Z-80 ASSEMBLER II

1680

the digital group

po box 6528 denver, colorado 80206 (303) 777-7133

296-000-A-17

FIXES TO ASM II/32:

LOC		WAS	CHANGE TO
H	L		
Ø11	Ø72	Ø52 352 Ø11	315 Ø3Ø Ø14
Ø14	Ø3Ø	0——0	Ø52 352 Ø11 Ø53 311

Z-80 ASSEMBLER II

The Digital Group Z-80 Assembler II provides the means to translate programs written in symbolic machine language into the precise numeric language required by the computer.

This Assembler allows the programmer to:

- Specify machine instructions symbolically
- Represent storage locations by alphameric symbols of his choice
- Define data areas and constants
- Control the operation of the Assembler

The Assembler program itself translates the symbolic instructions into machine code, assigns storage locations, and generates the user's executable machine language program. Machine instructions are symbolically represented as Assembler language statements. Instruction formats are as defined in the *Zilog Z-80 CPU Technical Manual* shipped with each Digital Group Z-80 system.

This Assembler uses an extensive line-oriented editor which automatically generates line numbers and automatically tabs to the proper field. Other features of the Assembler are:

- Auto file management
- Source code write on cassette
- Object code write on cassette
- Source code read from cassette
- Resequence line numbers (all lines, block of lines)
- File listing (all lines, single line, block of lines)
- Line deletion (single line, block of lines)
- Linking ability between files
- Plain English error messages
- Split octal, true octal, hexadecimal, decimal, and ASCII constants
- Relative addressing
- Source code reduction
- File merge
- Octal or hexadecimal listings

System Required to Run this Assembler

- Digital Group Z-80 system with 18K or more memory
- Digital Group Audio Cassette interface (standard with most systems)
- Digital Group 512 or 1024 character TV Readout board
- Standard ASCII keyboard attached to Port Ø. Upper and lower case character set, control codes, and cursor keys helpful, but not required.

General Assembler Design

The Assembler program occupies from address 0 to split octal address 057377 (2FFF in Hex). The programmer's source and object code may be placed at any address above the Assembler (060000 Octal or 3000 Hex). The exact memory utilization is:

000000 - 000377	EROM
001000 - 005377	General Z-80 OP SYS
006000 - 010377	Optional hardcopy area
011000 - 015377	TV scroll area
016000 - 060000	ASSEMBLER II
060000 - 377377	User area for source and object files

Several versions of page 6 have already been written for various hardcopy devices. The 256 bytes in this area should be adequate for most hardcopy devices.

Using the Assembler

The Digital Group Z-80 Assembler II is distributed on an audio cassette recorded in 1100 baud Suding format. The 64 character version is contained in the first audio file. The 32 character version is in the second file. It is recommended that working copies of the program be made from the distribution tape.

1. Place the cassette containing the Assembler program in the cassette recorder. When the low tone begins, turn on your Digital Group Z-80 system.

2. When the data burst ends, the system will display an option select list. A modified version of the Z-80 OP System is utilized for options 3 and 4 (see Appendix A).

3. Option 7 begins the Assembler operation. If hardcopy is desired, option 8 may be selected, but an initial selection of hardcopy wastes paper and slows the operation. Option 8 is recommended for source code listing of large programs and assembly listings.

4. The screen will initially display "Assembler II" followed by

   ```
   FILE 060000 060000
   READY
   ```

   and an underline character (_) at the beginning of the next line.

5. Precise entries to the Assembler are required to obtain proper results. The following abbreviations will be referred to in the text:

   (sp) is a space bar depression.
   (cr) is a carriage return key depression.
   (X *ctl*) is a letter X entry while the "Control" key is depressed (deletes the line of characters currently being entered).
   (@ *ctl*) is a letter @ entry while the "Control" key is depressed (exits from the auto numbering mode).
   ! begins the auto numbering mode (with a line number of 0100).
   (RUB) is a delete key depression (backspaces to delete the previous character just entered — used for error correction).

   The scrolling system's input routine automatically modifies any lower case alpha entry to upper case.

   Note: Certain illegal operation codes or invalid instruction formats may cause unpredictable and sometimes catastrophic results. The *Zilog Z-80 CPU Technical Manual* should be referenced for correct op codes and instruction formats.

6. The system is now ready to assist you in building your source file. To begin auto numbering type the character "!".

7. The system responds with 0100; you may now build your program. The system has been designed so that five fields are utilized:

   The **number** field contains a four digit number between 0001 and 9999. The auto numbering system increments by ten. The number entry *must* contain four digits.

   The **label** field is optional. If desired, an entry consisting of 1-6 alphanumeric characters, the first of which is alpha, may be placed in this field. A statement which begins with an * in this field will be treated as a comment statement.

   The **operation** field must be one of the various Z-80 op codes shown in the *Zilog Z-80 CPU Technical Manual*, or one of the Pseudo ops described below.

   The **operand** field is required for most, but not all, op codes. The included listing shows samples. Symbolic labels may be utilized if desired.

   A **comment** may be included if desired. A short, meaningful comment can be a great help for later analysis of the function of each instruction.

   As statements are being typed, depressing the space bar will automatically position the cursor at the start of the next field. Corrections may be made by depressing the RUB key to backspace one character or by (X *ctl*) which deletes the whole line.

   A (cr) terminates the line entry, and the system will automatically enter the next line number. If no further entry is desired, an (@ *ctl*) will exit from the auto numbering system.

8. After the source file has been entered, several options are open to the programmer prior to assembling the source file.

   LIST (cr) results in a listing of the source file you have been building.

   RSEQ (sp) xxxx (sp) yy (cr)
   reorders source statement numbering to start at line xxxx and increment in yy steps. RSEQ entered without operands will renumber starting at 0100 in 10 steps. RSEQ (sp) xxxx (cr) will renumber starting at line xxxx in 10 steps.

   SPLIT (cr) resequences part of a source file. The assembler will display the following prompting messages on the screen:
   START LINE #Type in the first line number desired to be changed and (cr). (Leading zeros need not be typed.)
   END LINE #Type in the last line number desired to be changed and (cr).
   NEW START #Type in the new number for first resequenced line and (cr).
   STEPType in the offset number of lines between one line and the next and (cr).
   Note: SPLIT will not reorganize the source file, but will only "rename" the line numbers.

   HEX (cr) displays all output in a hexadecimal mode.

   OCT (cr) displays all output in a split octal mode.

9. The assembly of the source code can take five forms.

   ASSM (cr) assembles the source file starting at the first available address.

   ASSME (cr) is similar to ASSM (cr) format, but only errors are listed.

   ASSME (sp) xxxx (sp) yyyy (cr)
   is the same as ASSME except lines xxxx to yyyy plus the rest of the block will be printed.

   ASSML (cr) permits linking the assembly of a number of source files through a common label table. Common labels must contain a period (e.g. LAB2. or LAB2.3). This permits utilizing the same limited amount of memory for each source file. The cassette recorder is utilized to SAVE and LOAD the various source files. The ASSML instruction then builds the desired object code.

   ASSMLE (cr)
   is similiar to ASSML format, but only errors are listed.

