What is

SCSI?

Fourth Edition

Table of Contents

INTRODUCTION

SCSI (pronounced "skuzzy") is an acronym for Small Computer System Interface. SCSI originated from the Selector Channel on IBM-360 computers, and in 1981 was scaled down by the Shugart Associates Company to make an universal, intelligent disk drive interface. It was called the Shugart Associates Systems Interface, or SASI, pronounced "sassy". The Small Computer Systems Interface, or SCSI, became an ANSI standard in 1986, after about four years of committee discussions.

The word "small" in SCSI is just a leftover from the not-so-distant past. SCSI is now being used in computer systems of all sizes, from small ones all the way up to big systems.

Before we start talking about SCSI, let's define first what a "bus" is, and what kind of bus SCSI is: Websters dictionary defines a bus as: "A large motor-driven passenger vehicle operating usually according to a schedule along a fixed route". In computer applications, a better definition is: "A conductor or an assembly of conductors used for collecting electric currents and distributing them to outgoing feeders".

This is a very broad definition and we need to clarify the characteristics of the SCSI bus in order to understand where and how SCSI fits into a computer system. We can classify buses by their "intelligence" (or lack thereof), mode of operation (serial or parallel), and most importantly we need to clarify the position of the SCSI bus in a computer system (backplane bus, peripheral bus, or network bus). Using these classifications, we can say that SCSI is a peripheral bus, that is intelligent, parallel, with medium-to-high performance. With regard to polarity, SCSI using Single-Ended interface is a "negative" bus (LOW level represents the logical TRUE).

During the course of the ANSI committee meetings, the standard was expanded to use devices other than magnetic disk drives. The standard now recognizes magnetic disk and tape drives, various types of optical disk drives, printers, scanners, processors, communications devices, medium changers, and other types. The SCSI standard has also changed to take advantage of newer hardware and more intelligent controllers. Caching is recognized. Intelligent command queuing is allowed for. There are provisions for intelligent self-testing by the peripheral. The data path has been widened from 8 bits up to a possible 32 bits. Transfer speeds have risen from a common 1 MHz to a maximum of 40 MHz. Work is underway on even higher speeds up to 80 MHz and more.

You may ask "Why would anyone use SCSI”

One of the possible answers lies in the ease with which new peripheral devices can be added to a computer system. Without SCSI, for each new peripheral you must teach the computer how to manipulate the hardware to accomplish the reading or writing of data to and from the device. In the past, by the time the hardware and software design of a computer was complete, a new generation of peripherals was often available which the computer could not use. The development cycle had to start again. As hardware capabilities grew at increasing rates, it took even longer to develop the computer-to-peripheral interfaces. As a result of all this, the peripherals attached to a computer system were a generation or more behind the computer itself.

Another good reason for using SCSI is the high performance requirements. That is why you will find SCSI used almost exclusively in medium and large systems.

With SCSI, the intelligence moved from the host to the peripheral device. The computer now can use a standard set of commands (about a dozen commands for most devices) to accomplish moving data back and forth to the peripheral. The development cycle for SCSI implementations is short. Thanks to the relative simplicity of the SCSI interface, it is possible to connect a new peripheral device to an existing computer with no hardware changes or additional hardware parts. With SCSI, the designer only needs to write a new I/O driver (which may be just a modification of an existing one) - a task which often can be done in two months or less.

SCSI is both a bus hardware specification, and a command set to optimize use of that bus. The core of the SCSI idea is to give the computer complete device independence. In other words, to the host, all magnetic disks look alike, except for their total capacity. All printers look alike, all CD-ROMs look alike, and so on. Within any one device type, the system should not need modifications when removing one manufacturer's device and replacing it with a different manufacturer's device.
By separating the computer from the specific details of the peripheral's hardware, it is much easier to add a new peripheral device to the computer's repertoire. SCSI peripherals are being used on most computer systems, ranging from MS-DOS based personal computers through UNIX based workstations and bigger systems.

This document is a brief overview designed to help the reader understand the basic functionality of SCSI, and its position in a computer system. For reasons of clarity it was decided to keep this document short and simple. Intentionally, we do not present a complete description of all the possible SCSI functions here. The reader is referred to the SCSI specification to obtain detailed definitions of all individual functions.

SCSI TERMS, BUZZWORDS, FUNCTIONS

If we want to understand functionality of the SCSI bus, we need to understand the terms and definitions. We also need to understand the language used to describe it. In the following, we will try to explain the SCSI functions and terminology.

INITIATOR - TARGET

There are two kinds of devices on the SCSI bus: the SCSI Initiators start the I/O process, and the Targets respond to a request to perform an I/O process. In other words, Initiators are devices which request that commands be carried out, and Targets are devices which carry out commands. The "master" and "slave" function moves back and forth between the two. The Initiator starts the arbitration and selects a Target. The Target takes over and requests a command from the Initiator. The Initiator responds by sending the command code (Command Descriptor Block - CDB) to be executed.

SCSI host adapters are Initiators, but at times, the host adapter may need to act as a Target for some commands. SCSI peripheral devices are Targets, but for some commands (e.g., a COPY command), the peripheral may need to act as an Initiator temporarily. The single-byte (NARROW) SCSI bus supports up to eight devices (the 16-bit WIDE bus supports 16 devices), in any mix of Initiators and Targets, with the limitation that at least one Initiator and at least one Target must be present.

The SCSI host adapter is connected to the SCSI bus on one side, and to the host computer bus on the other. The host computer is built around a certain internal backplane bus: it may use the PC/ISA, EISA, PCI, VME, etc. The host adapter acts as an intermediary, that translates the signals and protocols on the host computer internal bus to signals and protocols on the SCSI bus, to accomplish the task assigned by the host computer's operating system. SCSI bus adapters are designed for a certain backplane bus: for the PC/ISA, or EISA, or PCI, or VME, etc.

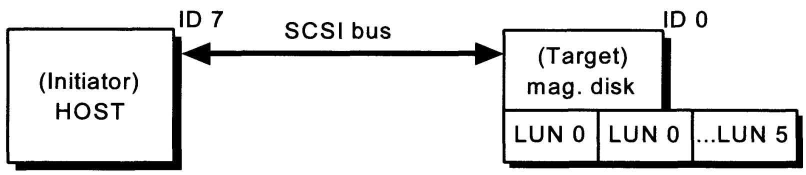

Figure 1. SCSI system: one Initiator, and one Target with six LUNs

A SCSI system may be as simple as a single computer with a SCSI host adapter connected by cable to a single SCSI Target device, such as a disk drive. More peripherals may be added to the same SCSI bus, such as a SCSI tape backup unit or a SCSI printer, etc. Each Target device can also be subdivided into several Logical Units (LUNs). It is also possible to connect several computers, each with one or more SCSI host adapters to a shared peripheral, such as a SCSI scanner. The most complex systems would have two or more computers, each with its own SCSI host adapter, connected to several different types of shared peripheral devices. On a single SCSI cable, any combination of one to eight devices (single-byte SCSI) is allowed, with at least one Initiator and one Target.

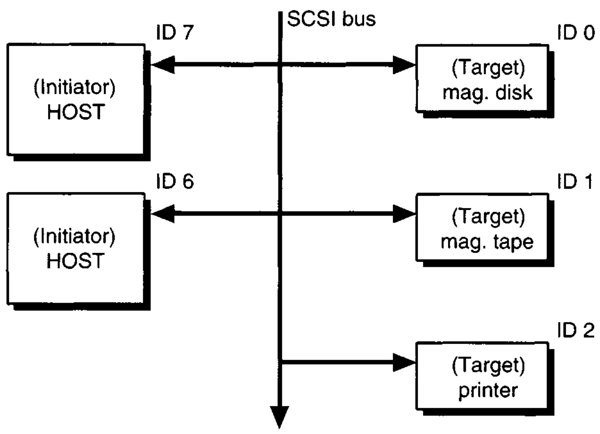

Figure 2. SCSI system with two Initiators and three Targets.
(LUNs in Targets not shown)

DEVICE ADDRESSING

The limitation for the number of devices on a single SCSI bus is a result of the need for a unique identification number (SCSI ID) for each device. Since the SCSI bus began with eight data lines, and because it does not allow for encoded ID, the maximum number of devices (SCSI IDs) on a single-byte SCSI bus is eight. In addition to this, each of the devices, except the Initiator, can have up to eight logical units (LUNs). This brings the theoretical maximum number of devices/LUNs on the 8-bit SCSI bus to 57 : 1 Initiator + 7 Targets x 8 LUNs.

SCSI devices connected to the bus must use unique identification addresses, each one with one ID bit assigned to it. The SCSI device ID is used during the Arbitration and Selection phases of each command. Higher IDs have higher priority. On the WIDE SCSI bus in SCSI-3, the low-byte IDs have higher priority over high-byte IDs. This is to allow the 8-bit devices to be always recognized by all other devices. See the table on the next page.

