Arquitecturas de Almacenamiento

Tema III

Interfaz SCSI sobre bus paralelo SCSI

Contenidos

I – Interfaz SCSI: Introducción

- Interfaz SCSI aparece en 1978, y se estandariza en 1981
  - Desarrollado por Shugart Associates
  - SCSI = "Small Computers System Interface"
  - Originalmente llamado SASI = "Shugart Associates System Interface"
- Diseñado para actuar como interface para discos duros de altas prestaciones
  - Implementando sobre un bus paralelo de 8 bits
- Tuvo gran aceptación industrial y comercial
  - Su uso se extendió a otros tipos de dispositivos de almacenamiento

Interfaz SCSI: Introducción

- Versión original del interfaz se estandariza como SCSI-1
- Desarrollos posteriores sobre bus paralelo dieron lugar a la estandarización SCSI-2
  - Premura de fabricantes por comercializar productos SCSI-2 produjeron problemas:
    - Implementaciones incompletas del estándar
    - Uso confuso de nombres propietarios para anchos de datos y velocidades de transmisión.
- Estandarización SCSI-2 no cubría adecuadamente el uso del SCSI sobre otros buses distintos al SCSI paralelo

Estandarización SCSI-3

- Para solucionar estos problemas, a principios de los años 2000 se desarrolla la estandarización SCSI-3

- SCSI-3 ya no es un único estándar, sino un conjunto de estándares interrelacionados ments defined in this standard and the appropriate SCSI implementation standards. In the event of a conflict between this document and other SCSI standards under the jurisdiction of technical committee T10, the requirements of this standard shall apply.

1.3 SCSI standards family

Figure 2 shows the relationship of this standard to the other standards and related projects in the SCSI family standards as of the publication of this standard.

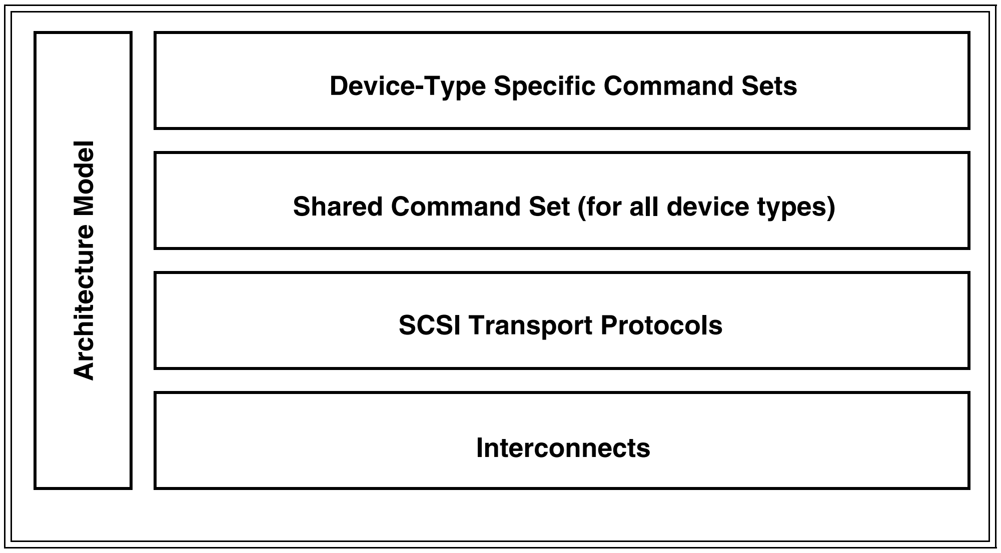

Architecture Model
Device-Type Specific Command Sets
Shared Command Set (for all device types)
SCSI Transport Protocols
Interconnects

Figure 2 — SCSI document structure

The roadmap in figure 2 is intended to show the general applicability of the documents to one another. Figure 2 is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture.

The functional areas identified in figure 2 characterize the scope of standards within a group as follows:

Architecture Model: Defines the SCSI systems model, the functional partitioning of the SCSI standard set and requirements applicable to all SCSI implementations and implementation standards.

Device-Type Specific Command Sets: Implementation standards that define specific device types including a device model for each device type. These standards specify the required commands and behavior that is specific to a given device type and prescribe the requirements to be followed by a SCSI initiator device when sending commands to a SCSI target device having the specific device type. The commands and behaviors for a specific device type may include by reference commands and behaviors that are shared by all SCSI devices.

Shared Command Set: An implementation standard that defines a model for all SCSI device types. This standard specifies the required commands and behavior that is common to all SCSI devices, regardless of device type, and prescribes the requirements to be followed by a SCSI initiator device when sending commands to any SCSI target device.

SCSI Transport Protocols: Implementation standards that define the requirements for exchanging information so that different SCSI devices are capable of communicating.

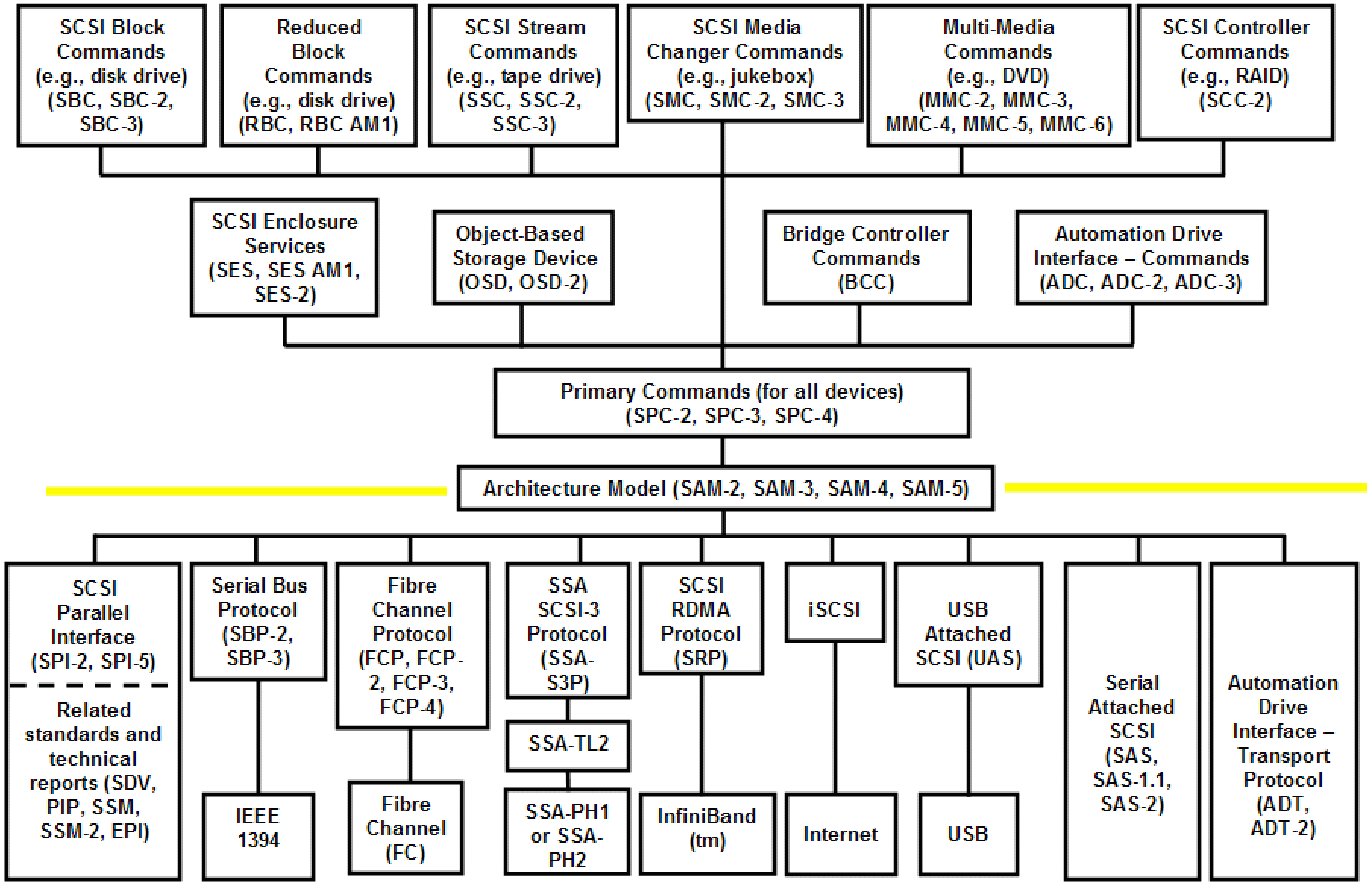
Estandarización SCSI-3

- Modelo de estándares SCSI-3 contempla una arquitectura de torre de protocolos

  - Comandos son enviados a los dispositivos, y ejecutados, en los niveles más altos.

  - Comandos, y datos de respuesta, son movidos mediante protocolos de transporte.

  - Los datos físicos pasan a través de un bus de interconexión, existiendo múltiples posibilidades para elegir el tipo de bus más adecuado.


SCSI Block Commands (e.g., disk drive) (SBC, SBC-2, SBC-3)
Reduced Block Commands (e.g., disk drive) (RBC, RBC AM1)
SCSI Stream Commands (e.g., tape drive) (SSC, SSC-2, SSC-3)
SCSI Media Changer Commands (e.g., jukebox) (SMC, SMC-2, SMC-3
Multi-Media Commands (e.g., DVD) (MMC-2, MMC-3, MMC-4, MMC-5, MMC-6)
SCSI Controller Commands (e.g., RAID) (SCC-2)
SCSI Enclosure Services (SES, SES AM1, SES-2)
Object-Based Storage Device (OSD, OSD-2)
Bridge Controller Commands (BCC)
Automation Drive Interface – Commands (ADC, ADC-2, ADC-3)
Primary Commands (for all devices) (SPC-2, SPC-3, SPC-4)
Architecture Model (SAM-2, SAM-3, SAM-4, SAM-5)
SCSI Parallel Interface (SPI-2, SPI-5)
Related standards and technical reports (SDV, PIP, SSM, SSM-2, EPI)
Serial Bus Protocol (SBP-2, SBP-3)
IEEE 1394
Fibre Channel Protocol (FCP, FCP-2, FCP-3, FCP-4)
Fibre Channel (FC)
SSA SCSI-3 Protocol (SSA-S3P)
SSA-TL2
SSA-PH1 or SSA-PH2
SCSI RDMA Protocol (SRP)
InfiniBand (tm)
iSCSI
Internet
USB Attached SCSI (UAS)
USB
Serial Attached SCSI (SAS, SAS-1.1, SAS-2)
Automation Drive Interface – Transport Protocol (ADT, ADT-2)

(*) This chart reflects the currently approved SCSI project family. T10 is responsible for all projects, except: IEEE is responsible for IEEE 1394; T11

4 SCSI architecture model

4.1 Introduction

The purpose of the SCSI architecture model is to:

a) Provide a basis for the coordination of SCSI standards development that allows each standard to be placed into perspective within the overall SCSI Architecture model;
b) Identify areas for developing standards and provide a common reference for maintaining consistency among related standards so that independent teams of experts may work productively and independently on the development of standards within each functional area; and
c) Provide the foundation for application compatibility across all SCSI interconnect and SCSI transport protocol environments by specifying generic requirements that apply uniformly to all implementation standards within each functional area.

The development of this standard is assisted by the use of an abstract model. To specify the external behavior of a SCSI system, elements in a system are replaced by functionally equivalent components within this model. Only externally observable behavior is retained as the standard of behavior. The description of internal behavior in this standard is provided only to support the definition of the observable aspects of the model. Those aspects are limited to the generic properties and characteristics needed for host applications to interoperate with SCSI devices in any SCSI interconnect and SCSI transport protocol environment. The model does not address other requirements that may be essential to some I/O system implementations such as the mapping from SCSI device addresses to network addresses, the procedure for discovering SCSI devices on a network, and the definition of network authentication policies for SCSI initiator devices or SCSI target devices. These considerations are outside the scope of the architecture model.

The set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

The SCSI architecture model is described in terms of objects (see 3.1.71), protocol layers and service interfaces between objects. As used in this standard, objects are abstractions, encapsulating a set of related functions, data types, and other objects. Certain objects, such as an interconnect, are defined by SCSI, while others, such as a task, are needed to understand the functioning of SCSI but have implementation definitions outside the scope of SCSI. That is, although such objects exhibit well-defined and observable behaviors, they do not exist as separate physical elements. An object may be a single numeric parameter, such as a logical unit number, or a complex entity that performs a set of operations or services on behalf of another object.

Service interfaces are defined between distributed objects and protocol layers. The template for a distributed service interface is the client-server model described in 4.2. The structure of a SCSI I/O system is specified in 4.4 by defining the relationship among objects. The set of distributed services to be provided are specified in clause 5 and clause 7.

Requirements that apply to each SCSI transport protocol standard are specified in the SCSI transport protocol service model described in 5.4 and 7.10. The model describes required behavior in terms of layers, objects within layers and SCSI transport protocol service transactions between layers.

Estandarización SCSI-3

- El modelo arquitectural SCSI (*SAM = SCSI Architecture Model*) es la estandarización que define:
  - El modelo de sistemas SCSI
  - La partición funcional (división de funciones) del juego de protocolos SCSI
  - Los requerimientos aplicables a todas las implementaciones SCSI
- Todos los dispositivos SCSI actualmente disponibles, independientemente de su complejidad o bus, se atienen a esta estructura jerárquica
  - Obligatoriamente, implementarán un conjunto común de comandos de servicio
    - SCSI Primary Commands (SPC)
  - Después, impementarán los comandos específicos de su tipo de dispositivo.

1.1.1 Scope of SCSI standards

Figure 1 uses a representative set of specifications to show the functional partitions and the relationships among SCSI standards applicable to drives covered by this manual.

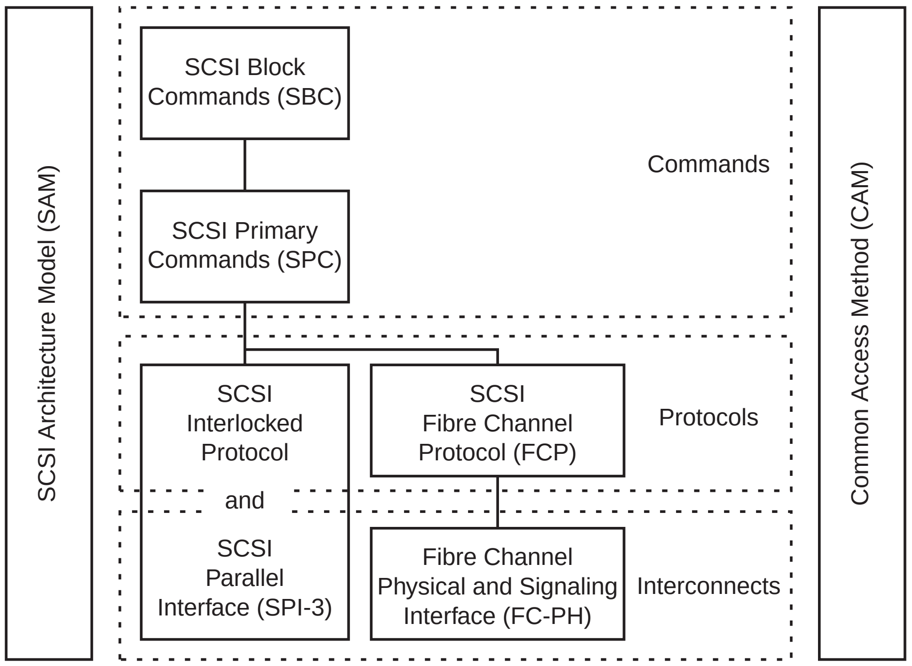

SCSI Architecture Model (SAM)
SCSI Block Commands (SBC)
SCSI Primary Commands (SPC)
Commands
SCSI Interlocked Protocol
and
SCSI Parallel Interface (SPI-3)
SCSI Fibre Channel Protocol (FCP)
Protocols
Fibre Channel Physical and Signaling Interface (FC-PH)
Interconnects
Common Access Method (CAM)

Figure 1. Functional scope of SCSI[1] standards

The functional areas define the scope of each standard as follows:

- SCSI Architecture Model: Defines the SCSI systems model, the functional partitioning of the SCSI standard set and requirements applicable to all SCSI implementations and implementation standards.
- Commands: Implementation standards which define classes including a device model for each class. These standards specify the required commands and behavior that is common to all devices or unique to a given class of devices and prescribe the rules to be followed by a SCSI initiator port when sending commands to a device.
- Common Access Method: Implementation standard which defines a host architecture and set of services for device access.
- Protocols: Implementation standards which define the rules for exchanging information so that different SCSI devices can communicate.
- Interconnects: Implementation standards which define the electrical and signaling rules essential for devices to interoperate over a given physical interconnect.

[1] The diagram of Figure 1 shows how the standards listed below fit within each category. The standards included in the diagram are meant to serve as examples and may not reflect the full set of standards currently in force.

4.2 The SCSI distributed service model

Service interfaces between distributed objects are represented by the client-server model shown in figure 5. Dashed horizontal lines with arrowheads denote a single request-response transaction as it appears to the client and server. The solid lines with arrowheads indicate the actual transaction path through the service delivery subsystem. In such a model, each client or server is a single thread of processing that runs concurrently with all other clients or servers.

