

By the way, the 8K Basic works best under NAS-SYS, because you can correct lines of program using its editing facilities. Otherwise, you have to re-enter the whole line in error. I have got a version of the 2K Tiny Basic which runs under NAS-SYS, but it's not really worth bothering with - the 8K Basic is so tremendous. I am patiently waiting for ZEAP to be converted to run under NAS-SYS - come on!!

Send in any queries about monitors and what they do or don't have, and I will try to remember the answers and print replies, Good Luck out there!

Richard Beal

Note: Address contributions to "Software Section, INMC"

### Idea for a Program

Suppose you have a program from a magazine which is assembled to run at address 0000-0100, and it uses a data area at F000-F020. You want to run it on a NASCOM, with the program at 0C80-0D80 and the data area at 0E00-0E20. Now you could convert all the addresses, effectively reassembling the program. But suppose you don't have the source listing, and don't even know that it uses a data area. It would still be possible to disassemble and examine the program, but this would take a long time.

What I propose is a program to watch over the program that is being run. The unaltered program would be loaded into the NASCOM at 0C80. The control program would be told:-

All addresses 0000-0100 are to be converted to 0C80-0D80. Any other addresses are to be reported out before being executed.

The control program would then be run. It would look at each instruction in turn and then execute it. Addresses in instructions would be examined, and depending on their value, either left alone, be adjusted according to instructions given to the control program, or else be reported out and the program paused.

This suggested control program would also be excellent for debugging ordinary programs. For example, on an unexpanded NASCOM, it would be told: Run this program. Report any attempt to use an address in the range 1000-FFFF.

The control program could carry out the single step execution of the other program by copying the instruction to another part of RAM and following it with a RET, and then calling it as a subroutine. Or perhaps NMI single step feature could be used.

Please send in any ideas for how to do this, or indeed if it is possible.

Better still, try writing a program to do it!

Richard Beal  
Software Co-ordinator  
INMC Committee