For example, suppose a user wishes to assemble two source files, TEST1 and TEST2, each requiring 4K of storage. The user has an 18K system, so making them into a single 8K source file (plus the 12K of the Assembler code) would exceed the storage capacity of the system. TEST1 references labels in TEST2 and vice versa. The user would assemble TEST1 using the ASSM instruction. "LABEL" errors will result, but are temporarily ignored. The TEST1 source file is then SAVEd on a cassette. TEST2 is next made current and LOADed. This time the user assembles TEST2 with the ASSML instruction. There should be no "LABEL" errors on this run if this is the last source file. TEST1 is then made current and reLOADed. Reassemble TEST1 using ASSML. The assembly is now complete.

Other Operations

Several other features of the Z-80 Assembler II are as follows:

LOADS (cr) loads source code from cassette which was saved under SAVES.

SAVES (cr) saves a source file on cassette.

SAVEO (cr) saves an object file on cassette (if no ORG statement was used).

EXEC (cr) or EXEC (sp) HHHH (cr) or EXEC (sp) 000000 (cr)
causes the Assembler to branch to the indicated address. This address is normally the beginning address of the object code generated by the assembly operation. If no address is present then the editor will jump to the first address after the source file to be executed. This is very useful when an ASSM has been generated without ST or ORG statements in the source file.

LTABL (cr) prints the label table data for the last assembly in alphabetical order (do not use before assembly).

NEWF (cr) clears all RAM source areas and label table areas. While not generally necessary, this operation is recommended when running successive assemblies which are not linked.

ZERO (cr) zeros memory from the end of the file to the top of memory.
CAUTION: This command destroys the label table. This is especially important for ASSML runs.

MERGE (cr) merges a file on tape with a file in memory. The file on tape is merged by line number, with no check for duplicate line numbers.

Arguments

Arguments consist of register names, conditions (for CALL, JR,JP, and RET), constants, and variables, with or without offset.

Register names: A,B,C,D,E,H,L,AF,BC,DE,HL,SP,IX,IY. On a LD operation M may be used instead of (HL).

Conditions: Z,NZ,C,NC,PO,PE,P,M.

Constants:	Split Octal	Indicator: None Default if no other indicator is specified. Example: 000377
	True Octal	Indicator: Q Example: 1000Q (equivalent to 002000, split octal).
	Hexadecimal	Indicator: H Example: 0FC32H (Must start with a numeric character.)
	Decimal	Indicator: D Example: 12357D
	ASCII	Indicator: Single quotes (MSB = 1) Example: 'B' (equivalent to 302, octal) Indicator: Double quotes (MSB = 0) Example: "B" (equivalent to 102, octal).

Variables: The $ sign is used to indicate the 16 bit value of the program counter at the first byte of code on the line. Values may be added to or subtracted from this point.

The "less than" sign (<) preceding a label indicates the least significant 8 bits. The "greater than" (>) may be used to indicate the most significant 8 bits. Example:

```
 LABEL = 123345
<LABEL = 000345
>LABEL = 123000
```

All constants and variables may be combined by + or - signs up to a limit of 64 characters in one line.

Pseudo Ops

ST resets the original pointer of the assembly, but will not put the code there. (This pseudo op is very useful when an assembly is desired for an area in which the Assembler itself resides or when the desired memory is not on the system performing the assembly.) To recover this object code, SAVEO command should be used. Example:

```
0100            ST    005000
```

ORG resets the origin of the assembly at that point to the value indicated. Examples:

```
0100            ORG   4000H    HEX ORG
0110            ORG   100123   OCTAL ORG
```

EQU gives a value to a symbol. Examples:

```
0120   THREE    EQU   003
0130   ERASE    EQU   000346
```

DC, DB defines a data constant.

DS leaves the indicated number of bytes unchanged. Example:

```
0140   BUFFER   DS    512D
```

DW defines a number or a label without the single quote marks. A 16-bit number is stored corresponding to the address of the label or the value of the number. Examples:

```
0170   HERE     DW    123321
0180   WHERE    DW    THERE
```

END indicates end of the assembly.

Error Messages

Error diagnostics are given in plain English after each error is encountered. Some messages will occur at command times such as:

???&?	The command given was illegal.
PROGRAM TOO LONG	The source code does not allow enough space in memory for the object code to be placed. Object code too long.
LINE NUMBER TOO LONG	Line numbers only allowed to 9999.

Saving the Object Code on Cassette

Often the Asembler will be used to generate object code designed to be run immediately with the Assembler and source file resident. In this case, no special object code saving is necessary. However, the code will generally be designed to run in the storage area occupied by the Assembler, hence, temporary cassette loading may be desirable. Similarly, large programs may be constructed from smaller object code modules. The main operating system can be used as the new system's operating system, or the Assembler can be used to generate a "new" operating system.

Either way, the programmer will generally temporarily put the new data on cassette. If an ORG statement is used then the programmer must use the basic cassette reading system in the EROM area (000000 to 000377). Otherwise, the SAVEO command will allow the programmer to save off the object code directly onto cassette. To use SAVEO, type this in as a command, start the tape recorder, hit the carriage return. The screen will blank with the word WRITING appearing on it. When it is finished writing object code onto your cassette, the screen will come back with the SAVEO typed plus the starting and ending addresses of the object code. The programmer can now load this object code into another program to build the file that was being worked on. This will work with an ST command or without an ST or ORG command. However, if more than one ST command is used then the program should be broken into smaller pieces and each should be saved independently of the others.

ASSEMBLER II REFERENCE SHEET

EXECUTIVE COMMANDS

LIST	LIST ALL SOURCE STATEMENTS
LIST X	LIST LINE X
LIST X Y	LIST FROM LINE X TO LINE Y
ASSM	ASSEMBLE SOURCE STATEMENTS
ASSME	ASSEMBLE, BUT LIST ONLY STATEMENTS WITH ERRORS
ASSME X Y	ASSEMBLE, LIST ONLY LINES X TO Y AND ERRORS
ASSML	ASSEMBLE, KEEPING IN LABEL TABLE ANY LABEL CONTAINING A PERIOD
ASSMLE	SAME AS ASSML, BUT LISTING ONLY ERRORS
NEWF	CLEAR EXISTING FILE
DELT X	DELETE LINE X
DELT X Y	DELETE FROM LINE X TO LINE Y
RSEQ	RESEQUENCE SOURCE WITH STARTING LINE 1ØØ, 1Ø STEPS BETWEEN LINES
RSEQ X	RESEQUENCE SOURCE WITH STARTING LINE X, 1Ø STEPS BETWEEN LINES
RSEQ X Y	RESEQUENCE SOURCE WITH STARTING LINE X, Y STEPS BETWEEN LINES
SPLIT	RESEQUENCE PART OF SOURCE CODE, BY LINE NUMBER
HEX	DISPLAY ALL OUTPUT IN HEXADECIMAL MODE
OCT	DISPLAY ALL OUTPUT IN SPLIT OCTAL MODE
SAVES	SAVE SOURCE CODE ON CASSETTE
SAVEO	SAVE OBJECT CODE ON CASSETTE
LOADS	LOAD SOURCE CODE FROM CASSETTE
EXEC	EXECUTE OBJECT CODE - ASSUMES NO 'ORG' OR 'ST' STATEMENT USED
EXEC X	EXECUTE OBJECT CODE - X=ADDRESS IN HEX (4 DIGITS) OR OCT (6 DIGITS) - NEITHER WITH AN 'H' OR 'Q' SUFFIX
LTABL	PRINT LABEL TABLE
MERGE	MERGES EXISTING FILE TO FILE COMING IN FROM CASSETTE SIMILAR TO LOADS, BUT MERGES WITH EXISTING FILE BY LINE NUMBER
ZERO	ZERO ALL MEMORY ABOVE SOURCE FILE

