X3T9.2/92-34

Quantum

EXTERNAL MEMO

DATE: February 14, 1992

TO: X3T9.2

FROM: Jim McGrath
408-894-4504
fax 408-894-3208

SUBJ.: DMA Meeting at HP-PCG on 1/24/92

Attendees

Tom Newman	Adaptec
Joe Chen	Cirrus Logic
Chris Runawitz	HP
Larry Lamars	Maxtor
Jim McGrath	Quantum
Clyde Nagakura	Quantum
Gene Milligan	Seagate
Tom Hannan	Western Digital

Background

This meeting was called to discuss Chris Runawitz's proposal (distributed during the January working group) to adopt new timings for ATA DMA. In evaluating various IDE devices implementing DMA, Chris found that no drive complying with the ATA standard can achieve transfer rates greater than approximately 2 MBytes/sec. This is true even when the actual data transfer takes zero time. The bottleneck was in backplane (i.e. system) arbitration time (> 1 us), since ATA requires the drive to arbitrate for the backplane for every word transferred.

The object of this exercise is to devise a way to implement DMA that can yield at least 4 MBytes/sec, and should be expandable up to 8 MBytes/sec. This will be incorporated into the enhanced ATA standard. Since implementing this fast DMA will require chip changes, it is higher desirable that an early consensus be reached. This will allow chips to be turned and devices tested before the standard has to be forwarded to X3T9.

With this in mind, most of the meeting was spent discussing Western Digital's implementation of DMA. While they violate the ATA standard, their approach does appear to yield a truly fast DMA.

Fast DMA

Tom Hannan (WD) asserted that the only thing that has to change in the ATA document to allow for fast DMA is the requirement that the DMA REQ and ACK times go through a complete handshake for every work transferred. Since this approach will avoid the need for a backplane arbitration cycle, it can yield high speed DMA transfers.

While Tom also believes that the standard must be enhanced with more detailed timing and protocol information, the key concept can be agreed to quickly. The drive should continue to assert DMA REQ for every word of a multiple word transfer. Only when the transfer is to be stopped should the signal be deasserted. This allows for long period of data transfer with few arbitration cycles. Both modeling and empirical testing indicate that this technique achieves performance close to a totally dedicated channel.

The question then arises as to when the DMA REQ line should be deasserted? In their first implementation, WD had to stop every 16 bytes due to a bug in a customer's system hardware (backplane DRMA refresh was not being done during drive data transfers). This bug has long since been fixed, and Tom recommends a mandatory halt only every sector. At this point systems can allow other devices to easily gain access to the backplane.

There is an edge timing condition. Deasserting DMA REQ at the drive end may result in the system seeing the signal deassert only after several tens of ns. In this case, the system could already be committed to another DMA cycle. This would result in either a fatal data corruption problem or a system hang.

This problem can be avoided if the drive simply deasserts one transfer time early. If the system still asks for another DMA cycle, then data will be available. After than cycle the deassertion should have long been noted by the system, and DMA properly shut down. If another cycle is not requested, then the drive simply does one more transfer after going through the normal arbitration cycle. Note that this can decrease performance, but the cost is still only two arbitration cycles over possible 512 bytes of data transfer - still an acceptable low overhead.

The advantage of this approach is that is is tolerant of propagation delays. Given that different chip sets will have different delays, that the cables can be 0 to 18 inches long, and that DMA speeds will be increased in the future, this is a desirable (perhaps even necessary) attribute.

There are still some timing issues that require agreement. For instance, how long should the drive wait for this last DMA cycle before assuming that the system has agreed to shut down DMA? Since DMA cycles are normally < 1 us, but arbitration will be > 1 us, a 1 us cutoff seems reasonable. The same technique can be used to allow the system to signal to the drive at any time that the current chain of DMA cycles should be broken (i.e. the system needs to take to another device on the backplane).

There appeared to be general support for this technique, although Joe Chen (Cirrus Logic) had some concerns over the cost of implementation in ASIC logic.

Even Faster DMA

We also discussed Mode 0, 1, and 2 timings. While modes 0 and 1 are straightforward to support, mode 2 required a DMA cycle every 2 (8 MHz) backplane clocks. This might create problems for some chipsets. Moreover, there is no standard way to have the drive and system agree to the Mode to be used. The drive can tell the system what mode is supported via IDENTIFY DRIVE, but there is no standard way for the system to tell the drive which mode it supports. This is a real problem with Mode 2 and any faster modes we may define in the future.

Finally, it is also desirable to respec some of the bus timing to make them more compliant with EISA DMA (this will make system integration easier).

Next Steps

I propose we meet in two weeks to discuss this topic further. Friday February 28 or March 6 would be logical dates. Location would be in the Bay Area (Quantum can host). Note that the SCSI Forum is in Sunnyvale during the first week in March. At that time we should reach agreement on the following:

- o the WD approach should, in principle, be followed
- o assign an editor for the draft document
- o begin documenting timings required to implement the WD approach
- o examine the current mode timing for EISA compliance
- o discuss how different modes should be invoked
- o make provision for backplanes > 8 MHz (local buses)

Please contact me regarding this meeting.