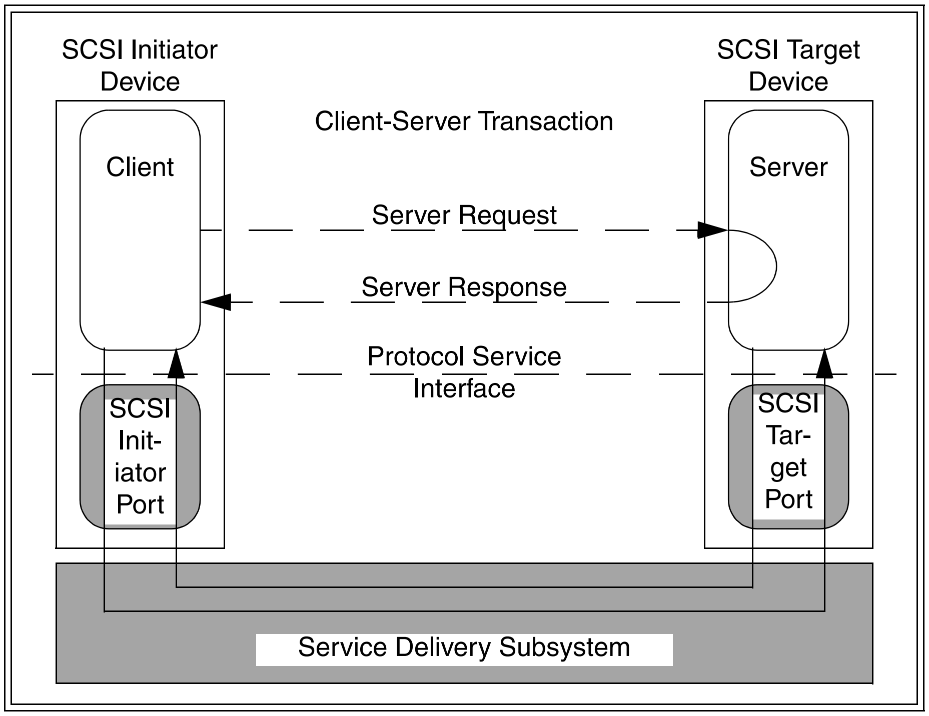


Figure 5 — Client-Server model

A client-server transaction is represented as a procedure call with inputs supplied by the caller (i.e., the client). The procedure call is processed by the server and returns outputs and a procedure call status. A client directs requests to a remote server via the SCSI initiator port and service delivery subsystem and receives a completion response or a failure notification. The request identifies the server and the service to be performed and includes the input data. The response conveys the output data and request status. The function of the service delivery subsystem is to transport error-free copies of the requests and responses between sender and receiver. A failure notification indicates that a condition has been detected (e.g., a reset or service delivery failure) that precludes request completion.

As seen by the client, a request becomes pending when it is passed to the SCSI initiator port for transmission. The request is complete when the server response is received or when a failure notification is sent. As seen by the server, the request becomes pending upon receipt and completes when the response is passed to the SCSI target port for return to the client. As a result there may be a time skew between the server and client's perception of request status and server state. All references to a pending command or task management function in this standard are from the application client's point of view (see 5.5 and 5.6).

Client-server relationships are not symmetrical. A client may only originate requests for service. A server may only respond to such requests.

The client requests an operation provided by a server located in another SCSI device and waits for completion, which includes transmission of the request to and response from the remote server. From the client's point of view, the behavior of a service requested from another SCSI device is indistinguishable from a request processed in the same SCSI device. In this model, confirmation of successful request or response delivery by the sender is not required. The model assumes that delivery failures are detected by the SCSI initiator port or within the service delivery subsystem.

4.3 The SCSI client-server model

As shown in figure 6, each SCSI target device contains one or more logical units and provides services performed by device servers and task management functions performed by task managers. A logical unit is an object that implements one of the device functional models described in the SCSI command standards and processes commands (e.g., reading from or writing to the media). Each pending command or series of linked commands defines a unit of work to be performed by the logical unit. Each unit of work is represented within the SCSI target device by a task that may be externally referenced and controlled through requests issued to the task manager.

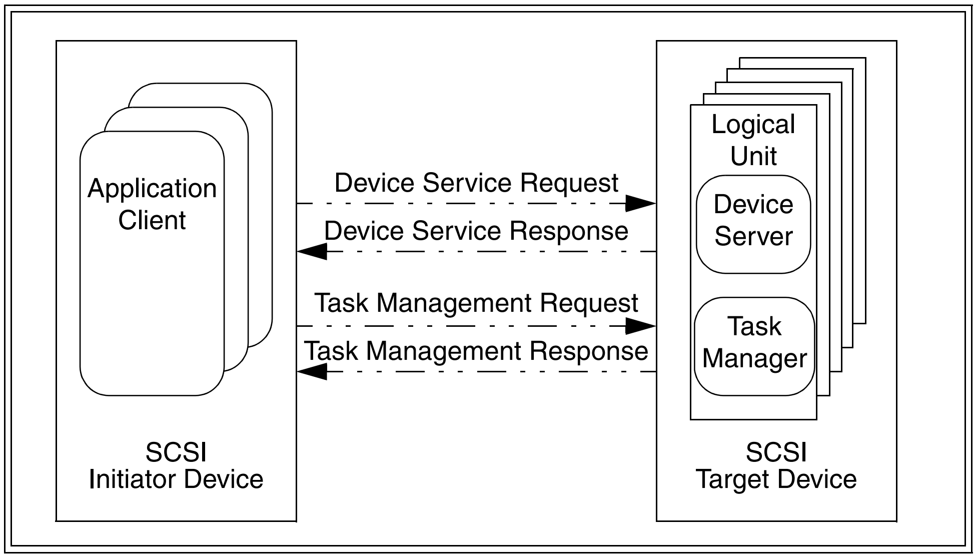


Figure 6 — SCSI client-server model

All requests originate from application clients residing within a SCSI initiator device. An application client is independent of the interconnect and SCSI transport protocol (e.g., an application client may correspond to the device driver and any other code within the operating system that is capable of managing I/O requests without requiring knowledge of the interconnect or SCSI transport protocol). An application client creates one or more application client tasks each of which issues a single command, a series of linked commands, or a task management function. Application client tasks are part of their parent application client. An application client task ceases to exist once the command, series of linked commands, or task management function ends.

As described in 4.2, each request takes the form of a procedure call with arguments and a status to be returned. An application client may request processing of a command through a request directed to the device server within a logical unit. Each device service request contains a CDB defining the operation to be performed along with a list of command specific inputs and other parameters specifying how the command is to be processed. If supported by a logical unit, a series of linked commands may be used to define an extended I/O operation.

A task is an object within the logical unit representing the work associated with a command or series of linked commands. A new command or the first in a series of linked commands causes the creation of a task. The task persists until a task complete response is sent or until the task is ended by a task management function or exception condition. For an example of the processing for a single command see 5.8.1. For an example of linked command processing see 5.8.2.

Arquitectura cliente-servidor

- Modelo de operación SCSI es una arquitectura cliente-servidor
  - Dispositivo periférico (disco, cinta, intercambiador robótico, etc) es el servidor
    - Ofrece acceso a un servicio para una aplicación cliente
  - Dispositivo que envía el comando es el cliente
    - Solicita hacer uso del servicio ofertado
- Por tanto:
  - Iniciador = genera comando del interfaz SCSI = cliente que accede al servicio
  - Diana (target) = ejecuta comando = servidor que ofrece la funcionalidad

Arquitectura cliente-servidor

- Una consecuencia de esta arquitectura cliente-servidor es que es la diana la que gestiona:
  - El control de flujo
  - El orden de ejecución de los comandos

An application client may request processing of a task management function through a request directed to the task manager within the logical unit. The interactions between the task manager and application client when a task management request is processed are shown in 7.9.

4.4 The SCSI structural model

The SCSI structural model represents a view of the elements comprising a SCSI I/O system as seen by the application clients interacting with the system. As shown in figure 7, the fundamental object is the SCSI domain that represents an I/O system. A domain is made up of SCSI devices and a service delivery subsystem that transports commands, data, task management functions, and related information. A SCSI device contains clients or servers or both and the infrastructure to support them.

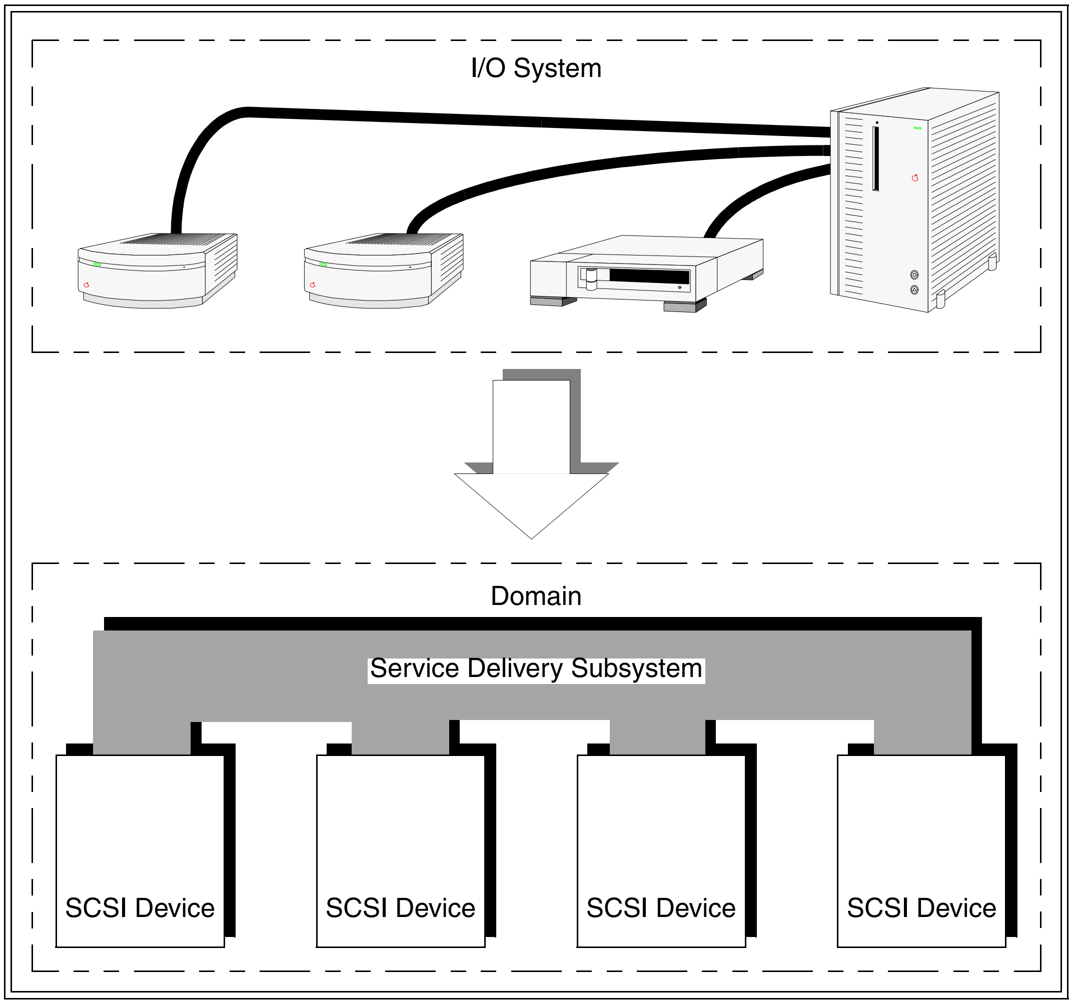


Figure 7 — SCSI I/O system and domain model

Device ID y LUN

- Llamamos *dominio SCSI* al conjunto de dispositivos SCSI interconectados por un mismo subsistema de transporte físico de datos.
  - Es una definición más genérica que usar el nombre “bus”
- Dentro de un dominio SCSI, tanto iniciador como diana deben tener un identificador físico únivoco
  - Es llamado “Device Name”
- En el bus paralelo SCSI, es el Device ID, número asignado de forma única entre 0-7 (o 0-15)
  - Cuando tenemos más de un bus SCSI en el mismo ordenador, el “Device Name” es la combinación de:
    - El número de bus al que está conectado el dispositivo
    - El Device ID del dipositivo

4.7 SCSI devices

A SCSI device is a SCSI target device, a SCSI initiator device, or a SCSI target/initiator device.

A SCSI initiator device contains at least one SCSI initiator port and is capable of originating commands and task management requests (see 4.7.1). A SCSI target device contains at least one SCSI target port, at least one logical unit, and is capable of processing commands and task management requests (see 4.7.2). A SCSI target/initiator device contains at least one SCSI target/initiator port, at least one logical unit, and is capable of originating and processing commands and task management requests (see 4.7.3).

To be functional, a SCSI domain is required to contain a SCSI target port or a SCSI target/initiator port operating as a SCSI target port and a SCSI initiator port or SCSI target/initiator port operating as a SCSI initiator port.

4.7.1 SCSI initiator device

A SCSI initiator device (see figure 10) contains:

a) Zero or more initiator device names;
b) One or more SCSI initiator ports, each containing an:
   A) Initiator port identifier; and
   B) Optional initiator port name;
c) One or more application clients; and
d) Zero or more application client tasks.

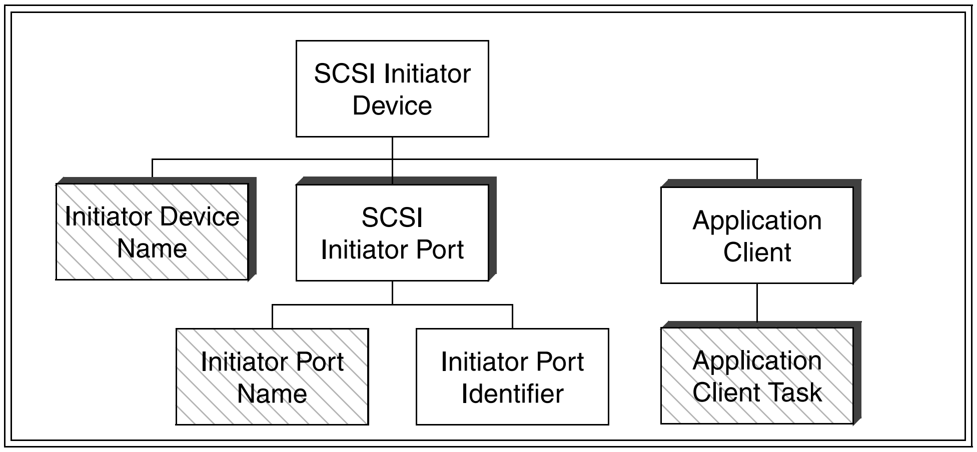


Figure 10 — SCSI initiator device model

The initiator port identifier is a value that is the SCSI port identifier (see 4.7.4) for a SCSI initiator port.

An initiator device name is a name (see 3.1.65) that is a SCSI device name (see 4.7.7) for a SCSI initiator device. For each supported SCSI transport protocol, a SCSI initiator device shall have no more than one (i.e., zero or one) SCSI initiator device name that is not in the SCSI name string format (see SPC-3). A SCSI initiator device shall have no more than one (i.e., zero or one) SCSI initiator device name in the SCSI name string format regardless of the number of SCSI transport protocols supported by the SCSI initiator device. If a SCSI initiator device has a SCSI device name in the SCSI name string format then the SCSI initiator device should have only one SCSI initiator device name. A SCSI transport protocol standard may place additional requirements on initiator device names.

The initiator port name is a name (see 3.1.65) that is the SCSI port name (see 4.7.8) for the SCSI initiator port. A SCSI transport protocol standard may place additional requirements on initiator port names.

Application clients are the sources of commands and task management functions.

An application client task is the source for a single command, series of linked commands, or a single task management function.

4.7.2 SCSI target device

A SCSI target device (see figure 11) contains:

a) Zero or more target device names;
b) One or more SCSI target ports, each containing:
    A) A task router;
    B) A target port identifier;
    C) An optional relative port identifier; and
    D) An optional target port name;
    and
c) One or more logical units.

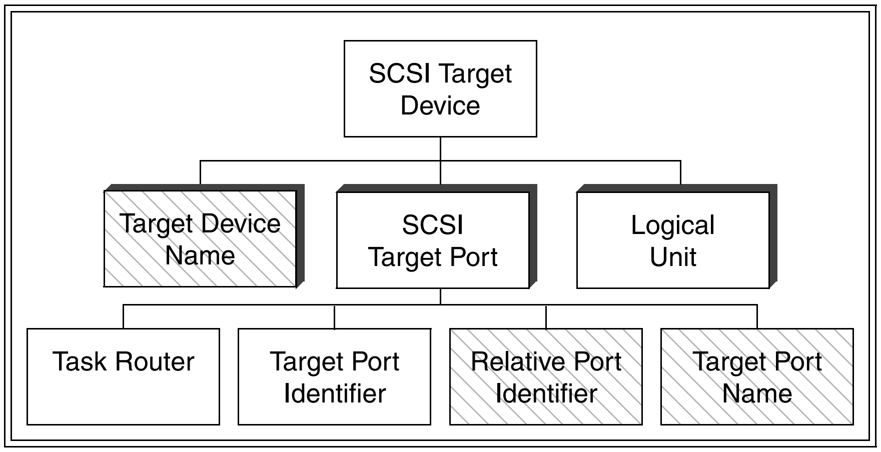


Figure 11 — SCSI target device model

The target port identifier is a value that is a SCSI port identifier (see 4.7.4) for a SCSI target port.