VARIABLES

123345	SPLIT OCTAL
1234H	HEXADECIMAL
123456Q	TRUE OCTAL
123446D	DECIMAL
'A'	ASCII VALUE OF A, MSB=1
"A"	ASCII VALUE OF A, MSB=Ø
$	VALUE OF FIRST BYTE ON LINE WHERE $ EXISTS - CHANGES FOR EVERY LINE
LABEL	VALUE OF LABEL
<LABEL	LEAST SIGNIFICANT BYTE OF LABEL
>LABEL	MOST SIGNIFICANT BYTE OF LABEL
+ OR -	ALL EXPRESSIONS MAY HAVE AS MANY + or - AS BUFFER ALLOWS (64 CHAR/LINE)

PSEUDO OPS

ST X	ASSEMBLE CODE FOR ADDRESS X, BUT DO NOT PUT CODE THERE
ORG X	ASSEMBLE CODE FOR ADDRESS X, AND PUT CODE THERE
DC	DEFINE CONSTANT (8 BIT VALUE)
DB	SAME AS DC DB OR DC ARGUMENTS ARE SEPARATED BY COMMAS. STRINGS OF ASCII CHARACTERS MAY BE QUOTED WITH SINGLE QUOTE (') OR DOUBLE QUOTES (").
DW	DEFINE WORD (16 BIT VALUE) MAY BE SEPARATED BY COMMAS
DS	DEFINE STORAGE. LEAVES ARGUMENT VALUE OF BYTES UNCHANGED.
END	STOPS ASSEMBLY AT THIS POINT. NOT NECESSARY IF COMPLETE ASSEMBLY DESIRED.

```
FILE  060000  060000
READY
0100 KEY     IN    0   GET DATA FROM PORT 0
0110         BIT  7,A
0120         JR   Z,KEY
0130         PUSH AF .  SAVE DATA
0140 STROBE IN   000
0150         BIT  7,A
0160         JR   Z,KEY
0170         POP  AF    RESTORE DATA
0180         RET

FILE  060000  060256
READY
0090 * KEYBOARD INPUT SUBROUTINE :
0160         JR   NZ,KEY
RSEQ


FILE  060000  060321
READY
LIST
0100 * KEYBOARD INPUT SUBROUTINE
0110 KEY     IN    0   GET DATA FROM PORT 0
0120         BIT  7,A
0130         JR   Z,KEY
0140         PUSH AF    SAVE DATA
0150 STROBE IN   000
0160         BIT  7,A
0170         JR   NZ,KEY
0180         POP  AF    RESTORE DATA
0190         RET

FILE  060000  060321
READY
ASSM

060322                   0100 * KEYBOARD INPUT SUBROUTINE :
060322 333 000           0110 KEY     IN    0   GET DATA FROM PORT 0
060324 313 177           0120         BIT  7,A
060326 050 372           0130         JR   Z,KEY
060330 365               0140         PUSH AF    SAVE DATA :
060331 333 000           0150 STROBE IN   000
060333 313 177           0160         BIT  7,A
060335 040 363           0170         JR   NZ,KEY
060337 361               0180         POP  AF    RESTORE DATA
060340 311               0190         RET

NO ERRORS FOUND

FILE  060000  060321
READY
HEX
FILE  3000  30D1
READY
ASSM

30D2               0100 * KEYBOARD INPUT SUBROUTINE
30D2 DB 00         0110 KEY     IN    0   GET DATA FROM PORT 0
30D4 CB 7F         0120         BIT  7,A
30D6 28 FA         0130         JR   Z,KEY
30D8 F5            0140         PUSH AF    SAVE DATA
30D9 DB 00         0150 STROBE IN   000
30DB CB 7F         0160         BIT  7,A
30DD 20 F3         0170         JR   NZ,KEY
30DF F1            0180         POP  AF    RESTORE DATA
30E0 C9            0190         RET

NO ERRORS FOUND

FILE  3000  30D1
READY
0050         ST   4000H
ASSM

4000               0050         ST   4000H
4000               0100 * KEYBOARD INPUT SUBROUTINE
4000 DB 00         0110 KEY     IN    0   GET DATA FROM PORT 0
4002 CB 7F         0120         BIT  7,A
4004 28 FA         0130         JR   Z,KEY
4006 F5            0140         PUSH AF    SAVE DATA
4007 DB 00         0150 STROBE IN   000
4009 CB 7F         0160         BIT  7,A
400B 20 F3         0170         JR   NZ,KEY
400D F1            0180         POP  AF    RESTORE DATA
400E C9            0190         RET

NO ERRORS FOUND

FILE  3000  30E0
READY
SPLIT
START LINE#
150
END LINE#
170
NEW START#
165
STEP
1


FILE  3000  30E0
READY
LIST
0050         ST   4000H
0100 * KEYBOARD INPUT SUBROUTINE
0110 KEY     IN    0   GET DATA FROM PORT 0
0120         BIT  7,A
0130         JR   Z,KEY
0140         PUSH AF    SAVE DATA
0165 STROBE IN   000
0166         BIT  7,A
0167         JR   NZ,KEY
0180         POP  AF    RESTORE DATA
0190         RET

FILE  3000  30E0
READY
RSEQ 9000 1


FILE  3000  30E0
READY
ASSM

NO ERRORS FOUND
```

3.2

```
FILE  3000  30E0  :
READY
LIST
9000          ST   4000H
9001 * KEYBOARD INPUT SUBROUTINE
9002 KEY      IN   0   GET DATA FROM PORT 0
9003          BIT  7,A
9004          JR   Z,KEY
9005          PUSH AF    SAVE DATA
9006 STROBE   IN   000
9007          BIT  7,A
9008          JR   NZ,KEY
9009          POP  AF    RESTORE DATA :
9010          RET :

FILE  3000  30E0  :
READY
DELT 9006 9009 :

FILE  3000  3094
READY
LIST
9000          ST   4000H
9001 * KEYBOARD INPUT SUBROUTINE
9002 KEY      IN   0   GET DATA FROM PORT 0
9003          BIT  7,A
9004          JR   Z,KEY
9005          PUSH AF    SAVE DATA
9010          RET :

FILE  3000  3094  :
READY
LIST 9001
9001 * KEYBOARD INPUT SUBROUTINE

LIST 9003 9005 :
9003          BIT  7,A
9004          JR   Z,KEY
9005          PUSH AF    SAVE DATA

ASSME


NO ERRORS FOUND :

FILE  3000  3094
READY
SAVED 4000  4007
FILE  3000  3094  :
READY
LTABL
KEY    4000    :
FILE  3000  3094
READY
OCT
FILE  060000  060224
READY
LTABL
KEY    100000    :
FILE  060000  060224
READY
NEWF

FILE  060000  060000
READY
0100 TEST     ST   4000H
0110          LD   HL,123345
0120          LD   HL,1234H
0130          LD   HL,1234560
0140          LD   HL,12789D
0150          LD   HL,$
0160 TEST1    LD   A,'A'
0170          LD   A,"A"
0180          LD   A,<TEST1
0190 TEST2    LD   HL,TEST1
0200          LD   HL,<TEST1
0210          LD   HL,>TEST1
0220          LD   A,TEST2-TEST1
```