SCSI version	SCSI Addr.	SCSI ID D31 – D24	D23 – D16	D15 – D8	D7 – D0	Prioriy	
SCSI 2 & 3	7	---- ----	---- ----	---- ----	1--- ----	1	highest priority
	6	---- ----	---- ----	---- ----	-1-- ----	2	
	5	---- ----	---- ----	---- ----	--1- ----	3	
	4	---- ----	---- ----	---- ----	---1 ----	4	
	3	---- ----	---- ----	---- ----	---- 1---	5	
	2	---- ----	---- ----	---- ----	---- -1--	6	
	1	---- ----	---- ----	---- ----	---- --1-	7	
	0	---- ----	---- ----	---- ----	---- ---1	8	
SCSI 3 only	15	---- ----	---- ----	1--- ----	---- ----	9	
	14	---- ----	---- ----	-1-- ----	---- ----	10	
	13	---- ----	---- ----	--1- ----	---- ----	11	
	12	---- ----	---- ----	---1 ----	---- ----	12	
	11	---- ----	---- ----	---- 1---	---- ----	13	
	10	---- ----	---- ----	---- -1--	---- ----	14	
	9	---- ----	---- ----	---- --1-	---- ----	15	
	8	---- ----	---- ----	---- ---1	---- ----	16	
	23	---- ----	1--- ----	---- ----	---- ----	17	
	22	---- ----	-1-- ----	---- ----	---- ----	18	
	21	---- ----	--1- ----	---- ----	---- ----	19	
	20	---- ----	---1 ----	---- ----	---- ----	20	
	19	---- ----	---- 1---	---- ----	---- ----	21	
	18	---- ----	---- -1--	---- ----	---- ----	22	
	17	---- ----	---- --1-	---- ----	---- ----	23	
	16	---- ----	---- ---1	---- ----	---- ----	24	
	31	1--- ----	---- ----	---- ----	---- ----	25	
	30	-1-- ----	---- ----	---- ----	---- ----	26	
	29	--1- ----	---- ----	---- ----	---- ----	27	
	28	---1 ----	---- ----	---- ----	---- ----	28	
	27	---- 1---	---- ----	---- ----	---- ----	29	
	26	---- -1--	---- ----	---- ----	---- ----	30	
	25	---- --1-	---- ----	---- ----	---- ----	31	lowest
	24	---- ---1	---- ----	---- ----	---- ----	32	priority

Figure 3. SCSI adresses, IDs, and arbitration priorities

SCSI does not specify which ID should be assigned to Initiators or Targets; however in most systems it works best with Initiators using the highest IDs starting from 7 down, and Targets using IDs from 0 and up (on an 8-bit bus). The bootable disk is usually set to ID = 0. Assigning higher IDs to slower Targets (e.g., tapes) is recommended, to give them higher priority over faster Targets (e.g., disks) and to prevent the high performance devices from hogging the bus.

Logical Units (LUN)

As we described earlier, each device on the SCSI bus has its own unique address – the SCSI ID. In addition to this, each Target is also sub-divided into Logical Units (LUN). The maximum number of Logical Units in SCSI-2 is eight: LUN-0 to LUN-7. The SCSI-3 allows an almost unlimited number of LUNs (SAM reserves a 64-bit field for LUN). The actual number of LUNs used depends on each specific carrier protocol.

Usage of LUNs varies. Most magnetic disks have only a single LUN (LUN-0), but LUNs can also represent zones on a disk. Using LUNs we can address multiple peripheral devices which may share a common controller (LUN-0), etc.

The concept of LUNs is defined only for the Targets, therefore if a SCSI device implements both the Initaitor and the Target role, only the Target can have LUNs.

Asynchronous – Synchronous

There are two handshaking protocols controlling the transfer of information. They are called "*asynchronous*" and "*synchronous*". The two modes differ in the way the strobe signals REQ and ACK handshake. The ymay either interlock (*asynchronous* mode) or not (*synchronous* mode).

The "*asynchronous*" alternative requires that each byte be requested and acknowledged before the next one begins. Let's consider an Initiator connected to a Target in a point-to-point configuration. The REQ is sent by the Target from one end of the cable, and the ACK is the response by the Initiator from the other end of the cable. As indicated by arrows in Figure 4, the timing of leading and trailing edges of the REQ and ACK strobes, being interlocked, will be affected by propagation delay in the SCSI cable. As a result, slower transfer speeds are to be expected with longer distances between Initiator and Targets, due to propagation delays in the hardware (cable, driver, and receiver). The "*asynchronous*" mode is used for speed matching of devices: the slower device slows down the faster one and thus enables communication.

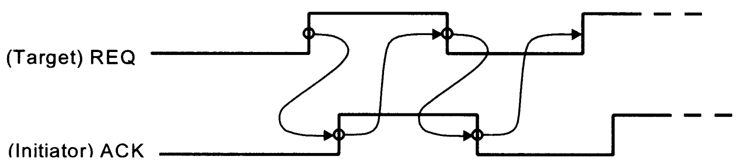

Figure 4. Asynchronous timing

The *"synchronous"* alternative allows the maximum transmission speed (crystal controlled), as long as a certain maximum offset between requests (REQ signal from Target) and acknowledgements (ACK from Initiator) is respected. The offset is usually given by the depth of the FIFO in the receiving device.

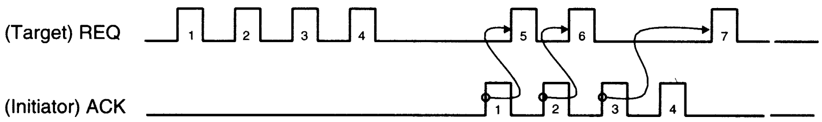

Figure 5. Synchronous timing

Actually, the *synchronous* devices use the *synchronous* mode for data transmission phases only, and use *asynchronous* mode for all other phases within the same command. Hosts can communicate with some devices in *synchronous,* with other devices in *asynchronous* mode, while both these devices are residing on the same SCSI bus. The host and the target exchange SDTR messages to negotiate the *synchronous* (optional) mode.

WIDE & FAST & UltraSCSI

The SCSI-1 specification defines the width of the data on the SCSI cable to be one byte wide, plus parity. This is called the "NARROW" SCSI bus. The term "WIDE" in SCSI-2 defines the two-byte or four-byte wide data interface. Each data byte has its corresponding parity bit.

Transfer rates above 5MHz and below 10MHz in SCSI-2 are defined as "FAST", recently often called FAST/10. The minimum transfer period (100 through 200 nsec) must be established beforehand, using the SDTR (Synchronous Data Transfer Request) message interchange individually between each Initiator and each Target (see Figure 6). There are also physical and electrical aspects to the implementation of FAST SCSI: it is recommended that only the differential interface be used.

The SCSI-2 FAST is also referred to as FAST/10, the "10" meaning maximum transfer rate of 10MHz. In 1993, the SCSI-3 proposed higher rates called FAST/20 or UltraSCSI, with maximum rate of 20MHz. See the draft ANSI standards document X3T10/1071D called "SCSI-3 FAST/20". The FAST/20 used on a WIDE bus will allow up to 40MBps transfer rate. And as if this was not enough, the ANSI standards committee defined FAST/40 (called Ultra-2) , and FAST/80 (called Ultra-3). With increasing rates the timing requirements get progressively tighter and only shorter cables are allowed. The very high rates, if accepted, will not be used with cables, but most likely will be used on backplanes where the transmission characteristics can be better controlled. All rates starting with FAST/20 will be used only in connection with active termination on the SE (Single-Ended) and LVD (Low Voltage Differential) interfaces. The FAST/40 will be used with LVD interface only.

Compatibility between devices using different speeds and timings is guaranteed by a rule which requires that all devices (SCSI-1, SCSI-2 or SCSI-3) start up in a single byte (i.e. not WIDE) and in *asynchronous* mode (i.e. not synchronous nor FAST). Before using *synchronous*, FAST, or WIDE options, the Initiator and the corresponding Target must exchange messages negotiating the width used, *synchronous* mode, *minimum transfer period,* and REQ-ACK *offset*. The WDTR (Wide Data Transfer Request) and SDTR (Synchronous Data Transfer Request) negotiation is executed for each Target individually, allowing for a mix of transfer modes on the same bus.

Synchronous transfer mode as well as FAST and WIDE are optional and are used for data phases only. All other phases (message, command, or status) always use the standard *asynchronous* mode. When enabled, the *synchronous* FAST and WIDE are used in data phases of all commands, not only in Read or Write, but also in Inquiry, Request Sense, Mode Select, etc.

Note, that the term "*synchronous*" as used in SCSI, refers to the synchronous transfer rate, and does NOT mean or imply synchronization between the Initiator and Target.