The relative port identifier (see 4.7.5) identifies the SCSI target port relative to other SCSI ports in the SCSI target device.

A target device name is a name (see 3.1.65) that is a SCSI device name (see 4.7.7) for a SCSI target device. For each supported SCSI transport protocol, a SCSI target device shall have no more than one (i.e., zero or one) SCSI target device name that is not in the SCSI name string format (see SPC-3). A SCSI target device shall have no more than one (i.e., zero or one) SCSI target device name in the SCSI name string format regardless of the number of SCSI transport protocols supported by the SCSI target device. If a SCSI target device has a SCSI device name in the SCSI name string format then the SCSI target device should have only one SCSI target device name. A SCSI transport protocol standard may place additional requirements on target device names.

The target port name is a name (see 3.1.65) that is the SCSI port name (see 4.7.8) for the SCSI target port. A SCSI transport protocol standard may place additional requirements on target port names.

The task router routes commands and task management functions between the service delivery subsystem and the appropriate logical unit's task manager (see 4.7.6).

A logical unit is the object to which commands are sent. One of the logical units within the SCSI target device shall be accessed using the logical unit number zero or the REPORT LUNS well-known logical unit number. See 4.8 for a description of the logical unit.

Device ID y LUN

- Para un iniciador basta con su Device Name
- Para la diana, sin embargo, no basta con el identificador físico
  - Puede ofrecer varias funciones distintas
    - Ej: varias particiones de un disco físico, ofrecidas como unidades lógicas distintas
  - Cada función es un servidor distinto ofrecido por la diana
- Para diferenciar entre los diversos servidores (funciones) ofrecidos por una misma diana, necesitamos un segundo identificador
  - Llamado LUN = Logical Unit
- Diana siempre ofrecerá, por tanto, una o más LUNs
  - LUN0 siempre debe existir, y es la usada para acceder a funciones de gestión o informativas.

Figure 8 shows the main functional components of the SCSI domain. This standard defines these components in greater detail.

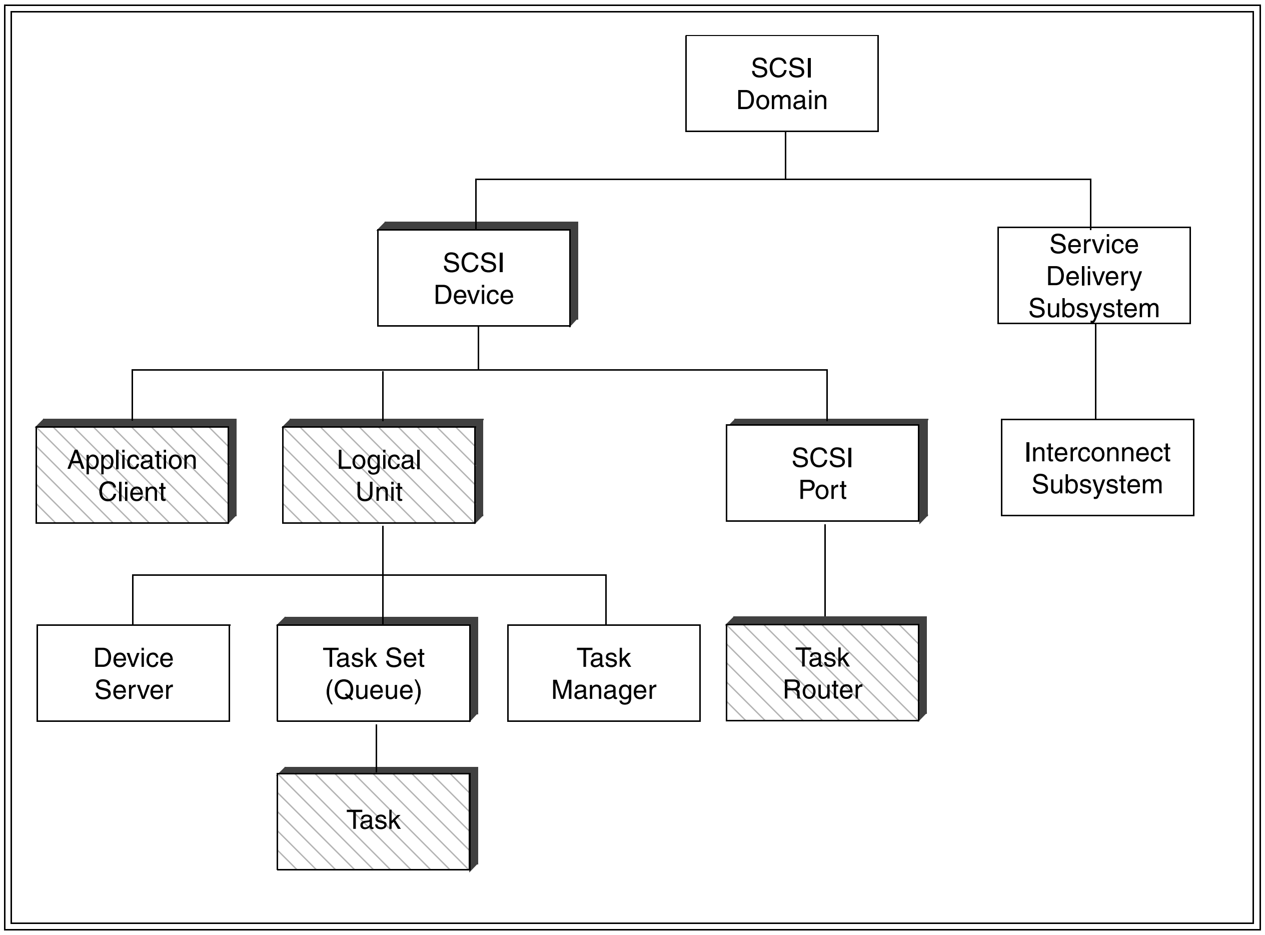


Figure 8 — Overall SCSI domain model

Device ID y LUN

- Cada LUN de la diana tendrá una o varias colas de tareas (comandos)
  - Le permitirán, si lo soporta, reordenar la ejecución de los comandos.
  - Por tanto, si un disco físico ofrece varias LUNs (particiones), podría serle requerido que ejecute concurrentemente varios comandos.
  - Es necesario tener ésto en cuenta a la hora de valorar el rendimiento del disco.

- Presencia de colas requiere que cada comando tenga un identificador único en la cola
  - Este identificador es llamado *Tag*

Modelo distribuido del SCSI

- En el estándar SCSI-3, las comunicaciones entre iniciador y diana se consideran implementadas en un modelo de capas de protocolo.

- Los comandos y datos generados por aplicaciones en iniciador o diana son encapsulados en un protocolo de transporte
  - Este protocolo de transporte, a su vez, es encapsulado en el protocolo usado en el bus de interconexión

- Concepción como capas de protocolo, y encapsulamiento, permite usar el mismo comando SCSI sobre cualquier bus
  - FC, SAS, USB, Firewire, ...; todos usan el mismo juego básico de comandos SCSI

4.15 The SCSI model for distributed communications

The SCSI model for communications between distributed objects is based on the technique of layering as shown in figure 25.

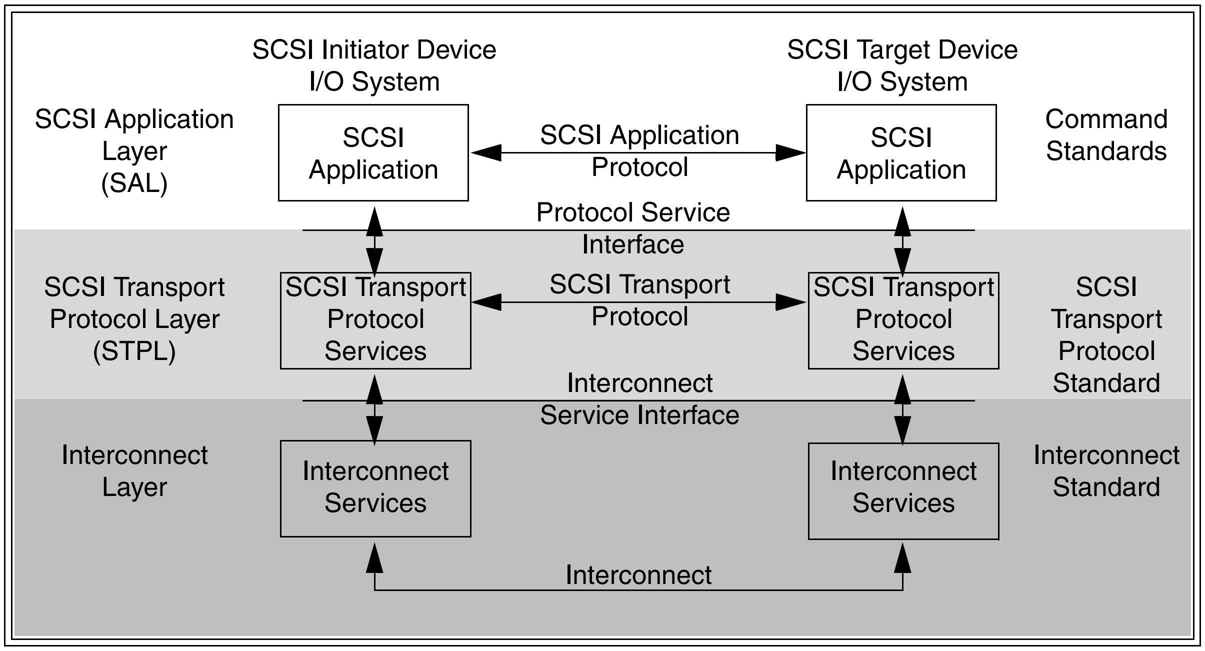

SCSI Initiator Device I/O System
SCSI Target Device I/O System
SCSI Application Layer (SAL)
SCSI Application
SCSI Application Protocol
SCSI Application
Command Standards
Protocol Service Interface
SCSI Transport Protocol Layer (STPL)
SCSI Transport Protocol Services
SCSI Transport Protocol
SCSI Transport Protocol Services
SCSI Transport Protocol Standard
Interconnect Service Interface
Interconnect Layer
Interconnect Services
Interconnect Services
Interconnect Standard
Interconnect

Figure 25 — Protocol service reference model

The layers comprising this model and the specifications defining the functionality of each layer are denoted by horizontal sequences. A layer consists of peer entities that communicate with one another by means of a protocol. Except for the interconnect layer, such communication is accomplished by invoking services provided by the adjacent layer. The following layers are defined.

SCSI application layer (SAL): Contains the clients and servers that originate and process SCSI I/O operations by means of a SCSI application protocol.

SCSI transport protocol layer (STPL): Consists of the services and protocols through which clients and servers communicate.

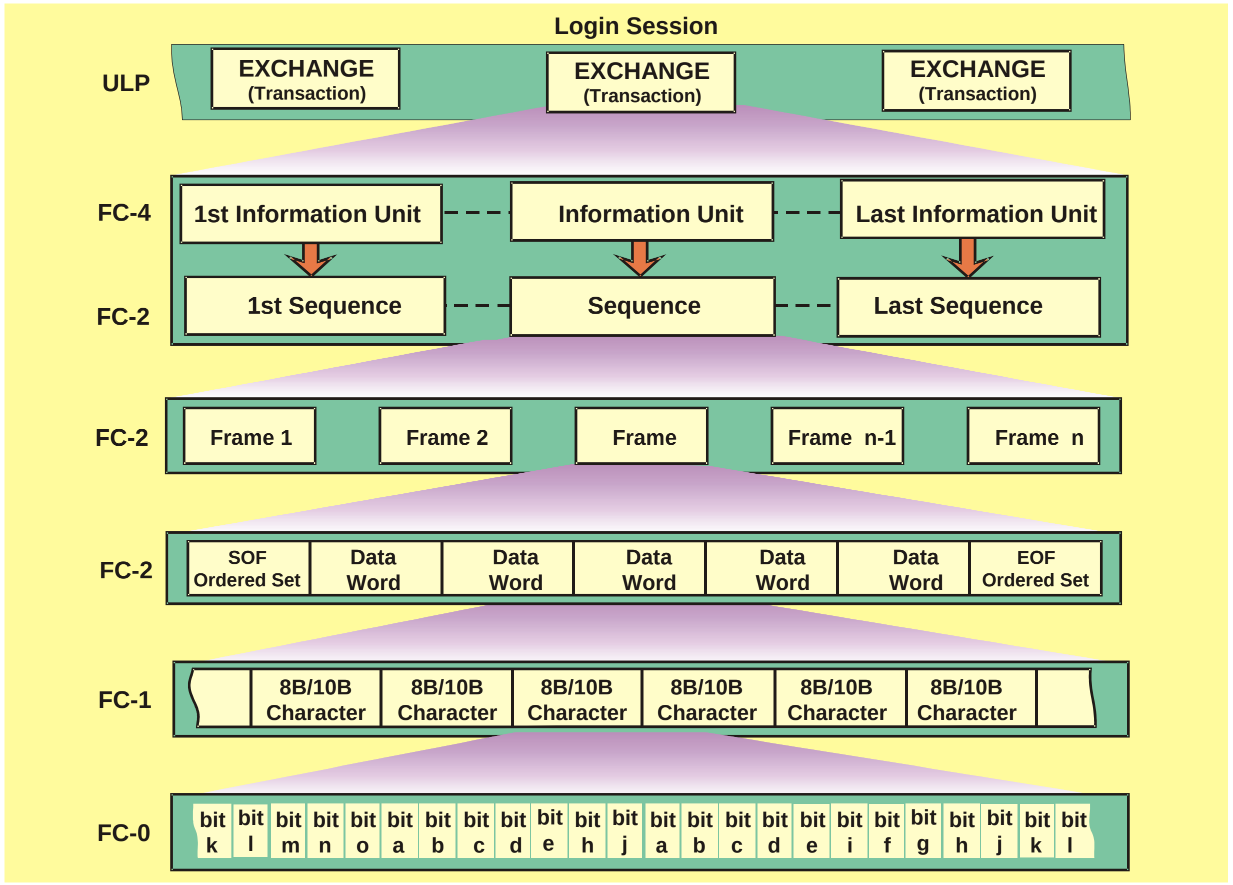
Interconnect layer: Comprised of the services, signaling mechanism and interconnect subsystem used for the physical transfer of data from sender to receiver. In the SCSI model, the interconnect layer is known as the service delivery subsystem.

The set of SCSI transport protocol services implemented by the service delivery subsystem identify external behavioral requirements that apply to SCSI transport protocol standards. While these SCSI transport protocol services may serve as a guide for designing reusable software or firmware that is adaptable to different SCSI transport protocols, there is no requirement for an implementation to provide the service interfaces specified in this standard.

The SCSI transport protocol service interface is defined in this standard in representational terms using SCSI transport protocol services. The SCSI transport protocol service interface implementation is defined in each SCSI transport protocol standard. The interconnect service interface is described as appropriate in each SCSI transport protocol standard.

