



Memory Expansion Card

#1: Questions and Answers

Revised by: Mike Askins & Matt Deatherage

November 1988

Written by: Cameron Birse

April 1986

This Technical Note documents many of the questions and answers concerning the Memory Expansion Card which are not covered in its manual.

Question: What screen holes does the Memory Expansion Card firmware use?

Answer: The Memory Expansion Card uses the following screen holes:

\$478 + slot	numbanks	number of 64K banks (256K = \$04, 512 = \$08)
\$4F8 + slot	powerup	powerup byte (\$A5)
\$578 + slot	power2	

These screen holes are not cast in concrete and may change with a new revision of the firmware.

Question: Why does RESET turn off the Memory Expansion Card registers until an access to the \$Cn00 space?

Answer: The reason \$Cn00 enables the registers was to optimize speed and the number of pins and logic on the custom gate array. The boot scan hits \$Cn00 anyway and enables the registers.

Question: Will any access (read, write, or status) to the firmware cause the Memory Expansion Card to format itself?

Answer: Yes, any access to the firmware will cause it to format itself to the current operating system (DOS 3.3, Pascal, or ProDOS), assuming it is not already formatted.

Question: Why isn't the Memory Expansion Card marked as a non-interruptible device? What if an interrupt occurred during access to the card and the interrupt handler also accessed the card?

Answer: The Memory Expansion Card is not marked as a non-interruptible device because it would not be fatal to have an interrupt occur during an access to the device. Obviously, the interrupt handler would have to save and restore the registers as well as update the "free block" bitmap, so when the handler returns control the

program does not overwrite the new data. The reason other devices are marked as non-interruptible is due to timing dependent read and write requirements.

Question: Why does the Memory Expansion Card fail to format if the powerup screen hole contains the value \$A0?

Answer: The firmware checks the screen holes for \$A0 values, and if they are all \$A0, it assumes that someone made a mistake and cleared the screen improperly, filling the screen holes with spaces. In this case, the firmware does not want to reformat and lose all the files on the RAM disk.

Question: The code at \$Cn5A has the following sequence, and does not seem to make sense:

```
LDA #$1  
LDY $42  
CMP #4  
BCS Cn8E
```

Shouldn't the `CMP #4` be a `CPY #4`?

Answer: Yes, this is a known bug that will be fixed if the ROMs are ever revised. The bug by itself was not considered significant enough to justify a revision. Note that this is corrected in the Memory Expandable Apple IIc.

Question: If DOS formats the Memory Expansion Card, ProDOS cannot reformat it without a power down or using a ProDOS application which formats disks. In other words, it does not reformat itself when I boot into a new operating system. Isn't that a bit severe?

Answer: This is no different than any other disk device. ProDOS does not have a format command, so you cannot just format from ProDOS without having the formatter installed and some means for calling it. Additionally this was done intentionally so that you could load DOS files into the RAM card and be able to boot ProDOS and use the CONVERT program to convert the DOS files to ProDOS.