The following example shows message exchange between an Initiator and a Target, where the WDTR message was rejected by the Target, and the SDTR message negotiated synchronous mode with offset of 8 and maximum rate of 5 MHz (200 nsec).

```
Arbitration   /80 (7)
Select w/ATN  /81 (0,7)
   Message-Out/C0 (Identify : LUN 0 Disconnect OK)
   Message-Out/01 (Ext Msg)
               02 (Length)
               03 (WDTR)
               01 (Size 2 bytes)
   Message-In /07 (Msg Reject)
   Message-Out/01 (Ext Msg)
               03 (Length)
               01 (SDTR)
               28 (Period 160 nsec)
               08 (Offset 8 bytes)
   Message-In /01 (Ext Msg)
               03 (Length)
               01 (SDTR)
               32 (Period 200 nsec)
               08 (Offset 8 bytes)
   Command    /12 00 00 00 30 00 (Inquiry)
      Data-In /00 00 01 01 1F 00 00 00 41 4E 43 4F 54 20 20 20
               44 53 43 2D 33 30 32 2F 4A 20 20 20 20 20 20 20
               33 2E 34 30
   Status     /00 (Good)
   Message-In /00 (Command Complete)
Bus Free
```

Figure 6. Example of message interchange between Initiator and Target

This example shows negotiation for SCSI-2 WIDE and Synchronous transfer mode. Message-out is sent by the Initiator, message-in by the Target.

Note that the WDTR cancels the synchronous (SDTR) negotiation (according to SCSI specification), and therefore the WDTR should always be followed by SDTR. It is also recommended that the SDTR and WDTR extended messages be attached to the Inquiry and Request Sense commands, positioned within the command right after the Identify message-out as shown above. However, it is not illegal if the SDTR and WDTR are attached to other commands or if they are positioned in other parts of a command.

When is the WDTR or SDTR used? Typically, you will find the WDTR and SDTR messages used in the bootstrap sequence or following a bus reset (catastrophic condition). Once agreed upon, the width and speed parameters are remembered in the I/O drivers, and need not be repeated.

DISCONNECT – RECONNECT

When considering overall performance of the system, an important feature of SCSI is the ability to *disconnect* and *reconnect.* This feature allows slow operations (like seek in disks, or rewind in tapes) to be executed off line. Using *disconnect* and *reconnect,* several operations in a system may progress concurrently.

As with WDTR and SDTR above, the *disconnect* and *reconnect* feature has to be mutually agreed between the Initiator and each Target. The Initiator has to grant a Disconnect privilege by setting bit 6 in the IDENTIFY message at the beginning of each command.

LOGICAL BLOCK ADDRESSING (LBA)

Uniformity of data addressing, e.g. in magnetic disks, is accomplished by the concept of logical blocks. If the computer wants to access the data in a certain Logical Block Address (LBA), it does not need to know the physical geometry or layout of the device. All knowledge of the physical details (e.g. track, cylinder, head, sector address) is embodied within the device itself, where the logical address is internally converted to a physical address. This feature should be mostly an advantage for the system designer, however it may not always be so: with the logical addressing scheme, it may not be possible to accomplish maximum performance tune-up of some systems.

The Initiator uses the Read Capacity command (usually in the bootstrap sequence) to find the maximum LBA on a magnetic disk.

ASYNCHRONOUS EVENT NOTIFICATION (AEN)

The Asynchronous Event Notification (AEN) is an optional protocol which allows the Target to notify its Initiator that an asynchronous event occurred. Without the AEN, the Initiator would use the polling method to find out whether an event occurred "outside of a command", i.e. not within a command but rather during the time period when the Target and Initiator were not communicating.

There are several situations where the AEN could be useful: devices with removable media may need to notify the Initiator of a medium change, or of switching the write protect on/off. Another example may be the completion of a rewind in a tape drive.

The AEN is performed by the Target (acting temporarily as an Initiator) issuing a SEND command to the Initiator (acting temporarily as a Target). The receiving device must respond as a Processor type device.

SCSI DEVICE TYPES

There are many different kinds of peripheral devices used in computer systems today. They range from simple ones, used only for buffer-to-buffer communication, to the more complex ones using a memory-like configuration, which requires addressing. Other peripherals have mechanical parts and need to be programmed for mechanical movement, etc. SCSI divides all these devices into ten groups, called SCSI *device types.* All device types use some commands which are common to all, and additional commands unique to each individual type.

In SCSI-1 there are six SCSI device types defined:

1 - Random access	(LBA addr, Tx.Length)
2 - Sequential access	(read next record)
3 - Printer	(page layout control)
4 - Processor	(simple Send & Receive)
5 - WORM	(big size, removable)
6 - Read Only random access	

SCSI-2 has five additional device types:

6 - CD-ROM	(replacing Read Only random access)
7 - Scanner	(page layout like printer)
8 - Magneto Optical	
9 - Medium Changer(Jukebox)	(mechanical control)
10 - Communication	

SCSI-3 redefined the device types as follows:

1 - Block devices (SBC)	(magnetic & optical disks)
2 - Stream devices (SSC)	(tapes, processors, communications)
3 - Graphic devices (SGC)	(printers, scanners, pre-press)
4 - Medium changer devices (SMC)	(jukeboxes)
5 - RAID Controllers (SCC)	(controllers)
6 - CD-ROMS (MMC)	(multimedia)
7 - Enclosures (ESC)	(environmental services)

The SCSI-3 command sets are defined in eight separate specifications documents. The SCSI-3 Primary Commands (SPC) document is for the commands common to all devices. The SBC, SSC, SGC, etc. documents list additional commands specific to each of the device types.

SCSI COMMANDS AND THEIR FORMAT

A SCSI command is generated by the Initiator (in the host) and is sent to the Target in the command phase. A command and its parameters are sent as a block several bytes long called the Command Descriptor Block (CDB). In SCSI-2 or SCSI-3, there are three groups of commands, depending on the length of their CDBs:

Group	- 0	uses 6-byte CDBs
	- 1,2	uses 10-byte CDBs
	- 5	uses 12-byte CDBs

Command groups 3 and 4 are reserved, and groups 6 and 7 are "vendor specific" (a synonym for non-standard).

SCSI was initially designed for magnetic disks which in the late seventies had relatively small capacities compared with those available today. For this reason, the commands needed only small address and length fields and the six-byte CDBs were sufficient.

The generic structure of the CDB of all groups is as follows:

The first byte of the CDB is the Operation Code (OP code). It is followed by the Logical Unit Number (LUN) in the upper three bits of the second byte, and by the block address (LBA) and transfer length fields (Read and Write commands) or other parameters. The last byte of each CDB is the Control byte. This byte contains two important bits, the LINK and FLAG; these bits are used for controlling the linked commands mechanism.

CDB EXAMPLE TAKEN FROM THE SCSI SPECIFICATION

The following illustration shows exact format and definition of a READ command CDB (Command Descriptor Block). A definition of individual fields and values of the CDB follows. In the EXAMPLES OF SCSI COMMANDS section, the CDBs are shown as the *Command phase* (on the fifth line) in each of the examples.

READ command CDB (Command Descriptor Block)

READ command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code (08h)							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The READ(6) command requests that the target transfer data to the initiator. The most recent data value written in the addressed logical block shall be returned.

The cache control bits are not provided for this format. Targets with cache memory may have values for the cache control bits that affect the READ(6) command; however, no default value is defined by this command. If explicit control is required, the READ(10) command should be used.

The logical block address field specifies the logical block at which the read operation shall begin.

The transfer length field specifies the number of contiguous logical blocks of data to be transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

SCSI COMMAND PHASES

Individual commands execute in subsequent parts called *phases*. Some *phases* are set by the Initiator, some by the Target. Following is a list of SCSI command *phases:*

arbitration
selection
message out
message in
command data out
data in
status
bus free

The SCSI specification allows execution of command phases in any order, with the exception that each command must start with arbitration and selection. In practice however, we will see SCSI commands executing in *phases* as though in a certain predefined sequence. Some *phases* are missing in some commands, but the general sequence is always the same; it is usually:

arbitration
selection (with Attention)
message out
command data in/out
status
message in

COMMAND LINKING

The LINK and FLAG bits in the Control byte of a CDB (the last byte) allow linking of two or more successive commands together. If the LINK bit is set, the current command does not end with Bus Free phase, but attaches the following command by starting with its Command phase instead. Command linking is used for example in data base applications when a Read command must be followed immediately by a Write command, without interruption from another Initiator (host).

```
Bus Free
   Arbitration   /80 (7)
   Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
      Command    /08 00 00 00 01 01 (Read)
         Data-In /00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  .......
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      Status     /10 (Intermediate)
      Message-In /0A (Linked Command Complete)
      Command    /0A 00 00 00 01 00 (Write)
      Data-Out   /00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  .......
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      Status     /00 (Good)
      Message-In /00 (Command Complete)
Bus Free
```

Figure 7. READ and WRITE commands linked together.

Note the LINK bit set in the CDB, the Intermediate status, and the Linked Command Complete message-in in the READ command.

COMMAND QUEUING

Command Queuing is the capability of a SCSI Target to accept multiple commands (I/O Processes), and to execute them in an optimum sequence. This may sometimes be useful, for example, in a high performance system to minimize latency, access, or seek times in disks. It is an optional function which may or may not be supported by a Target.