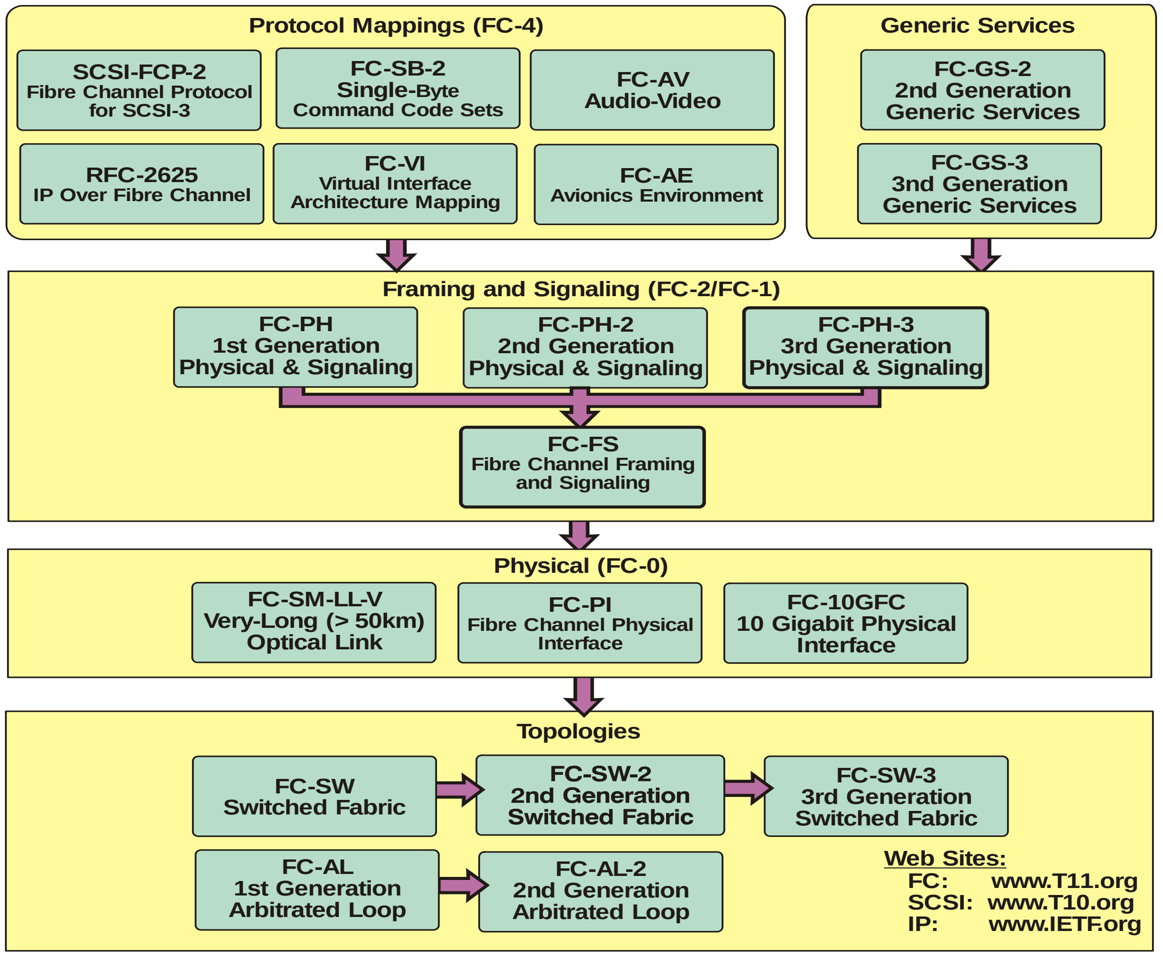
General Information: Fibre Channel Protocol Levels

EDUCATION



General Information:
Fibre Channel Documentation Structure

SNIA
STORAGE NETWORKING INDUSTRY ASSOCIATION
EDUCATION



Protocol Mappings (FC-4)
SCSI-FCP-2 Fibre Channel Protocol for SCSI-3
FC-SB-2 Single-Byte Command Code Sets
FC-AV Audio-Video
RFC-2625 IP Over Fibre Channel
FC-VI Virtual Interface Architecture Mapping
FC-AE Avionics Environment
Generic Services
FC-GS-2 2nd Generation Generic Services
FC-GS-3 3nd Generation Generic Services
Framing and Signaling (FC-2/FC-1)
FC-PH 1st Generation Physical & Signaling
FC-PH-2 2nd Generation Physical & Signaling
FC-PH-3 3rd Generation Physical & Signaling
FC-FS Fibre Channel Framing and Signaling
Physical (FC-0)
FC-SM-LL-V Very-Long (> 50km) Optical Link
FC-PI Fibre Channel Physical Interface
FC-10GFC 10 Gigabit Physical Interface
Topologies
FC-SW Switched Fabric
FC-SW-2 2nd Generation Switched Fabric
FC-SW-3 3rd Generation Switched Fabric
FC-AL 1st Generation Arbitrated Loop
FC-AL-2 2nd Generation Arbitrated Loop

Web Sites:

FC:	www.T11.org
SCSI:	www.T10.org
IP:	www.IETF.org

Figure 27 shows how SCSI transport protocol services may be used to process a client-server request-response transaction at the SCSI application layer.

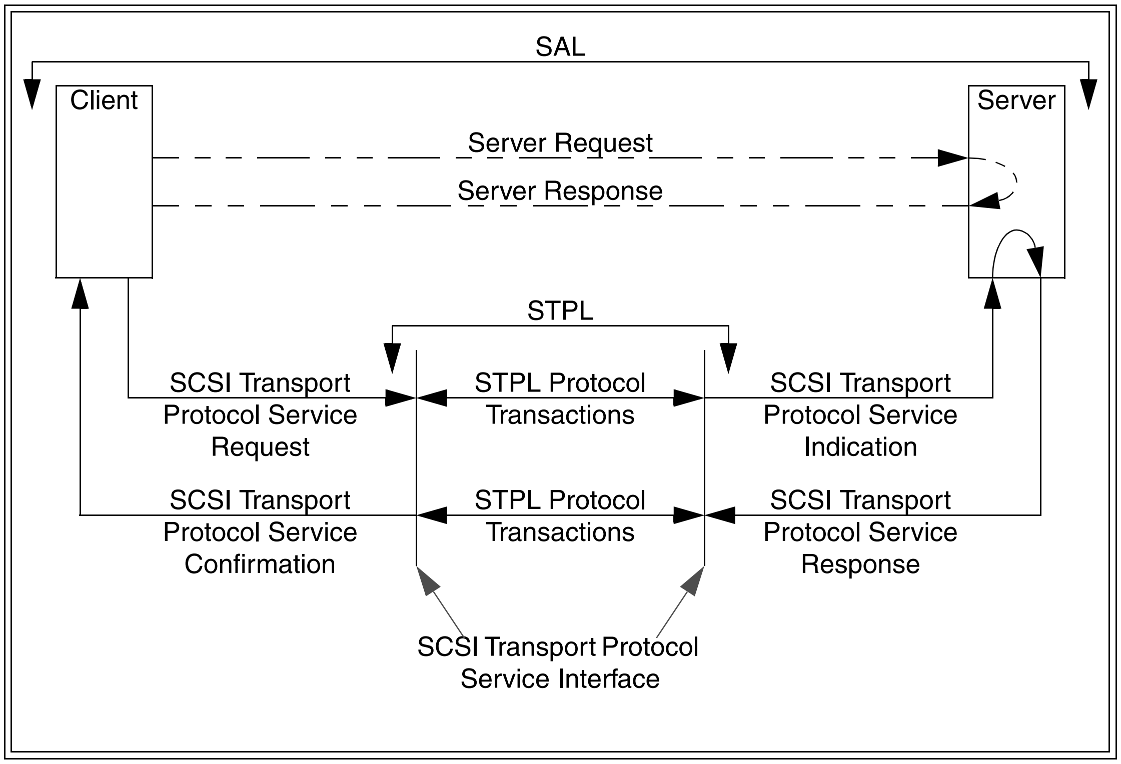

SAL
Client
Server
Server Request
Server Response
STPL
SCSI Transport Protocol Service Request
STPL Protocol Transactions
SCSI Transport Protocol Service Indication
SCSI Transport Protocol Service Confirmation
STPL Protocol Transactions
SCSI Transport Protocol Service Response
SCSI Transport Protocol Service Interface

Figure 27 — Request-Response SAL transaction and related STPL services

The dashed lines in figure 27 show a SCSI application protocol transaction as it may appear to sending and receiving entities within the client and server. The solid lines in figure 27 show the corresponding SCSI transport protocol services and STPL transactions that are used to transport the data.

Comandos y respuestas SCSI

- Iniciador envía comandos (peticiones de acceso a servicio) a la diana (servidor)

- Comando es comunicado enviando un CDB (*Command Descriptor Block*)
  - Conjunto estructurado de bytes, que será interpretado por la diana como un requerimiento de ejecución de un comando, y sus parámetros.
  - Pueden ser de longitud fija (6, 10, 12 o 16 bytes), o variable
  - Deben ir precedidos de un mensaje que indique a qué cola de tareas en la LUN van destinados

- A la terminación de un comando la diana devolverá un código de estado

2.1.2 The fixed length CDB formats

All fixed length CDBs shall have an OPERATION CODE field as their first byte and a CONTROL byte as their last byte. Table 2 shows the typical format of a 6-byte CDB. Table 3 shows the typical format of a 10-byte CDB. Table 4 shows the typical format of a 12-byte CDB. Table 5 shows the typical format of a 16-byte CDB. Table 6 shows the format of a 16-byte CDB for commands that provide for a long LBA.

Table 2. Typical CDB for 6-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			(MSB)				
2	LOGICAL BLOCK ADDRESS (if required)							
3								(LSB)
4	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
5	CONTROL							

Table 3. Typical CDB for 10-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								(LSB)
6	Miscellaneous CDB information							
7	(MSB)	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
8								(LSB)
9	CONTROL							

Table 4. Typical CDB for 12-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								(LSB)
6	(MSB)							
7	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
8								
9								(LSB)
10	Miscellaneous CDB information							
11	CONTROL							

Table 5. Typical CDB for 16-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)]				
2	(MSB)	LOGICAL BLOCK ADDRESS (if required)						
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								(LSB)
6	(MSB)	Additional CDB data (if required)						
7	Additional CDB data (if required)							
8								
9								(LSB)
10	(MSB)	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
11	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
12								
13								(LSB)
14	Miscellaneous CDB information							
15	CONTROL							

Table 6. Typical CDB for long LBA 16-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information							
2	(MSB)							
3	LOGICAL BLOCK ADDRESS							
4								
5								
6								
7								
8								
9								(LSB)
10	(MSB)							
11	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
12								
13								(LSB)
14	Miscellaneous CDB information							
15	Control							

2.1.3 The variable length CDB formats

The first byte of a variable length CDB shall contain the operation code 7Fh. The CONTROL byte is the second byte in the variable length CDB (see table 7)

Table 7. Typical variable length CDB

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Miscellaneous CDB information							
3	Miscellaneous CDB information							
4	Miscellaneous CDB information							
5	Miscellaneous CDB information							
6	Miscellaneous CDB information							
7	ADDITIONAL CDB LENGTH (n–7)							
8	(MSB)	SERVICE ACTION						
9								(LSB)
10 : n	Service Action specific fields							

ADDITIONAL CDB LENGTH field

The ADDITIONAL CDB LENGTH field specifies the number of additional CDB bytes. This value in the ADDITIONAL CDB LENGTH field shall be a multiple of 4. If the number of CDB bytes delivered by the service delivery subsystem is not sufficient to contain the number of bytes specified by the ADDITIONAL CDB LENGTH field, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

SERVICE ACTION field

The SERVICE ACTION field specifies the action being requested by the application client. The SERVICE ACTION field is required in the variable length CDB format and is described in 4.3.4.2. Each service action code description defines a number of service action specific fields that are needed for that service action.

A 32-byte variable length CDB format is defined for long LBA operations (see table 8)

Table 8. Typical variable length CDB for long LBA 32-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Miscellaneous CDB information							
3	Miscellaneous CDB information							
4	Miscellaneous CDB information							
5	Miscellaneous CDB information							
6	Miscellaneous CDB information							
7	Additional CDB Length (n–7) [9]							
8	(MSB)	SERVICE ACTION						
9								(LSB)
10	Miscellaneous CDB information			DPO	FUA	Miscellaneous CDB information		
11	Miscellaneous CDB information							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	Miscellaneous CDB information							
27								
28	(MSB)	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
31								(LSB)

2.2 Common CDB fields

2.2.1 Operation Code

The first byte of a SCSI CDB shall contain an operation code identifying the operation being requested by the CDB. Some operation codes provide for modification of their operation based on a service action (see 2.1.4.2). In such cases, the operation code and service action code combine to identify the operation being requested. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

The OPERATION CODE (see table 10) of the CDB has a GROUP CODE field and a COMMAND CODE field. The three-bit GROUP CODE field provides for eight groups of command codes. The five-bit COMMAND CODE field provides for thirty-two command codes in each group. A total of 256 possible operation codes exist. Operation codes are defined in this manual and other command standards. The group code value shall determine the length of the CDB (see table 11).

Table 9. OPERATION CODE byte

Bit	7	6	5	4	3	2	1	0
	GROUP CODE			COMMAND CODE				

The value in the GROUP CODE field specifies one of the groups shown in Table 10.

Table 10. Group Code values

Group Code	Meaning	Typical CDB Format
000b	6 byte commands	see Table 2
001b	10 byte commands	see Table 3
010b	10 byte commands	see Table 3
011b	Reserved [a]	
100b	16 byte commands	see Table 5 and Table 6
101b	12 byte commands	see Table 4
110b	Vendor Specific	
111b	Vendor Specific	
[a] The format of the commands using operation code 7Fh is described in 2.1.3. With the exception of operation code 7Fh, all group code 011b operation codes are reserved.		

2.2.2 SERVICE ACTION

All CDB formats except the 6-byte format provide for a SERVICE ACTION field containing a coded value identifying a function to be performed under the more general command function specified in the OPERATION CODE field. While the SERVICE ACTION field is defined for CDB formats, it is used as described in this subclause only in those CDB formats that contain a SERVICE ACTION field. When the specific field SERVICE ACTION is not defined in a CDB format, the bits identified as the SERVICE ACTION field in a CDB shall be used or reserved as specified by the particular CDB format.

2.2.3 Logical block address

The logical block addresses on a logical unit or within a volume or partition shall begin with block zero and be contiguous up to the last logical block of that logical unit or within that volume or partition.

A six-byte CDB may contain a 21-bit LOGICAL BLOCK ADDRESS field. The ten-byte and the twelve-byte CDBs may contain 32-bit LOGICAL BLOCK ADDRESS fields. The sixteen-byte CDB has two formats: one allows a 32-bit LOGICAL BLOCK ADDRESS field (see Table 5) and the other allows a 64-bit LOGICAL BLOCK ADDRESS field (see Table 6). LOGICAL BLOCK ADDRESS fields in additional parameter data have their length specified for each occurrence. See the specific command descriptions.

2.2.4 TRANSFER LENGTH

The TRANSFER LENGTH field specifies the amount of data to be transferred, usually the number of blocks. Some commands use transfer length to specify the requested number of bytes to be sent as defined in the command description.

Commands that use one byte for the TRANSFER LENGTH field may allow up to 256 blocks or 256 bytes of data to be transferred by one command.

In commands that use multiple bytes for the TRANSFER LENGTH field, a transfer length of zero specifies that no data transfer shall take place. A value of one or greater specifies the number of blocks or bytes that shall be transferred. Refer to the specific command description for further information.

2.2.5 PARAMETER LIST LENGTH

The PARAMETER LIST LENGTH field is used to specify the number of bytes sent from the Data-Out Buffer. This field is typically used in CDBs for parameters that are sent to a device server (e.g., mode parameters, diagnostic parameters, log parameters). A parameter list length of zero specifies that no data shall be transferred. This condition shall not be considered as an error, unless otherwise specified.

2.2.6 ALLOCATION LENGTH

The ALLOCATION LENGTH field specifies the maximum number of bytes that an application client has allocated in the Data-In Buffer. An allocation length of zero specifies that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the Data-In Buffer when the number of bytes specified by the ALLOCATION LENGTH field have been transferred or when all available data have been transferred, whichever is less. The allocation length is used to limit the maximum amount of variable length data (e.g., mode data, log data, diagnostic data) returned to an application client. If the information being transferred to the Data-In Buffer includes fields containing counts of the number of bytes in some or all of the data, then the contents of these fields shall not be altered to reflect the truncation, if any, that results from an insufficient ALLOCATION LENGTH value, unless this manual describes the Data-In Buffer format states otherwise.

If the amount of information to be transferred exceeds the maximum value that the ALLOCATION LENGTH field is capable of specifying, the device server shall transfer no data and terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

7.3 Status

A Status byte shall be sent from the target to the initiator during the STATUS phase at the termination of each command as specified in Tables 63 and 64 unless the command is cleared by one of the following conditions:

1. an Abort message
2. a Bus Device Reset message
3. a hard reset condition
4. an unexpected Bus Free condition (see Section 3.1.1)
5. an ABORT TASK message
6. a CLEAR TASK SET message

Table 63: Status byte

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved		STATUS BYTE CODE					Reserved

Table 64: Status byte code bit values

Status byte	Status represented	Task Ended
00h	Good	Yes
02h	Check Condition	Yes
04h	Condition Met/Good	Yes
08h	Busy	Yes
10h	Intermediate/Good	No
14h	Intermediate/Condition Met	No
18h	Reservation Conflict	Yes
22h	Obsolete [2]	Yes
28h	Queue Full/Task Set Full [1]	Yes
30h	ACA Active	Yes
40h	Task Aborted	Yes
All other codes	Reserved	

[1] What was formerly called a “Command Queue” is now called a “Task Set.”

[2] Formerly “Command Terminated.”

A description of the status byte codes is given below.

Good. This status indicates that the Device Server has successfully completed the task.

Check Condition. This status indicates that an auto contingent allegiance or contingent allegiance condition has occurred (see Section 7.6.1). Optionally, autosense data may be delivered (see Section 7.6.4.2).

Condition Met. This status shall be returned whenever the requested operation specified by an unlinked command is satisfied (see the PREFETCH commands in ANSI SCSI Block Commands-2, T10/1417D).

Busy. This status indicates that the logical unit is busy. This status shall be returned whenever a logical unit is unable to accept a command from an otherwise acceptable initiator (i.e., no reservation conflicts). The recommended initiator recovery action is to issue the command again at a later time.

Intermediate. This status or Intermediate-Condition Met shall be returned for each successfully completed command in a series of linked commands (except the last command), unless the command is terminated with Check Condition, Reservation Conflict, Task Set Full, Busy status. If Intermediate or Intermediate-Condition Met status is not returned, the series of linked commands is terminated and the task is ended.

Intermediate–Condition Met. This status is returned whenever the operation requested by a linked command is satisfied (see the PREFETCH commands in ANSI SCSI Block Commands-2, T10/1417D), unless the command is terminated with Check Condition, Reservation Conflict, Task Set Full, Busy status. If Intermediate or Intermediate-Condition Met status is not returned, the series of linked commands is terminated and the task is ended.