3-4

```
FILE  060000  060371
READY
ASSM

100000                     0100 TEST    ST    4000H
100000 041 345 123         0110         LD    HL,123345
100003 041 064 022         0120         LD    HL,1234H
100006 041 056 247         0130         LD    HL,1234560
100011 041 365 061         0140         LD    HL,12789D
100014 041 014 100         0150         LD    HL,$
100017 076 301             0160 TEST1   LD    A,'A'
100021 076 101             0170         LD    A,"A"
100023 076 017             0180         LD    A,<TEST1
100025 041 017 100         0190 TEST2   LD    HL,TEST1
100030 041 017 000         0200         LD    HL,<TEST1
100033 041 000 100         0210         LD    HL,>TEST1
100036 076 006             0220         LD    A,TEST2-TEST1 :

NO ERRORS FOUND

FILE  060000  060371
READY
```

```
FILE  060000  117360
READY
ASSM

120000                  0100        ORG  120000
120000                  0110 *
120000                  0120 *8 BIT LOAD GROUP
120000                  0130 TEST   EQU  1234H
120000                  0140 THREE  EQU  3
120000 355 127          0150        LD   A,I
120002 355 137          0160        LD   A,R
120004 177              0170        LD   A,A
120005 170              0180        LD   A,B
120006 171              0190        LD   A,C
120007 172              0200        LD   A,D
120010 173              0210        LD   A,E
120011 174              0220        LD   A,H
120012 175              0230        LD   A,L
120013 176              0240        LD   A,(HL)
120014 012              0250        LD   A,(BC)
120015 032              0260        LD   A,(DE)
120016 335 176 002      0270        LD   A,(IX+2)
120021 375 176 003      0280        LD   A,(IY+THREE) :
120024 072 064 022      0290        LD   A,(TEST)
120027 076 003          0300        LD   A,THREE
120031 107              0310        LD   B,A
120032 100              0320        LD   B,B
120033 101              0330        LD   B,C
120034 102              0340        LD   B,D
120035 103              0350        LD   B,E
120036 104              0360        LD   B,H
120037 105              0370        LD   B,L
120040 106              0380        LD   B,(HL)
120041 335 106 002      0390        LD   B,(IX+2)
120044 375 106 003      0400        LD   B,(IY+THREE) :
120047 006 003          0410        LD   B,THREE
120051 117              0420        LD   C,A
120052 110              0430        LD   C,B
120053 111              0440        LD   C,C
120054 112              0450        LD   C,D
120055 113              0460        LD   C,E
120056 114              0470        LD   C,H
120057 115              0480        LD   C,L
120060 116              0490        LD   C,(HL)
120061 335 116 002      0500        LD   C,(IX+2)
120064 375 116 003      0510        LD   C,(IY+THREE) :
120067 016 003          0520        LD   C,THREE
120071 127              0530        LD   D,A
120072 120              0540        LD   D,B
120073 121              0550        LD   D,C
120074 122              0560        LD   D,D
120075 123              0570        LD   D,E
120076 124              0580        LD   D,H
120077 125              0590        LD   D,L
120100 126              0600        LD   D,(HL)
120101 335 126 002      0610        LD   D,(IX+2)
120104 375 126 003      0620        LD   D,(IY+3)
120107 026 003          0630        LD   D,THREE
120111 137              0640        LD   E,A
120112 130              0650        LD   E,B
120113 131              0660        LD   E,C
120114 132              0670        LD   E,D
120115 133              0680        LD   E,E
120116 134              0690        LD   E,H
```

```
120117 135              0700        LD   E,L
120120 136              0710        LD   E,(HL)
120121 335 136 002      0720        LD   E,(IX+2)
120124 375 136 003      0730        LD   E,(IY+3)
120127 036 003          0740        LD   E,3
120131 147              0750        LD   H,A
120132 140              0760        LD   H,B
120133 141              0770        LD   H,C
120134 142              0780        LD   H,D
120135 143              0790        LD   H,E
120136 144              0800        LD   H,H
120137 145              0810        LD   H,L
120140 146              0820        LD   H,(HL)
120141 335 146 002      0830        LD   H,(IX+2)
120144 375 146 003      0840        LD   H,(IY+THREE)
120147 046 003          0850        LD   H,THREE
120151 157              0860        LD   L,A
120152 150              0870        LD   L,B
120153 151              0880        LD   L,C
120154 152              0890        LD   L,D
120155 153              0900        LD   L,E
120156 154              0910        LD   L,H
120157 155              0920        LD   L,L
120160 156              0930        LD   L,(HL)
120161 335 156 002      0940        LD   L,(IX+2)
120164 375 156 003      0950        LD   L,(IY+THREE)
120167 056 003          0960        LD   L,3
120171 167              0970        LD   (HL),A
120172 160              0980        LD   (HL),B
120173 161              0990        LD   (HL),C
120174 162              1000        LD   (HL),D
120175 163              1010        LD   (HL),E
120176 164              1020        LD   (HL),H
120177 165              1030        LD   (HL),L
120200 066 003          1040        LD   (HL),THPEE
120202 002              1050        LD   (BC),A
120203 002              1060        LD   (BC),A
120204 335 167 002      1070        LD   (IX+2),A
120207 335 160 002      1080        LD   (IX+2),B
120212 335 161 002      1090        LD   (IX+2),C
120215 335 162 002      1100        LD   (IX+2),D
120220 335 163 002      1110        LD   (IX+2),E
120223 335 164 002      1120        LD   (IX+2),H
120226 335 165 002      1130        LD   (IX+2),L
120231 335 066 002 003  1140        LD   (IX+2),THREE :
120235 375 167 003      1150        LD   (IY+3),A
120240 375 160 003      1160        LD   (IY+3),B
120243 375 161 003      1170        LD   (IY+3),C
120246 375 162 003      1180        LD   (IY+3),D
120251 375 163 003      1190        LD   (IY+3),E
120254 375 164 003      1200        LD   (IY+3),H
120257 375 165 003      1210        LD   (IY+3),L
120262 375 066 003 003  1220        LD   (IY+3),THREE
120266 062 064 022      1230        LD   (TEST),A
120271 355 107          1240        LD   I,A
120273 355 117          1250        LD   R,A
120275                  1260 * END 8 BIT LOAD
120275                  1270 *
120275                  1280 * BEGIN 16 BIT LOAD
120275 361              1290        POP  AF
120276 001 064 022      1300        LD   BC,TEST
120301 355 113 064 022  1310        LD   BC,(TEST)
120305 301              1320        POP  BC
120306 021 064 022      1330        LD   DE,TEST
120311 355 133 064 022  1340        LD   DE,(TEST)
120315 321              1350        POP  DE
120316 041 064 022      1360        LD   HL,TEST
120321 052 064 022      1370        LD   HL,(TEST)
```

