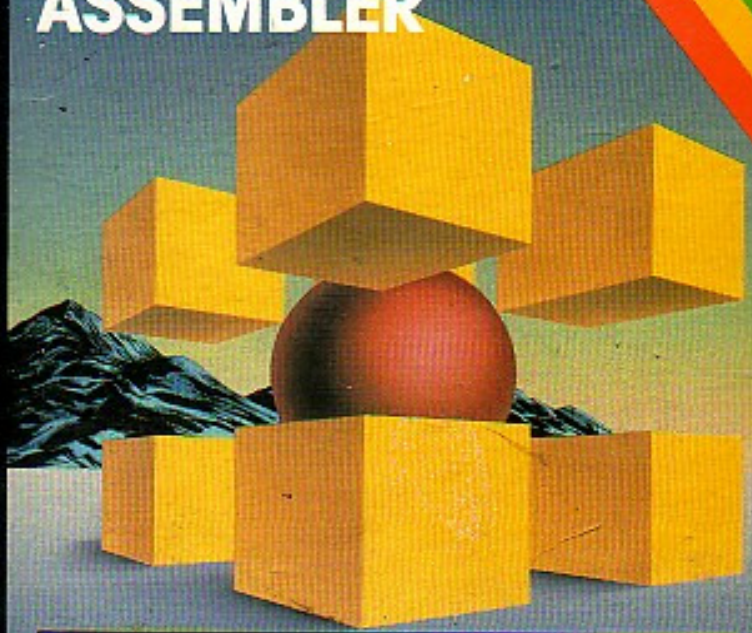
sinclair®

ZX Spectrum®

# ZEUS ASSEMBLER



SOFTWARE BY Crystal Computing

CASSETTE
48K RAM

# USER MANUAL FOR ZEUS ASSEMBLER

First published in 1983 by
Sinclair Research Limited
25 Willis Road  Cambridge  CB1 2AQ  England

ISBN 0 85016 004 X



Printed in the UK

# Contents

# 1 Introduction

Welcome to ZEUS, the assembler-based machine code operating system for the ZX Spectrum.

ZEUS provides the optimum method for writing Z80 machine code on your computer and yet, as you will shortly discover, it is even easier to use than BASIC!

The heart of ZEUS is a high speed two-pass assembler designed to allow the convenient symbolic representation of your machine code program: lines of assembly language mnemonics (machine code instructions) are entered as simply as you would enter a BASIC program. But here the similarity with other assemblers ends, for the true power of ZEUS stems from the unique versatility of its operating system. Just as the Sinclair ROM enables you to write BASIC, so ZEUS provides all the facilities you need to produce machine code programs quickly and efficiently.

# 2 Definitions

The following terms are used throughout this manual; we give their definitions here for those unfamiliar with assembler terminology.

| | |
|---|---|
| SOURCE FILE: | The text, including all assembly language mnemonics, labels and comments. |
| OBJECT CODE: | The machine code produced by ZEUS from the source file. |
| SYMBOL: | A string of alphanumeric characters used to represent a numerical value (either data or an address). |
| LABEL: | A special type of symbol, the value of which corresponds to the address of the instruction which immediately follows it. |
| DIRECTIVE: | An assembler directive is not an actual Z80 instruction but represents an order given by the programmer to the assembler which results in the storing of values either into symbols or into memory. |

# 3 Loading and running Zeus

If you wish to use your ZX Printer with ZEUS, remember to connect it before plugging in your Spectrum.

1. Rewind the cassette to the beginning of side A.

2. Set the volume on your recorder to about ¾ maximum and the tone to full treble.

3. Connect the EAR socket on your recorder to the EAR socket at the rear of the computer.

4. Type LOAD "zeus" or simply LOAD "". Press ENTER and start the tape.

5. Stop the tape when you see the copyright message and flashing cursor.

# 4 Writing machine code with ZEUS

This section is a description of the entry of a short machine code subroutine which, along with the program itself, has been written to illustrate some of the features of ZEUS.

