

CROSS-ASSEMBLER FOR INTEL-8080

January 1977  
Alf-Martin Jensen

1

## Introduction

The INTEL-8080 cross-assembler for the NORD-1 and NORD-10 computers is a two pass assembler. It accepts source code in the format described in "8080 Assembly Language Programming Manual" (INTEL Corp. 1974). The object output produced is in standard INTEL hexadecimal format. The symbol table of the assembly may also be output to the object file for later use by a debugging program. The assembly listing consists of a table of contents, the assembled program, the symbol table of the assembly and an alphabetically sorted cross reference table. The user has extensive control over the format of the assembly listing and may also specify what parts of the listing are to be output to the listing device.

The assembler occupies about 8.5 K words of memory plus symbol table, macro storage and stack area. It runs under the operating systems NORD-TSS and SINTRAN-III without modification.

## 2 Assembler Operating Procedure

### 2.1 The Command Processor

To start the assembler from TSS or SINTRAN-III one types the following:

@INTEL-8080-ASSEM cr

INTEL-8080 ASSEMBLER V01.02

£

The command processor is now ready to accept commands. A command consists of a command name followed by zero or more parameters. Several commands, with the required parameters, may be written on a single line. Commands are separated in the same way as parameters. The command name consists of one or more parts separated by hyphens ("-"). Each part of the command name may be abbreviated as long as the command can be distinguished from all other permissible commands. Hyphens preceeding the part of the command name to be written must be included. Consider as an example the commands PRINT-SYMBOL and WRITE-SYMBOL. The first command may be typed as P, PR, PRINT- or P-S or in a number of other ways. However if -SYMBOL or -S is typed the command processor will indicate that the command is ambiguous. While typing commands, the editing characters ctrl-A (backspace one character), ctrl-W (backspace one word) and ctrl-Q (delete whole line) are available. The collection of parameters is done in a standardized way as follows.

Parameters are separated by either a comma or any number of spaces or a combination of comma and spaces.

Parameters may be null in which case a default value is assigned.

When a parameter is missing (as opposed to null) it is asked for and the command processor expects the user to supply the required parameter plus more parameters if he so wishes.

When a parameter syntax error is detected an error message is printed and the parameter is asked for.

Excess parameters are ignored.

If the the operating system used is SINTRAN-III, commands can be given directly to the SINTRAN-III command processor by preceeding them with an @ sign. In this case commands to the local command processor following the SINTRAN-III command are ignored.

## 2.2 Description of the Available Commands

### 2.2.1 HELP command name

Lists available commands on the terminal. If no command name is specified, a list of all the available commands is printed on the terminal. If one of the legal command names is specified, this command, including parameters, is printed. The command name may be abbreviated just as ordinary command names. If the specified command name is "ALL", all commands are printed with parameters.

### 2.2.2 EXIT

Returns control to the monitor.

### 2.2.3 ASSEMBLE source file, list file, object file

Assembles the specified source file with listing on the list file and object output to the object file. If no source file is specified, the files in the file name list specified by the SOURCE-FILES command are used (see below). If no list file is specified, no listing is produced, but error messages are printed on the terminal. If no object file is specified, no object output is produced. The default file type of the object output file is :HEX.

### 2.2.4 SOURCE-FILES file name list

This command is used when a source program consisting of several files is to be assembled into a single object file. Only the last file may contain an END directive. The command takes a list of file names, terminated by an empty file name, as its parameter. The file name list is cleared when assembly terminates.

### 2.2.5 PRINT-SYMBOL-TABLE output file

Prints the symbol table on the specified output file. The default output file is the terminal.

### 2.2.6 PRINT-CROSS-REFERENCE output file

Prints the cross-reference table on the specified output file. This command has no effect if the cross-reference table has not been built, i.e. the command LIST CROSS-REFERENCE-TABLE has not been given. The default output file is the terminal.

### 2.2.7 PRINT-MACROS macro name, output file

Prints the body of the macro specified by macro name on the specified output file. If no macro name is specified, the name and body of all currently defined macros are printed. The parameters are printed as a percent sign (%) followed by a number. The number refers to the position of the parameter in the MACRO statement of the macro definition. The first parameter is numbered 1. The default output file is the terminal.