11-1

171_8 =

11-2

01 111 001
F 9

```
120324 341              1380         POP  HL
120325 371              1390         LD   SP,HL
120326 335 371          1400         LD   SP,IX
120330 375 371          1410         LD   SP,IY
120332 061 064 022      1420         LD   SP,TEST
120335 355 173 064 022  1430         LD   SP,(TEST)
120341 335 041 064 022  1440         LD   IX,TEST
120345 335 052 064 022  1450         LD   IX,(TEST)
120351 335 341          1460         POP  IX
120353 375 041 064 022  1470         LD   IY,TEST
120357 375 052 064 022  1480         LD   IY,(TEST)
120363 375 341          1490         POP  IY
120365 355 103 064 022  1500         LD   (TEST),BC
120371 355 123 064 022  1510         LD   (TEST),DE
120375 042 064 022      1520         LD   (TEST),HL
121000 355 163 064 022  1530         LD   (TEST),SP
121004 335 042 064 022  1540         LD   (TEST),IX
121010 375 042 064 022  1550         LD   (TEST),IY
121014 365              1560         PUSH AF
121015 305              1570         PUSH BC
121016 325              1580         PUSH DE
121017 345              1590         PUSH HL
121020 335 345          1600         PUSH IX
121022 375 345          1610         PUSH IY
121024                  1620 * END OF 16 BIT LOAD
121024                  1630 *
121024                  1640 * END OF LOAD TEST
121024                  1650 * EXCHANGES
121024                  1660 *
121024 010              1670         EX   AF,AF
121025 331              1680         EXX
121026 353              1690         EX   DE,HL
121027 343              1700         EX   (SP),HL
121030 335 343          1710         EX   (SP),IX
121032 375 343          1720         EX   (SP),IY
121034                  1730 *
121034                  1740 * BLOCK TRANSFER
121034                  1750 *
121034 355 240          1760         LDI
121036 355 260          1770         LDIR
121040 355 250          1780         LDD
121042 355 270          1790         LDDR
121044                  1800 *
121044                  1810 * BLOCK SEARCH
121044                  1820 *
121044 355 241          1830         CPI
121046 355 261          1840         CPIR
121050 355 251          1850         CPD
121052 355 271          1860         CPDR
121054                  1870 *
121054                  1880 * 8 BIT ARITHMETIC & LOGIC
121054 207              1890         ADD  A
121055 200              1900         ADD  B
121056 201              1910         ADD  C
121057 202              1920         ADD  D
121060 203              1930         ADD  E
121061 204              1940         ADD  H
121062 205              1950         ADD  L
121063 206              1960         ADD  (HL)
121064 335 206 002      1970         ADD  (IX+2)
121067 375 206 003      1980         ADD  (IY+THREE)
121072 306 003          1990         ADD  THREE
121074 217              2000         ADC  A
121075 210              2010         ADC  B
121076 211              2020         ADC  C
121077 212              2030         ADC  D
121100 213              2040         ADC  E
```

```
121101 214              2050         ADC  H
121102 215              2060         ADC  L
121103 216              2070         ADC  (HL)
121104 335 216 002      2080         ADC  (IX+2)
121107 375 216 003      2090         ADC  (IY+THREE)
121112 316 003          2100         ADC  THREE
121114 227              2110         SUB  A
121115 220              2120         SUB  B
121116 221              2130         SUB  C
121117 222              2140         SUB  D
121120 223              2150         SUB  E
121121 224              2160         SUB  H
121122 225              2170         SUB  L
121123 226              2180         SUB  (HL)
121124 335 226 002      2190         SUB  (IX+2)
121127 375 226 003      2200         SUB  (IY+THREE)
121132 326 003          2210         SUB  THREE
121134 237              2220         SBC  A
121135 230              2230         SBC  B
121136 231              2240         SBC  C
121137 232              2250         SBC  D
121140 233              2260         SBC  E
121141 234              2270         SBC  H
121142 235              2280         SBC  L
121143 236              2290         SBC  (HL)
121144 335 236 002      2300         SBC  (IX+2)
121147 375 236 003      2310         SBC  (IY+3)
121152 336 003          2320         SBC  THREE
121154 247              2330         AND  A
121155 240              2340         AND  B
121156 241              2350         AND  C
121157 242              2360         AND  D
121160 243              2370         AND  E
121161 244              2380         AND  H
121162 245              2390         AND  L
121163 246              2400         AND  (HL)
121164 335 246 002      2410         AND  (IX+2)
121167 375 246 003      2420         AND  (IY+3)
121172 346 003          2430         AND  3
121174 257              2440         XOR  A
121175 250              2450         XOR  B
121176 251              2460         XOR  C
121177 252              2470         XOR  D
121200 253              2480         XOR  E
121201 254              2490         XOR  H
121202 255              2500         XOR  L
121203 256              2510         XOR  (HL)
121204 335 256 002      2520         XOR  (IX+2)
121207 375 256 003      2530         XOR  (IY+THREE)
121212 356 003          2540         XOR  THREE
121214 267              2550         OR   A
121215 260              2560         OR   B
121216 261              2570         OR   C
121217 262              2580         OR   D
121220 263              2590         OR   E
121221 264              2600         OR   H
121222 265              2610         OR   L
121223 266              2620         OR   (HL)
121224 335 266 002      2630         OR   (IX+2)
121227 375 266 003      2640         OR   (IY+THREE)
121232 366 003          2650         OR   THREE
121234 277              2660         CP   A
121235 270              2670         CP   B
121236 271              2680         CP   C
121237 272              2690         CP   D
121240 273              2700         CP   E
121241 274              2710         CP   H
121242 275              2720         CP   L
```