The program loads byte 255 into each address (memory location) on the screen. The screen is 6144 bytes in length and starts at address 16384. The outline of the program along with its BASIC equivalent is as follows:

```
        LD HD,16384         5000 LET HL=16384
        LD DE,16385         5010 LET DE=16385
        LD A,255            5020 LET A=255
        LD (HL),A           5030 POKE HL,A
        LD BC,6143          5040 LET BC=6143
loop    LD A,(HL)           5050 LET A=PEEK HL
        LD (DE),A           5060 POKE DE,A
        INC HL              5070 LET HL=HL+1
        INC DE              5080 LET DE=DE+1
        DEC BC              5090 LET BC=BC–1
```

```
LD A,B                  5100 IF BC<>0 THEN GOTO 5050
OR C                    5110 RETURN
JR NZ,loop
RET
```

First clear the screen by pressing CAPSHIFTED '9' (equivalent to entering 'graphics' mode in Sinclair BASIC).

Entering lines of text is very similar to writing a BASIC program. Each line must be prefixed by a number which may take any value from 0 to 65534 inclusive. It is important to remember that, providing the line the cursor is on begins with a valid line number, pressing ENTER will insert that line at the correct position in the source file so replacing any previously existing line which had the same number.

The command to tell ZEUS to provide line numbers automatically is of the form:

```
                I x y
where           x=line number from which to start
                y=increment between successive line numbers
```

x and y are called 'parameters' because they specify the particular values which the command will use when it executes. For example, in a long BASIC program you may use

```
                LIST 1000
```

Here, the parameter 1000 causes the command to start listing from line 1000.

To illustrate the use of parameters type:

```
                I 100 100
```

and press ENTER several times. To exit from auto-line number mode simply delete the current line number (eg. with CLEARLINE: CAPSHIFTED '1') and press ENTER.

Clear the screen (CAPSHIFTED '9'), type 'I' and press ENTER several times. As you have not specified particular values for the parameters x and y, ZEUS has used the 'default' values which are x = 10 and y= 10. To take an example from BASIC, the default of the LIST command parameter is the first line number.

To enter the first few lines of text:

1. Clear the screen (CAPSHIFTED '9')

2. Type 'I' followed by ENTER (once only)

3. Type in the following lines of text, pressing ENTER after each to insert the line in the source file and to produce the next line number:

```
LD HL,16384
LD DE,16385
LD A,255
LD (HL),A
LD BC,6143
```

   If you enter a character incorrectly use ERASE (CAPSHIFTED '0') and re-type.

4. Exit from auto-line number mode (CAPSHIFTED '1' followed by ENTER).

You may now use the full screen editor which, as the name suggests, allows the text to be edited by altering it as it is displayed on the screen.

First obtain a listing of the source file. Clear the screen and enter 'L' (short for LIST). The LIST command takes up to three parameters of the form:

```
                L x y z
where           x=line number to list from
                y=line number to list up to
                z=number of lines to list before pausing
```

To alter a particular character in the text use the cursor control keys (CAPSHIFTED '5', '6', '7' & '8') to move the cursor to the desired point in the listing and simply type in the replacement character. The cursor will move automatically to the next character position. Remember that the modified line will be inserted into the text on pressing ENTER only when the cursor is sitting on that line.

For example, to change line 30 to read:

```
00030 LD A,254
```

first obtain a listing of the text and using the cursor control keys place the cursor in the same character square as the right hand '5' digit. Type '4' and press ENTER, CLEARSCREEN and LIST. You may now change the number back to 255 either by the same method or by entering

30 LD A,255

Before proceeding, CLEARSCREEN and LIST.

The next part of the program involves a loop. It is often desirable to indent loops in order to make a program more readable. To speed the process of entering sections of indented text, ZEUS provides a 'user definable' TAB facility. The TAB stops may be set up from within the Monitor. Type 'M' and press ENTER. The word 'Monitor' appears to remind you of the mode you are in. Next type 'S' and move the cursor along until it is lined up with the column of Ls in the listing at the top of the screen. Now type a non-space character (eg. 1) followed by, say, 5 spaces, a further 1 and ENTER. CAPSHIFTED '2' will now move the cursor to the stops you have just defined.

