Overlap Proposal
Starters

• Phase in the Changes
  • Allow features to be added over time

• Provide solutions for the Customers
  • Chipset providers (e.g., Intel PCI DMA, IBM, CMD, OPTI, Symphony, Adaptec)
  • BIOS (Phoenix)
  • OS & Drivers (Microsoft, IBM)
  • System Vendors

• There is a whole world of peripherals that must be considered
  • Disk, CD-ROM, CD-R/E, Floptical, Tape, Others?
Assumptions

- Always be backward compatible.
- Don’t alter the key value that IDE currently has (Cost & Performance)
- Allow first implementations with no hardware changes to the drives and minimal or no hardware changes to the host.
  - Don’t obsolete existing systems
- Phase in the improvements (Mixed Environments are OK)
- ATAPI & ATA Features should be implemented identically where possible.
- Improve Performance only where complexity and cost are not increased.
Phased Approach

- **Phase 0** DSC Based Overlap
  - Being done today for both tape and CD-ROM

- **Phase 1** ATAPI Overlap
  - Included in 8020 and 157 and is being implemented by CD-ROM manufactures
  - Only one ATAPI Device provides overlap and only using PIO

- **Phase 2** ATA Overlap
  - ATA and ATAPI devices know how to overlap
  - Interrupt must be shared
  - DMA signals must be shared

- **Phase 3** Interleaved DMA
  - PCI DMA Bridge H/W filters interrupts
  - Device releases bus after each DMA xfer

- **Phase 4** Queuing
  - DSC status and Immediate Commands
  - Arbitration of Task File Registers via Service Command, Status and Release
  - Sharing of INTRQ via Proxy
  - Arbitration of DMARQ via INTRQ/DMA Ready and Service before using DMARQ
  - Byte Count for DMA is communicated using Service
  - Host / Driver Interlock
  - Tags

³ 4, 4/18/98, ATA Overlap v4 5/25/95
Why Phased Implementation

- A tangled web is created when Host Hardware, Device Hardware and Drivers must all change to implement new features
  - Devices with new capabilities sold into existing systems can not use any of the these new features
  - Systems with new features will have to wait until new Devices and Driver/OS upgrades become available
  - OS & Driver upgrades will have to wait until new Systems and New Devices are available

- To untangle this web, improvements should be independent
  - Feature improvements in the Devices are accomplished most easily when not impacted by bridge logic, and OS upgrades
Basic Building Blocks

- Arbitration of the ATA Registers
  - Release of Task File Registers and Service Command (A2h)

- Arbitration of Interrupts
  - Proxy Interrupts / Service Status

- Arbitration of DMA Control Signals
  - Interrupt before asserting DMARQ with DMA Ready Status, allowing host to select device via Service Command

- New Overlap Read & Write Commands for ATA

- Interleave Capable PCI DMA State Machine
  - Overlap PCI DMA reads & writes Task File Registers directly
  - Host Interlocks

- Command Queuing (Communication of Tags)
  - IDE Feature Register / ATAPI Tag Register
ATAPI Overlap

Diagram showing a host connected to a disk and a CD.
ATAPI Overlap

- Allows a CD-ROM Drive to be attached to Primary Cable with little Performance Penalty

- Uses existing Host & Drive (ATAPI) Hardware, no changes required

- ATAPI Drive Releases the Task File Ownership after acceptance of an ATAPI command
  - Overlap Mode is enabled on each command via ATAPI Features Register bit

- Commands are issued to an ATA (Legacy) Drive while an ATAPI Command is still processing. These commands proceed to completion before any processing on the Overlapped ATAPI Command is performed

- ATAPI Device reports capability in Identify Drive Data Command
  - ATAPI Device reports nominal time required for release operations
ATAPI Overlap Timing

Overlap Capable Drive

[Diagram showing overlap capable drive]

ATA Drive

[Diagram showing ATA drive]

DRV Bit

INTRQ

BSY

DRQ
ATA Overlap
ATA Overlap

- Drive Releases Task File after acceptance of Overlap enabled command
- Task File arbitration performed by the Host (Driver) via the Service Command
- Sharing of INTRQ
  - Proxy Interrupts used to signal Service to Host
- Sharing of DMARQ
  - DMARQ is not driven until the Device is issued the Service Command
  - When DMA is needed the Device issues a Service + DMA Ready Interrupt
- For non-ATAPI Devices, two new commands for Read and Write Overlap
  - DMA or PIO specified in the IDE Feature Register
- ATAPI Devices use ATAPI Overlap with addition of INTRQ & DMARQ sharing techniques
New Interrupts and Status for Overlap

- Drive uses Interrupt & Service Status to gain Host’s attention
- Service Status set when any service is needed by the Device
- For ATAPI Devices the Interrupt Reason RELEASE bit is used to indicate a Release Interrupt
- DMA Ready is used for Overlapped and Interleaved DMA operations