HOW SCSI COMMANDS EXECUTE

CONTROL OVER THE BUS

The wires of the SCSI bus are shared among all the devices connected. Because the wires can carry only one piece of information at a time, a device which wishes to use the SCSI bus must obtain permission from all the other devices *(Arbitration phase)*. After winning the arbitration, the device controlling the bus must get the attention of the device with which it wishes to communicate *(Selection phase).* These actions together makeup the initial part of each command. Following the *selection phase,* information transfer starts, and the data is sent between the two devices. We say that the commands execute in phases. These phases alternate until the command has completed. Furthermore, when more than two devices are connected on a SCSI bus, all of the phases from one command may not be contiguous in time. Phases from other commands (to other devices) may be interleaved. However, following arbitration and selection, if a connection is successfully made, a certain sequence of phases follows through to completion of the command.

NEGOTIATING FOR THE BUS (Starting a SCSI Command)

The negotiation stage begins with an idle bus. The idle condition is called the Bus *Free phase*, and is detected when both the BSY and SEL control lines are released by all devices. During a Bus Free phase, a device needing to use the bus begins an *Arbitration phase* by asserting the BSY control line and its own ID data line. Note that the device drives only one data bit representing its own ID number. At the same time, other data bits (lines) may be driven by the devices that own them. If more than one device attempts to gain control during the same Arbitration phase, the one with the highest priority ID number (see Figure 3) wins control of the bus, and the losing devices must release all their lines within a specified time. The winning device then attempts to get the attention of its Target by beginning a *Selection* or *Reselection* phase. The SEL control line is asserted, and the ID data line for the device being addressed is added to the data bus; at this point, two data lines (Initiator ID and Target ID bits) are driven "true" (Low) and with good (odd) parity. After this, the BSY control line is released. When the addressed device notices that it is being addressed, it asserts the BSY control line. Finally, the calling device releases the SEL control line and the data bus. There are some minor differences depending on whether the caller is an Initiator or a Target, but the statements here hold regardless.

INFORMATION TRANSFER

An information stage begins when the selection phase ends. During the information stage, the Target controls the quantity and variety of information transmitted, and the direction of transfer for each byte. As in the arbitration and selection phases, the information stage is divided into individual phases called *Information Transfer phases: Msg-Out phase, Command phase, Data-In/Out phase, Status phase,* and *Msg-Inphase.* Although there is no set order for the phases in an information stage, this is the usual phase sequence. The information stage lasts until the Target releases the BSY control line, thus returning to the *Bus Free phase.*

Phase is determined by the Target which holds the BSY control line active, and asserts a particular pattern on the MSG, C/D, and I/O control lines. Each pattern has its own phase name. If the Target determines that the next piece of information will take a long time to prepare for, it can end the connection by issuing the DISCONNECT message and return to the Bus Free phase, intending to reestablish connection later (re-connect) using new Arbitration and Reselection phases. Thus the transfer of data for a command can be spread over several pairs of connections.

There is a relatively standard sequence of phases by which a command is executed. The command starts with *Arbitration phase* followed by *Selection phase*.

If during selection the ATN signal is asserted (this is mandatory in SCSI-2), the next is the *Message-Out phase,* where the Initiator tells the Target which part of the Target (which LUN) should act on the command. The Initiator may also enable the *disconnect* mechanism in the Target, and establish the mode of data transfer during the *Data-In* and *Data-Out phases* that follow later in the command.

In SCSI-1 it was allowed to skip the *Msg-Out phase* (the ATN signal not asserted during selection), and proceed directly to the *Command phase.*
Next is the *Command phase.* This is the time when the Initiator tells the Target what to do; it sends the Command Descriptor Block (CDB) to the Target. The particular command may require that more information be transferred between the devices. This is done during a *Data-In* or *Data-Out phase,* but never both within the same command. The Target can give interim progress report to the Initiator using a *Message-In phase*; this is a requirement just before terminating an information stage. Also, the Target tells the Initiator whether the command was successful or not by using the *Status phase.* Finally, the command is completed by sending final progress report using a *Message-In phase.*

THE MESSAGE SYSTEM

The message system in SCSI is used to allow communication between the Initiators and Targets for the purpose of interface management. Messages can be sent in both directions: message-out from Initiator to Target, or message-in from Target to Initiator. There are three types of message formats:

1. single-byte,
2. two-byte, and
3. extended messages - which can be three bytes or longer.

We can also classify messages by their usage:

1. Identification messages identify the I/O process as part of an initial connection by a host adapter or during reconnection by a target.
2. I/O Process Management messages are used to manage I/O processes, logical units, and targets.
3. Connection Management messages manage the physical transfer of information between host adapters and targets.

MESSAGE NAME	CODE (Hex)	Direction	Initr	Target
Abort	06	Out	O	M
Abort Tag	0D	Out	O	O
Bus Device Reset	0C	Out	O	M
Clear Queue	0E	Out	O	O
Command Complete	00	In	M	M
Disconnect	04	In/Out	O	O
Identify	80+	In/Out	M	M
Ignore Wide Residue (2 bytes)	23	In	O	O
Initiate Recovery	0F	In/Out	O	O
Initiator Detected Error	05	Out	M	M
Linked Command Complete	0A	In	O	O
Linked Command Complete(w.Flag)	0B	In	O	O
Message Parity Error	09	Out	M	M
Message Reject	07	In/Out	M	M
Modify Data Pointer	Ext.Msg	In	O	O
No Operation	08	Out	M	M
Queue Tag Messages(2 bytes):				
Head of Queue Tag	21	Out	O	O
Ordered Queue Tag	22	Out	O	O
Simple Queue Tag	20	In/Out	O	O
Release Recovery	10	Out	O	O
Restore Pointers	03	In	O	O
Save Data Pointer	02	In	O	O
Synchr.Data Tx.request (SDTR)	Ext.Msg	In/Out	O	O
Terminate I/O Process	11	Out	O	O
Wide Data Tx.Request (WDTR)	Ext.Msg	In/Out	O	O

M - Mandatory   In - Target to Initiator
O - Optional   Out - Initiator to Target
80+ - Codes 80Hex through FFHex used for IDENTIFY Messages

Figure 8. SCSI-2 Message codes

The most common message is the single byte with value = 00, which means "command complete". This is the I/O Process Management message which is sent in the last phase of a command.

The Target controls the phases (by driving the C/D, I/O, and MSG lines), and therefore, when an Initiator needs to send a message to the Target, it has to signal the Target by asserting the ATN line. As a response, the Target sets the message-out phase (by setting the C/D, I/O, and MSG lines accordingly), and receives the message.

Figure 8 lists SCSI-2 messages in alphabetical order. Most of the messages are self explanatory, however, the reader is referred to the SCSI specification for detailed individual explanations.

THE STATUS

Status of the Target is reported to the Initiator on two levels: brief status in the status phase of each command, or more detailed status in SENSE DATA.

The status byte at the end of each SCSI command (in status phase) reports to the Initiator whether execution of a command was successful (value = 00), whether an error occurred (value = 02), or reports other information like BUSY (value = 08), etc. This status byte does not give any detailed information however. The following table lists SCSI-2 values used:

STATUS DESCRIPTION	VALUE (Hex)
Good	00
Check Condition	02
Condition met	04
Busy	08
Intermediate	10
Intermediate-Condition met	14
Reservation conflict	18
Command terminated	22
Queue full	28
All other codes are reserved.	

Figure 9. SCSI-2 status codes

If a Check Condition status is reported for a command (value = 02) it means that some condition exists which prevented successful completion of that command. It could be an error in the CDB, a hardware problem, or some external problem. The Initiator can obtain detailed information regarding this condition by issuing a REQUEST SENSE command. The Target will respond by returning SENSE DATA in the data-in phase of the same command.

The SENSE DATA is a set of flags and indicators within the Target, which are continuously updated to reflect current status in full detail. The SENSE DATA is presented in a hierarchical manner by the Target. It starts with general conditions and proceeds to specific conditions as follows:

1. SENSE-KEY
2. ADDITIONAL SENSE CODE (ASC)
3. ADDITIONAL SENSE CODE QUALIFIER (ASCQ)

The Sense Key defines the class of error or condition information. There are fifteen Sense Keys which classify the device condition as hardware or software problem, a fatal or recoverable error, etc. The ASC and the ASCQ provide more detailed information. Consult the SCSI specification and your particular device manual for specific descriptions.

THE RULE ABOUT USING THE REQUEST-SENSE COMMAND

Whenever an Error or other unusual situation occurs, the Sense Data in the Target is updated, and a Check Condition status is reported to the Initiator at the end of the current command. When the

Initiator receives a Check Condition, it should issue a Request-Sense command to inquire about the (potential) error. It is important to remember that the first command following a Check Condition also resets the Sense Data in the Target and therefore, if a REQUEST SENSE command is not used at this point (following the Check Condition), the information about the error will be lost.