Return to ZEUS (enter "Z"). Now you are ready to add the loop. First enter "I 60" to invoke auto-line numbering from line 60. Type in the word 'loop' (which you may wish to put in lower case characters by using the CAPSHIFT key). Use CAPSHIFTED '2' to put the cursor at the next TAB stop. Now type:

LD A,(HL)

and press ENTER. Enter the following lines remembering to indent by using CAPSHIFTED '2':

```
LD (DE),A
INC HL
INC DE
DEC BC
LD A,B
OR C
JR NZ,loop
```

Note that each reference to a particular label must be identical, character for character, to that label. Hence if you have used lower case in line 60 then you must do the same in line 130.

Enter the last line without the indent, ie:

00140 RET

Exit from auto-line number mode (CAPSHIFTED '1' followed by ENTER).

Two additional lines are necessary before you can assemble and test the routine. First, you must tell ZEUS where to place the assembled code by using the directive ORG.

Enter 5 ORG 30000

to assemble the machine code from memory location 30000.

Second, you must set an entry point, ie. the point from which the execute command ('X') will run the code. Enter:

```
6 ENT
```

followed by CLEARSCREEN and LIST.

You will notice that it is necessary to press ENTER twice to list the complete source file. This is because the default value for the z parameter (number of lines to list before pausing) is 14 and the source listing is now 16 lines long.

ZEUS also features a renumber command of the form:

```
R x y z
```

See ZEUS Assembler commands (Section 6) for definition of the parameters.

To renumber the source file, enter 'R'. Now list the first 5 lines by typing:

```
L,,5
```

and pressing ENTER once only. Note that the use of a comma to replace a parameter causes the command to use the default value of that parameter. You can see that the source has been renumbered in accordance with the default values of the 'R' command.

Check carefully that the source file is identical to the following:

```
00010 ORG 30000
00020 ENT
00030 LD HL,16384
00040 LD DE,16385
00050 LD A,255
00060 LD (HL),A
00070 LD BC,6143
00080 loop LD A, (HL)
00090      LD (DE) ,A
00100      INC HL
00110      INC DE
00120      DEC BC
00130      LD A,B
00140      OR C
00150      JR NZ,loop
00160 RET
```

Now you are ready to assemble the source file to produce the object code, commonly called the machine code. Type 'A' (short for 'assemble') and press ENTER.

If all is well the cursor will simply move below the 'A'. If you have made a mistake, an error report will be issued followed by the offending line. If the error is not apparent, consult the appendix on error report codes.

At this stage the easiest mistake to make is to omit a space from within an instruction. This will produce error 0 (incomplete statement). You may find it worthwhile to purposely introduce an error by, for example, removing the space in line 30. (CLEARSCREEN, LIST, place the cursor in the space between 'D' and 'H', DELETE (CAPSHIFTED '3'), ENTER, CLEARSCREEN, ASSEMBLE).

You should see:

```
Error 0
0030 LDHL,16384
```

To correct the error, move the cursor to the 'H' and use INSERT (CAPSHIFTED '4') to add the space, ASSEMBLE, CLEARSCREEN and LIST.

It is advisable to save the source file on tape prior to execution. Loading and saving is accomplished by using the BASIC commands and treating the source file as a block of data bytes.

The BASIC command to save a block of bytes is of the form:

```
SAVE "filename" CODE x,y
```

where x=address of the first data byte
y=length of data

The values of the parameters x and y are given by the 'T' command. Enter 'T and you should see

```
start of source =32768
length          =00144
```

Return to BASIC by entering 'Q' (short for QUIT) and use

```
SAVE "source 1" CODE 32768,144
```

to save the source to tape. Check your recording with

```
VERIFY "" CODE
```

To return to ZUES enter

```
PRINT USR 57344
```

Before you can list the source you must enter 'O' to tell ZEUS to retrieve the old source file. The 'O' command can also take a

parameter, that being the start address of the file it is to retrieve.

You may create a source file at a different address by either:

— Using the 'N' (short for NEW) command before you enter the program. The parameter specifies the address from which the source will be built, or

— Reloading an old source file to a different location from that from which it was saved (see Section 6, 'T' Command).

When you began to enter lines of text, ZEUS automatically started the source from the default of the 'N' command, ie. address 32768 (or 8000 hex.)

