

## V &T Assembler

*I learned about assembly language and the Z80 using this assembler but the documentation was never more than a few scruffy photocopies. Anybody interested will find the following useful. Most of the numbers are in hexadecimal. I used to use it mostly for controlling hardware which isn't an issue for the emulator but it may still be of interest.*

### Setting up

The assembler contains a 45 byte jump table at address 207E containing jumps to addresses in B-BUG/T-MONITORS and a similar table for NASSYS at 20AE. To run under NASSYS overwrite the first table with the command

```
C 20AE 207E 2D
```

If you want to use your own print routine put in a jump to it at address 20AB by modifying addresses as follows (I don't think emulator users will find this useful).

```
20AB C3
20AC LL
20AD HH
```

where LL and HH are the low and high bytes of the address of your print routine. Otherwise replace 20AB with a C9 RET instruction.

The assembler is supplied located at address 1000. You can make a version that sits anywhere you like (so it need never be in the way). The relocater is at 20DB so to make as assembler at 7000 type

```
E 20DB 7000
```

and the assembler will reply

```
**** Relocated to 7000
```

### Running the assembler

I'll do this with examples.

If the assembler is at 7000 type, for example

```
E 7000 1000 2000
```

the 1000 is the address where you want the source to be and the 2000 is the amount of memory you wish to allocate to source.

The assembler will then display its prompt and show the memory it will be using for source.

You MUST now tell it where you want the object code and where you want it to use for its symbol table as follows :

```
O 3000 1000
```

will select 3000 as the address for the object code of maximum length 1000

```
W 8400 C00
```

will select a work area for the symbol table at 8400 of length C00.

The source editor uses the concept of a current line

The command

```
A 10
```

will cause a prompt for a line of source which will become line 10 and the current line.

The command

```
X
```

will repeatedly prompt for lines which will be appended after the current line. Enter a single full stop to exit this mode.

To change line 120 use

```
R 120
```

To delete line 120 use

```
D 120
```

to delete lines 120 to 200 use

```
D 120 200
```

The L command lists the source and the N command renumbers the lines. There are commands for getting source from tape and saving it to tape but I used to drop out of the assembler and use NASSYS.

There is also a 'find and replace' command with the syntax

```
F LLLL MMMM NNNN DSSSSS... .SSSSD
```

```
or F LLLL MMMM NNNN DSSSSS... .SSSSDTTTTTT.....D
```

LLLL means the line number to start at, MMMM is the line to end at, NNNN is the number of times to find or find and replace, D is and non-searched for character used as a delimiter. SSSS and TTTT are the strings to find and the replacement. All the numbers have sensible defaults.

After you have entered some source the H command will show how much of each area has been used like this

```
1000 10AB 2000 3000 3000 4000 8400 8400 9000
```

This means that source occupies from 1000 to 10AB and that the workspace, where the symbol table will be created is at 8400. The symbol table is currently empty because no source has been assembled. Similarly

the object code area at 3000 is empty. If you assemble your source with the command T1 (which means assemble without printing) then the display will change, typically as follows :

```
1000 10AB 2000 3000 3024 4000 8400 8410 9000
```

showing that you now have 24 bytes of object code at address 3000 and that the symbol table occupies 10 bytes at 8400.

The T1 command assembles to screen, T2 assembles to printer if you have a print routine as mentioned above, and T3 does both.

*Here's an example, with commentary of the assembler being used to put a message on the top line of the screen.*

```
E 7000 1000 1000
>A10
:TOPLN EQU 0BCA
10 TOPLN EQU 0BCA ;address of top line
>X
:;This program puts a message on the top line
: LD HL TOPLN; address of top line in HL
: LD DE MESS ; address of message in DE
:LOOP LD A,(DE);get one character
: OR A ; check if it's the end
: JR Z DONE;if so jump out
: LD (HL),A;otherwise move it to top line
: INC HL;bump pointers
: INC DE
: JR LOOP; jump round for next character
:DONE RST 18;NAS SYS end of program hook
: DEFB 5B
:MESS DEFM 'This is a message'
: DEFB 00 ; flag the end with a null
:.
DEFB 00
>N
>T1
```

Start the assembler with source at 1000

Put in line 10 then use the X command for subsequent lines.

Assembler echos current line

use N command to renumber the lines and T1 to compile

The assembler will produce a listing, with or without errors. If there are no errors you will want to exit with the M command and try your program with E 3000 (if that's where you put it).

**Note that**

- Comments begin with a semicolon
- If the first character of a line isn't a semicolon the assembler assumes that a line begins with a label, if the first character is a space it assumes that the line hasn't got a label
- You can use either a space or comma to separate arguments
- There is an EQU directive to tidy up use of literals
- DEFB means define byte and needs an 8 bit argument
- DEFW means define word and needs a 16 bit argument as in DEFW 000D
- DEFM means define message
- If you restart the assembler with E 7000 and no more arguments it uses the previously defined work areas
- All numbers are hexadecimal and you don't have to say so
- Character and string literals use single quotes
- The assembler will do some arithmetic as in TOP EQU 'Z'+1
- Labels can be up to six characters and are not followed with a colon
- There's no lower case except in comments

If you get this far and want clarification please send me an e-mail at [john@ntparry.freereserve.co.uk](mailto:john@ntparry.freereserve.co.uk). I'll be happy to try to help with either how the assembler works or explaining Z80 code.