## 2.2.8 WRITE-MNEMONICS output file

Prints the name and value of all operation codes and directives on the specified output file. The default output file is the terminal.

## 2.2.9 LINES lines/page

This command enables the user to specify the number of lines per page on the assembly listing.

## 2.2.10 SYMBOLS

This command specifies that the symbol table should be output to the object output file at the end of the assembly. The symbols may then later be loaded into the debugger. The assembler is initially in this mode.

## 2.2.11 NO-SYMBOLS

This command specifies that the symbol table should not be output to the object file.

## 2.2.12 LIST list directive

## 2.2.13 NO-LIST list directive

These commands give the user extensive control of the format of the assembly listing. The user may also specify which parts of the assembly listing are to be output to the listing device. List directives work in the same way as commands; they may be abbreviated and can have parameters. If no list directive is specified for the LIST command, the listing mode is reset to the default value. A NO-LIST command with an empty parameter will cause all output, except error messages, to be suppressed. The following are legal list directives.

### HELP command name

Lists available list directives on the terminal. Read the section about the HELP command (2.2.1) for further information.

### SEQUENCE-NUMBERS

Controls the listing of the source line sequence numbers. Default is LIST.

### LOCATION-COUNTER radix

Controls the listing of the assembly location counter field. The assembly location counter will be listed in the radix specified. The default radix is 16. Default is LIST.

#### GENERATED-CODE radix

Controls the listing of the generated binary code. The generated code will be listed in the radix specified. The default radix is 16. Default is LIST.

#### BINARY-EXTENSION

Controls the listing of binary extension; that is, those locations and binary contents beyond the first byte. This is a subset of the GENERATED-CODE directive. Default is LIST.

#### SOURCE-CODE

Controls the listing of the source code. Default is LIST.

#### TABLE-OF-CONTENTS

Controls the listing of the table of contents during pass one of the assembly. The table of contents consists of the page number and the text for every TITLE directive in the program. Default is LIST.

#### SYMBOL-TABLE

Controls the listing of the symbol table for the assembly. Default is LIST.

#### CROSS-REFERENCE-TABLE

Controls the generation of and printing of an alphabetically sorted cross-reference table at the end of the assembly. The cross-reference table consists of all the user defined symbols and for each of them a list of line numbers. The number of a line where the symbol is defined is followed by a "f" sign. Default is NO-LIST.

#### MACRO-CALLS

Controls the listing of macro calls. Default is LIST.

#### MACRO-EXPANSIONS

Controls the listing of macro expansions. Default is NO-LIST.

#### MACRO-DEFINITIONS

Controls the listing of the macro body during macro definition. Default is LIST.

PAGE first page, last page

Enables the user to specify which pages of the assembly listing to output. Several (a maximum of eight in the current version of the assembler) PAGE directives may be specified, resulting in several areas that will be output. PAGE is only legal as a directive to the LIST command.

LOCATION first location, last location

Enables the user to specify which parts of the assembly listing to output addressed by the assembly location counter. The parameters may be expressions containing symbolic references. The expressions are evaluated between the execution of first and second pass. In this way it is possible to use labels or equated symbols. Several (a maximum of eight in the current version of the assembler) LOCATION directives may be specified, resulting in several areas that will be output. LOCATION is only legal as a directive to the LIST command.

3 Object Output Format

The hexadecimal output format consists of records containing the following information: record length, memory address, record type, data and checksum. A frame by frame description follows:

## Frame 0

Record mark. Signals the start of a record. The ASCII character colon (":", 03AH) is used as the record mark.

## Frames 1 to 2

Record length. Two ASCII characters representing a hexadecimal number in the range 0 to OFFH (0 to 255). This is the number of data bytes in the record. A record length of zero indicates end of file.

## Frames 3 to 6

Load address. Four ASCII characters that represent the initial memory location where the data following will be loaded.

## Frames 7 to 8

Record type. Always zero in code records.

Frames 9 to  $9+2*(\text{record length})-1$ 

Data. Each 8 bit byte is represented by two ASCII characters which are the hexadecimal value of the byte.

Frames  $9+2*(\text{record length})$  to  $9+2*(\text{record length})+1$ 

Checksum. The checksum is the negative of the sum of all 8 bit bytes in the record since the record mark evaluated modulo 256.