U-U

305
11000101
C 5

```
121243 276              2730          CP   (HL)
121244 335 276 002      2740          CP   (IX+2)
121247 375 276 003      2750          CP   (IY+THREE)
121252 376 003          2760          CP   THREE
121254 074              2770          INC  A
121255 004              2780          INC  B
121256 014              2790          INC  C
121257 024              2800          INC  D
121260 034              2810          INC  E
121261 044              2820          INC  H
121262 054              2830          INC  L
121263 064              2840          INC  (HL)
121264 335 064 002      2850          INC  (IX+2)
121267 375 064 003      2860          INC  (IY+THREE)
121272 075              2870          DEC  A
121273 005              2880          DEC  B
121274 015              2890          DEC  C
121275 025              2900          DEC  D
121276 035              2910          DEC  E
121277 045              2920          DEC  H
121300 055              2930          DEC  L
121301 065              2940          DEC  (HL)
121302 335 065 002      2950          DEC  (IX+2)
121305 375 065 003      2960          DEC  (IY+THREE)
121310                  2970 * :
121310                  2980 * 16 BIT ARITHMETIC
121310                  2990 * :
121310 011              3000          ADD  HL,BC
121311 031              3010          ADD  HL,DE
121312 051              3020          ADD  HL,HL
121313 071              3030          ADD  HL,SP
121314 335 011          3040          ADD  IX,BC
121316 335 031          3050          ADD  IX,DE
121320 335 071          3060          ADD  IX,SP
121322 335 051          3070          ADD  IX,IX
121324 375 011          3080          ADD  IY,BC
121326 375 031          3090          ADD  IY,DE
121330 375 071          3100          ADD  IY,SP
121332 375 051          3110          ADD  IY,IY
121334 355 112          3120          ADC  HL,BC
121336 355 132          3130          ADC  HL,DE
121340 355 152          3140          ADC  HL,HL
121342 355 172          3150          ADC  HL,SP
121344 355 102          3160          SBC  HL,BC
121346 355 122          3170          SBC  HL,DE
121350 355 142          3180          SBC  HL,HL
121352 355 162          3190          SBC  HL,SP
121354 003              3200          INC  BC
121355 023              3210          INC  DE
121356 043              3220          INC  HL
121357 063              3230          INC  SP
121360 335 043          3240          INC  IX
121362 375 043          3250          INC  IY
121364 013              3260          DEC  BC
121365 033              3270          DEC  DE
121366 053              3280          DEC  HL
121367 073              3290          DEC  SP
121370 335 053          3300          DEC  IX
121372 375 053          3310          DEC  IY
121374                  3320 *
121374                  3330 * GENERAL PURPOSE OPS
121374                  3340 * :
121374 047              3350          DAA
121375 057              3360          CPL
121376 355 104          3370          NEG
122000 077              3380          CCF
122001 067              3390          SCF
```

```
122002                  3400 *
122002                  3410 * JUMP, CALL, & RETURN :
122002                  3420 * :
122002 303 064 022      3430          JP   TEST
122005 332 064 022      3440          JP   C,TEST
122010 322 064 022      3450          JP   NC,TEST
122013 312 064 022      3460    4-6   JP   Z,TEST
122016 302 064 022      3470          JP   NZ,TEST
122021 352 064 022      3480          JP   PE,TEST
122024 342 064 022      3490          JP   PO,TEST
122027 372 064 022      3500          JP   M,TEST
122032 362 064 022      3510          JP   P,TEST
122035 030 376          3520 HERE     JR   HERE
122037 070 374          3530          JR   C,HERE
122041 060 372          3540          JR   NC,HERE
122043 050 370          3550          JR   Z,HERE
122045 040 366          3560          JR   NZ,HERE
122047 351              3570          JP   (HL)
122050 335 351          3580          JP   (IX)
122052 375 351          3590          JP   (IY)
122054 315 064 022      3600          CALL TEST
122057 334 064 022      3610          CALL C,TEST
122062 324 064 022      3620          CALL NC,TEST
122065 314 064 022      3630          CALL Z,TEST
122070 304 064 022      3640          CALL NZ,TEST
122073 344 064 022      3650          CALL PO,TEST
122076 354 064 022      3660          CALL PE,TEST
122101 374 064 022      3670          CALL M,TEST
122104 364 064 022      3680          CALL P,TEST
122107 020 324          3690          DJNZ HERE
122111 311              3700          RET
122112 330              3710          RET  C
122113 320              3720          RET  NC
122114 310              3730          RET  Z
122115 300              3740          RET  NZ
122116 340              3750          RET  PO
122117 350              3760          RET  PE
122120 370              3770          RET  M
122121 360              3780          RET  P
122122 355 115          3790          RETI
122124 355 105          3800          RETN
122126                  3810 *
122126                  3820 * RESTART
122126                  3830 *
122126 307              3840          RST  0
122127 317              3850          RST  8D
122130 327              3860          RST  16D
122131 337              3870          RST  24D
122132 347              3880          RST  32D
122133 357              3890          RST  40D
122134 367              3900          RST  48D
122135 377              3910          RST  56D
122136                  3920 *
122136                  3930 * INPUT
122136                  3940 * :
122136 333 003          3950          IN   A,THREE
122140 355 170          3960          IN   A,(C)
122142 355 100          3970          IN   B,(C)
122144 355 110          3980          IN   C,(C)
122146 355 120          3990          IN   D,(C)
122150 355 130          4000          IN   E,(C)
122152 355 140          4010          IN   H,(C)
122154 355 150          4020          IN   L,(C)
122156 355 160          4030          IN   F,(C)
122160 355 242          4040          INI
122162 355 262          4050          INIR
122164 355 252          4060          IND
```

```
122166 355 272           4070         INDR
122170                   4080 *
122170                   4090 * OUTPUT
122170                   4100 *
122170 323 003           4110         OUT  THREE
122172 355 171           4120         OUT  (C),A
122174 355 101           4130         OUT  (C),B
122176 355 111           4140         OUT  (C),C
122200 355 121           4150         OUT  (C),D
122202 355 131           4160         OUT  (C),E
122204 355 141           4170         OUT  (C),H
122206 355 151           4180         OUT  (C),L
122210 355 243           4190         OUTI
122212 355 263           4200         OTIR
122214 355 253           4210         OUTD
122216 355 273           4220         OTDR
122220                   4230 *
122220                   4240 * CPU CONTROL
122220                   4250 *
122220 000               4260         NOP
122221 166               4270         HALT
122222 363               4280         DI
122223 373               4290         EI
122224 355 106           4300         IM   0
122226 355 126           4310         IM   1
122230 355 136           4320         IM   2
122232                   4330 *
122232                   4340 * ROTATES & SHIFTS
122232                   4350 *
122232 313 007           4360         RLC  A
122234 313 000           4370         RLC  B
122236 313 001           4380         RLC  C
122240 313 002           4390         RLC  D
122242 313 003           4400         RLC  E
122244 313 004           4410         RLC  H
122246 313 005           4420         RLC  L
122250 313 006           4430         RLC  (HL)
122252 335 313 002 006   4440         RLC  (IX+2)
122256 375 313 003 006   4450         RLC  (IY+THREE)
122262 313 017           4460         RRC  A
122264 313 010           4470         RRC  B
122266 313 011           4480         RRC  C
122270 313 012           4490         RRC  D
122272 313 013           4500         RRC  E
122274 313 014           4510         RRC  H
122276 313 015           4520         RRC  L
122300 313 016           4530         RRC  (HL)
122302 335 313 002 016   4540         RRC  (IX+2)
122306 375 313 003 016   4550         RRC  (IY+THREE)
122312 313 027           4560         RL   A
122314 313 020           4570         RL   B
122316 313 021           4580         RL   C
122320 313 022           4590         RL   D
122322 313 023           4600         RL   E
122324 313 024           4610         RL   H
122326 313 025           4620         RL   L
122330 313 026           4630         RL   (HL)
122332 335 313 002 026   4640         RL   (IX+2)
122336 375 313 003 026   4650         RL   (IY+THREE)
122342 313 037           4660         RR   A
122344 313 030           4670         RR   B
122346 313 031           4680         RR   C
122350 313 032           4690         RR   D
122352 313 033           4700         RR   E
122354 313 034           4710         RR   H
122356 313 035           4720         RR   L
122360 313 036           4730         RR   (HL)
122362 335 313 002 036   4740         RR   (IX+2)
```

```
122366 375 313 003 036   4750         RR   (IY+THREE)
122372 313 047           4760         SLA  A
122374 313 040           4770         SLA  B
122376 313 041           4780         SLA  C
123000 313 042           4790         SLA  D
123002 313 043           4800         SLA  E
123004 313 044           4810         SLA  H
123006 313 045           4820         SLA  L
123010 313 046           4830         SLA  (HL)
123012 335 313 002 046   4840         SLA  (IX+2)
123016 375 313 003 046   4850         SLA  (IY+THREE)
123022 313 057           4860         SRA  A
123024 313 050           4870         SRA  B
123026 313 051           4880         SRA  C
123030 313 052           4890         SRA  D
123032 313 053           4900         SRA  E
123034 313 054           4910         SRA  H
123036 313 055           4920         SRA  L
123040 313 056           4930         SRA  (HL)
123042 335 313 002 056   4940         SRA  (IX+2)
123046 375 313 003 056   4950         SRA  (IY+THREE)
123052 313 077           4960         SRL  A
123054 313 070           4970         SRL  B
123056 313 071           4980         SRL  C
123060 313 072           4990         SRL  D
123062 313 073           5000         SRL  E
123064 313 074           5010         SRL  H
123066 313 075           5020         SRL  L
123070 313 076           5030         SRL  (HL)
123072 335 313 002 076   5040         SRL  (IX+2)
123076 375 313 003 076   5050         SRL  (IY+THREE)
123102 355 157           5060         RLD
123104 355 147           5070         RRD
123106 007               5080         RLCA
123107 017               5090         RRCA
123110 027               5100         RLA
123111 037               5110         RRA
123112                   5120 *
123112                   5130 * BIT MANIPULATION
123112                   5140 *
123112 313 107           5150         BIT  0,A
123114 313 100           5160         BIT  0,B
123116 313 101           5170         BIT  0,C
123120 313 102           5180         BIT  0,D
123122 313 103           5190         BIT  0,E
123124 313 104           5200         BIT  0,H
123126 313 105           5210         BIT  0,L
123130 313 106           5220         BIT  0,(HL)
123132 335 313 002 106   5230         BIT  0,(IX+2)
123136 375 313 003 106   5240         BIT  0,(IY+THREE)
123142 313 117           5250         BIT  1,A
123144 313 120           5260         BIT  2,B
123146 313 131           5270         BIT  3,C
123150 313 142           5280         BIT  4,D
123152 313 153           5290         BIT  5,E
123154 313 164           5300         BIT  6,H
123156 313 175           5310         BIT  7,L
123160 313 176           5320         BIT  7,(HL)
123162 335 313 002 176   5330         BIT  7,(IX+2)
123166 375 313 003 176   5340         BIT  7,(IY+THREE)
123172 313 207           5350         RES  0,A
123174 313 200           5360         RES  0,B
123176 313 201           5370         RES  0,C
123200 313 202           5380         RES  0,D
123202 313 203           5390         RES  0,E
123204 313 204           5400         RES  0,H
123206 313 205           5410         RES  0,L
123210 313 206           5420         RES  0,(HL)
```

```
123212 335 313 002 206  5430         RES  0,(IX+2)
123216 375 313 003 206  5440         RES  0,(IY+THREE)
123220 313 217          5450         RES  1,A
123224 313 220          5460         RES  2,B
123226 313 231          5470         RES  3,C
123230 313 242          5480         RES  4,D
123232 313 253          5490         RES  5,E
123234 313 264          5500         RES  6,H
123236 313 275          5510         RES  7,L
123240 313 276          5520         RES  7,(HL)
123242 335 313 002 276  5530         RES  7,(IX+2)
123246 375 313 003 276  5540         RES  7,(IY+THREE)
123252 313 307          5550         SET  0,A
123254 313 300          5560         SET  0,B
123256 313 301          5570         SET  0,C
123260 313 302          5580         SET  0,D
123262 313 303          5590         SET  0,E
123264 313 304          5600         SET  0,H
123266 313 305          5610         SET  0,L
123270 313 306          5620         SET  0,(HL)
123272 335 313 002 306  5630         SET  0,(IX+2)
123276 375 313 003 306  5640         SET  0,(IY+THREE)
123302 313 317          5650         SET  1,A
123304 313 320          5660         SET  2,B
123306 313 331          5670         SET  3,C
123310 313 342          5680         SET  4,D
123312 313 353          5690         SET  5,E
123314 313 364          5700         SET  6,H
123316 313 375          5710         SET  7,L
123320 313 376          5720         SET  7,(HL)
123322 335 313 002 376  5730         SET  7,(IX+2)
123326 375 313 003 376  5740         SET  7,(IY+THREE)
123332                  5750 * END :

NO ERRORS FOUND :

FILE  060000  117360
READY
LTABL
HERE   122035   TEST   022064   THREE  000003
FILE  060000  117360
READY
```