If you have a ZX Printer attached then you may make a 'hard copy' of the source file. To turn the printer 'on', enter 'P1'. Now enter

```
L,,16
```

When the printer is on, all output sent to the VDU is also sent to the printer. To turn the printer off enter 'P0'. You may interrupt the printer by using the BASIC BREAK key. Return to ZEUS in the normal way.

You are now ready to test the routine. Enter 'X' (short for EXECUTE). If all is well, the screen (apart from the border) will fill with the current ink colour (ie. white). If this does not happen, list the source and check it carefully against the listing above. If the computer crashes, simply reload ZEUS, QUIT, and load the source file using LOAD "" CODE. Remember to use the 'O' command to retrieve the old source file.

In this section you will have gained an appreciation of the way ZEUS can be used to produce a simple subroutine. There are many additional features at your disposal to aid the programming of more complex routines. The next section provides a comprehensive description of these and also consolidates the features already mentioned.

# 5 Entering and editing text

## 5.1 General format

ZEUS uses the ASCII character set. To enter assembly language instructions you must first type in a line number and follow it with the required line of text. The line number must lie in the range 0–65534 inclusive.

The text must consist of one or more statements separated by colons. Each statement comprises:

1. An optional label,

2. An instruction,

3. An optional comment.

### 5.1. 1 Optional label

The following rules govern the use of labels:

— A label may contain upper and lower case letters and digits.
— A label must start with a letter.
— A label can be up to 14 characters in length.
— A label must not be identical to a reserved word (eg. using 'LD' as a label is not permitted). However, a label may contain reserved words (eq, 'HELD' would be valid). For a list of reserved words see Appendix three.
— A label must be separated from an instruction by a space. Note that every reference to a particular label must be identical, character for character, to that label.

### 5.1. 2 Instruction

An instruction may be any of the standard Zilog Z80 instructions or it may be an assembler directive (see below). An extra facility is provided by ZEUS for referring to the parity/overflow flag. When using ZEUS you may refer to 'jump overflow' instead of 'jump parity even'. Thus:

JP PE,... may be written JP V,....
and JP PO,... may be written JP NV,....
The two versions are completely interchangeable but their use can help to make a routine more readily comprehensible.

### 5.1.3 Optional comment

A comment may be appended to the end of any instruction. It must be separated from the instruction by a semicolon. Lines containing comments only are also allowed.

## 5.2 Constants

Constants may be expressed either in decimal or hexadecimal form as follows:

| | | | | |
|---|---|---|---|---|
| Decimal: | 1 | 99 | 234 | 4096 |
| Hexadecimal: | #A | #FE | #6843 | #5C00 |

Literals can be expressed as follows:

| | | | | |
|---|---|---|---|---|
| "A | "7 | "? | "£ | "= |

For example, to load the literal 'C' into the accumulator, use:

```
LD A,"C
```

Note that in Z80 convention, parentheses denote an address eg. LD A,(200) will load the accumulator with the contents of memory location 200 (decimal).

Finally, there is also a system constant '$' which is set to the current assembly address, eg:

DJNZ $ is equivalent to loop DJNZ loop

## 5.3 Operators

In order to further facilitate the writing of symbolic programs, ZEUS allows the use of logical operators:

| | |
|---|---|
| + | addition |
| – | subtraction |
| & | logical AND |
| ! | logical OR |

No operator priority is observed: expressions are evaluated strictly from left to right.

## 5.4 Expressions

Wherever a constant is required in an instruction, an expression may be used in its place. Expressions are built from labels and/or constants separated by operators, eg:

```
LD A,(ADDRESS+offset)
LD HL,START–3
IN A,(PORT & #FF)
LABEL !#FF00
```

Note that expressions will be computed by the assembler to determine the actual memory location which will be inserted into the object code. Expressions are computed at assembly time, not at

program execution time.

## 5.5 Assembler directives

The following 'pseudo-operations' either provide parameters for the operation of ZEUS or instruct the assembler to store values either as symbols or directly into memory:

ORG nnnn — Short for ORIGIN. This directive instructs ZEUS to assemble the block of machine code (as translated from the source listing after the ORG statement) from address nnnn (provided that the current DISP value is zero–see below). Multiple ORGs within the same source file are allowed. Each ORG statement will redirect the address from which subsequent code is assembled.

