

make simple additions of one new byte requiring two bytes of data (or to delete such an instruction), I have included a routine that allows you to add one byte to this table.

First, using Key D, select a SHOW FROM address 0000. Note that 0000 contains a 91 instruction. Press Key 0. Write the address 0000 then the byte D4 and press Key E. D4 is now at ML 0000/ Press Key A. An exclamation point comes up, the display rolls and something wierd has happened at ML 0000. Before going further, change ML 0000 back to 91 using Key 0.

Now SHOW FROM ML 001F again, and check ML 0025 against the typed listing. The D4 is correctly read as a call to the sub routine at ML 0330 and the 69 has also been correctly disassembled as an input instruction.

I wish that were the end of the problem, but it isn't. In the case where you have a block of pure data, DISASSEMBLER-7 will not be able to tell it from instructions. This is not so bad as long as neither of the last two bytes of data resemble instructions needing two or one byte arguments. If they do, the program may pick up the next bytes in your program and treat them as arguments. Knowing where the blocks of data are supposed to be will greatly aid you in viewing the listing.

With the ability to add your own mnemonics