New Status Register Bits

- D7: BSY
- D6: DRDY
- D5: DMA Ready
- D4: Service
- D3: DRQ
- D2: CORR
- D1: Index
- D0: Error or Check
New Status Combinations

- Host driver interprets three status bits for state information

<table>
<thead>
<tr>
<th>Service</th>
<th>DMA Rdy</th>
<th>DRQ</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Bus is Released</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Non Overlapped PIO Command with data ready</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Transition from Data ready to Idle (Delayed deassertion)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Device has been selected via Service and DMA data ready</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Some Service is needed (Status)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Overlapped Command with PIO data ready</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Illegal</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Overlapped Command with DMA data ready</td>
</tr>
</tbody>
</table>
Overlap Commands for ATA Style Devices

- Two new commands added to provide overlap capability
  - Read overlapped A6h
  - Write overlapped A7h

- Overlapped Commands use new definition for Task File Registers

- All overlapped commands use the Service Status and Service Command

- Tags are communicated back to host via Service Command

- Tag of zero (0) indicates an overlapped type command
New Overlap Capable Commands Vs Existing

- **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, but does allow new accelerations
  - Allows for Tag data for Command Queuing
  - Does not break existing Drive Auto DRQ logic when using Queuing
  - Provides for simple Drive Hardware Decode and Sequence logic
Overlapped Commands

Host

Overlap Driver

Disk

CD
Overlap basics

- **Sharing of Registers**
  - ATA Devices use Overlap Commands and ATAPI Devices use the Overlap Enable bit in the Features Register to enable Overlapped operation
  - Each Device releases the bus after receiving an Overlap enabled command
  - Each device generates an interrupt with Service Status bit set when data or status needs to be sent/received to/from the Host
  - Each device starts PIO or DMA (also sets DMA Ready Status when Interrupting) data transfers after receiving the Service Command

- **Sharing of Interrupt Signal**
  - Proxy Interrupts must be enabled via Set Features before overlapping commands

- **Sharing of DMA Signals**
  - For Overlapped DMA there is an interrupt at the beginning of DMA operation and one at the end for status
  - For Interleaved DMA there is a byte count reported and the Device releases the bus after sending/receiving that amount of data
Overlapped Command Timing

Drive 1

Drive 0

DRV Bit

PDIAG-

INTRQ

BSY

DRQ
Interrupt Sharing

- **Two basic schemes**
  - Add a new signal
    - Creates 4 INTRQ signals for the Bridge to deal with
    - New Status to indicate which is interrupting must be created (Outside the Task File)
  - Share the existing signal
    - New Status needed to indicate an interrupt has been generated (Inside the Task File)

- **Four basic flavors have been explored**
  - New Signal used to allow one interrupt for each device (IORDY)
  - Electrically sharing the existing INTRQ signal (Inverted INTRQ)
    - Changes the Host hardware and causes interrupts at times where they can not be processed
  - Proxy generation of INTRQ using the existing ATA Definition
    - Uses only one signal (PDIAG) and does not affect the Host hardware
    - Interrupts only occur when the driver /bridge logic can process them
  - Arbitration between devices to allow only one device to use the INTRQ signal
    - Complicated and uses at least 2 signals, but does not change the Host hardware
Proxy Interrupts

- **PDIAG-** used to ask selected Drive to generate an Interrupt for the non-selected drive
- Both Drives on the Cable must support the Capability
- Function is enabled via the SET FEATURES Command
Proxy Rules

1. When a selected device is not actively using the Task File e.g. Device not interrupting (SERVICE bit is clear); and BSY is clear; and DRQ is clear; and Proxy Interrupts are enabled using set features command

   Then the selected device shall:

   Assert INTRQ if PDIAG- has transitioned from a logic high (deasserted) to a logic low (Proxy Interrupt is being requested). The sensing of a Proxy Interrupt Request shall be “Edge Sensitive”.

2. When a selected device releases the Task File (BSY & DRQ Cleared) the device shall discontinue requesting a Proxy Interrupt, before BSY & DRQ are cleared, by tristating the device’s PDIAG- driver and pulling PDIAG- high through the existing 10k pull up resistors on each device.

3. When a non-selected device, with overlap enabled, wishes to interrupt the host, the device shall set the SERVICE and optionally the DMA READY status bits, Enable PDIAG- and drive PDIAG- active low.

4. When a selected device receives a Service command, it shall clear any pending Service request and discontinue requesting a Proxy Interrupt (Tristate PDIAG-, allowing it to be pulled up to +5 volts)

5. The ATA EXECUTE DRIVE DIAGNOSTICS command shall cause Proxy Interrupts to be disabled. The host shall re-enable Proxy Interrupts following each diagnostic command using the enable Proxy Interrupts set features command.