DISP nnn — Short for DISPLACEMENT. A DISP instruction alters the place from which subsequent code is generated even though the code so produced is assembled to run at the address specified by the current ORG parameter. It is sometimes convenient to be able to generate code at a different location from the one at which it is ultimately intended to run (for example if the latter is occupied at present). The monitor may be used to relocate the code to the ORG address.

For example, given the following two statements at the start of a source file:

```
ORG 30000
DISP 10000
```

the source following would be assembled from 40000 but would only normally run at 30000.

ENT — Set an entry point. The 'X' command executes the assembled code from the last ENT directive in the source file.

EQU — Short for EQUATE or EQUALS. A label (symbol) may have a value assigned to it using a statement of the form:

```
label EQU value
```

DEFB nn,nn,.. — Inserts bytes nn at the current assembly address.

DEFW nnnn,nnnn, — Inserts words (addresses) nnnn at the current assembly address.

DEFM/string/ — The text enclosed in the '/' delimiters will be inserted at the current assembly address.

As with all assembly language instructions, directives may be prefixed by a label, eg:

```
DATA1 DEFW 40000,3456,6789
OFFSETS DEFB 3,2,14,20,9,2
USRADDR ENT
```

## 5.6 Screen editor

ZEUS provides a full screen editor to facilitate modification of the source file.

First list the text by using the 'L' command. The cursor may now be moved to any point in the listing by using the cursor control keys (CAPSHIFTED '5', '6', '7', '8').

To replace the character at the current cursor position simply enter the new character. The cursor will move automatically to the next character position.

Space may be inserted into a line at the current cursor position by using the inverse video key (CAPSHIFTED '4'). The character at the edge of the screen is lost.

The character at the cursor position may be deleted by using the true video key (CAPSHIFTED '3').

Having altered the line, pressing ENTER will insert the new line in the correct place in the text.

Other commands are:

CLEARSCREEN: The entire screen may be cleared and the cursor

placed at the top left hand corner by using the 'graphics' key (CAPSHIFTED '9')

CLEARLINE: The current line may be cleared and the cursor placed at the start of the line using the 'edit' key (CAPSHIFTED '1')
N.B. This command does not remove the current line from the text but simply from the screen.

TAB: The 'Caps lock' key (CAPSHIFTED '2') moves the cursor to the next tab stop on the current line. The tab stops may be set up from within the Monitor (See Section 7).

# 6 ZEUS Assembler Commands

A command consists of a command letter followed by any number of numerical or string parameters.

Numerical parameters consist of either a decimal or a hexadecimal constant. Entry of numerical parameters overwrites the default parameters for that command.

If it is desired to alter a later default parameter without upsetting earlier ones, this can be achieved by entering a comma for every parameter to be skipped, eg:

| | |
|---|---|
| L | List the text from start to finish |
| L 10 50 | List the text from line 10 to line 50 |
| L , 50 | List the text from the start to line 50 |

String parameters are represented by enclosing the string within delimiters. Delimiters can be any character except ',' or '#' or any digit, eg:

F "LDIR" finds all occurrences of the string LDIR
F /LDIR/ ditto

Commands

A x Assemble the source file and display error messages. After x errors have been encountered and displayed, assembly will stop. To continue assembly, press ENTER.
DEFAULT: x=14

D x y — Delete all lines between line x and line y inclusively. If no parameters are given, no action will be taken.

F "string" x y z — This command searches the source file for all occurrences of the given string. The search commences at line x and finishes at line y. Any lines containing the string are displayed and a pause initiated after z lines have been displayed.
DEFAULT: x=lowest line number
y=highest line number
z=14
"string" = " "

I x y — After entry of this command, ZEUS will automatically generate line numbers, starting with line number x. Any text entered will be inserted in the correct place in the source file and another line number generated. Parameter y is the spacing between successive line numbers. Deleting the line number and pressing ENTER will halt auto line numbering.
DEFAULT: x=10 y=10

L x y z — List the source file from line x to line y. Parameter z gives the number of lines to be listed before pausing.
DEFAULT: x=start y=finish z=14

M — Enter the Monitor (See Section 7).

