

Debug Command Synopsis

This handout is intended as a handy reference guide for performing the more common operations in DEBUG. Not all commands are covered.

Debug understands only hex. Neither decimal or binary numbers are accepted. Consequently, a trailing "H" to denote base 16 is not accepted.

1. Help. Type -? at the - prompt to get a complete list of DEBUG commands and syntax.

2. Specifying addresses. Many debug commands allow an address to be specified. Addresses can be specified as offsets or segment:offset combinations. The segment and offset can be specified with either numeric addresses or segment registers. Examples:

```
U ffff:0000  starts disassembly at address FFFF:0000
U 121       starts disassembly at offset 121 from the CS register.
U DS:3F00   starts disassembly at offset 3F00 from the DS register.
```

3. Specifying ranges. Many debug commands (for example the D command) allow an optional range to be specified. Ranges can be specified as either *startAddr endAddr* or as *StartAddr L Length*. Examples:

```
D 100          displays 128 bytes (the default value) from DS:100 on
D 100 200      displays 257 bytes between DS:100 and DS:200 inclusive
D 100 L 100    displays 256 bytes starting at DS:100
D 503 L 2      displays 2 bytes starting at DS:503
```

4. Flags. T is not displayed.

	O	D	I	S	Z	A	P	C
Set:	OV	UP	EI	NG	ZR	AC	PE	CY
Clear:	NV	DN	DI	PL	NZ	NA	PO	NC

5. Starting DEBUG.

A. To start DEBUG without loading a program into memory, just type DEBUG at the command prompt and press Enter. DEBUG assumes that you want to assemble a .COM program and will start with all segment registers having the same value, IP = 0100, and SP = FFFE.

B. To debug an assembled program type DEBUG path\fn.ext at the command prompt, where "path\fn.ext" represents the path and filename of the file to debug. Examples:

```
debug hw1.com          starts debug and loads specified program
debug D:\230\hw\prog1.exe  starts debug and loads specified program
```

C. To debug an assembled program that takes command-line parameters, simply add the parameters to the debug command line. Examples:

```
debug hw1.com param1 param2  starts debug and loads hw1.com with
                                command line parameters param1 and param2
```

6. Commands. Only the A, D, E, G, P, Q, R, T and U commands are covered in this handout.

See text appendix B for other commands.

Assemble (A [address])

Default segment is CS. If [address] is specified, assembly will start at specified address, otherwise it will start at the current value of IP. Debug will display the address and wait for input. When done with assembly, press Enter and the hyphen prompt will return

Display (or Dump) (D [range])

Default segment is DS. D by itself displays 128 bytes; starting address is determined by a fairly complex set of rules (for .COM files, it is 0100). If only one number is specified for the range (e.g., D 3A0) then 128 bytes are displayed. Successive D commands will display sequentially from the last D command. See Ranges above for examples of DEBUG ranges.

Enter (E address [list]). Address MUST be specified. If [list] is provided, the command will execute and terminate. E 100 12 34 56 78 will enter four bytes into memory starting at 0100 and the hyphen prompt will return. If [list] is omitted, the E command will display the current contents of memory at address followed by a period and will wait for data entry. Pressing Enter will terminate E while entering data and press the spacebar will continue the command for the next byte in memory.

Go (G [=address] [addresses]). Executes current program starting at IP if [=address] is not specified, or at [=address] if specified. [addresses] is a space-delimited list of breakpoints. Examples:

```
g =100 1A3 1D4      starts execution at address 100, will break if
                    addresses 1A3 or 1D4 are encountered during execution
g 23F               starts at current IP; breakpoint at 23F
g                  start at current IP, no breakpoints.
```

If the program terminates and returns to DOS, debug displays the message "Program terminated normally." You should not attempt to restart a terminated program. Instead, Q and reload the program.

Proceed (P [=address] [number]). Most often used to execute CALL, INT, or LOOP or REP x instructions without single stepping. If tracing with T you do NOT want to trace through an INT call especially; so hit P instead of T. [=address] has the same function as the G command. [number] if specified will execute the specified number of instructions; e.g., P 8 will execute the next 8 instructions from the current IP.

Note: if you accidentally T into an INT, your best bet will be to Q and start over. However, if you can figure out the values for CS and IP for instruction following the INT, you can use the G command to set a breakpoint there and continue debugging.

Quit (Q). Terminates debug. Note that there are never any warnings about possibly losing your work.

Register (R [register]). If [register] is not specified, all registers are displayed and you cannot change their contents. If [register] is specified, the specified register will be displayed and debug will display a colon prompt. Enter a new hex value or press enter to leave the specified register unchanged. Note that IP can be specified in the R command (e.g., R IP), but 8 bit registers cannot (e.g., R AH). To change an 8-bit register specify the full 16-bit register (e.g., R AX).

Trace (T [=address] [value]). T by itself executes the instruction at IP, displays all registers and the next instruction. It also displays any memory potentially loaded or stored by the next instruction. [=address] serves the same function as in the G command. [value] serves the same function as [number] in the P command; i.e., execute the next n instructions without stopping (but traces are displayed).

Unassemble (U [range]). If range is not specified, U starts at current IP and normally disassembles about 32 bytes. It is often necessary to observe the target address of a JMP or Jxx instruction and then start disassembly again at that target address. Note that because all versions of DEBUG understand ONLY 8086 instructions, valid instructions for the 186 and later processors (such as PUSH immediate) will be disassembled as DB directives.