

MAC/65 BUG SHEET #1

This is the first bug sheet for MAC/65 which has been published.

Before we get into the various bugs that have been reported for MAC/65 and the related products, here is some useful information which should enhance your understanding of MAC/65 and make your programming easier.

IMPROVED I/O AND FORWARD FIXES

In the I/O macros (either the file "IOMAC.LIB" for the disk version or the listing in the back of the cartridge version manual), we cautioned you that you must not make forward references to a buffer. For example:

```
PRINT 0,BUFFER
```

```
BUFFER .DS 40
```

will NOT work! Instead, we warn you to place the buffer definition before the usage of the macro, thus:

```
BUFFER .DS 40
```

```
PRINT 0,BUFFER
```

which works just fine, thank you.

The reason for this is that the macros check for a literal string by code such as

```
.IF %1<256 ; if not a buffer, assume string
```

Which works because MAC/65 returns the length of a string as its numeric value. For example, in the line

```
PRINT 0,"HI THERE!"
```

The value of %1 (parameter 1) is zero (as coded), but the value of %2 is nine (9) because the length of **,"HI THERE!"** is 9.

The problem with placing the buffer definition after the macro usage is that MAC/65 gives any undefined label a value of zero (especially for macro evaluation purposes). Thus it sees

```
PRINT 0,BUFFER
```

(where BUFFER is defined after its usage here) as almost the same as

```
PRINT 0,"BUFFER"
```

(believe it or else!) on the first pass of the assembly! On the second pass, though, BUFFER has been defined, so it takes on its proper value and is not treated as a string. Presto! Phase errors!

What can you do about it? The general solution is to always define a label before using it in a macro (a good idea with 6502 code anyway, because of zero-page ambiguities). However, for the special case of the OPEN and PRINT macros as we supplied them to you, you may make a minor modification which will allow the buffer to be defined after the macro usage. To make this fix, we depend on the fact that the value of an undefined label is zero and test for the specific value of zero. Thus, after applying these patches,

```
PRINT 0,BUFFER
```

will work fine, but

```
PRINT 0,BUFFER+2
```

will NOT work, since MAC sees that as 0+2 giving 2 which (once again) looks like the length of a literal string.

Anyway, the changes to be made are shown below. Simply add the underlined portion to each line shown:

For OPEN:

```
1730 .IF %1<256 .AND %1<>0
```

For PRINT:

```
3260 .IF %2<128 .AND %2<>0
```

HIDDEN FEATURES!

It's true. We "found" two hidden features in the cartridge version of MAC/65 that were not well documented, if at all. And they can make editing easier.

FIND with exact match.

When you use the FIND or REP edit commands, the format is given as follows (using REP as the example):

```
REP /<old>/<new>/[!no1[, !no2]]
      [C, A] [C, Q]]
```

and the manual notes that the delimiter (shown as / here) may be any character except a space. (Actually, the manual only states this for FIND, but it's true for REP as well.) All that is true. But...

Because of the way MAC/65 works, any characters you place between the delimiters will be converted to upper case. Searching for comments in lower case letters is thus well nigh impossible. But (and you saw this coming, didn't you) if you use a quote (") as your delimiter, the search (and/or replace) string is not converted at all, and the search is for an exact match.

Enhanced LIST command.

Although not mentioned in the MAC/65 manual, you can give the cartridge a LIST command and end it with a minus sign like this:

```
LIST 1234 -
```

and MAC/65 will list from the first line (1234 in this case) to the end of your program. Thus, you never again need type "LIST 1234,9999".

BUGS IN THE MAC/65 CARTRIDGE

Very few bugs have been reported in the MAC/65 cartridge. A few users have, however, reported some problems which we could not duplicate. If you experience a problem with MAC/65, please try to find a method of reliably demonstrating it so we at O.S.S. can more easily investigate it. As there is to date only one version of the MAC/65 cartridge, all bugs described here apply to your version.

1. DOS 3 PROBLEMS

MAC/65 does not operate properly when assembling a file from disk under Atari DOS 3.