N x — Create a new, empty source file at address x.
NB: care must be taken when using this command as placing a source file at certain locations may cause a crash.
DEFAULT: x=32768 (#8000)

| | |
|---|---|
| O x | This command causes the 'old' source file at address x to become the current source; it is usually used to retrieve a source file which has been loaded to address x.<br>DEFAULT: x=32768 (#8000) |
| P x | Printer operation. x=1 turns the printer on. All output to VDU is also sent to the ZX Printer. Pressing BREAK during printer operation causes a return to BASIC. Re-enter ZEUS as normal (the printer is automatically switched off on re-entry). x=0 turns the printer off.<br>DEFAULT: x=0 |
| Q | QUIT. Exit to BASIC. To return to ZEUS, enter<br>PRINT USR 57344 |
| R x y z | Renumber the source file. The parameters are as follows:<br>x=first new line number<br>y=spacing between successive new line numbers<br>z=line to start re-numbering from.<br>DEFAULT: x=10 y=10 z=lowest current line number |
| S x | Print the symbol table. x gives the number of symbols/labels and their associated hexadecimal values listed before pausing.<br>DEFAULT: x=15 |
| T | Display the start address and length of the current text file in bytes. The file may be saved by returning to BASIC ('Q') and using:<br>SAVE "filename" CODE start,length<br>To reload an old source file to<br>— the same address as it was saved from, use:<br>LOAD "filename" CODE<br>— a different address, use:<br>LOAD "filename",start address |

Having loaded an old source file, re-enter ZEUS (PRINT USR 57344) and enter the command

O start address

to retrieve the source file at 'start address'

X — Execute the machine code produced by the last assembly. Execution begins at the ENT statement nearest to the end of the source file.
If no ENT statement existed, an error message is issued and no execution occurs.

# 7 The monitor

The co-resident Monitor allows you to directly inspect and manipulate memory and the I/O ports and also set up several options within ZEUS.

MONITOR COMMANDS

A x — This command displays both the decimal and the hexadecimal representation of the value x which may be of either form, eg:

A 59 prints HEX= #3B DECIMAL=59
A #2A prints HEX= #2A DECIMAL=42

C x y z — Copy a block of memory. This command takes three parameters, as follows:

x=start address of block to be copied from
y=start address of block to be copied to
z=the number of bytes to copy

E x — Execute machine code from address x.

I x — Print the value of I/O port x.

K x y — This command allows the border, paper and ink colours to be altered. x is the byte which is used to fill the attributes table: it is made up of four fields:

Bits 0–2 control the ink colour
Bits 3–5 control the paper colour
Bit 6 is set for 'bright'
Bit 7 is set for 'flash'
y is the required border colour

eg: K F 1: white ink on blue background with blue border

K 38 7: black ink on white background with white border.

M x

Enter modify mode starting at address x.
The modify command provides an extremely flexible and hence powerful memory manipulation tool, eg:

M 6000

will print:

6000 XX (note the position of the cursor)

XX is the present contents of address 6000. You may now proceed in four ways:

1. Alter the contents of the address
Type the new value over the old one and press ENTER.
Modify prints the next address, 6001, along with its present contents.

2. Alter the contents of the address and subsequent addresses simultaneously.
Type the new value for the present address followed by the values to be written in the subsequent addresses. On pressing ENTER, modify will print the address and value of the byte after the last one modified.

3. Alter the modification address.
Type '/' followed by the new address. On pressing ENTER, modify will move to the new address.

4. Exit from modify mode.
Type a full stop and press ENTER.

Notes: Characters can be entered by the use of quotes, eg:
after entering M 6000 overwrite the present value with "C. The ASCII value of the character 'C' is now at address 6000.
More complex formats can be entered, eg:
6000 43 41 40 "A "C "D.
enters the six values into 6000 to 6005 and leaves modify mode.

