Apple IIGS
#21: DMA Compatibility for Expansion RAM

Revised by: Glenn A. Baxter November 1988
Written by: Jim Merritt August 1987

This Technical Note discusses the Apple IIGS Extended Memory Slot specification.

The Apple IIGS Extended Memory Slot specification provides for DMA access to no more than four rows of RAM on a single board through the CROW0 and CROW1 signals. Expansion board designs that involve more than four rows of RAM are not compatible with DMA accesses. Each of the four rows can hold either 256K or 1 MB of data. The designs of the Fast Processor Interface (FPI) and its successor, the CYA, impose this limit. Each row can be organized in any of the following configurations to yield the respective board capacities assuming there are no more than four rows:

Chips	Configuration	Board Capacity
8	256K x 1 DRAM	1 MB
8	1 MB x 1 DRAM	4 MB
2	256K x 4 DRAM	1 MB
2	1 MB x 4 DRAM	4 MB

The CROW0 and CROW1 signals properly decode the row addresses for both normal and DMA cycles. The Extended Memory Slot interface does not support the latching of bank address information off the data bus during a DMA cycle, and a card which attempts to latch the bank address will likely get the **last** CPU cycle's bank address. Getting the last address is not a problem if it accidently happens to be the bank to which you wish to talk, but this is rarely the case. The card gets the last CPU cycle's bank address because DMA essentially shuts off the CPU, so it cannot emit the bank address. The FPI and CYA, which contain the DMA bank address register ($C037), do not emit the DMA bank address either, thus preventing bus contention with the processor as it is being removed from that bus. The DMA bank address register inside the FPI affects the addressing and control information that the Extended Memory Slot sees; it does not affect the data bus. Therefore, during DMA, the bank address time is filled with what is essentially random bank address information. Using this random information could result in damaging the contents of the memory (destroying little things like the operating system).

Suppose a card were designed to latch the bank address directly from the data bus with the rising edge of the PH2 clock signal. It could use the bank address to derive the proper RAM row address and never bother with CROW0 and CROW1 at all. Directly latching the bank address

would permit the card to accommodate any desired RAM arrangement in 64K increments, including an odd number of rows. Although the technique is valid during CPU cycles, it does not work during DMA cycles since the FPI never emits the DMA bank address onto the data bus. During DMA cycles, any card that tries to latch the bank address directly, instead latches the bank address that was put on the data bus during the last CPU cycle, which is probably the wrong value.

Currently, there does not seem to be a solution for the DMA situation. There the possibility of "limited DMA compatibility." An example of a limited-compatibility card would be one with six banks of memory. It's lower four banks are DMA compatible since they use the CROW0 and CROW1 lines, but the upper two banks do not work properly with DMA. This limited approach should be safe, but it is not guaranteed since DMA cards are sometimes aware of the total system memory and may expect, quite reasonably, to have access to **all** of the memory when in fact it does not. There are currently no "memory intelligent" DMA cards, but that could change at any point. The best we can suggest at this time is for hardware developers to build only four-row cards allowing up to 4 MB of memory, which is sufficient for most current applications.

Further Reference

- *Apple IIGS Hardware Reference*