Reservation Conflict. This status shall be returned whenever a SCSI initiator port attempts to access a logical unit or an element of a logical unit that is reserved with a conflicting reservation type for another SCSI initiator. (See the RESERVE, RELEASE, PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands in ANSI SCSI Primary Commands - 4, T10/1731-D). The recommended initiator recovery action is to issue the command again at a later time. Removing a persistent reservation belonging to a failing initiator may require the execution of a PERSISTENT RESERVE OUT command with the Preempt or Preempt and Clear actions (see the SPC-4 standard, T10/1731-D).

Task Set Full. This status shall be implemented if the logical unit supports the creation of tagged tasks (see Section 7.7). This status shall not be implemented if the logical unit does not support the creation of tagged tasks.

When the logical unit has at least one task in the task set for a SCSI initiator port and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, Task Set Full shall be returned. When the logical unit has no task in the task set for a SCSI initiator port and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, Busy should be returned.

When the logical unit has at least one task in the task set and a lack of task set resources prevents entering a newly received untagged task in the task, Busy should be returned.

The logical unit should allow at least one queued command for each supported initiator that has identified itself to the target by a protocol specific procedure or by the successful transmission of a command.

ACA Active. This status shall be returned when an auto contingent allegiance exists within a task set and a SCSI initiator port issues a command for that task set when at least one of the following is true:

a. There is a task with the ACA attribute in the task set;
b. The initiator issuing the command did not cause the ACA condition; or
c. The task created to execute the command did not have the ACA attribute and the NACA bit was set to one in the CDB Control byte of the faulting command (see Section 7.6.1).

The initiator may reissue the command after the ACA condition has been cleared.

Task Aborted. This status shall be returned when a task is aborted by another SCSI Initiator and the Control mode page TAS bit is set to one.

7.3.1 Status precedence

If more than one condition applies to a completed task, the report of a Busy, Reservation Conflict, ACA Active or Task Set Full status shall take precedence over the return of any other status for that task.

Mensajes

- Además de comandos y códigos de estado, iniciador y diana pueden intercambiar mensajes
  - Permiten controlar la gestión del enlace
- Hay mensajes de 1 byte, 2 bytes o extendidos
- Se verán, conforme los necesitemos, consultando las tablas en la documentación
  - Su implementación y significado es específico y dependiente del tipo de mecanismo de transporte (bus) y estructuración en tramas usados.

1.3 Physical interface characteristics

The physical interface characteristics (cables, connectors, electrical descriptions, termination requirements, etc.) for the drives covered by this Interface Manual are found in each individual drive's Product Manual, since these features are not the same for all drives.

1.4 Summary of SCSI messages

Following is an alphabetical summary of the SCSI messages described in this manual. Details are given in Section 4.

Message Name	Hex Code	Page number
ABORT TASK	06	105
CLEAR QUEUE	0E	106
DISCONNECT	04	87
EXTENDED MESSAGE	01	84
IDENTIFY	80-FF	87
IGNORE WIDE RESIDUE	23	88
INITIATE RECOVERY	0F	Not described in this manual
INITIATOR DETECTED ERROR	05	89
MESSAGE PARITY ERROR	09	89
MESSAGE REJECT	07	89
MODIFY DATA POINTER	01, 05, 00	89 (extended message)
NO OPERATION	08	91
PARALLEL PROTOCOL REQUEST	01, 06, 04	92
QAS REQUEST	55	96
RELEASE RECOVERY	10	Not described in this manual
RESTORE POINTERS	03	96
SAVE DATA POINTERS	02	96
SYNCHRONOUS DATA TRANSFER REQUEST	01, 03, 01	97 (extended message)
TASK ATTRIBUTE MESSAGES		102
ACA (AUTO CONTINGENT ALLEGIANCE)	24	103
HEAD OF QUEUE TAG	21	104
LINKED COMMAND COMPLETE	0A	89
ORDERED QUEUE TAG	22	104
SIMPLE QUEUE TAG	20	104
TASK COMPLETE	00	99
TASK MANAGEMENT MESSAGES		105
ABORT TASK	00	105
ABORT TASK SET	06	106
CLEAR ACA	16	106
CLEAR TASK SET	0E	106
LOGICAL UNIT RESET	17	106
TARGET RESET	0C	106
WIDE DATA TRANSFER REQUEST	01, 03	100 (extended message)

Bus paralelo SCSI

- El bus paralelo SCSI (SPI) es un bus asíncrono, de 8 o 16 bits de ancho

- Aunque tiene modos de transferencia llamados *asíncrono* y *síncrono*, en realidad es siempre asíncrono

  - Diferencia entre modo asíncrono y síncrono es control de flujo REQ-ACK

    - Modo asíncrono: REQ requiere un ACK antes del siguiente REQ (Stop & Wait)

    - Modo síncrono: REQ no requiere ACK antes del siguiente REQ, sino que haya un número igual de REQs y ACKs (ventana deslizante)

Bus paralelo SCSI

- Mediante aumento en la frecuencia de la señalización REQ/ACK, y transfiriendo en ambos flancos del REQ, se consiguió aumentar la capacidad del bus desde 4 Mbytes/s a 320 Mbytes/s

- Bus paralelo SCSI ya está obsoleto, y ha sido substituido por el SAS (*Serial-Attached SCSI*) por la dificultad para superar fiablemente los 320 Mbytes/s computer.

- **SCSI-3**: This specification debuted in 1995 and included a series of smaller standards within its overall scope. A set of standards involving the **SCSI Parallel Interface** (SPI), which is the way that SCSI devices communicate with each other, has continued to evolve within SCSI-3. Most SCSI-3 specifications begin with the term **Ultra**, such as Ultra for SPI variations, Ultra2 for SPI-2 variations and Ultra3 for SPI-3 variations. The Fast and Wide designations work just like their SCSI-2 counterparts. SCSI-3 is the standard currently in use.

Different combinations of doubled bus speed, doubled clock speed and SCSI-3 specifications have led to lots of SCSI variations. The chart below compares several of them. Many of the slower ones are no longer in use -- we've included them for comparison.

Name	Specification	# of Devices	Bus Width	Bus Speed	MBps
Asynchronous SCSI	SCSI-1	8	8 bits	5 MHz	4 MBps
Synchronous SCSI	SCSI-1	8	8 bits	5 MHz	5 MBps
Wide	SCSI-2	16	16 bits	5 MHz	10 MBps
Fast	SCSI-2	8	8 bits	10 MHz	10 MBps
Fast/Wide	SCSI-2	16	16 bits	10 MHz	20 MBps
Ultra	SCSI-3 SPI	8	8 bits	20 MHz	20 MBps
Ultra/Wide	SCSI-3 SPI	8	16 bits	20 MHz	40 MBps
Ultra2	SCSI-3 SPI-2	8	8 bits	40 MHz	40 MBps
Ultra2/Wide	SCSI-3 SPI-2	16	16 bits	40 MHz	80 MBps
Ultra3	SCSI-3 SPI-3	16	16 bits	40 MHz	160 MBps
Ultra320	SCSI-3 SPI-4	16	16 bits	80 MHz	320 MBps

In addition to the increased bus speed, Ultra320 SCSI uses **packeted** data transfer, increasing its efficiency. Ultra2 was also the last type to have a "narrow," or 8-bit, bus width.

All of these SCSI types are **parallel** -- bits of data move through the bus simultaneously rather than one at a time. The newest type of SCSI, called **Serial Attached SCSI (SAS)**, uses SCSI commands but transmits data serially. SAS uses a point-to-point serial connection to move data at 3.0 gigabits per second, and each SAS port can support up to 128 devices or expanders.

Bus paralelo SCSI

- Además de las 8 o 16 líneas de datos, el bus paralelo SCSI usa diversas líneas de control para gestionar las fases del bus asociadas a la ejecución de los comandos

2.1 SCSI bus signals overview

Information transfer on the SCSI bus is allowed between only two SCSI devices at any given time except during MESSAGE IN PHASE when QAS is enabled. All SCSI devices that have QAS enabled are required to monitor messages during a MESSAGE IN PHASE for a QAS REQUEST MESSAGE. The maximum number of SCSI devices is determined by the width of the data path implemented. The SCSI devices may be any combination of SCSI initiator ports (commonly called "initiators") and SCSI target ports (commonly called "targets"), provided there is at least one of each.

Each SCSI device has a SCSI address and a corresponding SCSI ID bit assigned to it. When two SCSI devices communicate on the SCSI bus, one acts as the initiator and the other acts as the target. The initiator originates an I/O process and the target receives the I/O process.

Some drive models have a single 80-pin I/O connector that contains additional interface lines that carry drive configuration select signals. These are peculiar to certain drives and are not SCSI standard signals. These are described in the individual drive's product manual, Volume 1.

The 28 SCSI standard signals are described as follows:

BSY (Busy)—An "OR-tied" signal to indicate the bus is being used.

SEL (Select)—An "OR-tied" signal used by a SCSI initiator port to select a SCSI target port, or by a SCSI target port to reselect a SCSI initiator port.

RST (Reset)—An "OR-tied" signal that indicates the bus reset condition (see Section 5.2).

C/D (Control/Data)—A signal sourced by a SCSI target port that indicates whether CONTROL or DATA PHASE information is on the data bus. Assertion indicates Control (i.e., COMMAND, STATUS, and MESSAGE phases).

I/O (Input/Output)—A signal sourced by a SCSI target port to control the direction of data movement on the Data Bus with respect to a SCSI initiator port. Assertion indicates input to the initiator. This signal also distinguishes between SELECTION and RESELECTION phases.

MSG (Message)—A signal sourced by a SCSI target port to indicate the MESSAGE phase or a DT DATA phase depending on whether C/D is true or false. Asserted indicates MESSAGE or DT DATA.

REQ (Request)—A signal sourced by a SCSI target port to indicate a request for an information transfer on the SCSI bus.

ACK (Acknowledge)—A signal sourced by a SCSI initiator port to respond with an acknowledgment of an information transfer on the SCSI bus.

ATN (Attention)—A signal sourced by a SCSI initiator port to indicate the Attention condition.

DIFFSENS (Differential Sense)/Multimode—SE or LVD alternative—"LW" and "LC" models have I/O circuits that can operate either in single-ended (SE) or low voltage differential (LVD) mode. When the interface DIFFSENS line is between -0.35 V and +0.5 V, the drive interface circuits operate single-ended. When DIFFSENS is between +0.7 V and +1.9 V, the drive interface circuits operate low voltage differential. This arrangement is not intended to allow dynamically changing transmission modes, but rather to prevent incompatible devices from attempting to interoperate. Drives must operate only in the mode for which the installation and interface cabling is designed. Multimode I/O circuits used by "LW" and "LC" devices do not operate at high voltage differential levels and should never be exposed to high voltage differential environments unless the command mode voltages in the environment are controlled to safe levels for single-ended and low voltage differential devices (see the ANSI SPI-5 specification). High Voltage Differential (HVD) is now an obsolete ANSI standard.

P_CRCA (Parity/CRC Available)—A signal identifying either parity or CRC available based on bus phase and negotiated settings.

During the SELECTION PHASE, RESELECTION PHASE, ST DATA PHASE, COMMAND PHASE, MESSAGE PHASE, and STATUS PHASE, this signal is referred to as DB(P_CRCA) and is sourced by the SCSI device port driving the Data Bus. The DB(P_CRCA) signal is associated with the DB(7-0) signals and is used to detect the presence of an odd number of bit errors within the byte. The DB(P_CRCA) bit is driven such that the number of logical ones in the byte plus the parity bit is odd.

Data group transfers are enabled (see Section 4.3.12) when this signal is referred to as P_CRCA and is sourced by the target to control whether a data group field is a pad field, pCRC field, or data field (see Section 2.11.1). When asserted, the data group field shall be pad or pCRC fields that shall not be transferred to the application client. When negated, the data group field shall be a data field that shall be transferred to the application client.

During DT DATA phases when information unit transfers are enabled, this signal is referred to as P_CRCA and sourced by the SCSI target. Depending on the negotiated condition of read streaming and write flow control, the SCSI initiator and target usage for P_CRCA is different. When information unit transfers are enabled, the SCSI target and initiator shall use the P_CRCA signal as indicated in Table 2.

Table 2: P_CRC signal usage requirements

Write flow control	Read streaming	DT Data phase	SCSI initiator response to P_CRCA	SCSI target usage of P_CRCA
Disabled	Disabled	All	Ignore	Continuously negated.
Enabled	Disabled	DT DATA IN	Ignore	Continuously negated.
		DT DATA OUT	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current write stream.
Disabled	Enabled	DT DATA IN	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
		DT DATA OUT	Ignore	Continuously negated.
Enabled	Enabled	DT DATA IN	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
		DT DATA OUT	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
A SCSI device is not required to use read streaming even if it is enabled A SCSI device is not required to use write flow control even if it is enabled				

P1 (Parity 1)—A signal normally sourced by the SCSI device driving the Data Bus. The P1 signal is associated with the DB(15–8) signals and is used to detect the presence of an odd number of bit errors within the byte The P1 bit is driven such that the number of logical ones in the byte plus the P1 bit is odd.

During the ST DATA PHASE with transfer length set for 8-bit transfers, COMMAND PHASE, MESSAGE PHASE, and STATUS phase, the P1 signal shall not be driven by any SCSI device.

During the SELECTION phase and the RESELECTION phase on a 16-bit wide bus segment the P1 signal shall be sourced by the SCSI device driving the DATA BUS.

When data group transfers are enabled (see Section 4.3.12), the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

When information unit transfers are enabled, the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

During DT DATA phases when information unit transfers and paced transfers are enabled the P1 signal shall be sourced by the SCSI device driving the DATA BUS. The P1 signal is used to indicate the data valid or data invalid state during paced transfers.

DB(7–0) (8-bit data bus)—Each data bit that forms the 8-bit data bus. Bit significance and priority during arbitration are shown in Table 1.

DB(15–0) (16-bit data bus)—16 data bit signals that form the 16-bit Data Bus. Bit significance and priority during arbitration are shown in Table 1.

Greater detail on each of the SCSI bus signals is found in the following sections.

2.1.1 Drive select

For SCSI ID selection, install drive select jumpers as shown in configuration selection figure in the individual drive's Product Manual. Refer to the "Physical interface" section of the individual drive's Product Manual for the location of the drive select header. Drives using the 8-bit data interface can have one of eight ID bits selected by installing 0 through 2 (3) jumpers in a binary coded configuration on the drive select header. Drives using the 16-bit data interface can have one of 16 ID bits selected by installing 0 through 3 (4) jumpers in a binary coded configuration on the drive select header. "LC" model drives (80-pin direct connect I/O connector) can be assigned their bus ID over the SCSI interface.

2.1.2 Signal values

Signals may assume true or false values. There are two methods of driving these signals. In both cases, the signal shall be actively driven true, or asserted. In the case of OR-tied drivers, the driver does not drive the signal to the false state, rather the bias circuitry of the bus terminators pulls the signal false whenever it is released by the drivers at every SCSI device. If any driver is asserted, then the signal is true. In the case of non-OR-tied drivers, the signal may be negated. Negated means that the signal may be actively driven false, or may be simply released (in which case the bias circuitry pulls it false), at the option of the implementor.

2.2 Signal states

2.2.1 SE signals

Signals may be in a true (asserted) or false (negated) state. Signals that are asserted are actively driven to the true state. Signals that are negated may either be actively driven to the false state or released to the false state. A signal that is released goes to the false state because the bias of the terminator pulls the signal false. OR-tied signals shall not be actively driven false.

Note. The advantage of actively negating signals false during information transfer is that the noise margin is higher than if the signal is simply released. This facilitates reliable data transfer at high transfer rates.

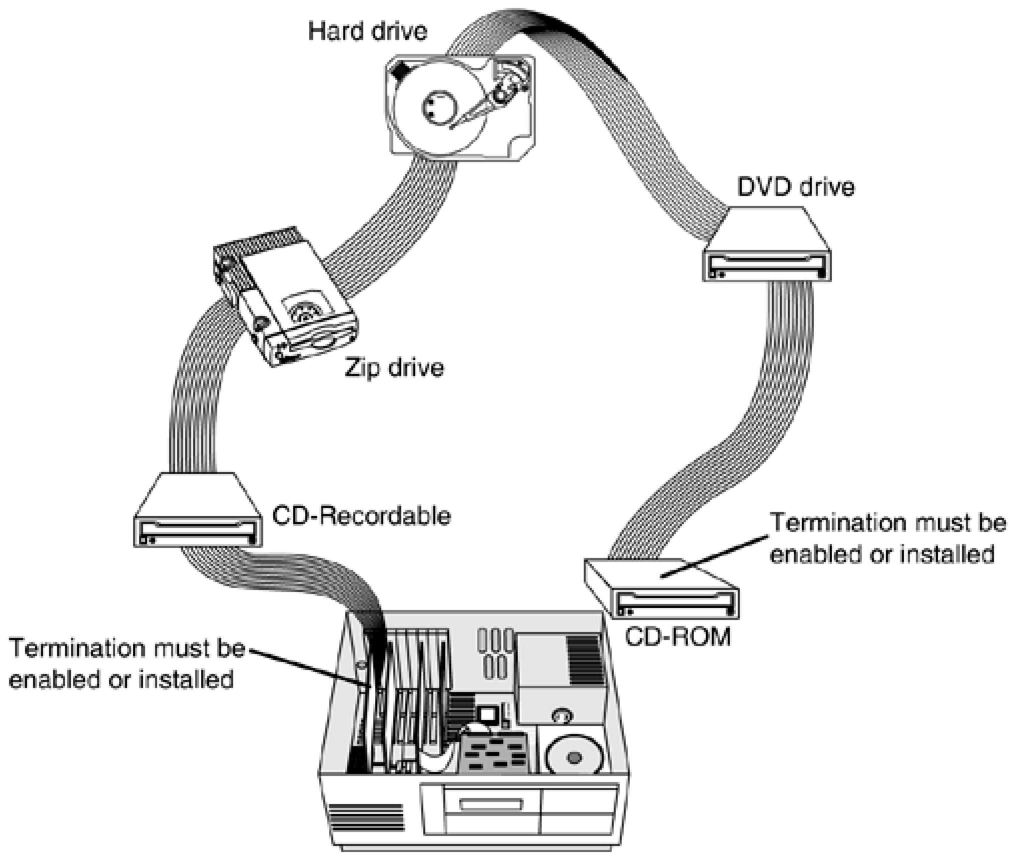
Bits of the data bus are defined as one when the signal is true and defined as zero when the signal is false.