If the symbol table is output to the object file, each symbol is output as a single record. In such a record, record length is always 5, frames 3 to 6 contain the value of the symbol, and the data is the name of the symbol. When the symbol table is output the format of the entire object file is as follows:

<code records><end of file><symbol records><end of file>

From this one can see that the file may be loaded by any program even if it has no use for the symbols.



#### 4 Assembly Listing Format

The assembly listing consists of four parts: the table of contents, the assembled program, the symbol table of the assembly and an alphabetically sorted cross reference table. Figure 1 shows an example of the listing format. Every page of the listing starts with a page heading. A detailed description of the format follows.

##### 3.1 Page Heading

The first four lines of a page constitute the page heading. Before the heading lines are printed, the listing device is advanced to a new page. If the listing device is the terminal, a blank line is printed instead of advancing it to the next page. The heading consists of the following fields: (refer to fig. 1)

- 1 The name of the file currently being assembled. The name appears exactly as written in the ASSEMBLE or SOURCE-FILES command.
- 2 Current date and time.
- 3 Page number. This field is left blank for the pages that are a part of the table of contents.
- 4 The current title. This field is blank if a TITLE directive has not yet been encountered.
- 5 Two blank lines.

##### 3.2 Table Of Contents

When listing the table of contents, the title is set to "TABLE OF CONTENTS". Each line of the table of contents contains the following information.

- 6 Page number.
- 7 The text of the TITLE directive that generated the entry.

##### 3.3 Program Listing

The program listing consists of four fields on each line as follows:

- 8 The source line number. This field is blank if the line was not read from the source input file.
- 9 The current location counter. This field is blank if the operation does not change the location counter or if the line is a binary extension line, i.e. the location counter is only printed for the first byte of a multibyte instruction or directive.
- 10 The generated code, one byte per line.
- 11 The source code.

When an error is found the error text is printed on the line preceeding the line in which the error was found. The error text is preceeded by three asterisks ("\*\*\*"). At the end of the listing the following two lines are printed:

13 Number of errors detected during the assembly.

14 Time used.

### 3.4 Symbol Table

When listing the symbol table, the title is set to "SYMBOL TABLE". The symbols are listed four to a line and the fields are as follows:

14 Symbol name.

15 Symbol value.

16 Symbol type. The types are: L=label, E=EQU symbol, S=SET symbol, M=macro name. If symbol type is M the value is the address of the macro body in the macro storage area.

### 3.5 Cross Reference Table

When listing the cross reference table the title is set to "CROSS REFERENCE TABLE". The cross reference table is an alphabetically sorted list of all symbols used in the program. Each symbol is followed by a list of line numbers. The line numbers of the lines where the symbol is defined are followed by a "f" sign.

# MULTIPLY TABLE OF CONTENTS

14 JANUARY 1977 1016:03

1 MULTIPLICATION ROUTINE  
MULTIPLY  
MULTIPLICATION ROUTINE  
14 JANUARY 1977 1016:05  
PAGE 1

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19  
; BC := D \* C  
ORG 100H  
MULT: MVI B,0 ; INITIALIZE MOST SIGNIFICANT BYTE  
MVI E,0 ; BIT COUNTER  
MULTO: MOV A,C ; ROTATE LEAST SIGNIFICANT BIT OF  
RAR ; MULTIPLIER AND SHIFT  
MOV C,A ; LOW-ORDER BYTE OF RESULT  
DCR E  
JZ DONE ; EXIT IF COMPLETE  
MOV A,B  
JNC MULT  
ADD C ; ADD IF BIT WAS ONE  
RAR ; CARRY=0 HERE; SHIFT RESULT  
MOV B,A  
JMP MULTO  
DONE:  
END

ERRORS DETECTED: 0  
TIME USED IS 4 SECS

MULTIPLY  
SYMBOL TABLE

14 JANUARY 1977 1016:21

PAGE 2

MULT 100 L MULTO 104 L MULT1 110 L DONE 115 L  
14 15 16

MULTIPLY  
CROSS REFERENCE TABLE

14 JANUARY 1977 1016:23

PAGE 3

A	7	9	12	16
B	5	12	16	
C	7	9	14	
DONE	11	18		
E	6	10		
MULT	5	13		
MULTO	7	17		
MULT1	15			

Figure 1, Example of Assembly Listing.