EXAMPLES OF SCSI COMMANDS AND USAGE RULES

In the following examples we show commands TEST UNIT READY, INQUIRY, REQUEST SENSE, SEND DIAGNOSTIC, RECEIVE DIAGNOSTIC RESULTS, MODE SELECT, MODE SENSE, READ, WRITE which use 6-byte CDB (Command Descriptor Block), and the READ CAPACITY command which uses a 10-byte CDB. Also 12-byte CDB commands are standard, although not shown here.

```
Bus Free
   Arbitration   /80 (7)
   Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
      Command    /00 00 00 00 00 00 (Test Unit Rdy)
      Status     /00 (Good)
      Message-In /00 (Command Complete)
Bus Free
```

The Initiator, ID 7 in the example above, uses the TEST UNIT READY command to find out if a certain peripheral (Target ID 0) is connected and powered up, and if it is operational.

```
Bus Free
Arbitration   /80 (7)
Select w/ATN  /81 (0,7)
   Message-Out/C0 (Identify : LUN 0 Disconnect OK)
   Command    /12 00 00 00 30 00 (Inquiry)
      Data-In /00 00 01 01 1F 00 00 00 41 4E 43 4F 54 20 20 20
               44 53 43 2D 33 30 32 2F 4A 20 20 20 20 20 20 20
               33 2E 34 30
   Status     /00 (Good)
   Message-In /00 (Command Complete)
Bus Free
```

In the example above, the Initiator uses the INQUIRY command to find out what kind of Targets are at individual IDs. Each selected Target uses the Data-In phase to identify itself and its "personality". This data must be in "readable ASCII" form. Note that the INQUIRY command in this example does not have the WDTR and SDTR extended messages, although it probably would show the WDTR and SDTR if this was the first INQUIRY in the bootstrap sequence in a synchronous and/or wide system.

```
Bus Free
Arbitration   /80 (7)
Select w/ATN  /81 (0,7)
   Message-Out/C0 (Identify : LUN 0 Disconnect OK)
   Command    /03 00 00 00 10 00 (Request Sense)
      Data-In /70 00 06 00 00 00 00 00
   Status     /00 (Good)
   Message-In /00 (Command Complete)
Bus Free
```

In the example above, the Initiator uses the REQUEST SENSE command whenever it receives a CHECK CONDITION status in a command. The SENSE DATA (received in Data-In phase) reports detailed status (information about failure or a certain condition) of the Target. By executing the REQUEST SENSE (or any other command), the SENSE DATA is reset in the Target.

```
Bus Free
Arbitration   /80 (7)
Select w/ATN  /81 (0,7)
   Message-Out/C0 (Identify : LUN 0 Disconnect OK)
   Command    /1C 00 00 00 30 00 (Receive Diagnostic Results)
      Data-In /40 00 00 0A 00 05 00 02 FD 03 00 00 00 27
   Status     /00 (Good)
   Message-In /00 (Command Complete)
Bus Free
```

In the example above, the SEND DIAGNOSTIC command provides a means to request the Target to perform a self test. If the self test successfully passes, the command returns GOOD status; otherwise, the command will be terminated with CHECK CONDITION status, and the Initiator can use the REQUEST SENSE command to request the SENSE DATA describing details about the failure.

```
Bus Free
Arbitration   /80 (7)
Select w/ATN  /81 (0,7)
   Message-Out/C0 (Identify : LUN 0 Disconnect OK)
   Command    /1D 04 00 00 00 00 (Send Diagnostic)
   Status     /00 (Good)
   Message-In /00 (Command Complete)
Bus Free
```

In the example above, the RECEIVE DIAGNOSTIC RESULTS command requests that analysis data be sent to the Initiator after completion of a SEND DIAGNOSTIC command. It must follow the SEND DIAGNOSTIC immediately, because execution of any command resets the results data. These two commands may be linked.

```
Bus Free
    Arbitration   /80 (7)
    Select w/ATN  /81 (0,7)
       Message-Out/C0 (Identify : LUN 0 Disconnect OK)
       Command    /15 00 00 00 18 00 (Mode Select)
          Data-Out/00 00 00 08 00 00 F5 D4 00 00 02 00 08 0A 01 00
                   00 00 00 00 00 00 00 00
       Status     /00 (Good)
       Message-In /00 (Command Complete)
 Bus Free
```

In the example above, the Initiator uses the MODE SELECT command to specify medium, LUN, or peripheral device parameters to the Target. Another possible use may be turning on caching in Random Access devices. The parameters in the Target are grouped in mode pages. The mode pages can mean different kinds of information in different device types.

Targets that implement the MODE SELECT command shall also implement the MODE SENSE command. Initiators should issue MODE SENSE prior to each MODE SELECT to determine supported pages, page lengths, and other parameters.

```
Bus Free
   Arbitration   /80 (7)
   Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
     Command    /1A 00 04 00 FF 00 (Mode Sense)
     Data-In    /23 00 00 08 00 00 00 00 00 00 02 00 04 16 00 0F
                 64 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                 15 18 00 00
      Status     /00 (Good)
      Message-In /00 (Command Complete)
 Bus Free
```

In the example above, the MODE SENSE command enables a Target to report parameters to the Initiator. This command can request only specific information, by specifying certain mode pages (use the Page Code field in the third byte of the CDB). A page code of 3F hex would indicate that all mode pages implemented by the Target shall be returned to the Initiator. The MODE SENSE command is a complementary command to the MODE SELECT command. During the boot process for DOS, the MODE SENSE command is used to get the physical layout of the disk, to be used by the INT 13H DOS Bios functions.

```
Bus Free
   Arbitration   /80 (7)
   Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
      Command    /25 00 00 00 00 00 00 00 00 00 (Read Capacity)
         Data-In /00 00 F5 D4 00 00 02 00
      Status     /00 (Good)
      Message-In /00 (Command Complete)
 Bus Free
```

In the example above, the Initiator uses the READ CAPACITY command to request information regarding the capacity of the logical units on random access devices. In this example, the Target (magnetic disk) reports in the Data-In its maximum LBA 000OF5D4 hex (62,932 dec) and block length = 200 hex (512 dec).

```
Bus Free
   Arbitration   /80 (7)
  Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
      Command    /0A 00 00 00 01 00 (Write)
         Data-Out/00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  .......
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      Message-In /04 (Disconnect)
Bus Free
   Arbitration   /01 (0)
   Reselection   /81 (0,7)
      Status     /00 (Good)
      Message-In /00 (Command Complete)
Bus Free
```

The instance above is an example of a WRITE command. This Target has a cache buffer and therefore accepts the data immediately, then it disconnects while seeking and writing out the data. It later reconnects to report completion. If the Target did not use cache, we would see the Data-Out

phase in the second part of the command, following Reselection, similar to the READ command example below.
When transferring long data blocks in the READ or WRITE commands, the disconnect is used to divide the data phase into several shorter ones (separated by Bus Free), to allow bus access to other devices.

```
Bus Free
   Arbitration   /80 (7)
   Select w/ATN  /81 (0,7)
      Message-Out/C0 (Identify : LUN 0 Disconnect OK)
      Command    /08 00 00 10 20 00 (Read)
      Message-In /04 (Disconnect)
Bus Free
   Arbitration   /01 (0)
   Reselection   /81 (0,7)
      Message-In /80 (Identify : LUN 0)
         Data-In /00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                  .......
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      Status     /00 (Good)
      Message-In /00 (Command Complete)
Bus Free
```

Above is an example of a READ command. This Target needs to perform a seek before reading data, and therefore it uses Disconnect followed by Reselection to disconnect itself from the SCSI bus while seeking, and to allow the Initiator to communicate with other Targets in the meantime.