APPENDIX A

New DG OP SYS Format

A new format of TV Storage Dump and keyboard program is included on the front end of this software system. You will notice the new wording of options 3 and 4, and that 5 and 6 are missing.

Pressing option 3 (octal propram) will initially result in the familiar register display. However, subsequent operations are somewhat different.

Press the Space Bar. You will notice the page of octal bytes is one line shorter. The major difference is an arrow at the top left pointing to byte 000000 presently. This pointer indicates the byte where programming might take place if desired (since 000000 is in read only memory, no change is possible.). This pointer may be present by entering the page (H) and byte (L) similar the H&L presetting operation of the older DG OP System's keyboard programming system. Try entering H070 and then L123. Since this is RAM area in a 16K or greater system, the observed byte may be changed by entering the desired data. e.g. 321 could be entered from the keyboard. Notice the bottom line "scratchpad effect". The actual data is not entered at the indicated address until after the final entry. Emergency abort may be done by pressing the "reset key" on the system prior to the final entry, with no effect on memory.

The Digital Group keyboard with cursor control keys allows the user to move the pointer in the direction indicated by the cursor keys. Keyboards different from this one can move the pointer about if a control H, contol J, contol K, or control L is entered.

The system will return to the OP SYS by pressing an R or r on the keyboard. Option 4 (Hex Program) is similar to Option 3 except that the display is in Hex.