6000 43 41 40 / 6500 83 81 80

enters bytes 43 41 40 into 6000 to 6002. It then moves the modify address to 6500 and enters 83 81 and 80 into addresses 6500 to 6502. Finally it prints 6503 followed by contents of that address.
The above sequences can be combined with the 'enter modify mode' command ('M'), eg:

M 6000 FF 42.

enters modify mode, puts FF into 6000 and 42 into 6001 and leaves modify mode.

O x y — Output the value y to I/O port X

S — Set TAB stops. Any non-space character on the same line as the 'S' will define a TAB stop (provided the line is entered). CAPSHIFTED '2' advances the cursor between the stops, eg:

S 1 1

sets TAB stops at the columns occupied by the '1' character.

T x y — Tabulate memory from address x. The contents of memory are displayed eight bytes at a time in hexadecimal form. Each line is prefixed by the address of the first byte on that line. After y lines have been displayed, tabulation will halt. To continue, press ENTER; any other key will return you to normal (command) mode.

Note that is is possible to modify memory by the following method: tabulate the required area of

memory as described above and after returning to command mode, enter modify mode ('M'). Using the cursor control keys, move the cursor up into the tabulated listing. After altering all the required values on a given line press ENTER to modify the actual memory locations.

Exit from modify mode as before by typing a full stop and pressing ENTER.

Z — Return control to the assembler.

Note that all constants entered in the Monitor must be hexadecimal (with the exception of the 'A' command which, like the assembler, defaults to decimal); prefix hex values with a '#' symbol.

# Appendix one

## Command List

### 1. Assembler

| | |
|---|---|
| A x | ASSEMBLE. Print x errors at a time. |
| D x y | DELETE from line x to line y inclusive. |
| F "string" x y z | FIND 'string' between line x and line y; print z occurrences at a time. |
| I x y | INSERT (auto line number) starting at line x with increments of y. |
| L x y z | LIST from line x to line y in steps of z lines. |
| M | Enter MONITOR. |
| N x | Create a NEW source file at address x. |
| O x | Recover OLD source file at address x. |
| P x | PRINTER control: x=0 for 'off'; x=1 for 'on'. |
| Q | QUIT. Return to BASIC. |
| R x y z | RENUMBER the source file from line z to commence with line x and increments of y. |
| S x | Print the SYMBOL TABLE x lines at a time. |
| T | Display the start address and length of the source file. |
| X | EXECUTE object code at last ENT statement. |

### 2. Monitor

| | |
|---|---|
| A x | Print x in decimal and hex. |
| C x y z | COPY a block memory from address x to address y for a length z bytes. |
| E x | EXECUTE a machine code routine from address x. |
| I x | Print the value of I/O port x. |
| K x y | Set up the paper, ink and border colours. |
| M x | Enter MODIFY mode at address x. |
| O x y | OUTPUT byte y to I/O port x. |
| S | SET up the TAB stops. |
| T x y | TABULATE memory from address x, y lines at a time. |
| Z | Return to the assembler. |

| | |
|---|---|
| CAPSHIFTED '1' ('edit') | Clear the line containing the cursor |
| CAPSHIFTED '2' ('caps lock') | Move the cursor to the next TAB stop |
| CAPSHIFTED '3' ('true video') | Delete the character under the cursor |
| CAPSHIFTED '4' ('inv. Video') | Insert a space character at the cursor position |
| CAPSHIFTED '5', '6', '7' & '8' | Cursor directional control keys |
| CAPSHIFTED '9' ('graphics') | Clear the screen and home the cursor |

# Appendix two

**Error report codes**

| | |
|---|---|
| 0 | Illegal character or incomplete statement |
| 1 | Label too long |
| 2 | ')' symbol expected |
| 3 | Truncation error or jump out of range |
| 4 | ',' expected |
| 5 | Context error |
| 6 | Redefined label |
| 7 | '('symbol expected |
| 8 | Illegal mnemonic |
| 9 | Undefined label |