In the following example, the same READ command is shown in a time domain format (as on a logic analyzer). Number (in Hex) in the "Addr:" column points to the acquisition memory location where the individual SCSI events are recorded. Individual SCSI signals are labeled on the first line. The column marked "Ph" shows the abbreviated SCSI phase designation. The timing is shown in the "Time" column on the right in nanoseconds. It is the time differential from the transition on the precedingline. Note the Disconnect message at address 00010(Hex) resulted in Bus Free phase which lasted 1 ms 377us and 400ns.

```
 Addr:   BSY SEL ATN RST MSG C/D I/O Ph  Data          Time
00000:   ---- Start Trace ----
00001:      |.║_ .║  .║  .║  .║  .║  .Arb  80             0
00002:      |   |  ║   ║   ║   ║   ║        80         2 360
00003:      |   |  ║_  ║   ║   ║   ║        80         1 080
00004:    _|   |   |  ║   ║   ║   ║        81           120
00005:   ║_ .   |.  |.║  .║  .║  .║  .Sel  81           120
00006:      |  _|   |  ║   ║   ║   ║        81           840
00007:      | ║   _|  ║   ║_  ║_  ║        00           200
00008:      |.║  .║  .║  .  |.  |.║  .MO   C0       132 360
00009:      | ║   ║   ║   _|   |  ║        C0            80
0000A:      |.║  .║  .║  .║  .  |.║  .Cmd  08       182 940
0000B:      | ║   ║   ║   ║    |  ║        00           600
0000C:      | ║   ║   ║   ║    |  ║        00           500
0000D:      | ║   ║   ║   ║    |  ║        10           500
0000E:      | ║   ║   ║   ║    |  ║        20           500
0000F:      | ║   ║   ║   ║_   |  ║_       00           500
00010:    _|.║  .║  .║  . _|. _|. _|.MI   04       984 560
00011:   ║_ .║  .║  .║  .║  .║  .║  .BFr           577 300
00012:      |.║_ .║  .║  .║  .║  .║  .Arb  01     1 377 400
00013:      |   |  ║   ║   ║   ║   ║        01         2 720
00014:    _|   |  ║   ║   ║   ║   ║_       01         1 000
00015:   ║_ .   |.║  .║  .║  .║  .   |.Rsl  81           520
00016:      |  _|  ║   ║   ║   ║    |       81           560
00017:      | ║    ║   ║   ║_  ║_   |       00       126 680
00018:      |.║  .║  .║  . _|. _|   |.MI   80        30 520
00019:      |.║  .║  .║  .║  .║  .  |.DI   00     1 164 860
0001A:      | ║   ║   ║   ║   ║     |       00       625 000
0001B:      | ║   ║   ║   ║   ║     |       00           220
0001C:      | ║   ║   ║   ║   ║     |       00           260
0001D:      | ║   ║   ║   ║   ║     |       00           240
0001E:      | ║   ║   ║   ║   ║     |       00           280
0001F:      | ║   ║   ║   ║   ║     |       00           220
00020:      | ║   ║   ║   ║   ║     |       00           260
   .
   .
   .
00219:      | ║   ║   ║   ║   ║_    |       00           260
0021A:      |.║  .║  .║  .║_ .   |.  |.St   00     4 332 260
0021B:    _|.║  .║  .║  . _|. _|. _|.MI   00       564 240
0021C:   ║  .║  .║  .║  .║  .║  .║  .BFr           408 200
00024:   ---- End of Trace ----
```

SCSI HARDWARE

SCSI SIGNALS

```
          CONTROL BUS:

Phase Control Signals:
     C/D     (Control/Data)          - driven by Target
     I/O     (Input/Output)          - driven by Target
     MSG     (Message)               - driven by Target

Data Handshake Strobes:
     REQ     (Request)               - driven by Target
     ACK     (Acknowledge)           - driven by Initiator

Other Control Signals:
     BSY     (Busy)                  - driven by Initiator / Target
     SEL     (Select)                - driven by Initiator / Target
     ATN     (Attention)             - driven by Initiator
     RST     (Reset)                 - driven by Initiator / Target
```

```
          DATA BUS:

     DB(7-0,P) (Data bus)            - bi-directional data bus driven by
            (low byte)               Initiator(data-out),Target(data-in)
                                         Data Parity is always ODD.
     DB(8-15,PH) (byte 2)            - used in SCSI-2 WIDE (16 bits)
     DB(16-31,2P) (byte3,4)          - used in SCSI-2 WIDE (32 bits)
```

```
          SCSI SIGNALS - OTHER:

     TERMPWR        - source of 4.25 Volt(min) for terminators
     DIFFSENSE      - indicate S-E, HVD, or LVD interface.
                      Can be used as an (active high) enable
                      for the differential drivers
```

Figure 10. SCSI bus signals

SIGNAL DESCRIPTIONS

BSY	an "OR-tied" signal indicating that the bus is being used. The Initiator drives this line while arbitrating for the bus. The Target drives this line as a response to selection, and releases it at the end of the command.
SEL	an "OR-tied" signal used by an Initiator to select a Target, or used by a Target to reselect an Initiator.
CD, I/O, MSG	signals driven by a Target to set a certain phase. In addition, the 1/0 line is driven by a Target during reselection.
REQ	a data strobe signal driven by a Target.
ACK	a data strobe signal driven by an Initiator. It is an acknowledge response to the REQ strobe from the Target.
ATN	a signal driven by an Initiator to indicate to a Target that the Initiator has a message ready. The Target should respond by switching to the *Message-Out* phase, using C/D, I/O, and MSG lines.
RST	an "OR-tied" signal that generates the RESET event when asserted. It is recommended that only one device on the bus, typically an Initiator, drive the RST.
DATA BUS	the data bus is bidirectional, and is used for data, command, status, or message transfers between an Initiator and a Target. The low data byte (DBO - 7) is used on all systems. WIDE configurations, in data phases only, may also use the higher data bytes, if WDTR negotiation between that Initiator and Target was successful. The data bus is also used for addressing during arbitration, selection, and reselection phases.
DATA PARITY	each data byte has its own PARITY bit. The PARITY is always ODD. PARITY is not used (not driven) during arbitration. PARITY was optional in SCSI-1; it is mandatory in SCSI-2 or in SCSI-3.
DIFFSENSE	is used to indicate Single-Ended (SE), High Voltage Differential (HVD), or Low Voltage Differential (LVD). The voltage levels are: less than 0.5V for SE between 0.7V and 1.9V for LVD, and greater than 2.4V for HVD. If LVD devices detect anything other than LVD, they will become high impedance.

ELECTRICAL INTERFACE

The SCSI standard provides for three possible electrical configurations. The lowest cost *"Single-Ended"* alternative was intended for devices in close proximity to each other, as in single cabinet implementations (specified up to 6 meters). The more expensive *"differential"* alternative called also *HVD (High Voltage Differential)* provides for distances up to 25 meters and better noise immunity. In 1995 the most recently introduced *LVD (Low Voltage Dift~rential)* (up to 12 meters) was introduced for SCSI-3. *Single-Ended, LVD,* and *HVD* interfaces can NOT be mixed on the same cable and be operational. All three alternatives require termination on both physical ends of the cable.

NOTE: The 6-meter, 12-meter, or 25-meter distances can be significantly extended in low noise environments, if slower transfer rates are used and if all design standards are respected.

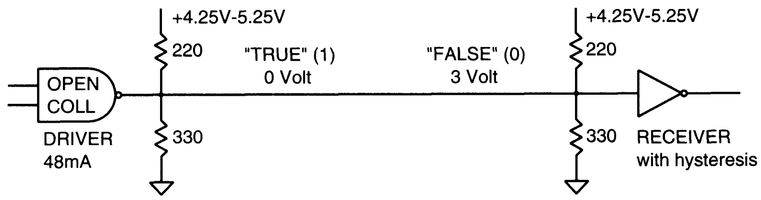

Figure 11. Single-Ended (SE) interface

The Single-Ended interface is also called unbalanced. The driver on the Single-Ended interface should be the "open drain" type, 48mA or equivalent. The receiver should have high hysteresis acting as a "glitch eater" to filter out line noise on longer cables and to prevent double triggering.

ACTIVE NEGATION DRIVERS

When using the FAST transfer mode and running higher rates on a Single-Ended bus, one may not rely on the terminators, working in tandem with open drain drivers, to pull up signals to their de-asserted state quickly enough. To speed up the trailing edge transitions, line drivers with capability to temporarily assist in returning the voltage level to the "off' state are used. This feature is called *active negation.* It may be more important on some signal lines than others. If enabled, *active negation* should always be used on the REQ and ACK, then with almost the same priority it should also be used on the data bus, including parity; it would rarely have noticeable effect on other signals. It should never be used on OR-tied signals like BSY, SEL, or RST.

New SCSI chips have an *active negation* feature selectable (register programmable) by signal groups:

1. REQ and ACK,
2. Data and Parity, and
3. all other signals.

When using active negation, expect that the drivers will consume more power, and that you may run into more (secondary) crosstalk problem. All newer SCSI driver chips are equipped with *active negation* drivers. Active termination is strongly recommended when using *active negation* drivers.

HIGH VOLTAGE DIFFERENTIAL (HVD) interface

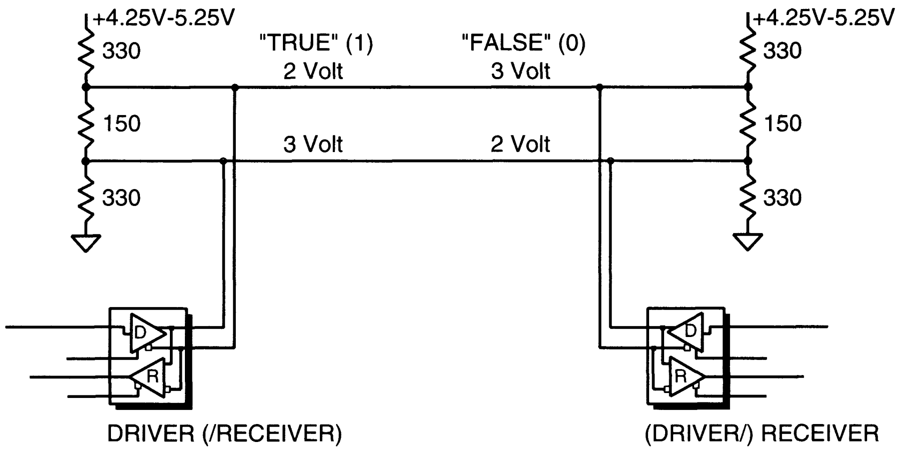

Figure 12. High Voltage Differential (HVD) interface

The differential interface uses transfers over two electrically symmetrical lines denoted +Signal and -Signal. The logical "TRUE" value is defined as +Signal more positive than -Signal. The logical "FALSE" value as +Signal more negative than -Signal.