2.2.2 LVD signals

Figure 3 defines the voltage and current definitions. A signal that is released goes to the false state because the bias of the terminator pulls the signal false.

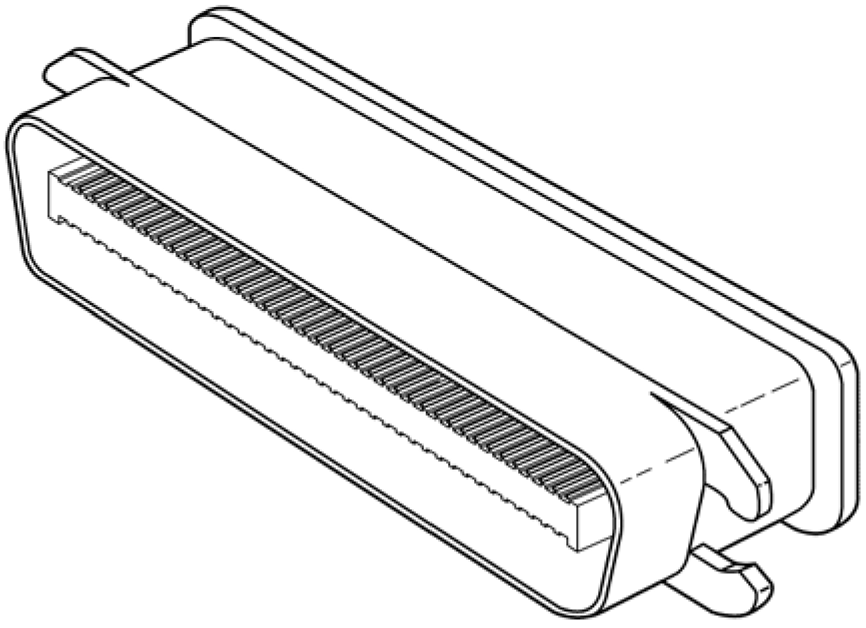
Bus paralelo SCSI

- El bus SCSI debe estar terminado en ambos extremos
  - Los terminadores pueden ser externos, o estar incorporados internamente en los propios dispositivos
  - Los terminadores pueden ser activos o pasivos
    - Para cualquier bus paralelo SCSI moderno, DEBE usarse terminación activa

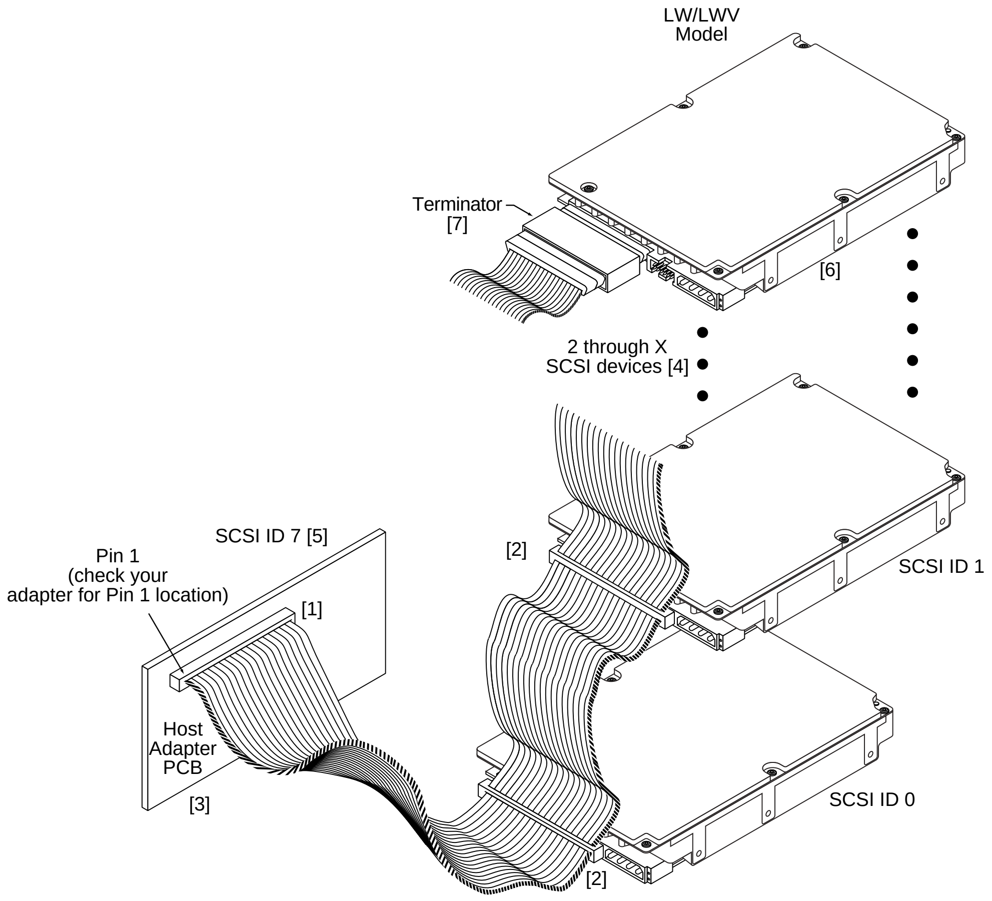

Hard drive
DVD drive
Zip drive
CD-Recordable
Termination must be enabled or installed
CD-ROM
Termination must be enabled or installed

When installing an external SCSI device, you usually find the device in a storage enclosure with both input and output SCSI connectors, so you can use the device in a daisy-chain. If the enclosure is at the end of the SCSI bus, an external terminator module most likely will have to be plugged into the second (outgoing) SCSI port to provide proper termination at that end of the bus (see Figure 8.17).

Figure 8.17. External SCSI device terminator.

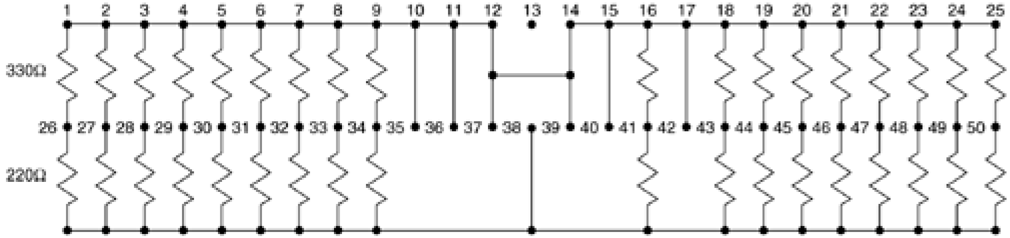


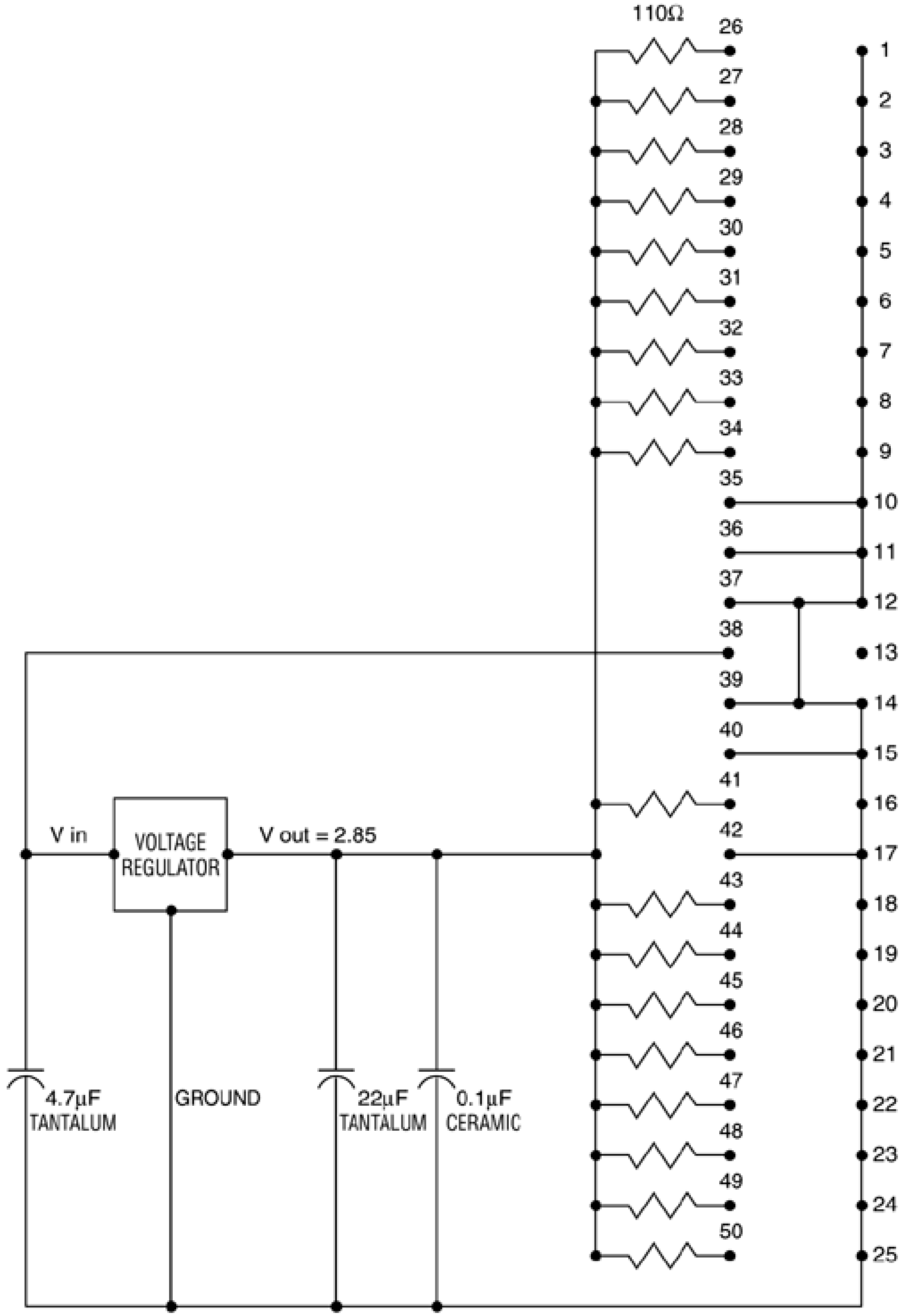
External terminator modules are available in a variety of connector configurations, including pass-through designs, which are necessary if only one port is available. Pass-through terminators also are commonly used in internal installations in which the device does not



[1] Closed end type 68-pin connector used. Terminators enabled.
[2] Open end type (in-line application) connector used.
[3] Host need not be on the end of the bus. Another device can be on the end with the terminator, the host having no terminator.
[4] Total interface cable length must not exceed that specified in ANSI document T10/1302D (including host adapter/initiator). The cable length restriction limits the total number of devices allowed.
[5] SCSI ID7 has highest arbitration priority, then ID15 to ID8 (ID 8 has the very lowest priority).
[6] Last drive on the bus.
[7] Open-end type 68-pin connector used. If end device, use external terminator and closed-end type 68-pin connector.

Figure 24. SCSI daisy chain interface cabling for LW drives


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
330Ω
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
220Ω


110Ω
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
V in
VOLTAGE REGULATOR
V out = 2.85
4.7µF TANTALUM
GROUND
22µF TANTALUM
0.1µF CERAMIC

Bus paralelo SCSI

- Bus SCSI tiene tres modos de señalización:
  - SE = Single Ended = señalización por nivel
  - LVD = Low-Voltage Differential = señalización diferencial
  - HVD = High-Voltage Differential = señalización diferencial de alto voltaje
- La señalización diferencial es más tolerante al ruido que la single-ended
- El tipo de señalización tiene implicaciones muy fuertes respecto a las velocidades de transferencia y distancias alcanzables

4 General

4.1 Overview

The SCSI Parallel Interface-2 Standard defines the cables, connectors, signals, transceivers, and protocol used to interconnect SCSI devices and the services provided to the application client.

4.1.1 Data transfer modes

SCSI parallel interface devices default to 8-bit asynchronous transfer. The 8-bit asynchronous information transfer mode is always used for all information transfers except DATA IN phases and DATA OUT phases. DATA IN phases and DATA OUT phases may use asynchronous or synchronous transfers that may be 8-bits, 16-bits or 32-bits wide, if a synchronous transfer agreement or a wide transfer agreement is in effect.

4.1.2 Cables, Connectors, Signals, Transceivers

SCSI parallel interface devices may be implemented with either 50, 68, or 80 pin connectors.

Table 1 defines the bus modes and transfer rates supported with the various transceivers defined within this standard.

Table 1 - Transceiver/speed support map

	Maximum transfer rate				
Transceiver	**Asynchronous**	**Fast-5**	**Fast-10**	**Fast-20**	**Fast-40**
SE	yes	yes	yes	yes	no
MSE (Note)	yes	yes	yes	yes	no
LVD	yes	yes	yes	yes	yes
HVD	yes	yes	yes	yes	yes
Key: yes = Transceiver/speed combination supported by this standard. No =Transceiver/speed combination not supported by this standard.					
Note-MSE is identical to SE except for the requirements in 7.3 and table 22.					

SCSI devices may connect to the bus via 8-bit, 16-bit, or 32-bit ports. The 8-bit and 16-bit ports shall connect to a primary bus that consists of an A cable or P cable via a single connector. The 32-bit port shall connect to a primary bus that consists of a P cable and a secondary bus that consists of a Q cable via two connectors; one to a P cable and the other to a Q cable. (see 5)

4.1.3 Physical architecture of bus

The position of the drivers, receivers, and terminators for a SE bus are shown in figure 3 and for a differential bus are shown in figure 2. The electrical properties of the drivers and receivers are all measured at the stub connections. Unless otherwise noted, all voltages are with respect to the signal ground of the SCSI device.

4 General

4.1 General overview

This standard defines the cables, connectors, signals, transceivers, terminators, and protocol used to interconnect parallel SCSI device ports and the services provided to the application client.

4.2 Cables, connectors, signals, transceivers

SCSI parallel interface devices may be implemented with either 50, 68, or 80 pin connectors.

Table 1 defines the bus segment modes and transfer rates supported with the various transceivers defined within this standard.

Table 1 - LVD transceiver/speed support map

Data Latching (ST/DT)	Maximum transfer rate							
	Async.	**Fast-5**	**Fast-10**	**Fast-20**	**Fast-40**	**Fast-80**	**Fast-160**	**Fast-320**
ST	yes	yes	yes	yes	yes	no	no	no
DT	no	no	yes	yes	yes	yes	paced	paced

Key:
yes = Transceiver/speed combination supported by this standard.
no =Transceiver/speed combination not supported by this standard.
paced =Transceiver/speed combination using paced transfers is supported by this standard (see 4.10).

SCSI devices may connect to the bus segment via 8-bit or 16-bit ports. The 8-bit ports shall connect to a bus segment with an A cable or equivalent (see clause 5). The 16-bit ports shall connect to a bus segment with a P cable or equivalent (see clause 5).

4.3 Physical architecture of bus segment

The position of the drivers, receivers, and terminators for a differential bus segment are shown in figure 2. The electrical properties of the drivers and receivers are all measured at the stub connections (see figure 3). Unless otherwise noted, all voltages are with respect to the signal ground of the SCSI device.

9.7.1.1 General cable characteristics

To minimize discontinuities and signal reflections, cables of different impedances should not be used in the same bus. Implementations may require trade-offs in shielding effectiveness, cable length, the number of loads, transfer rates, and cost to achieve satisfactory system operation. If shielded and unshielded cables are mixed within the same SCSI bus, the effect of impedance mismatch must be carefully considered. Proper impedance matching is especially important in order to maintain adequate margin at fast SCSI transfer rates.

9.7.1.2 Single-ended drivers/receivers

The maximum total cable length allowed with drives using single-ended I/O driver and receiver circuits depends on several factors. Table 18 lists the maximum lengths allowed for different configurations of drive usage. These values are from the SPI documents. All device I/O lines must have equal to or less than 25 pf capacitance to ground, measured at the beginning of the stub.

