MISOSYS

DISASSEMBLER

Exatron Version I

General:

The MISOSYS Disassembler is a machine language disassembler that functions with the Radio Shack TRS-80 microcomputer to produce an assembler source code from Z-80 machine language resident in memory. This disassembler operates in two passes in order to incorporate symbolic labels in the source output. The symbolic labels are generated for address and 16-bit numeric references within the start-to-end user disassembly request. References preceding the START address are output as equates (EQU) which can be optionally suppressed.

You are assumed to be familiar with Z-80 assembler mnemonics as specified in the RADIO SHACK EDITOR ASSEMBLER USER INSTRUCTION MANUAL #26-2002. Another good reference manual utilizing the standard ZILOG mnemonics is the Z-80 MICROCOMPUTER HANDBOOK by William J. Barden, Jr. (A Howard W. Sams & Co. publication available in most computer stores). Also, be aware of TRS-80 ASSEMBLY LANGUAGE PROGRAMMING also by William Barden, Jr., available at Radio Shack and other computer or program stores. Many other texts can be located which provide various insights into Z-80 assembly language programming. Do not overlook articles on assembler routines appearing in the magazine and journal media.

You should also understand that data elements and ASCII character strings within the range of disassembly will be intepreted as Z-80 machine instructions. These interpretations, in general, may require corrections to disassembled source after loading into the EDITOR ASSEMBLER.

This version also provides an interface to the Exatron Stringy Floppy. By using the "E" output command, the disassembly can be placed on a "stringy wafer" as a file or series of files. The files can then be subsequently loaded into an Editor Assembler modified by either EDTPAT or TAPMOD. EDTPAT is available from ESFOA, the Exatron Stringy Floppy Owners Association. TAPMOD should be available by September 1980 and will provide many enhancements to the Editor Assembler.

Finally, the MISOSYS Disassembler includes two commands to aid you in loading SYSTEM program tapes and locating where in memory the program loads. This provides quite an easy method to determine the START and ENDing locations for the disassembly

Disassembly Address Prompts:

Whenever you exit the Control Function prompts, you will be prompted to enter the storage locations of the program you want to disassemble. The addresses are entered in hexadecimal. Full line input control keys (backspace, line delete, etc.) are supported as in BASIC. In addition, you may enter the value without leading zeroes (0000 as 0, 06CC as 6CC, etc.). These prompts appear as follows:

START ADDRESS - Enter the memory address at which the disassembly should begin. This will be the first memory location that will be disassembled. If the "S" control command was used to load a SYSTEM program, this value would be automatically set to the program's START address,

END ADDRESS - The memory address at which disassembly should cease (Note that disassembly will run from START up to but not including END so END should be one memory position beyond where you want to stop the disassembly). Similar to START, this variable will be set to one greater than the "END" address if the program to disassemble was loaded with the "S" control command.

RELOC ADDRESS - If you had to move the program that you are disassembling (termed the target program) to an address area different from where it originally loaded because it woud have overlayed (loaded into the same region as) the Disassembler, the original START address should be entered here. For example, if the target program originally loaded from 5000H through 5500H and you moved it to load at 7000H to 7500H, then use START=7000, END=7500H, RELOC=5000H. This feature is useful to recover proper address references to code that may have been relocated to a higher or lower address in order to eliminate conflict with the load point of the Disassembler. Three different Disassemblers are provided to further ease conflicts with target programs under disassembly.

Address entries are retained by the program until changed by entering new values. Therefore, subsequent disassemblies using previously entered address information can be performed just by depressing the <ENTER> key. The SYMBOL table is regenerated only when the table is cleared using the control function, "C". Thus, shifting from one disassembly output command to another generates output rapidly. In addition, the Symbol table is not cleared when the Disassembler first loads. This provides a fast "review" capability since the first pass used to generate the symbol table, is bypassed.

Control Function Command Summary:

B - Return control to BASIC at X'06CC'

C - Clear the symbol table buffer

E - Switch the state of the EQUATE flag

S - Load a SYSTEM program & display load points

T - Determine load points of a SYSTEM program

ENTER - Exit from Control Function

Output Command Summary:

S - Output to Screen (Video Display)

T - Output to Tape Cassette

P - Output to Line Printer

E - Output to Exatron Stringy Floppy

R - Output review (continuous scroll until paused)

Special Control Functions:

CLEAR - Logical interrupt for a prompt

BREAK - Interrupt of command request entries

SHIFT @ - Pause during continuous scroll.

Load Instructions:

The MISOSYS Disassembler - Exatron Version I is a machine language program supplied on an Exatron stringy wafer. The program can be loaded using the Exatron "LOAD" command. The wafer contains three (3) versions of the Disassembler configured as follows:

File 1 - Loads from 4400H (17408D) to 5943 59A2H (22947D), with an ENTRY POINT at 4400H. Backup using SAVE parameters, 17408,5540,17408.

File 2 - Loads from 5400H (21504D) to 696C 69A2H (27043D), with an ENTRY POINT at 5400H. Backup using SAVE parameters, 21504,5540,21504.