Fix: This problem stems from a major difference in the way DOS 3 performs "note" and „point" operations. The latest version of DOS 3 has corrected this problem, which affects some Atari products as well, so if you just purchased your 1050 drive, you may not experience difficulties. In any case, the problem does not affect .INCLUDE statements, so another solution is to have a main file which INCLUDEs all your other files. Then just LOAD your main file and assemble from memory.

2. JSR, JMP TO ZERO PAGE

The MAC/65 cartridge does not generate code properly on a JSR or a JMP to a zero page location.

Fix: A temporary solution is to use macros in the source code to generate JSRs and JMPs to zero page. For example,

```
1000 .MACRO @JSR
1010 .BYTE $20
1020 .WORD %1
1030 .ENDM
1040 .MACRO @JMP
1050 .BYTE $4C
1060 .WORD %1
1070 .ENDM
```

To invoke the macro, type @JSR (or @JMP) in place of the JSR or JMP to zero page. The proper code will then be generated.

3. DOS XL DO.COM COMMAND

Due to a change in the memory usage of the MAC/65 cartridge version (as opposed to the older disk version), the DOS XL "DO" command does work properly with MAC/65.

Fix: In the file MACDO.M65 on the BBS is the MAC/65 source of a DO command which will work. Just assemble it to a file called DO.COM and use it instead of the file on your DOS XL disk to pass commands to MAC/65.

4. PRINTER TIMEOUT

Some printers, such as the RX-80 in particular, take so long to do a form feed that the computer times out. On a timeout, MAC/65 assumes you ran out of paper or the printer went off-line, etc., and stops assembling to wait for you to correct the problem, causing "lockup".

Fix: Whenever timeout occurs, MAC/65 waits for you to type CONTROL-1, so if your RX-80 locks up, just hit that key and MAC/65 will continue assembling. Also, you can pause a printer listing at any time with CONTROL-1, just as if you were printing to the screen.

BUGS IN THE DISK VERSION OF MAC/65

Surprisingly, only one bug has been reported in the latest version of MAC/65. As of this date, the latest version is 4.2. If you have an earlier version, please call or write for information on updating your disk.

1. DOS 3 PROBLEMS

MAC/65 does not operate properly when assembling a file from disk under Atari DOS 3.

Fix: This problem stems from a major difference in the way DOS 3 performs "note" and "point" operations. Atari is aware of this problem, and it affects some Atari products as well, but we don't know if Atari has or will ever fix it. In any case, the problem does not affect .INCLUDE statements, so another solution is to have a main file which includes all your other files. Then just LOAD your main file and assemble from memory.

2. PRINTER TIMEOUT

Some printers, such as the RX-80 in particular, take so long to do a form feed that the computer times out. On a timeout, MAC/65 assumes you ran out of paper or the printer went off-line, etc., and stops assembling to wait for you to correct the problem, causing "lockup".

Fix: Whenever timeout occurs, MAC/65 waits for you to type CONTROL-1, so if your RX-80 locks up, just hit that key and MAC/65 will continue assembling. Also, you can pause a printer listing at any time with CONTROL-1, just as if you were printing to the screen.

FIXES AND ENHANCEMENTS TO BUG/65

BUG/65 AND BASIC

Among debug programs available for Atari computers, BUG/65 has a few unique features. First and foremost is its intrinsic relocatability: you can place it almost anywhere in memory (or let it place itself at LOMEM automatically). Second but just as significant is the fact that it uses absolutely no zero page memory. The combination of these two features makes it ideal for a variety of uses. We will describe one: writing and debugging BASIC USR() subroutines with BUG/65.

The procedure we are about to describe is not for the faint of heart, those hesitant about experimenting with their machine. One slip could easily wipe out all programs in memory, so be sure and back yourself up carefully and often.

