

To: ATA Committee
From: Mehran Ramezani
 Emulex Corporation
 (714)668 5331 Tel
 (714)668 6654 Fax
Date: July 17, 1992
Subject: Proposal to define two new commands (READ DMA Scatter, WRITE DMA Gather)

Emulex Corporation proposes to define, two new commands for Scatter / Gather operations in conjunction with DMA.

Purpose:

To eliminate the possibility of "overrun condition" occurring when the System DMA Controller is programmed for chained mode (Scatter/ Gather mode .)

Problem:

Some Non-DOS Operating Systems do use the Scattered memory buffers which are not a problem for PIO operations or BUS Master adapters. For the Non-Bus Master DMA devices the EISA System DMA Controller provides a "Chained Mode" feature which is useful for transferring data from a peripheral to several different areas of memory within one transfer operation (from the DMA device's viewpoint). This is accomplished by causing the DMA controller to interrupt the CPU for more data transfer information while the previously programmed transfer is still in progress. The DMA controller then loads the new transfer information automatically when the previous transfer completes. This way the entire transfer can complete without interrupting the operation of the DMA device. This mode fails where the DMA transfer process does not allow enough time for the CPU to execute the interrupt routine, due to DMA device taking almost all the EISA Bus bandwidth and higher priority interrupts.

Solution:

One way to guarantee full DMA speed and the synchronized operation of the IDE drive and "Chained Mode" of the DMA Controller is, to make the drive aware of the Scatter/ Gather operation and amount of data transfer at each stage of the Scatter/ Gather list:

- a) Program the DMA Controller for the first two stages of the S/G List (stage #1 and #2).
- b) Issue a READ/ WRITE DMA Scatter/ Gather Command after loading the "Task File" with the starting sector address and full transfer length (traditional information).
- c) Write the length of the sum of stages #1 and #2 into the "Task File" after Status Register has indicated the ability to accept further information.
- d) Drive starts the data transfer via asserting DMARQ.
- e) The DMA Controller reaches the end of the first stage (#1) and interrupts (IRQ13) the CPU. Meanwhile, the DMA transfers are going on for the stage #2.
- f) The CPU programs the DMA controller for the next stage of the S/G list and loads the same information into the drive's "Task File." If this task is done before the drive reaches the terminal count for the previous stage, the DMA process goes on without pause. Otherwise, drive, in order to avoid overrun, knows when to drop the DMARQ and pause.
- g) The process of programming both the DMA Controller and the Task File at each stage of the S/G list continues until the total transfer is completed.