File 3 - Loads from 6400H (25600D) to 7962 79A2H (31139D), with an ENTRY POINT at 6400H. You can backup this version using SAVE parameters of 25600,5540 5485,25600.

Backup parameters are provided for your own personal use in maintaining safe copies of your MISOSYS Disassembler. You should not provide copies to others.

The memory region immediately preceding the program is used as a stack area. The memory region immediately following the program is used for storage of the program's variables and the symbolic label address buffer. A buffer area for Stringy Floppy use follows the symbol table buffer.

Control Function Details:

B - Command

This command is used to return control to the BASIC interpreter. A jump to address 6CCH is performed. The BASIC "READY" message will be displayed.

C - Command

Since this is a two-pass disassembler, the first pass is builds a table used for the generation of the symbolic labels. The first pass is performed only when the symbol table buffer region is "cleared". By issuing the "C" command the buffer is cleared and the following message is displayed:

Symbol table cleared

∴ Always CLEAR before starting ?
Might get Syntax Error

E - Command

It is common practice to define program constants and address references to other programs at the front end of a source program by means of equate statements (with the assembler pseudo-op, EQU). When the target program contains address references that precede the start-of-disassembly, these references will be output as EQU statements. You may choose to suppress the generation of equates in the disassembler's output by using this command. Equate generation will be either on or off. A flag control is used to indicate the ON or OFF mode. You reverse the flag's status each time you enter the "E".

Every referenced address is labelled "Z address"

S - Command

This command operates similar to BASIC's "SYSTEM" command. It will load a SYSTEM program into memory. However, in contrast to the nonexistant information supplied by the SYSTEM command, the "S" command will identify the program's FILENAME, its STARTing address, its ENDing address, and its TRANSFER address (the location that control will be transfered to after loading a SYSTEM program via the SYSTEM command and issuing the "/" <ENTER>). The program's FILENAME will be displayed as it is read from the tape. The address information will be displayed in the message:

START=xxxx, END=yyyy, TRANSFER=zzzz

where xxxx, yyyy, and zzzz are displayed in hexadecimal. Also, if the program loads without a chacksum error, the START and END variables will be retained for automatic use in the disassembly.

T - Command

The "Test" command operates just like the "S" control command. However, since you may want to discover the address load information without physically loading the program, this command will do just that. The information is identified but the program is not loaded into memory. The START and END variables are updated.

PRESS ENTER

To get out of Control Functions press Enter. Go back to 2nd Page to read what then happens. Then only does the

S Command - Screen (Video Monitor) output:

The screen output is directed to the CRT. Output is scrolled for 16 lines, then paused. The next 16 lines commence scrolling upon depression of any keyboard key except <CLEAR> and <BREAK>.

Depressing <CLEAR> will interrupt output and return you to the prompting message.

The output consists of the following references:

1. Effective memory address of the instruction.

2. Contents of memory starting from the instruction's physical memory location for as many bytes as the instruction's length. Output is in hexadecimal.

3. Sequential line number, in decimal, starting from 00001 and incremented by one (1).

4. A SYMBOLIC LABEL, where referenced as a 16-bit or relative value by the program to be disassembled, consisting of the address referenced preceded by the letter "Z".

5. Disassembled instruction using RADIO SHACK (same as ZILOG) mnemonics. The tab character between the OP code and the OPERAND is expanded for screen display.

6. Character output (in ASCII) of the instruction's hexadecimal values. Bit 7 is stripped from each byte prior to display in order to better identify character strings that utilize bit 7 for "begin-string" or "end-string" detection. Non-printable characters are converted to a period.

T - Tape Output:

This command will create a source cassette tape suitable for loading into the RADIO SHACK Editor Assembler using its' "L" command (using a version that still permits cassette tape input). After entering the Tape command, you will be prompted to prepare the cassette for writing with the message:

Ready cassette

Depression of the <ENTER> key will cause the disassembly to start. The output consists of:

1. The 5-digit ASCII line number,

2. The SYMBOLIC LABEL (or tab if a label is not required),

3. The disassembled instruction. The tab character between the OP code and the OPERAND is not expanded.

The tape is created in blocks consisting of 256 lines of output per block. File names are assigned sequentially. The first is "BLOCKA", the second is "BLOCKB", etc., incrementing the sixth character by one letter for each block. A five (5) second blank segment is written between each block to provide a manual search capability. An asterisk (*) blinks in the upper right hand corner of the screen (3C3FH) for every two lines of output. The starting address of the block will be output to the screen. Depressing the <CLEAR> key will interrupt the tape output only during the period of asterisk blinking.

P - Printer Output:

This command will provide the same output as the "S" command except that the output is directed to the LINE PRINTER. The output is printed 50 lines per page. Each page is numbered sequentially starting from one (1) and incremented by one (1). A heading which labels each column is provided on each page. The MISOSYS Disassembler fully supports the Centronic 779 Line Printer as used by Radio Shack. Any other printer compatible to the Centronic 779, should also function.

