Presentation on Western Digital's Enhanced IDE 95
Key Points

■ Phase in the Changes
  
  Start in 1995
  
  More advanced capabilities in 1996

■ Provide solutions for the Customers
  
  - PCI, DMA (Intel)
  - BIOS (Phoenix)
  - Drivers (Microsoft, IBM)
  - System Vendors

■ Focus on Overlapped operation

■ Allow but de-emphasize more advanced Server and Array capabilities
Assumptions

- Don’t alter the key value that IDE currently has (Cost & Performance)
- Phase in the improvements
- Allow first implementations without any hardware changes
- Always be backward compatible
- ATAPI & IDE Features should be implemented identically where possible
- Improve Performance only where complexity and cost are not increased
- Overlapped operations will be mainstream, Queuing will not
- Reducing number of host interrupts will be beneficial
- System suppliers, Motherboard manufactures, BIOS and OS suppliers will decide what is acceptable and what is not
Phased approach

■ ATAPI Overlapped Commands
  • Allow CD-ROM and Tape Drives to release the Task File Registers and commands to be sent to the faster IDE Disk Drives

■ IDE and ATAPI overlap using PIO and existing DMA
  • Add Overlapped command capability to the IDE Drives

■ New Overlapped PCI DMA Capability
  • Allow each Drive on the IDE cable to use the DMA Controller without any Host (Driver) intervention

■ Command Queuing and Tagging
  • Multiple commands for each Drive on the IDE cable

■ Advanced DMA for Servers and Arrays
  • Allow Tag information to be processed by the DMA Controller
Other Possible Areas for improvement

- One Interrupt for both Primary and Secondary Channels
- Parity on Data Transfers
- Reduced Command Set
- 3.3v Interface
Basic Building Blocks

- Arbitration of the Task File Registers
  - Release of Task File Registers
  - Selection Command (A2h)

- Arbitration of Interrupts
  - Shared Interrupts / Service Status

- Overlap Capable PCI DMA State Machine
  - DMA Ready Status
  - Host Interlocks

- Arbitration of DMA Control Signals
  - Overlap PCI DMA reads & writes Task File Registers directly

- Communication of Tags
  - IDE Feature Register / ATAPI Tag Register

- Advanced PCI DMA
Capability Pyramid

- Tag Capable PCI DMA
- Queuing Capable Driver
- Overlap Capable Driver
- PCI DMA
- Selection Command
- Release
- Overlapped ATA
- Overlapped IDE & ATA
- Overlapped DMA
- Command Queuing
- One interrupt per Command
- Command Tags
- Queuing Firmware
- Overlapped DMA Firmware
- IDE Overlap Command
ATAPI Overlap

- Allows a CD-ROM Drive to be attached to Primary Cable with no Performance Penalty
- Uses existing Host & Drive (ATAPI) Hardware
- ATAPI Drive Releases the Task File Ownership after acceptance of an ATAPI command
  
  Overlap Mode is enabled on each command via ATAPI Features Register
- Overlapped Commands are issued to an IDE (Legacy) Drive while an ATAPI Command is still processing
ATAPI Overlap Interrupts and Status

- Interrupts are not Shared
- Drive uses Interrupt & Service Status to gain Host’s attention
- Service Status set when any service is needed by the Device
- The Interrupt Reason and DRQ value of IO=1, CoD=1 and DRQ=0 (Message In to Host) is used to indicate a Release Interrupt
ATAPI Overlap and DMA

- Use existing PCI DMA

- When DMA used, the Drive Interrupts when DMA is complete, only when there is still more data to be transferred (That command still has more data to transfer)

- When DMA used, Host (Driver) uses Byte Count from Selection Command (A2h) results to program DMA Controller
IDE & ATAPI Overlap

- Two new commands for Read and Write Overlapped
- DMA or PIO specified in the IDE Feature Register
- IDE Drive Releases Task File after acceptance of Overlap command
- Task File arbitration performed by the Host (Driver)
- Shared Interrupts used to signal Service to Host
- nIEN used to prevent interrupts while BSY or DRQ is set
Shared interrupts

- IDE INTRQ signal becomes Open Collector
- Both IDE Drives on the Cable must support the Capability
- Pulsed Interrupt is used for non-PCI systems (Generates a Rising Edge for normal IDE interrupt logic)
- PCI Systems use a Low Level Interrupt signal
- Function is enabled at Post via the SET FEATURES Command
PCI DMA Overlap Capability

- Intercepts the Shared Interrupt from the IDE/ATAPI Device
- New Status bit in the IDE Status Register to indicate DMA ready
- Sequencer selects each Drive, senses DMA Ready & Service status bits
- Arbitrates and Selects a Drive by issuing A2 command
- Uses an Interlock to prevent Host (Driver) and Sequencer collisions
- Blocks Interrupts while processing INTRQ or DMA transfers
- Interrupts Host for unknown interrupt reasons
- Host (Driver) Performs same function on systems that do not have the
  Hardware support for the PCI DMA Overlap
- No change in the existing Drive DMA or DMARQ/DMACK logic

WESTERN DIGITAL
Advanced DMA

- Layered on Overlapped PCI DMA & Overlapped IDE / ATAPI Command capability
- Advanced DMA Controller reads Tag from Task File
Comparisons with Other Proposals

■ Shared Interrupt vs. Drive to Drive Arbitration

- Requires very little drive hardware changes
- Not as complicated as Drive to Drive Arbitration
- Allows PCI DMA controller to trap interrupt and perform arbitration or pass the interrupt through to the Host Driver when a function is not supported in hardware
- Reduces communication from the drive to the DMA controller after arbitration
- Allows some Overlapped commands in a mixed Legacy and Overlap Capable environment
Comparison with Other Proposals (Cont.)

- New Opcodes vs. using all Existing Opcodes in different "Modes"

  Using new Opcodes prevents any older driver from breaking
  Allows some redefinition of the Task File Registers
  Only two commands simplifies the Drive Firmware
  Enables the Function on a command by command basis automatically
  Does not break existing prefetch hardware
  Does not break existing Drive Auto DRQ logic when using Queuing
  Provides simple Drive Hardware Decode and Sequence logic
Comparison with Other Proposals (Cont.)

- PCI DMA hardware Arbitration vs. Drive to Drive Arbitration
  - Only the Host Hardware changes
  - When the Host Hardware does not support the feature, the Host Driver can provide the capability transparently to the IDE Drive
  - Allows for various levels of performance without changing the Drive Hardware or Firmware
  - Advanced PCI DMA can be implemented without any Drive Hardware Changes
  - No potential race condition on the DMARQ line during arbitration
  - Can be used in mixed Legacy and Enhanced environment