Command Summary

Space - New memory display page
H 000 (HH) - Preset page (octal or hex)
L 000 (HH) - Preset byte (octal of hex)
R - Return to OP SYS
H CTRL - Move pointer backward
J CTRL - Move pointer down
K CTRL - Move pointer up
L CTRL - Move pointer forward
000 (HH) - Insert (octal or hex) code at indicated byte

```
FILE  060000  074376
READY
ASSM

003346                  0001         ST   003346
003346                  0100 **************************
003346                  0110 *     POINTER   OCTAL/HEX
003346                  0120 *      DUMP AND PROGRAM
003346                  0130 **************************
003346                  0140 *
003346                  0150 *   REPLACES BYTES:
003346                  0160 *   003346-004377
003346                  0170 * 001233-001245
003346                  0180 *     MOVE BYTES FROM
003346                  0190 *      005225... TO
003346                  0200 *      005124...
003346                  0210 *
003346                  0220 ***************************
003346 061 000 002      0230 BEGIN  LD   SP,002000
003351 041 000 000      0240        LD   HL,000000
003354 345              0250        PUSH HL
003355 315 250 001      0260 KEY    CALL 001250
003360 346 337          0270        AND  337
003362 127              0280        LD   D,A
003363 376 200          0290 PTEST  CP   200   *SPACE FOR NEW PAGE
003365 040 003          0300        JR   NZ,RTEST
003367 321              0310        POP  DE   *GET RID OF OLD HL
003370 030 114          0320        JR   DCONV
003372 341              0330 RTEST  POP  HL
003373 376 322          0340        CP   322   *R RETURN TO OP SYS
003375 312 000 005      0350        JP   Z,005000
004000 376 310          0360 HTEST  CP   310   *H
004002 040 006          0370        JR   NZ,LTEST
004004 315 233 001      0380        CALL HLOUT
004007 147              0390        LD   H,A
004010 030 074          0400        JR   DCONV
004012 376 314          0410 LTEST  CP   314   *L
004014 040 006          0420        JR   NZ,STEST
004016 315 233 001      0430        CALL HLOUT
004021 157              0440        LD   L,A
004022 030 062          0450        JR   DCONV
004024                  0460 *RIGHT ARROW OR CONTROL L FOR SPACE RIGHT
004024 376 214          0470 STEST  CP   214
004026 040 003          0480        JR   NZ,BTEST
004030 043              0490        INC  HL
004031 030 053          0500        JR   DCONV
004033                  0510 *LEFT ARROW OR CONTROL H FOR BACKSPACE
004033 376 210          0520 BTEST  CP   210
004035 040 003          0530        JR   NZ,UTEST
004037 053              0540        DEC  HL
004040 030 044          0550        JR   DCONV
004042 247              0560 UTEST  AND  A   *CLEAR CARRY
004043 021 006 000      0570        LD   DE,000006
004046                  0580 *UP ARROW OR CONTROL K FOR LINE UP
004046 376 213          0590        CP   213
004050 040 004          0600        JR   NZ,DTEST
004052 355 122          0610        SBC  HL,DE
004054 030 030          0620        JR   DCONV
004056                  0630 *DOWN ARROW,LINE FEED,OR CONTROL J FOR LF
004056 376 212          0640 DTEST  CP   212
```

```
004060 040 004          0650        JR   NZ,NTEST
004062 355 132          0660        ADC  HL,DE
004064 030 020          0670        JR   DCONV
004066 366 040          0680 NTEST  OR   040   *RESTORE NUMBER
004070 365              0690        PUSH AF
004071 006 011          0700        LD   B,011
004073 315 370 000      0710 SKIP   CALL 000370
004076 020 373          0720        DJNZ SKIP
004100 361              0730        POP  AF
004101 315 251 004      0740        CALL ASCIIS
004104 167              0750        LD   (HL),A
004105 043              0760        INC  HL
004106 345              0770 DCONV  PUSH HL
004107 315 346 000      0780        CALL 000346  *ERASE TV
004112 321              0790        POP  DE  *GET HL INTO DE
004113 325              0800        PUSH DE   *BACK TO NORMAL
004114 142              0810        LD   H,D  *POINTER ON DISPLAYED PAGE
004115 173              0820        LD   A,E
004116 376 132          0830 PAGE1  CP   132
004120 060 004          0840        JR   NC,PAGE2
004122 056 000          0850        LD   L,000
004124 030 012          0860        JR   PSTART
004126 376 264          0870 PAGE2  CP   264
004130 060 004          0880        JR   NC,PAGE3
004132 056 132          0890        LD   L,132
004134 030 002          0900        JR   PSTART
004136 056 264          0910 PAGE3  LD   L,264
004140 134              0920 PSTART LD   E,H
004141 315 106 002      0930        CALL 002106   *CHARACTER
004144 135              0940        LD   E,L
004145 315 106 002      0950        CALL 002106
004150 315 370 000      0960        CALL 000370   *SPACE
004153 315 370 000      0970        CALL 000370   *SPACE
004156 006 006          0980        LD   B,006
004160 321              0990 BYTE   POP  DE   *PUT STACK HL IN DE
004161 345              1000        PUSH HL
004162 325              1010        PUSH DE
004163 355 122          1020        SBC  HL,DE   *SEE IF POINTER HERE?
004165 030 005          1030        JR   POINTR
004167 315 370 000      1040        CALL 000370
004172 030 005          1050        JR   CONTIN
004174 076 232          1060 POINTR LD   A,232   *ARROW
004176 315 372 000      1070        CALL 000372
004201 321              1080 CONTIN POP  DE
004202 341              1090        POP  HL
004203 325              1100        PUSH DE
004204 136              1110        LD   E,(HL)
004205 315 106 002      1120        CALL 002106   *PRINT BYTE
004210 043              1130        INC  HL
004211 175              1140        LD   A,L
004212 376 132          1150        CP   132
004214 312 355 003      1160        JP   Z,KEY
004217 376 264          1170        CP   264
004221 312 355 003      1180        JP   Z,KEY
004224 376 000          1190        CP   000
004226 040 012          1200        JR   NZ,NBYTE
004230 006 010          1210        LD   B,010
004232 315 370 000      1220 SKIP7  CALL 000370
004235 020 373          1230        DJNZ SKIP7
004237 303 355 003      1240        JP   KEY
004242 020 314          1250 NBYTE  DJNZ BYTE
```

```
004244 030 272           1260        JR   PSTART
004246 315 250 001       1270 ASCII  CALL 001250  *KEYBOARD # ENTRY
004251 107               1280 ASCIIS LD   B,A
004252 072 247 001       1290        LD   A,(001247)
004255 376 310           1300 HEXCK  CP   ´H´
004257 170               1310        LD   A,B
004260 050 044           1320        JR   Z,HEX
004262 315 372 000       1330 OCTAL  CALL 000372
004265 170               1340        LD   A,B
004266 017               1350        RRCA
004267 017               1360        RRCA
004270 346 300           1370        AND  300
004272 117               1380        LD   C,A
004273 315 250 001       1390        CALL 001250
004276 107               1400        LD   B,A
004277 315 372 000       1410        CALL 000372
004302 170               1420        LD   A,B
004303 007               1430        RLCA
004304 007               1440        RLCA
004305 007               1450        RLCA
004306 346 070           1460        AND  070
004310 201               1470        ADD  C
004311 117               1480        LD   C,A
004312 315 250 001       1490        CALL 001250
004315 107               1500        LD   B,A
004316 315 372 000       1510        CALL 000372
004321 170               1520        LD   A,B
004322 346 007           1530        AND  007
004324 201               1540        ADD  C
004325 311               1550        RET
004326 315 370 000       1560 HEX    CALL 000370
004331 170               1570        LD   A,B
004332 315 352 004       1580        CALL HEXERS
004335 007               1590        RLCA
004336 007               1600        RLCA
004337 007               1610        RLCA
004340 007               1620        RLCA
004341 107               1630        LD   B,A
004342 315 347 004       1640        CALL HEXER
004345 200               1650        ADD  B
004346 311               1660        RET
004347 315 250 001       1670 HEXER  CALL 001250
004352 376 340           1680 HEXERS CP   340
004354 070 002           1690        JR   C,UCASE
004356 326 040           1700        SUB  040
004360 365               1710 UCASE  PUSH AF
004361 315 372 000       1720        CALL 000372
004364 361               1730        POP  AF
004365 376 272           1740        CP   272
004367 070 002           1750        JR   C,NUMBER
004371 326 007           1760        SUB  007
004373 326 260           1770 NUMBER SUB  260
004375 311               1780        RET
001233                   1790        ORG  001233
001233 315 346 000       1800 HLOUT  CALL 000346    * ERASE TV
001236 172               1810        LD   A,D
001237 315 372 000       1820        CALL 000372
001242 315 246 004       1830        CALL ASCII   *GET AND PRINT PAGE/BYTE
001245 311               1840        RET

NO ERRORS FOUND
```