6. When a Device that is driving PDIAG- low is selected, the Device shall tristate PDIAG- and Drive INTRQ high.

7. The Host shall disable Proxy Interrupts before any power saving modes are commanded.
Advantages & Tradeoffs of Proxy

**Advantages**

- *Does not change the Host Hardware*
- *Simple to implement in most devices using only firmware*
- Prevents Interrupts when they can not be processed
- Accelerations in the Devices will require very few gates

**Tradeoffs**

- When Interrupt Filtering is implemented, the Bridge Logic must read the status register for two devices to determine which device is interrupting
- Does not help solve the multiple channels sharing the INTRQ signal model, unless the PDIAG- signal becomes the common interrupt for all four devices in the future
Proxy Interrupt Timing
Proxy Timing (Cont.)
Overlapped and Interleaved DMA

Host

DMA Bridge Logic

DMARQ, INTRQ

Disk

CD
Overlap and DMA

- When DMA is used in an overlapped command, the device must release the bus while processing the command.

- When DMA data is to be transferred, a Service Interrupt is first generated.

- When the Service command is processed, the Device asserts DMARQ to begin the DMA operation.

- When all the data has been transferred, a Status Interrupt is generated.

- Once the transfer begins, the device maintains ownership of the Task File Registers until all data has been transferred.
Overlapped DMA

Service Interrupt and Service Command

Service

Command to Device 0
Release

Command to Device 1
Release

DMA Data, Device 0
Status

DMA Data, Device 1
Status

Completion Interrupt
Interleaved DMA

- Uses existing PCI DMA

- DMA in Interleaved Mode is performed very similar to PIO, with individual transfers interrupting the Host, rather than just at the end of the transfer

- When DMA used, Host (Driver) uses Byte Count from Service Command (A2h) results to program DMA Controller

- When the Device transfers the amount of data specified in the Byte Count it releases the bus

- The Device only transfers data, and owns the bus, when there is data to be transferred
Interleaved DMA Timing

Drive 1

Drive 0

DRV Bit
PDIAG-
INTRQ
BSY
DMARQ
Advantages and Tradeoffs of Interleaved DMA

- **Advantages**
  - Interleaved DMA can be performed without any Host hardware changes
  - Long and slow data transfers from devices such as CD-ROM and QIC Tape will not penalize the faster peripherals (i.e. Hard Drives)
  - Provides a simple foundation for Host hardware accelerations

- **Tradeoffs**
  - Without Host accelerations, the Driver must field extra interrupts
PCI DMA Bridge
with Overlap and
Interrupt Filtering
PCI DMA Bridge Logic Overlap Capabilities

- Intercepts the Shared Interrupt from the ATA/ATAPI Device
- Uses DMA Ready, Service and DRQ Status bits to determine when the drive is ready to transfer DMA data
- Sequencer selects each Drive, senses DMA Ready & Service status bits
- Arbitrates and Selects a Drive by issuing Service (A2h) command
- Uses an Interlock to prevent Host (Driver) and Sequencer collisions
- 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
- May not require changes in existing Drive DMA or DMARQ/DMACK logic
Interrupt Filtering Example Timing

Drive 1

Drive 0

- DRV Bit
- Host Int
- PDIAG-
- INTRQ
- BSY
- DMARQ
- Interlock

PCI DMA Logic
Transitions From
Command Queuing

- Queuing just adds Tags to Overlap
- Tag as part of Command
  - ATA Tag is sent to Device in ATA Features Register
  - ATAPI Tag is send in the ATAPI Tag Register
- Tag as a qualifier for data transfer
  - Tag is sent to the host in the ATAPI Tag register after the Service Command is processed
- Only Simple Tags, Device can reorder any command
- Errors cause all queued commands to be removed from the queue
Capability Pyramid

Pyramid of Capabilities

- ATAPI Overlap Capable Driver
- Interleaved DMA Capable Driver
- Command Tags Queuing Firmware
- Queuing Capable Driver
- Interleaved PCI DMA
- PCI DMA Overlap Capable Driver

Future Enhancements
- Tag Capable PCI DMA
- Command Queuing
- Overlapped DMA
- Overlapped ATA & ATAPI
- One interrupt per Command
- Overlapped ATAPI

Service Command Release
- Proxy Interrupt ATA Overlap Commands Overlap Firmware
ATA Technology Philosophy

• **Evolutionary**
  - Compatibility with existing hardware standards (time to market)
  - Small incremental change to existing operating system drivers
  - Allows phased implementation of capabilities into operating system driver
    - Implementation and distribution by third party developers
    - Distribution of third party drivers by OS vendors
    - Inclusion in operating system release by OS vendors

• **Revolutionary**
  - Requires synchronized release of host and peripheral hardware features
  - Increased risk due to increased scope of change
  - Large scale changes may not result in perceivable benefits
Logical ATA encompassing all four devices

- If in the future the ATA Interface will be two cables but with only one actual hardware interface, then it might make sense to try to move to a model that has only one logical cable

- Sharing of the Interrupt should then be used
  - Electrical sharing of INTRQ would be best bet

- Sharing of DMARQ should be directed by the Host hardware

- Overlap should always be used and only one data transfer processed at a time