We describe the process step by step:

1. Remove all cartridges and boot your DOS XL disk with BUG/65 on it (hold down the OPTION button if using an 800 XL).

2. From the CP's D1: prompt, type BUG65, thus loading and running BUG/65.
3. Note the contents of LOMEM (location \$2E7); write them down. Use either the BUG/65 W# command or the SAVE command from CP to save a new, non-relocatable version of BUG/65 to disk.
Remember, the start address to save is \$200 higher than what was in LOMEM and the end address is \$2000 higher. (Thus, if \$2E7 contained \$2100, a likely value, you could use the CP command "SAVE BUG2100.COM 2300 4100" to obtain a new, custom version of BUG/65.)
4. Turn off the power and re-boot with BASIC (or BASIC XL) present. From CP, give a command to load and run your modified BUG/65 (e.g., "BUG2100" if you used our example above).
5. Once in BUG/65, change the contents of LOMEM to be at least \$2000 higher than the value you noted earlier. (We recommend changing LOMEM upwards by \$2400 at first. You can refine this value later.) This effectively protects BUG's memory from being used by BASIC.
6. Use BUG to load or mini-assemble your subroutine. Presuming that you wish to single-step, etc., through the routine, you should place a breakpoint at the beginning of the routine and exit to BASIC. If using Atari BASIC, you can do this with a command such as "G A000 @4100" (if your routine was at \$4100).
7. Now, when you call that routine from BASIC (via a USR usage), BUG/65 will get control at the breakpoint location. You will probably have to experiment quite a bit to discover how best to make BUG/65 work for you, but it certainly beats POKEing in routines from BASIC!

A "FIX" FOR BUG/65

If you have any version of BUG/65 other than the latest one (the one now supplied with DOS XL-- it has a manual with a yellow cover), this info is for you:

As you are probably aware, BUG/65 supports only one breakpoint (via the "G" command).

Sometimes, though, you would like to set another breakpoint. Or you might simply like it better if BUG would handle code which jumped to zeroed memory (zeroed memory appears to your 6502 CPU as a bunch of BRK instructions). BUG/65, though, is too smart for its own good.

When BUG gets control as the result of a BRK instruction being executed, it actually checks to see if the BRK instruction is one which it placed (as a result of a "G" command). If not, it assumes that it is a user breakpoint, to be handled by a user routine! Guess what? Most of us don't go around adding user BRK handlers to Atari's OS, so the machine goes off into never-never-land.

Quick fix (again, not for the faint of heart): After loading BUG/65 from CP, perform the following actions:

1. Display the contents of location \$2E7 (LOMEM).
2. Use the "H" command to add \$17EE to those contents. (Example: if \$2E7 contains 2200, use "H 2200 17EE" to find the sum.) Call this value ADDSUM.
3. Display the memory at ADDSUM (via the "D" command. You should see the following bytes (if not, stop now):

```
39EE E9 02 4D ...
```

("39EE" is arbitrary--value of ADDSUM if LOMEM is at \$2200.)

4. Use BUG/65 commands as follows:

```
Z 600
LDA #B0
STA addsum (use actual address instead of name!)
LDA #1E
STA addsum+1 (again, use an actual address)
BRK
```

(hit the BREAK key to exit miniassembler)

```
G 600
```

If you did everything correctly, BUG/65 should allow a "USER RUN" and then print a breakpoint message for the BRK at \$60A. If it didn't work, you may have an already-fixed version of BUG/65, so ignore all this.

PATCH FOR BUG/65 NOT WORKING WITH ATARI XL MACHINES

Again, if your BUG/65 manual has a yellow cover, ignore this section as your copy has already been fixed.

Well, we got bit. After preaching for so many years about „properly“ calling operating system routines, we produced a product which does a no-no.

BUG/65 calls the printer for output by using the Put-A-Character address from the device handler table. This works on 400/800 machines, but the 800 XL supports multiple printers (e.g., P1: and P2:), so it needs an open IOCB to refer to for the device number.

Didn't follow all that? Don't worry about it. The proper fix is to open a channel for the printer and use it for all printer output. But that violates BUG/65's philosophy of not touching IOCB's (thus reserving all of them for your program's use). So we cheat a little. We tell the printer driver that it is open on channel 0 (which is, of course, the screen editor, E:). It works. Don't knock it.