Table 18: Cable characteristics for single-ended circuits

I/O transfer rate	Maximum number of devices on the bus	maximum length between SCSI S.E. terminators	Transmission line impedance	
			REQ/ACK	Other signals
≤10M transfers/s (Fast 10)	16 (wide SCSI bus)	3 meters (9.8 ft)	90 ± 6 Ohms	90 ± 10 Ohms
≤20M transfers/s (Fast 20)	4 (wide SCSI bus)	3 meters (9.8 ft)	90 ± 6 Ohms	90 ± 10 Ohms
≤20M transfers/s (Fast 20)	8 (wide SCSI bus)	1.5 meters (4.9 ft)	90 ± 6 Ohms	90 ± 10 Ohms

A stub length of no more than 0.1 meter (0.33 ft) is allowed off the mainline interconnection with any connected equipment. The stub length is measured from the transceiver to the connection to the mainline SCSI bus.

Single-ended I/O cable pin assignments for LW drives are shown in Table 15.

Single-ended I/O pin assignments for LC models are shown in Table 16. The LC model does not require an I/O cable. It is designed to connect directly to a back panel connector.

9.7.1.3 Cables for low voltage differential drivers/receivers

The SPI-3 and SPI-4 specification for differential impedance for LVD cables is 122.5 ± 12.5 ohms.

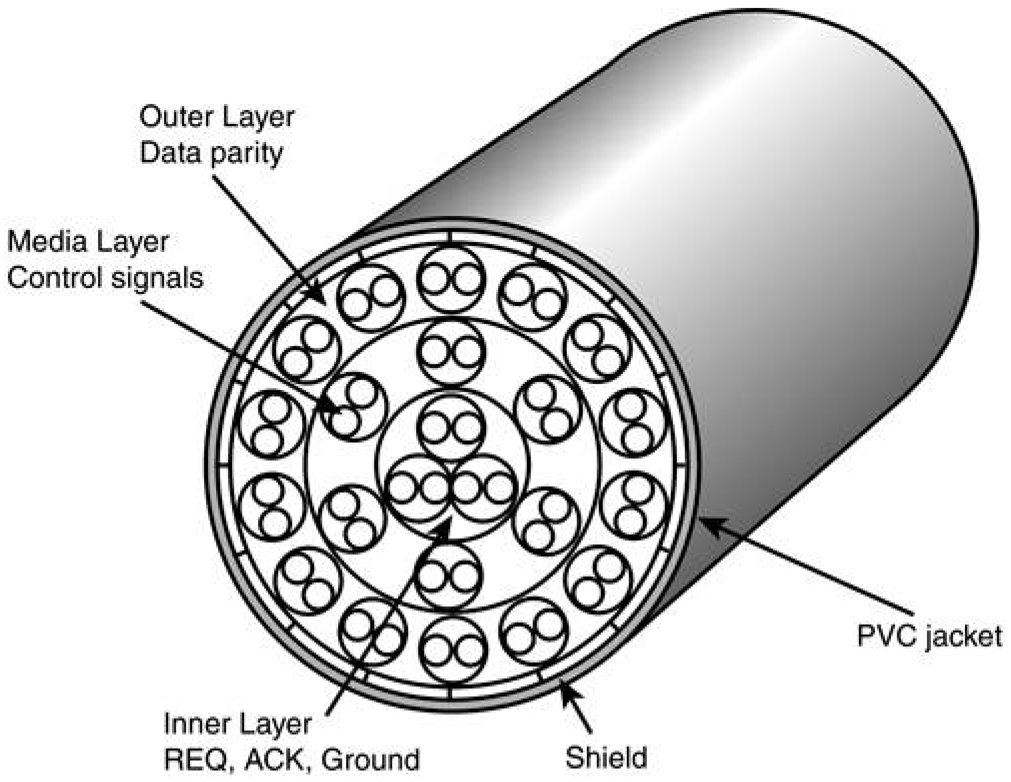
The maximum SCSI bus length between the terminators for a SCSI LVD point-to-point system (one initiator and one target) is 25 meters (82 feet).

The maximum SCSI bus length between the terminators for a SCSI LVD multidrop system (up to 16 total SCSI devices) is 12 meters (39.37 feet). However, implementation of this 12-meter total length is dependent on the configuration of the system and the choice of cable. See Tables 19 and 20 for examples of recommended SCSI LVD cable types and lengths.

It should be noted from the data in Tables 19 and 20 that shielded round twisted-pair cables can be used to implement a 12-meter system, while Twist-n-flat cables cannot be used to implement a 12-meter system due to higher levels of attenuation and crosstalk. In addition, twist-n-flat cables with shorter sections of twist allow greater crosstalk and their lengths must be reduced as shown in Table 20.

Table 19: Example shielded round twisted-pair cables - into a multi-drive backplane

Cable description	Differential impedance, nominal	Single-ended impedance, nominal	Single-ended capacitance, maximum	Time delay, nominal	Conductor DC resistance, nominal	Maximum shielded round twisted-pair cable length for U160/U320
30 AWG solid	130 ohms	90 ohms	17 pF/ft	1.36 ns/ft	0.113 ohms/ft	12 meters minus the SCSI trace length of the backplane
28 AWG stranded	123 ohms	80 ohms	19 pF/ft	1.54 ns/ft	0.067 ohms/ft	12 meters minus the SCSI trace length of the backplane
28 AWG stranded	132 ohms	85 ohms	20 pF/ft	1.50 ns/ft	0.065 ohms/ft	12 meters minus the SCSI trace length of the backplane



This specialized construction is what makes SCSI cables so expensive, as well as thicker than other types of cables. Note this specialized construction is necessary only for external SCSI cables. Cables used to connect devices inside a shielded enclosure (such as inside a PC) can use much less expensive ribbon cables.

The A cables can have pin-headerñtype (internal) connectors or external shielded connectors, each with a different pinout. The P cables feature the same connector pinout on either internal or external cable connections.

Single-Ended SCSI Cables and Connectors

The single-ended electrical interface is the most popular type for PC systems. Tables 8.3 and 8.4 show all the possible single-ended cable and connector pinouts. The A cable is available in both internal unshielded and external shielded configurations. A hyphen preceding a signal name indicates the signal is Active Low. The RESERVED lines have continuity from one end of the SCSI bus to the other. In an A cable bus, the RESERVED lines should be left open in SCSI devices (but may be connected to ground) and are connected to ground in the bus terminator assemblies. In the P and Q cables, the RESERVED lines are left open in SCSI devices and the bus terminator assemblies.

Table 8.3. A-Cable (Single-Ended) Internal Unshielded Header Connector

Signal	Pin	Pin	Signal
GROUND	1	2	-DB(0)
GROUND	3	4	-DB(1)
GROUND	5	6	-DB(2)
GROUND	7	8	-DB(3)
GROUND	9	10	-DB(4)

Table 20: Example Twist-n-flat cables - into a multi-drive backplane

Cable description	Differential impedance, nominal	Single-ended impedance, nominal	Single-ended capacitance, maximum	Time delay, nominal	Conductor DC resistance, nominal	Maximum twist-n-flat cable length	
						U320 transfer rate	U160 transfer rate
TPE, 22.25" twist, 1.75" flat, (24" flat to flat), 30 AWG solid tinned copper	131 ohms	93 ohms	15.3 pF/ft	1.45 ns/ft	0.105 ohms/ft	3.05 meters (10.0 ft)	6.1 meters (20.0 ft)
TPE, 8.1" twist, 1.75" flat, (9.85" flat to flat), 30 AWG solid tinned copper	131 ohms	93 ohms	15.3 pF/ft	1.45 ns/ft	0.105 ohms/ft	2.45 meters (8.33 ft)	4.9 meters (16.66 ft)
TPE, 4.25" twist, 1.75" flat, (6" flat to flat), 30 AWG solid tinned copper	131 ohms	93 ohms	15.3 pF/ft	1.45 ns/ft	0.105 ohms/ft	1.52 meters (5.0 ft)	3.04 meters (10.0 ft)

Annex I

(informative)

SCSI ICONS

These icons are provided as symbols to identify an SCSI port and to indicate whether the port is using:

a) single-ended transceivers (figure I.1),
b) LVD transceivers (figure I.2), or
c) SE/LVD multimode transceivers (figure I.3), or
d) HVD transceivers (figure I.4).

The icons illustrated in figure I.1, figure I.4, figure I.3, and figure I.4 may be enlarged or reduced as needed for the application. The text and graphic may be used together or separately. The text font and size may also be adjusted as required.

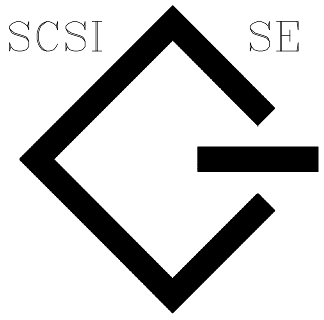


Figure I.1 - SE icon for SCSI

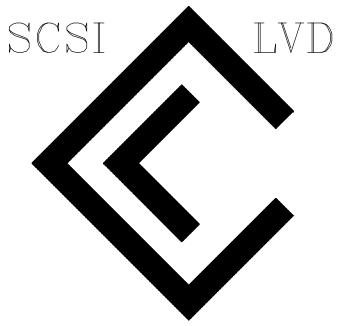


Figure I.2 - LVD icon for SCSI

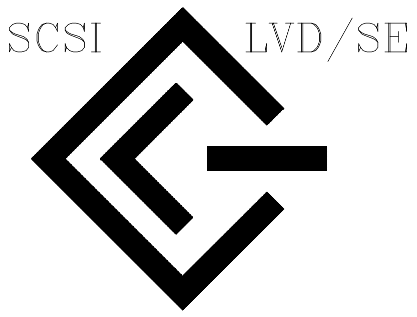


Figure I.3 - SE/LVD multimode icon for SCSI

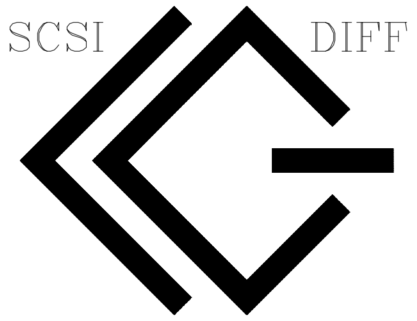


Figure I.4 - HVD icon for SCSI

Table 15: LW 68-conductor LVD P cable signal/pin assignments [11]

Note. Note: A minus sign preceding a signal name indicates that signal is active low.

Signal name [1]	Connector contact number [3]	Cable conductor number [2]		Connector contact number [3]	Signal name [1]
+DB12	1	1	2	35	–DB12
+DB13	2	3	4	36	–DB13
+DB14	3	5	6	37	–DB14
+DB15	4	7	8	38	–DB15
+DBP1	5	9	10	39	–DBP1
+DB0	6	11	12	40	–DB0
+DB1	7	13	14	41	–DB1
+DB2	8	15	16	42	–DB2
+DB3	9	17	18	43	–DB3
+DB4	10	19	20	44	–DB4
+DB5	11	21	22	45	–DB5
+DB6	12	23	24	46	–DB6
+DB7	13	25	26	47	–DB7
+DBP	14	27	28	48	–DBP
Ground	15	29	30	49	Ground
DIFFSNS [8]	16	31	32	50	Ground
TermPwr	17	33	34	51	TermPwr
TermPwr	18	35	36	52	TermPwr
Reserved	19	37	38	53	Reserved
Ground	20	39	40	54	Ground
+ATN	21	41	42	55	–ATN
Ground	22	43	44	56	Ground
+BSY	23	45	46	57	–BSY
+ACK	24	47	48	58	–ACK
+RST	25	49	50	59	–RST
+MSG	26	51	52	60	–MSG
+SEL	27	53	54	61	–SEL
+C/D	28	55	56	62	–C/D
+REQ	29	57	58	63	–REQ
+I/O	30	59	60	64	–I/O
+DB8	31	61	62	65	–DB8
+DB9	32	63	64	66	–DB9
+DB10	33	65	66	67	–DB10
+DB11	34	67	68	68	–DB11

Notes [ ]: See page following Table 17.

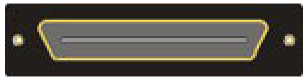
Products

New Products
Adapters
Apple/Mac Compatible
Audio/Video
Bulk Cable
Cases/Case Mods
CATV/SATV - New!
Closeout
Connectors
Firewire
Gender Changers
I/O Cards and Devices
Ink Jet Cartridges
KVMs/Switchboxes
Network Cables
Network Electronics
PC Cables
PDA/Cell Phone Cables
Power/Batteries
Premise Plus
SCSI Cables
Tools/Test
USB
Workstation

Log In/Register
What's New
Custom Cables
Where to Buy
Tech Support
About Us
Search

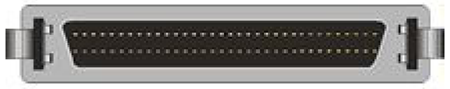
SCSI Connector Guide

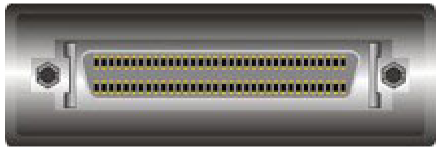
VHDCI .8mm 68-pin


VHDCI .8mm 68-pin Male

Used for scsi-3 applications: RAID. The VHDCI .8mm 68-pin connector has 68-pins arranged in two rows one on top of the other. The top row has 34 pins and the lower row has 34 pins. Also has been called scsi-5.

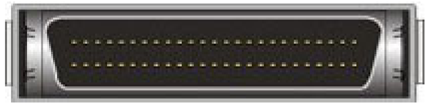
Micro DB68


Micro DB68 Male


Micro DB68 Female

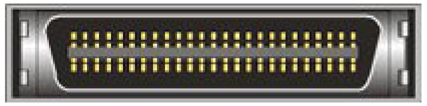
Used for scsi-3 applications: scanner, removable storage drive, controller, external cdr/cdrw, ultra/2. The Micro DB68 connector has 68-pins arranged in two rows one on top of the other. The top row has 34 pins and the lower row has 34 pins. Also known as HD(High-Density)68 & HP(Half-Pitch) DB68.

Micro DB50


Micro DB50 Male

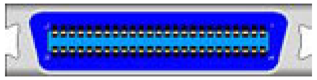
Used for scsi-2 applications: scanner, removable storage drive, controller, external cdr/cdrw. The Micro DB50 connector has 50-pins arranged in two rows one on top of the other. The top row has 25 pins and the lower row has 25 pins. Also known as HD(High-Density)50 & HP(Half-Pitch) DB50.

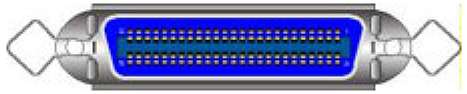
Micro Centronics 50


Micro Centronics 50 Male

Used for scsi applications: proprietary scsi-2 interface(rare). The Micro Centronics 50 connector has 50-pins arranged in two rows one on top of the other. The top row has 25 pins and the lower row has 25 pins.

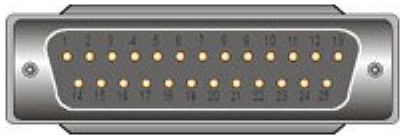
Centronics 50

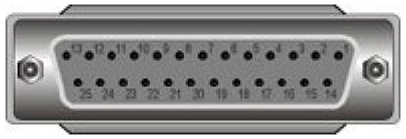

Centronics 50 Male


Centronics 50 Female

Used for scsi-1 applications: older scanners, controllers, external scsi device cases. The Centronics 50 connector has 50-pins arranged in two rows one on top of the other. The top row has 25 pins and the lower row has 25 pins.

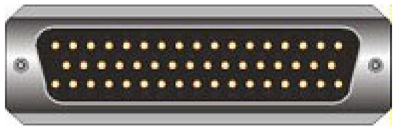
DB25


DB25 Male


DB25 Female

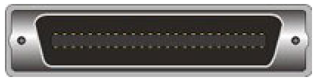
Used for parallel, serial or scsi applications: modem, null modem, laplink, printer, scanner, removable storage drive, Apple scsi. The DB25 connector has 25-pins arranged in two rows one on top of the other. The top row has 13 pins and the lower row has 12 pins.

DB50


DB50 Male

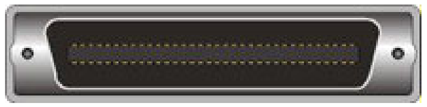
Used for early scsi applications: **older Sun Sparcstations.** The DB50 connector has 50-pins arranged in three rows one on top of the other. The top row has 17 pins, the middle row has 16 pins and the lower row has 17 pins.

Micro Centronics 60


Micro Centronics 60 Male

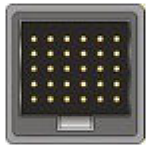
Used for scsi applications: IBM RS-6000. The Micro Centronics 60 connector has 60-pins arranged in two rows one on top of the other. The top row has 30 pins and the lower row has 30 pins.

Micro Centronics 68


Micro Centronics 68 Male

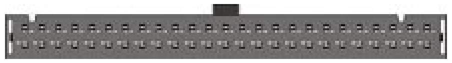
Used for scsi applications: IBM RS-6000. The Micro Centronics 68 connector has 68-pins arranged in two rows one on top of the other. The top row has 34 pins and the lower row has 34 pins.

HDI-30

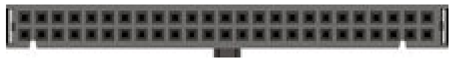

HDI-30 Male

Used for scsi applications: Apple PowerBook. The HDI-30 connector has 30-pins arranged in five rows one on top of the other.

