

Other points to remember;

Anything following a semicolon is a comment, and is ignored by the assembler.

The other directive ZEAP accepts is 'ORG', which is the location where ZEAP is to start the assembly. Sometimes there may be several ORG directives where parts of programs are assembled at different locations.

An 'EQU' is used where label or symbol not included in the program has been used. Most commonly these refer to routines in NASBUG/B-BUG which are used as part of the program. As these do not appear in the program, you have to tell ZEAP where they are. Also symbols like CR (for carriage return/new line) may be defined by an EQU as, you never know, someone may come along with a new monitor which redefines the symbols. In this way, ZEAP can be used as a sort of 'cross assembler'.

MEMORY PLAGUE !!

What is it?

How do I identify it?

How do I cure it?

"Memory Plague" is a euphemism for the unexplained failure of NASCOM memory boards. "Memory Plague" seems to affect about 10% of NASCOM issue 1 memory boards, and its causes are, to say the least, obscure. It would seem that noise caused by switching transients from the data latches (81LS97) and the address multiplexers (74LS157) may be breaking onto the data bus via IC2. Poor board layout would seem to be primarily to blame, in conjunction with chips that just happen to be on the low side of average for noise immunity.

Identifying "Memory Plague" is not as easy as it would at first appear because a memory suffering from mild plague will pass both memory test programs in the construction manuals with flying colours and will run Tiny Basic without problem. This is because both memory tests and Tiny Basic are loading operands to memory, and not actually executing M1 (op-code fetch) cycles. M1 cycles are more critical on timing and hence more susceptible to corruption caused by noise. Likewise "Memory modify" and "Copy" commands are unlikely to cause problems except in severe cases. So the only thing likely to reveal "Memory Plague" is a program with lots of M1 cycles and filling a sizeable chunk of memory. ZEAP is ideal for this. If you do not have ZEAP, then write a simple relocatable program that may be copied throughout memory, finishing with the printing of an * then looping back to the start. Leave this running as long as possible (preferably overnight). If the program "crashes", then provided you have eliminated next-doors' fridge (never your own, of course) as the cause, then it's likely that "Plague" has struck.

/