

The last line of the routine is a return instruction which it is usual to use at the end of a machine code routine, to return you to Basic.

When you are satisfied that you have entered the above program correctly, type 'X' to restore the prompt and cursor.

B — Breakpoint

This command allows you to temporarily interrupt a machine code program at any point, and return control to the MONITOR, so that you can inspect the values in the CPU registers, and in RAM, and make corrections as necessary.

It takes the form 'B aaaa', where B is the Breakpoint command mode, and aaaa is the address of the instruction that the break will replace. (aaaa must be the address of the first byte of a multi-byte instruction).

The Op. codes in the three addresses aaaa, aaaa+1 and aaaa+2 are automatically saved in data bytes within the MONITOR, and these locations are then loaded with CD OF F7 (for the 48K version) or CD OF 77 (for the 16K version), which constitutes a CALL to the entry point of the MONITOR. It must be a CALL to maintain correct operation of the Stack.

On entering the MONITOR at this address, the values in the CPU registers are stored in data bytes within the MONITOR at the addresses shown in Appendix A; the Stack Pointer is set to the MONITOR Stack; the message "Press Break for Monitor" is displayed on the bottom line of the screen, in addition to the screen display your program has created. The MONITOR will now wait until you press 'Break' when it will display the prompt and cursor on the bottom line of the screen.

You will now be able to use any of the MONITOR commands to check or alter the routine, before returning control to the routine at the point at which the break occurred. As the MONITOR uses its own integral Stack, separate from the Program Stack, there is no danger of over-writing the Program Stack during a Breakpoint.

Before running the example just entered, enter a Breakpoint at address 6009. This will have the effect of stopping the program after the registers BC, DE and HL have been cleared, but before they are incremented.

If the prompt is not visible on the bottom line, type X, otherwise,

Type B the breakpoint command mode
Type 6009 the breakpoint address

There is no need to type ENTER, as the Breakpoint is set after typing the fourth digit. The screen will scroll, and the prompt will appear on the bottom line.

Using the 'M' command, check that 6009 to 600B now contain CD OF F7 (for 48K) or CD OF 77 (for 16K) in place of 03 13 23, the latter having been stored for later replacement.

You are now in a position to run the routine up to the breakpoint.

J — Jump and execute

The Jump command allows you to jump out of the control of the MONITOR to the starting address of any routine that you write, and it takes the form 'J aaaa' where J is the Jump command mode, and aaaa is the start address of your program.

You can run your machine code programs either with the MONITOR 'J' command, or by returning to Basic and using the USR function. Either way, the MONITOR commands are available to you after a Breakpoint.

In this example, we will use the 'J' command.

Type X to restore the prompt and cursor.
Type J to enter the Jump command.
Type 6000 the start address.
Type ENTER

The screen is cleared and the routine will run, and then return to the MONITOR, with the "Press Break for Monitor" message.

The sequence of events on executing the J command is:

- i) the screen is cleared.
- ii) the Stack Pointer is set to the program Stack.
- iii) the start address is put into the Program Counter, and the program is executed.

The MONITOR uses its own integral Stack, which is set on entry to the monitor routine, therefore the Program Stack, which is set by