# USB
# Mass Storage Class Command Queuing
# (MSC-CQ)

Steve McGowan

3/11/08 V3

# Wish List

- **Command Queuing**
  - Multiple outstanding commands
    - Support up to 256
  - Out of order completions
- **First Party DMA**
  - Direct DMA Data to user/process address space
    - Not touched by Driver
- **Command Queue Bypass**
  - High priority commands automatically bypass lower priority
- **Core Targeting**
  - Completions can be targeted at individual cores
- **SAM-4 Compliant**
- **USB2 and USB3 compatible solution**
  - Graceful fallback to legacy Bulk Mode if MSC-CQ not supported by host

# Proposal

- **Command Pipe**
  - 1 to 16 prioritized OUT pipe for sending SCSI CDBs to an MSC-CQ device
  - Correlate to SCSI CDB Priorities
  - Higher priority pipes “Bypass” lower priority pipes
- **Status Pipe**
  - 1 to ? IN Pipes
  - Enables Core Targeting for completions
  - Target Status pipe identified by Command
- **Define pool of IN and OUT pipes**
  - 2 to 478 total, Up to 239 IN and 239 OUT
  - Allocated dynamically by driver
  - Target Data pipe identified by Command

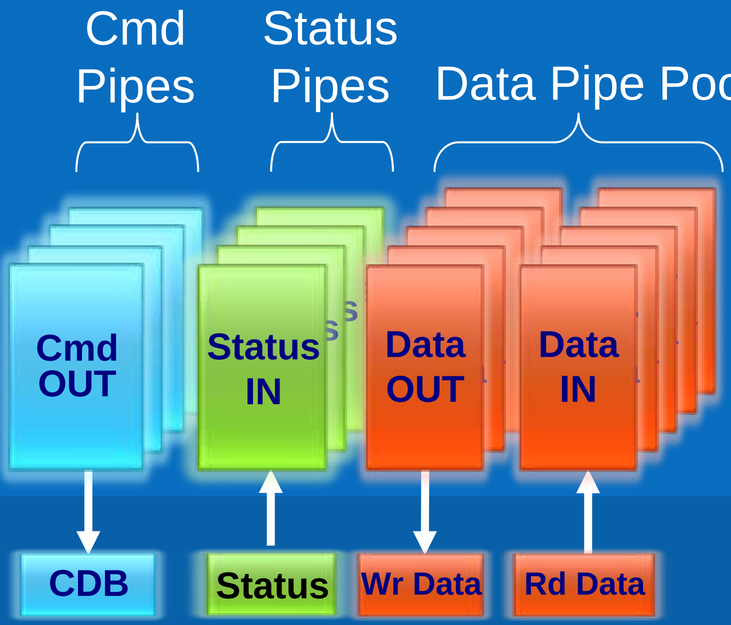
**A “pipe” may be an endpoint or a flow**

# General Approach

- **Support a variable number of Command Slots**
  - 1 to 478
  - Dynamic Pipe Allocation assigns a Data pipe from a pool of 239 IN and 239 OUT
- **Pipe per Command Slot**
  - A Pipe is an USB2 Endpoint or a USB3 Flow
  - IN and OUT Pipe
    - USB2 – IN and OUT Endpoints with the same *Endpoint Number* form a pair
    - USB3 – IN and OUT Flows with the same *Endpoint Number/Flow ID* form a pair
- **In-Order Transfers**
  - Don’t need offsets, Data transfers of a command always start at 0

# Pipe Allocation

- Init time Pipe allocation
- Command OUT Pipes Prioritized
- Status IN Pipes Command Directed
- IN and OUT Data pipes Command Directed
  - IN Data Pipe
    - Receives Read data
  - OUT Data Pipe
    - Transmits Write data



# Pipe Usage

- Individual Data Pipe IDs allow independent Command Slot Operations
- The Data for an individual Command Slots are ordered
- CDBs are independently streamed to the device as they are generated by the host
- CDBs are prioritized by device
- Status responses are transmitted by the device in order of completion
- Due to the Data pipe pool being 50% IN and 50% OUT, the number of outstanding commands will be limited by Read/Write mix
- Individual Data Pipe means that Data Overruns will not exceed allocated system Data buffer space for Cmd

# USB 2 to 3 Comparison

- **USB3**
  - 480 (15*2*16) available pipes
    - 15 Endpoints
    - 2 Directions
    - 16 Flows
  - 478 Max Command Slots
    - 2 pipes min required for Cmd and Status
  - No Polling
    - NRdy/FRdy handshake
- **USB2**
  - 30 (15*2) available pipes
    - 15 Endpoints
    - 2 Directions
  - 14 Max Command Slots
    - 2 pipes required for Cmd and Status
  - Polling
    - Ping/Nyet or IN/NAK handshake

# Recommendations

- **Multiple Alternate Interface Settings**
  - Default setting Bulk-Only Mode
    - Alternate IF 0
  - Limit Max Command Slot Count options
    - USB2 or 3
      - 1 (Bulk), 4, 8, 14
    - USB 3 only
      - 16, 32, 64, 128, 256, 478
- **Fixed Pipe to Command Slot Mapping**
  - Command pipe always uses EP1/Flow0 OUT
  - Status pipe always uses EP1/Flow0 IN
  - Data Pipes allocated in pairs
  - Pack EPs and Flow IDs within an EP
    - 16-478 Slot settings use all flows of respective EPs starting with EP 2-15 then remaining flows of EP 1
      - 16 Slot setting uses all Flows of EP2
      - 32 Slot setting uses all Flows of EP2 and EP3
      - etc.
      - The 478 Slot setting is a special case that uses all flows of EPs 2-15 and Flows 1-15 of EP1 IN/OUT
  - Minimize USB 3 to 2 mapping options
    - 4, 8, and 14 Slot options use EPs only (i.e. Flow 0 only on a USB3 device)