Introduction to Packetized SCSI

Charles Monia
Digital Equipment Corporation

April 27, 1992

Goals

- General upward compatibility with SCSI-2.
  - i.e.: No gratuitous changes
- Extend SCSI-2 Model to include packetized interconnects.
- Define behavior in an implementation-neutral way
- Facilitate implementation using a common code and hardware base that can be ported to different kinds of physical interconnects.

Goals (cont)

- Suitable for any interconnect technology that:
  - Provides "message" class delivery services.
    - i.e.: Sequenced delivery, free of corruption, loss or duplicate data
  - Supports "SCSI domain" topology
- Allow use of heterogeneous interconnect technology within a SCSI network.
  - Define payloads that are common to all packetized interconnects

Differences between packetized and interlocked protocols

- Interlocked Protocol
  - Target and initiator states are synchronized via bus control signals (phases).
  - Control, command and I/O data is distinguished by the phase in effect when the data was received.
- Packetized Protocol
  - Use of bus signals for state synchronization is impractical, i.e.: no bus phases. Synchronization must be based on packet type.
  - Control, command and I/O data is distinguished by "packet type"
Differences (cont)

- Interlocked Protocol defines behavior in terms of bus phases and data exchanged during each phase.

- Packetized protocol defines behavior in terms of packets passed between cooperating entities.

- Packetized behavior described by:
  - Objects
  - Functions
  - Protocols

Architecture Components

- Objects:
  - Can perform functions
  - Can define standard data types, e.g.: SCSI device address, Logical Unit Number.
  - May contain other objects

- Function - An interface between two objects residing on the same SCSI device

- Protocol - An interface between cooperating objects on different SCSI devices.

SAM Scope

- Defines:
  - Hierarchy of objects (includes data objects)
  - Protocols
  - Functions

- Standardizes:
  - Object types.
  - Object behavior specified by function and protocol semantics.
  - For objects within the scope of SAM, protocols between objects

- Avoids introducing a new 'object' when an existing SCSI-2 object can be extended or modified.

SCSI Objects

SCSI Domain

Interconnect

SCSI Domain - Set of SCSI Devices that can be accessed from an SCSI port. View of the network provided by a single SCSI Port

Interconnect - Pathway for the transfer of Commands and data, which provides sequenced, loss-free, duplicate-free transfers without data corruption.

Interconnect may be comprised of heterogeneous physical interconnects.

Each SCSI device has the same view of the domain.

Each SCSI device address references a unique physical device.
SCSI Objects

SCSI Device

To interconnect(s)

SCSI Port ......... SCSI Port

Transport Services

Initiator ...... Target LUN/TRN ...... Target LUN/TRN

SCSI Device - Physical device attached to the interconnect and referenced via a unique device address.

Each SCSI Device must have one or more SCSI ports, optional transport services and at least one of the following:

- One Initiator Object
- A Target LUN/TRN

Charles Monia, Digital Equipment Corporation
April 27, 1992

Multiport SCSI Devices

Multiple Domain

Port A ---- Port B

Single Domain

Port A ---- Port B

Charles Monia, Digital Equipment Corporation
April 27, 1992

SCSI Objects

Target LUN/TRN

Device Model

Target I/O Process

I/O Process Control

I/O Process Queue

Device Model - Conforms to one of the models described in the SBC, SSC or SCC specification.

Target I/O Process - Object which performs the SCSI I/O operation.

Target I/O Process Control- Creates, deletes and queues I/O processes

I/O Process Queue - Queue of uncompleted I/O Processes.

Charles Monia, Digital Equipment Corporation
April 27, 1992

Target LUN/TRN

I/O Process Control

I/O Process Queue

I/O Process

I/O Process Scheduler

Device-specific Model


- Must observe the SCSI queueing model constraints and any other constraints imposed by the device model.

Charles Monia, Digital Equipment Corporation
April 27, 1992
**SCSI Objects**

**Initiator**

- Application
  - Initiates request for I/O service.
- Initiator I/O Process Control
  - Creates I/O processes to service requests
  - Responds to target-requested ABORT messages
  - Transfers CDBs to target
  - Returns status to application
- Initiator I/O process
  - Performs data transfers as directed by target
  - Optionally, receives "autosense" data

---

**Transport Services**

- Perform Object-to-Object Communications
- Decomposes message into packets
- In multi-port systems, selects outgoing port
- Maps destination SCSI device address to destination port address
- Reconstructs message from packets
- Delivers Common Exchange Blocks, preserving the order in which they were received from the sender

---

**Initiator Model**

- Application
  - Commands and Status
- Status
  - CDB
  - Linked CDB
  - Linked CDB
- Data Buffer
- Sense Buffer (optional)
- Initiator I/O Process Control
- Initiator Model
SCSI Port

- Point of attachment to the physical interconnect.
- Decomposes packets into bit stream for transmission over the physical interconnect
- Recovers from transmission errors.
- Converts incoming bit stream into packets in memory free of detectable errors.
Functions

Target I/O Process

Data_out (buffer_offset, byte_count,...)
Data_in (buffer_offset, byte_count,...)
Sense_data_in (byte_count)
Command_out (...)

Initiator I/O Process

Data_out (buffer_offset, byte_count,...)
Data_in (buffer_offset, byte_count,...)
Sense_data_in (byte_count)
Command_out (...)

Note: Data buffer pointer is replaced by buffer_offset

Nexus

Nexus - An association between cooperating I/O processes that begins when a
command descriptor is sent and ends when one of the processes terminates

SPP Protocols

- Command - Response
- Protocol uses common Exchange Control Block
- Exchange block format is independent of interconnect type.

Variable Length Payload

Interconnect-specific addressing info

Exchange Control Block format is TBD.

Nexus Identifier

Function Code etc.