The following BASIC program will patch the BUG/65 disk file so that it will work on XL machines. PLEASE use this ONLY on a copy of your BUG/65 disk!

```

10 XIO 36,#1,0,0,"D:BUG65.COM":REM Unprotect the file
20 OPEN #1,12,0,"D:BUG65.COM"
30 FOR I=1 TO 2430:REM Move to proper position for
  patch
40 GET #1,C
50 NEXT I
60 PUT #1,230:REM Make the first patch
70 FOR I=2431 TO 3051:REM Move to next position
80 GET #1,C
90 NEXT I
100 FOR I=1 TO 18
110 READ C:PUT #1,C
120 NEXT I
130 CLOSE #1
140 END
150 DATA 134,148,132,149
160 DATA 170,160,1,177
170 DATA 148,72,136,177,148
180 DATA 72,138,162,0,96

```

ERRORS IN THE MAC/65 MANUAL

Ignoring typographical errors, there are only two bad errors in the MAC/65 manual. Each of these problems is described below with the corresponding page numbers in both the disk and cartridge version manuals.

.IF EXAMPLE

Cartridge version: page 44
Disk version: page 46

In the example shown, there are two .IF directives but only one .ENDIF. The example will not work properly as shown. To fix it, replace line 17 with:

```
17          .ENDIF
```

MACRO HINTS

Cartridge version: page 62
Disk version: page 65

The two examples shown are not equivalent, as the text professes. The first example is an example of unsafe coding, but it is not very meaningful as is. Perhaps a better example would be the following:

```

. IF %1<256
  . BYTE %1
. ENDF

```

In this case, if the macro were invoked with no parameters, %1 would have the value zero and would also generate an assembly error, and possibly incorrect code would be generated. A better way to code this macro is:

```
. IF %0<>1
```

```

.ERROR "Wrong number of params
.ELSE
.IF %1<256
  .BYTE %1
.ENDIF
.ENDIF

```

BUGS IN THE MAC/65 TOOLKIT

DISK INCOMPATIBILITIES

There exists an incompatibility between the MAC/65 Toolkit disk and some disk versions of MAC/65. If you are using the cartridge, ignore this section. If you are using the disk version, you must change four files on your toolkit diskette (or, if you prefer, just send back your diskette and we will upgrade the diskette). The MAC/65 cartridge version accepts some additional commands not present in all disk versions. In particular, ".ORG" may be used instead of „*=", and ".DS" may be used to replace „*= *+". In order to fix this problem, you must make some changes to four files on your Toolkit disk. For safety, please make a copy of your MAC/65 Toolkit disk before you start and perform the changes ONLY on the copy (that way, if something goes wrong, you can go back to the original and try again).

The following fix will only work with version 4.2 of MAC/65. If you own an earlier version, you must return your Toolkit disk for replacement, or upgrade your MAC/65 disk. If you own version 4.2, you may apply the following temporary patch to your MAC/65 program (do NOT save the patched version to disk):

1. Boot your MAC/65 disk and run the MAC65.COM program.
2. From the MAC/65 "EDIT" prompt, type the following command:
C 8898 < 2E,44,D3,2E,4F,52,C7,20

Now perform the necessary changes to four files on your MAC/65 Toolkit disk. The following files need to be changed:

```

COPY1.DEM
CPARSE.M65
FSCROLL.DEM
SYSEQU.M65

```

For each file, LOAD the file into memory. Then type the following two REPlace commands:

```

REP/.DS/*= *+/,A
REP/.ORG/*=/,A

```

Then, SAVE the new version back to your MAC/65 Toolkit disk. You may then use the Toolkit disk normally.

NAME CHANGES

Two macros in the PMGR.M65 file have names which conflict with 6502 instructions. These two macros, PLPFC and PLPLC, look to MAC/65 too much like the PLP instruction. In order to fix this problem, the names of the two macros must be changed. To perform the changes, LOAD the file PMGR.M65 into MAC/65 from your MAC/65 Toolkit diskette and type the following lines:

```

1141 .MACRO PPFC
1131 .MACRO PPLC

```

Then, SAVE the file back to your disk. From now on, use the name PPFC instead of PLPFC, and PPLC instead of PLPLC. Please change your manual accordingly on pages 22 and 23.