057A   2A  LARGE   TEXT '*** ENTRY TOO LARGE !!'
 0612 0590   00          BYTE 0
 0613 0591   20  MAX     TEXT ' ,MAX = '
 0614 0599   00          BYTE 0
 0615 059A   29  TIC     TEXT ') => '
 0616 059F   00          BYTE >0
 0617 05A0   0D  BYTWD   BYTE >D
 0618 05A1   32          TEXT '2. MODE (B/W) : '
 0619 05B1   00          BYTE 0
 0620 05B2   20  SPACE   TEXT '    '
 0621 05B5   00          BYTE 0
 0622 05B6   20  EPM     TEXT ' EPM '
 0623 05BB   00          BYTE 0
 0624 05BC   20  MEM     TEXT ' MEM '
 0625 05C1   00          BYTE 0
 0626 05C2   44  DATA    TEXT 'DATA= '
 0627 05C8   00          BYTE 0
 0628 05C9   2A  ERROR   TEXT '*** ERROR '
 0629 05D3   00          BYTE >0
 0630                    ****************
 0631                    * END MESSAGES *
 0632                    ****************
 0633                    END  START

0000 ERRORS
```

```
TXXREF 937542 *A     16:06:56     310/78     PAGE 0001

ABORT    0351       0223  0348
ABORT2   0353       0500
AREA     0052       0053  0054  0055  0181  0208  0269  0288  0322  0424
AREA1    0053       0197  0259  0281  0414
AREA2    0054       0182  0207  0268  0289  0321  0425
AREA3    0055       0201  0262  0304  0417
B        0031       0246
BADID    0228
BCMD     0048       0180
BOK      0491       0489
BYTWD    0617       0239
CHARCK   0525
CKOK     0499       0410  0479  0535
CMP      0475       0346  0395
CNCK     0482       0498
CNLP1    0444       0467
CNLP2    0439       0469
CONT     0581       0352
CONT1    0583       0353
CRLF     0575       0210  0263  0278  0301  0324  0342  0355  0427  0521
                    0534
CRU02    0042       0527
CRUADD   0045       0377  0431  0453  0484
CRUDTA   0043       0379  0446  0486
CRUID    0047       0212
DATA     0626       0508  0519
DFLT     0577       0203  0284  0316  0420
DFLT0    0209       0198
DFLT1    0277       0260  0311
DFLT2    0300       0279  0330
DFLT3    0323       0302
DFLT4    0426       0412
DLY      0457       0455
DLY1     0462       0463
DVCTYP   0564       0202
ECHO                0007
ENDPGM   0470       0441
EPM      0622       0505
EPMAD    0579       0283
ERASE1   0396       0382
ERR2     0512
ERRMSG   0503       0402  0490
ERROR    0628       0504
ERSCK    0570       0367
EXIT     0360       0357
FCN      0589       0340
FUNCT    0337
FUNCT0   0335       0326
GET      0256       0245  0257
HEXI                0005
HEXO                0006
HLIST    0591       0179
IDCK     0225       0220
JUMPER   0561       0229
LARGE    0611       0308  0327
```

```
TXXREF 937542 *A     16:06:56     310/78     PAGE 0003

START   0160        0164   0169   0633
STARTX  0178
STRTAD  0576        0264
TABEND  0553        0216
TABLE   0542        0184   0214
TIBUG   0004        0361
TIC     0615        0205   0266   0286   0318   0422
TSKCMP  0574        0499
TSTWS   0050        0163
TYPE    0193        0195   0230   0358
V       0034        0345
VER     0572        0476
W       0035        0244
Y       0036        0356

THERE ARE 0109 SYMBOLS
```

TEXAS INSTRUMENTS
INCORPORATED
Semiconductor Group
Post Office Box 1443, Houston, Texas 77001

MPB 12 Rev, *
1602028-9701

Printed in U.S.A.