Remember that the Single-Ended, the Low Voltage Differential (LVD), and the High Voltage Differential (HVD) alternatives are mutually exclusive, and cannot be mixed on the same bus system and be operational. The system is either all Single-Ended, LVD, or HVD; that applies to the devices and also to the terminators.

The S-E and LVD devices can be connected to the same SCSI bus, however on power up, if at least one S-E device is detected, the LVD devices must convert to S-E mode, and S-E mode will be used.

The Single-Ended interface is limited by poor signal quality at high speeds and long cables, due to transmission line effects aggravated by the wide variety of legal cable plants. The differential interface provides much higher signal quality, but its disadvantage is higher price. Higher price is caused by high power requirements for the differential line drivers, which prevents integrating them in the protocol chip.

The Low Voltage Differential (LVD) interface should resolve this dilemma; it shortens the differential maximum cable length by 50%, yet allows cable twice as long as that for the standard Single-Ended interface. See the following paragraph for details.

LOW VOLTAGE DIFFERENTIAL (LVD) interface

Only the Single-Ended interface was used with SCSI-1. Since relatively lower transfer rates were then used, this interface was satisfactory. With the introduction of the higher rates over 5MHz, the electrical quality of the interface became critical, and thus the differential interface became more popular. Besides having higher signal quality, it provided lower radiation and higher crosstalk immunity. The differential interface was more expensive than the Single-Ended, but the cost was not an important factor at that time.

Recently, with the transfer rates of 10MHz and higher, the differential interface became more popular, but its high cost became an issue. For this reason, the Low Voltage Differential (LVD), using 3.3 volts source rather than 5 volts, has been introduced.

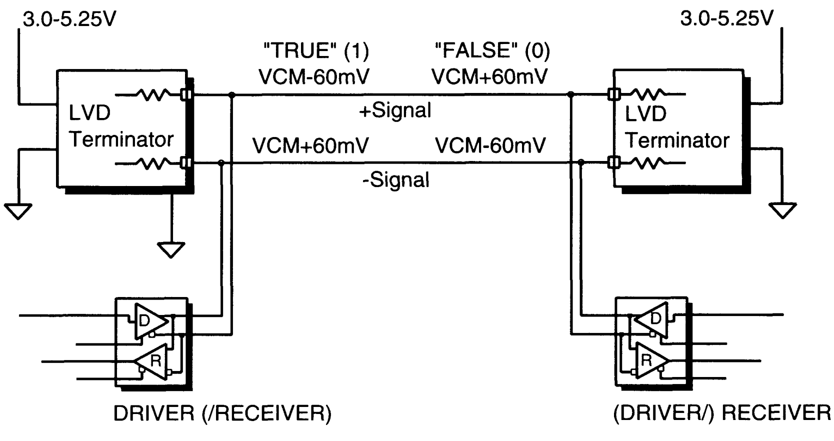

Figure 13: Low Voltage Differential (LVD) interface

The LVD reduces the allowable SCSI bus length by half, to 12 meters. It is not compatible with existing Single-Ended or HVD interfaces. The advantage of the LVD is that it allows building bus drivers inside the protocol chip, enabling the cost to be practically the same as with the Single-Ended drivers. The lower cost and high electrical quality will allow significant increase in popularity. The connector pinout of the LVD is compatible with the Single-Ended style. The HVD, mostly called simply *differential,* has data and control signals one position out from the center of the connector, making it physically incompatible with the S-E or LVD layout. The chip manufacturers are now able to build protocol chips with both interfaces, the Single-Ended and LVD, and with automatic detection of the attached interface. Only the LVD interface is being specified for the Fast/40 and higher rates.

MULTI-MODE

The S-E and LVD interfaces have compatible connector pin assignments. The odd numbered connector pins carry the negative lines for both the S-E and LVD; even numbered pins carry the positive lines for the LVD, and grounds for the S-E interface. Remember, this compatibility does NOT exist between the S-E and HVD interfaces! This compatibility allows building devices that can handle both interfaces. *Multi-mode* the name for a feature built in many new devices that can switch automatically between the S-E and LVD interface.

The rule for *multi-mode* operation: if one or more S-E devices is detected on the bus, S-E mode is enabled; if all devices on the bus are LVD, then LVD mode is enabled. Switching is done depending on the voltage level on the DIFFSENSE signal line: if the level is less than 0.5V, S-E mode is enabled; if it is between 0.7V and 1.9V, LVD mode is enabled; and if it is greater than 2AV, HVD mode is enabled. If an LVD device detects something other than LVD, it will become high impedance.

DEVICE ISOLATION & CAPACITIVE LOAD

In configured SCSI buses, the effect of capacitive loading of the signal lines is often overlooked. The SCSI-2 standard allows capacitance of up to 25pF per device per line. On a fully configured NARROW (8-bit) bus, the capacitance from the devices could add up to 200pF (8 x 25pF) in extreme cases. Then add the capacitance of the cable to it (we should use cables as short as possible!). This high capacitance may be tolerable at rates of 1MHz or slightly higher, however it is a serious problem at rates over 10MHz. Today, the integrators usually reduce the capacitance by reducing the number of devices connected.

One possible solution is being proposed, to design SCSI devices with drivers that would reduce their capacitive load to 5pF per line when the device is not selected. The BSY line would be monitored and the non-selected device would be disabled when BSY was active. Only the selected device would be loading the lines with 25pF.

The capacitive isolation is critical on REQ and ACK lines, and to a certain degree on the data lines during the data phase. The BSY line is the only one that would not use this mode; BSY would be used for controlling the enable/disable mechanism. The best capacitive loading situation is obviously in point-to-point configurations, when only two devices are connected to the bus, loading it with 50pF (2 x 25pF).

For more information on this subject, read the "SCSI Cable Length & Number of Devices" chapter on page 45.

SINGLE CONNECTOR ATTACHMENT (SCA)

When building disk arrays or other multiple device configurations, the cabling, termination and other mechanical aspects become important issues. The Single Connector Attachment (SCA) moves wiring of the SCSI bus directly to the backplane, and allows plugging of the drives into a single socket.

The connector mechanical definition specifies an 80-pin male connector on the SCSI device, compatible with an 80-pin receptacle, such as the AMP Champ #2-2-557103-1 on the backplane.

The SCA connector carries all SCSI signals, power, and auxiliary signals: The SCSI signal pins group supports the **NARROW or WIDE** (16-bit) bus, with Single-Ended or differential interface. The power contacts are for +5 volts and for +12 volts. The auxiliary signals are for spindle synch, SCSI ID selection, power fail, and spin-up control. The LED "Active" signal is for indication of SCSI operation. Note that no TERMPWR lines are included; it is assumed that the tenninator circuits are on the backplane.

The latest version, the SCA-2 specification adds *"hot plugging"* and motor control capability.

CABLES AND CONNECTORS

Several cables are defined in SCSI. However only two of these are being used in existing systems: the 50-wire "A" cable as defined in SCSI-1 and in SCSI-2; and the 68-wire known as the "P" cable, using a 68-pin connector. The "P" cable is used on SCSI-2 systems, although it was not defined in the SCSI-2; it is in the SCSI-3 specification.

The SCSI connector pinout is different for Single-Ended and HVD interfaces. In the 50-pin HVD connector, pairs 1 & 26 and pairs 25 & 50 are used for grounds, while in the Single-Ended these grounds would be on pairs 10 & 35 and 15 & 40 respectively. Consequently, signal groups in the HVD layout were shifted to the center of the connector. This fact should be taken into consideration when designing round cables with shielded REQ and ACK lines for Single-Ended or HVD interfaces.

