**M Core Word Set**

|  |  |  |
| --- | --- | --- |
| **Word** | **Level** | **Description** |
| + | 0 | Add B into A |
| .h | 0 | Display A on the screen in hexadecimal, prefixed by a space, at the current cursor position and move the cursor forward (low level debug output function) |
| 0< | 0 | Set A to -1 if A <0, 0 otherwise |
| 0= | 0 | Set A to -1 if A = 0, 0 otherwise |
| 2/ | 0 | Arithmetic shift right. |
| ab>r abc>r  r>abc r>ab  v>r r>v | 0 | Push and restore A, B, C or some combination to or from the return stack. Must unpick opposite – e.g. abc>r followed by r>abc must restore the abc registers. |
| c! ! | 0 | 8 and 16 bit writes of B to address A |
| c@ @ | 0 | 8 and 16 bit reads of address A into A |
| clr.screen | 0 | Clear Screen |
| cursor! | 0 | Set the cursor position to A |
| debug | 0 | Display A B C on the screen |
| halt | 0 | Stop, disable interrupts. |
| input@ | 0 | Basic input device. Reads a word where the lower 5 bits are Fire Up Down Left Right active high (the same as a Kempston Joystick). The upper 11 bits are undefined as more buttons may become available. |
| nand | 0 | Bitwise NAND B into A. |
| port@ port! | 0 | Read and write Z80 ports |
| r1>r2 | 0 | Copy r1 to r2 where rx is a b or c (not required if r1=r2) |
| screen! | 0 | Set the screen byte A to address B (2 + 6 bit) |
| swap | 0 | Exchange A and B |
| sys.info | 0 | Puts the address of the system info area into A *after having copied A to B first.* |
| validate | 0 | Check if A = B. If so display A, if not debug display and halt (this is a validator for checking the words work) |
| wordsize\* | 0 | Convert word count in A to overall byte size in A |
| - | 1 | Subtract A from B, result in A |
| 0- | 1 | Negate A (2’s complement) |
| 1+ 1- 2+ 2- | 1 | Adjust A by 1 or 2 |
| abs | 1 | Set A to abs(A) |
| and or xor | 1 | Binary operator; A:= A <op> B |
| not | 1 | Not A (1’s complement) |
| > = <  >= <> <= | 2 | Compare A against B, set A to -1 if true, 0 otherwise |
| 2\* 4\* 8\*  16\* 256\* | 2 | Shift A left |
| 4/ 16/ | 2 | Divide A by 2 and 4 and 16 (arithmetic not logical), note 222/ is part of the core set |
| max min | 2 | Set A to the larger or smaller of A and B |
| \* | 3 | Multiply A by B, result in A |
| / | 3 | Divide A into B, result in A (unpredictable if A = 0) |
| bswap | 3 | Byte swap A |
| copy | 3 | Copy C bytes from B to A (copes with overlap) |
| fill | 3 | Fill C bytes from B with constant A |
| mod | 3 | Divide A into B, modulus of result in A |