Internal 50-pin SCSI



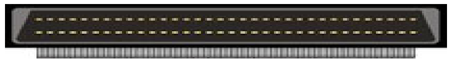
IDC50 Male



IDC50 Female

Used for internal scsi-1/scsi-2 applications: hard drive, cd-rom, removable storage drive.

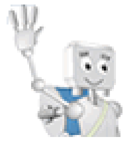
Internal 68-pin SCSI

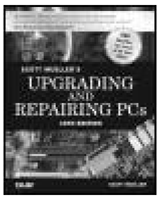


68-pin Male

Used for internal scsi-3/ultra2/lvd applications: hard drive, cd-rom, removable storage drive.

<< Return to Connector Guide





Upgrading and Repairing PCs, 15th Anniversary Edition
By Scott Mueller

Table of Contents

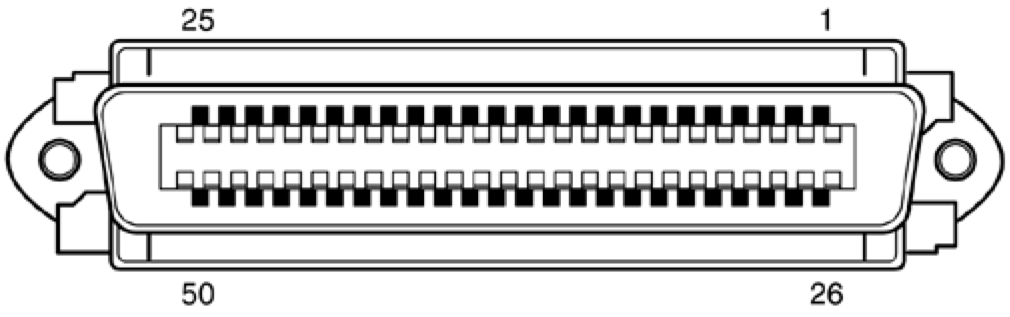
Chapter 8. The SCSI Interface

SCSI Cables and Connectors

The SCSI standards are very specific when it comes to cables and connectors. The most common connectors specified in this standard are the 50-position unshielded pin header connector for internal SCSI connections and the 50-position shielded Centronics latch-style connectors for external connections. The shielded Centronics-style connector also is called Alternative 2 in the official specification. Passive or Active termination (Active is preferred) is specified for both single-ended and differential buses. The 50-conductor bus configuration is defined in the SCSI-2 standard as the A cable.

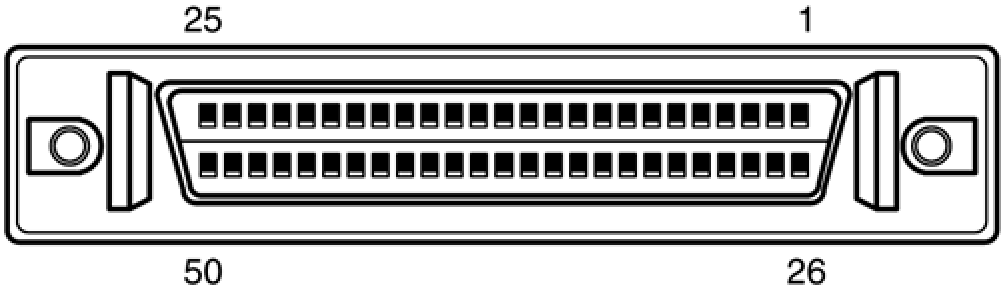
Older narrow (8-bit) SCSI adapters and external devices use a full-size Centronics-type connector. Figure 8.4 shows what the low-density, 50-pin SCSI connector looks like.

Figure 8.4. Low-density, 50-pin SCSI device connector.



The SCSI-2 revision added a high-density, 50-position, D-shell connector option for the A-cable connectors. This connector now is called Alternative 1. Figure 8.5 shows the 50-pin, high-density SCSI connector.

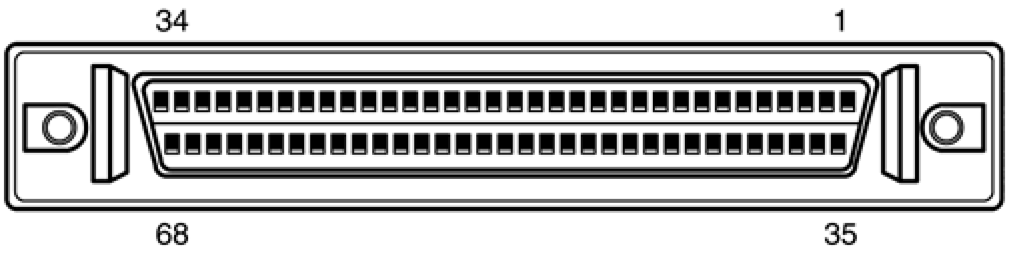
Figure 8.5. High-density, 50-pin SCSI device connector.



The Alternative 2 Centronics latch-style connector remains unchanged from SCSI-1. A 68-conductor B-cable specification was added to the SCSI-2 standard to provide for 16- and 32-bit data transfers; the connector, however, had to be used in parallel with an A cable. The industry did not widely accept the B-cable option, which has been dropped from the SCSI-3 standard.

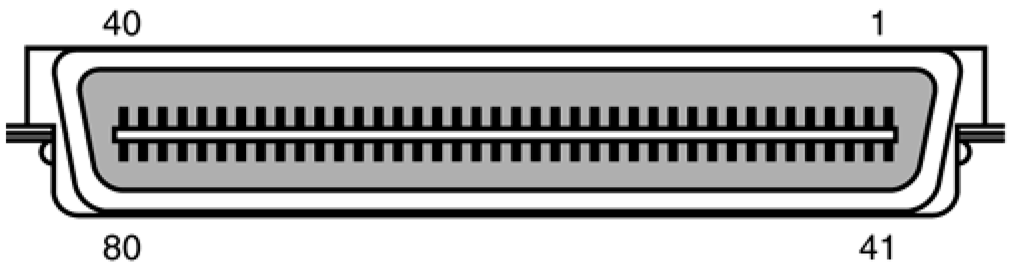
To replace the ill-fated B cable, a new 68-conductor P cable was developed as part of the SCSI-3 specification. Shielded and unshielded high-density D-shell connectors are specified for both the A and P cables. The shielded high-density connectors use a squeeze-to-release latch rather than the wire latch used on the Centronics-style connectors. Active termination for single-ended buses is specified, providing a high level of signal integrity. Figure 8.6 shows the 68-pin, high-density SCSI connector.

Figure 8.6. High-density, 68-pin SCSI device connector.



Drive arrays normally use special SCSI drives with what is called an 80-pin Alternative-4 connector, which is capable of Wide SCSI and also includes power signals. Drives with the 80-pin connector are usually *hot-swappable*—they can be removed and installed with the power on—in drive arrays. The 80-pin Alt-4 connector is shown in Figure 8.7.

Figure 8.7. 80-pin Alt-4 SCSI device connector.



Apple and some other nonstandard implementations from other vendors used a 25-pin cable and connector for SCSI devices. They did this by eliminating most of the grounds from the cable, which unfortunately results in a noisy, error-prone connection. These 25-pin connectors and cables are not compliant with any SCSI standard; you should avoid them if possible. The connector used in these cases was a standard female DB-25 connector, which looks exactly like a PC parallel port (printer) connector.

Fases lógicas del bus SCSI paralelo

- Para el envío y ejecución de un comando, el bus paralelo SCSI pasa por una serie de fases muy definidas y estructuradas.

3.0 Logical characteristics

The operations of the SCSI bus as described in this section are supported by the drive as specified in each individual drive's Product Manual. The drive always functions as the target unless otherwise stated.

3.1 SCSI bus phases overview

The drive responds to the following phases:

BUS FREE phase
ARBITRATION phase
SELECTION phase
RESELECTION phase

COMMAND phase Data (IN and OUT) STATUS (IN only) MESSAGE (IN and OUT)	$\left.\begin{matrix} \\ \\ \\ \end{matrix}\right\}$ These phases are collectively termed the Information transfer phases

The COMMAND, DATA, STATUS, and MESSAGE phases are collectively called the information transfer phases.

The SCSI bus can never be in more than one phase at a time. Signals that are not mentioned in a particular context shall not be asserted.

3.1.1 BUS FREE phase

The BUS FREE phase indicates that there is no current task and that the SCSI bus is available for a physical connection or physical reconnection. SCSI devices shall detect the BUS FREE phase after the SEL and BSY signals are both false for at least one bus settle delay.

SCSI devices shall release all SCSI bus signals within one bus clear delay after BSY and SEL are continuously negated (false) for one bus settle delay. If a SCSI device requires more than one bus settle delay to detect the BUS FREE phase, it shall release all SCSI bus signals within one bus clear delay minus the excess time to detect the BUS FREE phase. The total time to clear the SCSI bus shall not exceed one bus settle delay plus one bus clear delay.

During normal operation a SCSI target port enters the BUS FREE phase when it releases the BSY signal.

3.1.1.1 Unexpected and expected bus free phases

In some cases a SCSI target port (connected to a SCSI initiator port) unexpectedly reverts to the BUS FREE phase to indicate an error condition that it has no other way to handle. This is called an unexpected disconnect.

SCSI target ports shall create a BUS FREE phases after any of the following:

Fases lógicas del bus SCSI paralelo

- El interfaz SCSI soporta transacciones divididas
  - La diana compite también por el bus en la fase de arbitración
  - Se reconecta al iniciador, en la fase de reselección, para continuar la transferencia de datos asociada a un comando pendiente
  - Debe enviar al iniciador un mensaje con el tag del comando que se va a continuar
- Para completar un comando pueden ser necesarias varias fases de desconexión y reconexión

7.4 Command examples

7.4.1 Single command example

A typical operation on the SCSI bus is likely to include a single READ command to a peripheral device such as the drive. This operation is described in detail starting with a request from the initiator. This example assumes that no linked commands and no malfunctions or errors occur and is illustrated in Figure 19.

The initiator has active pointers and a set of stored pointers representing active disconnected SCSI devices (a SCSI initiator port without disconnect capability does not require stored pointers). The initiator sets up the active pointers for the operation requested, arbitrates for the SCSI bus, and selects the drive. Once this process is completed, the drive assumes control of the operation.

The drive obtains the command from the initiator (in this case a READ command). The drive interprets the command and executes it. For this command, the drive reads the requested data from the Disc Media and sends this data to the initiator. After sending the read data to the initiator, the drive sends a status byte to the initiator. To end the operation, the drive sends a Command Complete message to the initiator and then goes to the Bus Free state.

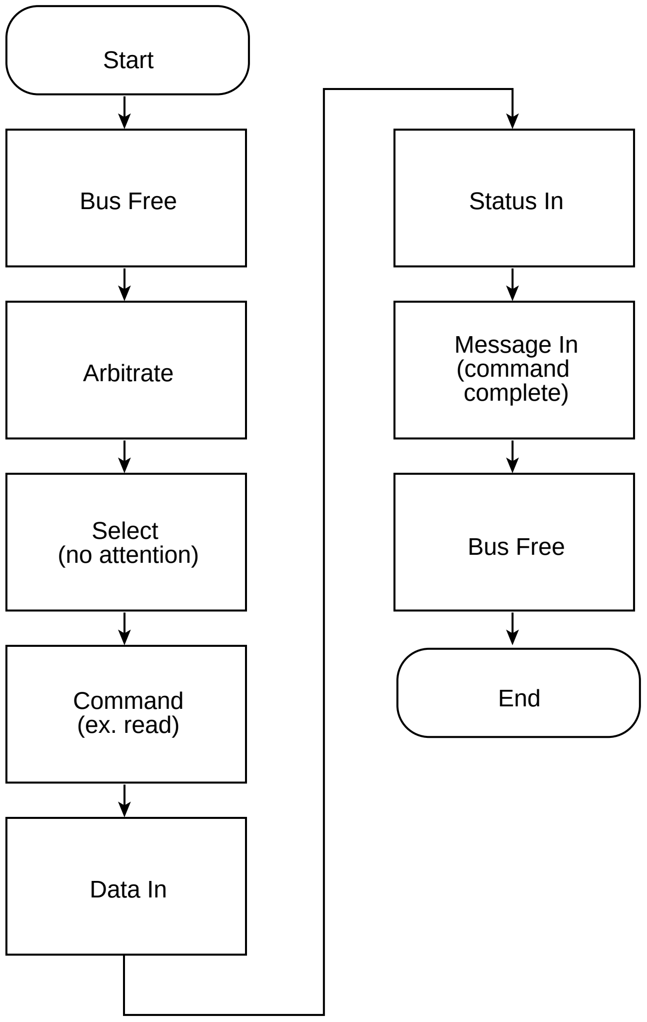


Figure 19. Single command example

7.4.2 Disconnect example

In the single command example, the length of time necessary to obtain the data may require a time consuming physical seek. In order to improve system throughput, the drive may disconnect from the initiator, freeing the SCSI bus to allow other requests to be sent to other SCSI devices. To do this, the initiator must be reselectable and capable of restoring the pointers upon reconnection. The drive must be capable of arbitrating for the SCSI bus and reselecting the initiator. See Figure 20.

After the drive has received the READ command (and has determined that there will be a delay), it disconnects by sending a DISCONNECT message and releasing BSY (goes to Bus Free state).

When the data is ready to be transferred, the drive reconnects to the initiator, the initiator restores the pointers to their most recently saved values (which in this case are the initial values), and the drive continues (as in the single command example) to finish the operation. The initiator recognizes that the operation is complete when a Command Complete message is received.

If the drive elects to disconnect after transferring part of the data (e.g., while crossing a cylinder boundary), it sends a Save Data Pointer message and a DISCONNECT message to the initiator and then disconnects. When reconnection is completed, the initiator restores the current data pointer to the value it was immediately before the Save Data Pointer message.

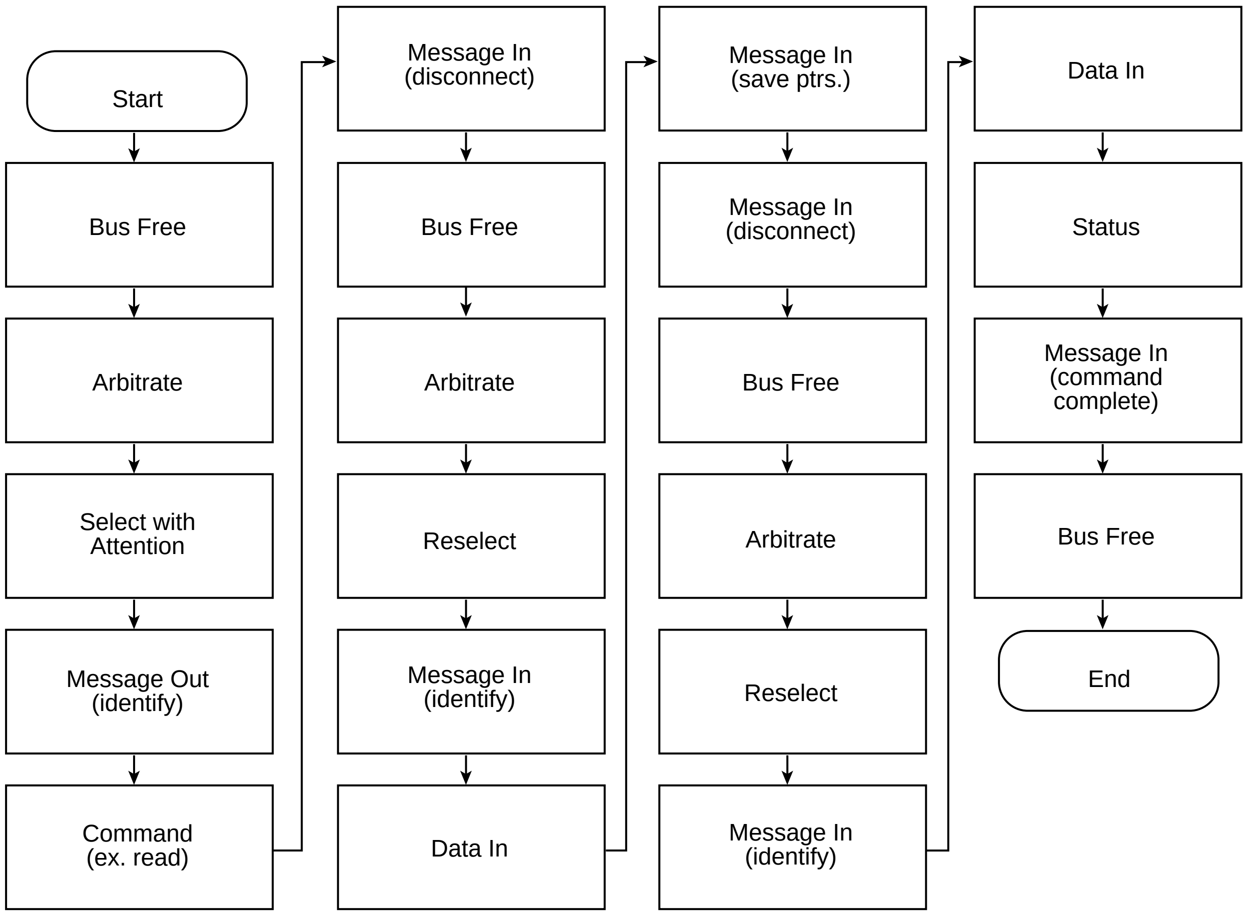


Figure 20. Disconnect example