RS232 serially driven printers may function only after ensuring that the driver routine maintains the line counter at address 4029H (16425D). The line counter is automatically incremented by the printer driver routine in ROM whenever a carriage return (0DH or 13D) is sent to the printer. Disassemble the ROM region from 58DH through 5D8H to examine the line printer driver routine. You must also patch your Printer driver routine into the standard Printer Data Control Block at 4026H & 4027H.

When the printer command is entered, the program will request you to enter a title and position the printer to receive. The prompt:

"Ready printer and enter title"

will be output. You may enter a title of up to twelve (12) characters which will be placed in the heading on each page of printed output. After depressing <ENTER> following the title, the disassembly will automatically start. By depressing the <CLEAR> key at any time during the printing, the output will be interrupted and you will return to the prompt message.

It has been ascertained that some printers (including the Centronic 779) print 67 lines prior to the form feed when reacting with the BASIC ROM printer driver routine. This is apparently due to some ROMs initializing the number of lines per page (4028H or 16424D) to a value of 43H (67D). If this is the case with your printer, perform the following operation when in BASIC:

```
POKE 16424,66      (4028H,42H)
```

This will condition the driver routine properly for your printer to print 66 lines per page (standard 11 inch pages). The RAM location (printer device control block) contains the number of lines per page which is initialized by BASIC to 67; 67 is the proper number (?) for only some printers.

Exatron Stringy Floppy attached, the error message:

Stringy Floppy error

will be displayed. The stringy floppy output is first written to a buffer region in memory. This region is located between the symbol table buffer and top of memory. The exact memory locations for the buffer are input by you. You also can optionally indicate the drive number that output is to be written to. The default drive is drive 0. The program will request these values with the prompt:

Buffer address? - End = XXXX

Where XXXX will be the top of memory for your system.

The top of memory value is maintained by BASIC in memory locations 40B1H & 40B2H. You cannot enter a value greater than the displayed value. Ideally, a large buffer region is more efficient than a smaller region. However, you may be restricted to a small region so that you do not set up the buffer to overlap the program you are disassembling. During the disassembly process, when the stringy floppy buffer becomes full, the Disassembler will automatically create a file and write the buffer to the file using the next sequential file number. Thus, even with a small buffer region (256 - 500 bytes), you can disassemble a large program in one operation. File 1 is always the first file created so be sure to start with a clean stringy wafer.

Once the buffer values are entered, they are retained by the program. For subsequent ESF requests, the prompt can be responded to by depressing only <ENTER> and the previous buffer values will be reused. The drive will re-default to drive 0.

If a buffer value is outside the allowable range, an error message:

Parm error

will be displayed and the prompt message will be repeated. You may exit the request by entering correct values or abort by depressing the <BREAK> key. Any error involving the stringy floppy (writing error, etc.) will cause the following message to be displayed:

Stringy Floppy error

and the operation will abort.

While the buffer region is loading, an asterisk will blink in the upper right hand corner of the video monitor. This blink occurs during disassembly into each line of source code. When the buffer is written to the stringy floppy, a message is displayed on the video monitor indicating the file number (BLOCKx, where x is 1, 2, 3, etc.) and the address of the last instruction in the buffer. If more than 9 files are generated, the file number will advance to 10, 11, 12, etc. The ESF system permits a maximum file number of 99 (it is not likely that you will reach this value). However, the message displayed while writing file 10 will read, "BLOCK:". For file 11, the character shown would be ";". Use the following table to convert the block letter to file number:

10 - :	11 - ;	12 - <	13 - =	14 - >
15 - ?	16 - @	17 - A	18 - B	19 - C
20 - D	21 - E	22 - F	23 - G	24 - H

Developing a Source tape:

The best way to employ the power of the MISOSYS Disassembler in order to create a "SOURCE" program, the following steps should be performed:

1. Determine the boundaries of the machine language program. This can be accomplished by loading the SYSTEM tape into memory with the "S" control command. Since it is possible that the target program may load into the same region as the Disassembler, you will find that the best procedure is to use the "T" control command first. The START and END values will automatically be initialized to those determined from the program tape itself. If you are disassembling your ROM, consult the memory map in your Level II manual.

2. Disassemble to the screen or printer to detect regions that may have been character strings or data. If you do not have a printer, make note of these regions on scratch paper.

3. Follow up with a "T" command disassembly to generate the SOURCE tape or use the "E" command to write a stringy floppy file.

4. Load the SOURCE tape into the Editor Assembler using its' "L" command.

5. Using the ASCII equivalents from a printed listing, ascertain any character strings and convert them to "DEFM" instuctions. If you do not have a printer, note logical ASCII sections as identified from a Screen output.

6. Make an attempt to scrutinize the listing for code sequences that make little sense. As you become more experienced with Z-80 assembler code, this will become an easier task. Illogical sequences are probably data areas. These "data areas" would best be cleaned up by converting them to "DEFB" or "DEFW" instructions.

Comments concerning this program may be directed to MISOSYS at the following address:

MISOSYS
5904 Edgehill Drive
Alexandria, Va. 22303
703-960-2998