# Appendix three

### Reserved words

The following is a list of reserved words, ie, those which must not be used as labels, although any label may contain them.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | ADC | ADD | AF' | AF | AND | B | BC | BIT |
| C | CALL | CCF | CP | CPD | CPDR | CPI | CPIR | CPL |
| D | DAA | DE | DEC | DEFB | DEFM | DEFS | DEFW | DI |
| DISP | DJNZ | E | EI | ENT | EQU | EX | EXX | H |
| HALT | HL | I | IM | IN | INC | IND | INDR | INI |
| INIR | IX | IY | JP | JR | L | LD | LDD | LDDR |
| LDI | LDIR | M | NC | NEG | NOP | NV | NZ | OR |
| ORG | OTDR | OTIR | OUT | OUTD | OUTI | P | PE | PO |
| POP | PUSH | RES | RET | RETI | RETN | RL | RLA | RLC |
| RLCA | RLD | RR | RRA | RRC | RRCA | RRD | RST | SBC |
| SCF | SET | SLA | SP | SRA | SRL | SUB | V | XOR |
| Z | | | | | | | | |

# Appendix four

### ZEUS memory map

The following areas of memory are used by ZEUS:

| | |
|---|---|
| A) ZEUS machine code and workspace | 57344–65279 (#E000–#FF00) |
| B) Source file | Initially starts at 32768 (#8000) but may be altered with the 'N' command. |
| C) Assembled code (Object code) | Defined by the user through the directives ORG & DISP |

D) Symbol table This is the table which contains the alphanumeric symbols and labels defined in the source file along with their associated values. It starts at 57343 and increases in size downwards at a rate of sixteen bytes per label used.

Note: if the symbol table (produced on assembly) becomes close enough to the source to be in danger of overwriting it, the message:

'Out of symbol space'

is issued. Similarly if, when entering text, the source file becomes too close to ZEUS the message:

'Out of memory'

is produced.

If either of these occurs, the source file must be moved down in memory. This can be done using the Monitor 'COPY' command. However, great care should be exercised to ensure that, if necessary, RAMTOP is redefined using the BASIC command CLEAR before copying down.

# Appendix five

### Useful Zeus routines

You may wish to call some of the following ZEUS subroutines from your own programs:

| NAME | ADDRESS | FUNCTION |
|---|---|---|
| INPUTCHAR | #F652 | Wait for a key to be pressed; result in A |
| PRINTCHAR | #F503 | Print char in A. May also be a control character (see below). If printer flag is non zero will also output to printer |
| PRINTDECIMAL | #E5A3 | Write HL in decimal followed by a space |
| PRINTHEXBYTE | #F2DF | Write A in hex followed by a space |
| PRINTHEXWORD | #E571 | Write HL in hex followed by a space |
| PRINTSTRING | #E4E3 | Write the string following the call; string must be terminated by byte zero. |
| INPUTLINE | #F6E2 | Enter screen editor, put line which cursor is on in buffer |
| BUFFER | #FE00 | 32 character buffer terminated by a byte zero |
| PRINTERFLAG | #F4CB | Non-zero to enable printer |
| PRINTDECNZ | #E59E | Print decimal number in HL without leading zeros |

Notes: All prints send characters to the printer as well as VDU if PRINTERFLAG is non-zero. PRINTCHAR also allows the use of screen editing commands. For example, printing byte #0F will clear the screen.

The list of editor control characters is as follows:

| | |
|---|---|
| #04 | Delete character under cursor |
| #05 | Insert space at cursor position |
| #06 | Move cursor position to next tab position |
| #07 | Clear line containing cursor |
| #08 | Move cursor left |
| #09 | Move cursor right |
| #0A | Move cursor down |
| #0B | Move cursor up |
| #0C | Move cursor left and blank character under cursor |
| #0D | Move cursor to beginning of next line<br>Scroll if on bottom line |
| #0F | Clear screen and cursor home |

# Appendix six

**Further machine code facilities from Crystal Computing:**

**ZX Spectrum Machine Code Monitor and Disassembler Package**

This program provides additional facilities to aid the inspection, analysis and debugging of your routines; with this and ZEUS co-resident in memory your Spectrum becomes a truly comprehensive Z80 programming unit.

NB. When you intend to use both programs at once, load ZEUS in FIRST, then QUIT and load in the 16K Monitor/Disassembler. You may enter both utilities separately, but it is convenient to enter ZEUS by using the Monitor command:

Goto E000

The use of the ZEUS command QUIT will then return control to the Monitor and Disassembler.