

Assembler Specifications

Write an assembler for the T.B.Inc. machine. The program should accept T.B. Inc. assembler code and produce an object file that the emulator can process. The assembler should also produce a user listing of both the original assembler program and the machine form of the program.

A. Source code requirements:

- (1) Implement the mnemonics of the T.B.Inc. machine.
- (2) In addition to (1), make it possible to use the following directives:
RESW, WORD, RESB, BYTE, CSECT, EQU, EXTDEF, and EXTREF.
 - (a) RESW: reserve space for one word.
 - (b) WORD: store the given value right justified in a word of memory.
 - (c) RESB: reserve space for the given number of bytes.
 - (d) CSECT: beginning of a new control section.
 - (e) EQU:
 - (f) EXTDEF: external definition
- (3) Programs may include decimal, hexadecimal, and character constant value expressions.
- (4) Assume that each statement must fit on one line (80 characters).

B. User listing:

For each assembler language source statement in a source code file, your user listing should contain, in the following order:

- the line number of the source statement (in decimal).
- the location counter value associated with the object code generated by the source statement (in hex).
- the source statement itself (verbatim).
- the object code generated for that statement.

C. Object Code

The machine code generated by the assembler should have the format required by the emulator.

D. Error Processing

The assembler should detect certain syntax errors/warnings in the source code and print appropriate messages on the line below the error, if possible. Each error message should identify the error/warning and its location.

E. Note:

Later you will write a linker to put together the output file from your assembler.

F. Additional Information about Statements in the Language

Each statement may have:

- optional label field
 - maximum length of a label: 6 characters
 - Labels must start with an alphabetic character and may be followed by up to 5 alphanumeric characters.
- operation field
 - opcode mnemonic or pseudo-op set off by spaces
- operand field
 - delimited by spaces on either side
 - may be any of the following:
 - label
 - constant
 - base ten ex. 27
 - hex ex. X'3C'
 - character ex. C'HELLO'
 - literal ex. =X'2A3B4C'
- optional address mode
 - comma followed by I,X,N, or XN
 - default is blank
- comment field

Pseudo-ops

CSECT indicates starting point and name for a code block
 BYTE generates constants in multiples of bytes
 WORD generates constants with length 1 word
 RESB reserves memory in bytes
 RESW reserves memory in words

A Sample Program

```

PROG      CSECT
          LDCH BCONST,XN      this is a comment
          STCH BVAR
          LDA WCONST
          STA  WVAR
          LDCH C'A',I
          HALT
. This is also a comment
BCONST    BYTE X'3'
WCONST    WORD      11
BVAR      RESB  2
WVAR      RESW  1

```