Single-Ended				High Voltage Differential (HVD)			
1	GND	26	D0-	1	GND	26	GND
2	GND	27	D1-	2	D0+	27	D0-
3	GND	28	D2-	3	D1+	28	D1-
4	GND	29	D3-	4	D2+	29	D2-
5	GND	30	D4-	5	D3+	30	D3-
6	GND	31	D5-	6	D4+	31	D4-
7	GND	32	D6-	7	D5+	32	D5-
8	GND	33	D7-	8	D6+	33	D6-
9	GND	34	DPAR-	9	D7+	34	D7-
10	GND	35	GND	10	DPAR+	35	DPAR-
11	GND	36	GND	11	DIFFSENS	36	GND
12	reserved	37	reserved	12	reserved	37	reserved
13	open	38	TRMPWR	13	TRMPWR	38	TRMPWR
14	reserved	39	reserved	14	reserved	39	reserved
15	GND	40	GND	15	ATN+	40	ATN-
16	GND	41	ATN-	16	GND	41	GND
17	GND	42	GND	17	BSY+	42	BSY-
18	GND	43	BSY-	18	ACK+	43	ACK-
19	GND	44	ACK-	19	RST+	44	RST-
20	GND	45	RST-	20	MSG+	45	MSG-
21	GND	46	MSG-	21	SEL+	46	SEL-
22	GND	47	SEL-	22	C/D+	47	C/D-
23	GND	48	C/D-	23	REQ+	48	REQ-
24	GND	49	REQ-	24	I/O+	49	I/O-
25	GND	50	I/O-	25	GND	50	GND

Figure 14. Example of 50-pin “A” cables (“NARROW” bus)

Single-Ended			
1	GND	35	D12-
2	GND	36	D13-
3	GND	37	D14-
4	GND	38	D15-
5	GND	39	DPARH-
6	GND	40	D0-
7	GND	41	D1-
8	GND	42	D2-
9	GND	43	D3-
10	GND	44	D4-
11	GND	45	D5-
12	GND	46	D6-
13	GND	47	D7-
14	GND	48	DPAR-
15	GND	49	GND
16	GND	50	GND
17	TRMPWR	51	TRMPWR
18	TRMPWR	52	TRMPWR
19	reserved	53	reserved
20	GND	54	GND
21	GND	55	ATN-
22	GND	56	GND
23	GND	57	BSY-
24	GND	58	ACK-
25	GND	59	RST-
26	GND	60	MSG-
27	GND	61	SEL-
28	GND	62	C/D-
29	GND	63	REQ-
30	GND	64	I/O-
31	GND	65	D8-
32	GND	66	D9-
33	GND	67	D10-
34	GND	68	D11-

High Voltage Differential (HVD)			
1	D12+	35	D12-
2	D13+	36	D13-
3	D14+	37	D14-
4	D15+	38	D15-
5	DPARH+	39	DPARH-
6	GND	40	GND
7	D0+	41	D0-
8	D1+	42	D1-
9	D2+	43	D2-
10	D3+	44	D3-
11	D4+	45	D4-
12	D5+	46	D5-
13	D6+	47	D6-
14	D7+	48	D7-
15	DPAR+	49	DPAR-
16	DIFFSENS	50	GND
17	TRMPWR	51	TRMPWR
18	TRMPWR	52	TRMPWR
19	reserved	53	reserved
20	ATN+	54	ATN-
21	GND	55	GND
22	BSY+	56	BSY-
23	ACK+	57	ACK-
24	RST+	58	RST-
25	MSG+	59	MSG-
26	SEL+	60	SEL-
27	C/D+	61	C/D-
28	REQ+	62	REQ-
29	I/O+	63	I/O-
30	GND	64	GND
31	D8+	65	D8-
32	D9+	66	D9-
33	D10+	67	D10-
34	D11+	68	D11-

Figure 15. Example of 68-pin “P" cables ("WIDE" bus)

Note that the pinouts of the Single-Ended and HVD 68-pin connectors are different; this is similar to that of the Single-Ended and HVD 50-pin connectors. Note too, that the LVD 50-pin and 68-pin connectors use pinouts, shown on the next page, analogous to the Single-Ended versions.

LVD Narrow ("A" cable)

1	D0+	26	D0-
2	D1+	27	D1-
3	D2+	28	D2-
4	D3+	29	D3-
5	D4+	30	D4-
6	D5+	31	D5-
7	D6+	32	D6-
8	D7+	33	D7-
9	DPAR+	34	PAR-
10	GND	35	GND
11	GND	36	GND
12	resrved	37	reserved
13	TRMPWR	38	TRMPWR
14	resrved	39	reserved
15	GND	40	GND
16	ATN+	41	ATN-
17	GND	42	GND
18	BSY+	43	BSY-
19	ACK+	44	ACK-
20	RST+	45	RST-
21	MSG+	46	MSG-
22	SEL+	47	SEL-
23	C/D+	48	C/D-
24	REQ+	49	REQ-
25	I/O+	50	I/O-

LVD Wide ("P" cable)

1	D12+	35	D12-
2	D13+	36	D13-
3	D14+	37	D14-
4	D15+	38	D15-
5	DPARH+	39	DPARH-
6	D0+	40	D0-
7	D1+	41	D1-
8	D2+	42	D2-
9	D3+	43	D3-
10	D4+	44	D4-
11	D5+	45	D5-
12	D6+	46	D6-
13	D7+	47	D7-
14	DPARL+	48	DPARL-
15	GND	49	GND-
16	DIFFSEN	50	GND
17	TRMPWR	51	TERMPWR
18	TRMPWR	52	TRMPWR
19	reserved	53	reserved
20	GND	54	GND
21	ATN+	55	ATN-
22	GND	56	GND
23	BSY+	57	BSY-
24	ACK+	58	ACK-
25	RST+	59	RST-
26	MSG+	60	MSG-
27	SEL+	61	SEL-
28	C/D+	62	C/D-
29	REQ+	63	REQ-
30	I/O+	64	I/O
31	D8+	65	D8-
32	D9+	66	D9-
33	D10+	67	D10-
34	D11+	68	D11-

Figure 16. Example of LVD for NARROW and WIDE bus

When designing a SCSI system, it is recommended that you use good quality cables from a reputable vendor. Always use cable with twisted pairs, whether the flat or the round type. NEVER use the non-twisted type for external cables; its sensitivity to electrical noise, and crosstalk often results in low signal quality and low electrical margin; that is not worth the small financial savings.

SIGNAL CROSSTALK, NOISE IMMUNITY, AND SKEW IN SCSI CABLES

To minimize radiation and crosstalk between signals in the cables, the following rules should be respected when building round SCSI cables:

- Use a three-layer cable with three twisted-wire pairs in the core (inner layer), nine pairs in the middle layer (50-pin cables), and the remaining pairs in the outer layer, all enclosed by a shield and a PVC jacket. Thethreelayers should be capacitively separated from each other by twisting the pairs together in the opposite direction for each layer. The insulation used should be of the Polyolefin type (no PVC!!).

- Place the two most critical pairs, the REQ and ACK, together with ground, in the core of the cable. Use the outside layer for the Data and Parity signals. Place the Control signals in the middle layer to insulate the REQ & ACK pairs from the Data pairs. Note that the Single-Ended and HVD connector pinouts are different, and therefore the optimal cable layout for each will be different. However, you may find that if you interchange the two, the Single-Ended cable will work on the HVD system most of the time (the HVD interface is more forgiving), but the HVD cable may not work on the Single-Ended system.

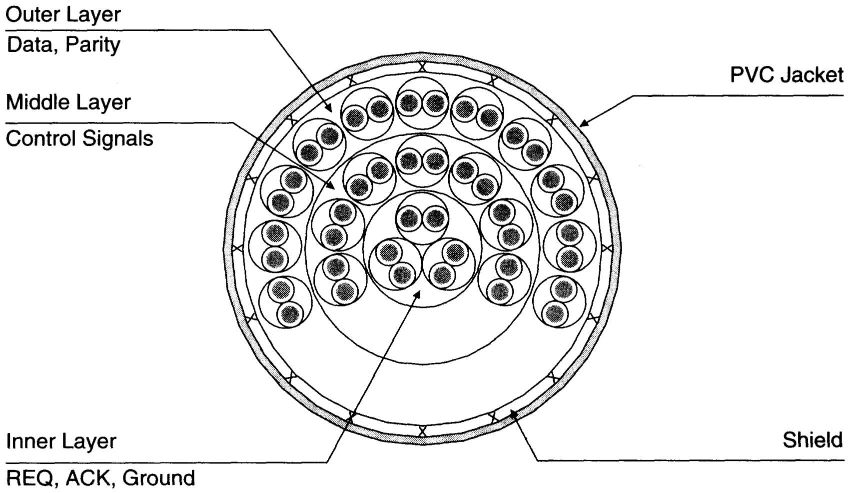

Figure 17. Three-layered cable scheme

Good quality cables should have uniform twisting of the individual wire pairs. Uniform wire length guarantees the same delays for all signals on the bus. If the difference of wire lengths in the cable is much different, the signals will propagate with different delays, which will show as excessive signal skew at the connectors.

The marginal quality of cable not designed using the rules above (high capacitance, crosstalk, skew, ..) will demonstrate itself by intermittent parity errors, and corrupted data. The symptoms will show more at higher transmission speeds and with longer cables. Very often the problems will be blamed on the devices, or even software, rather than the real source. It can never be emphasized enough that the cable is an important and sensitive part of a SCSI system.

SCSI cables are often rigid and heavy, and it is important that the connectors are well secured in their position. The Centronics type 50-pin connectors are generally very robust and wire straps are generally sufficient for hold-down. The 50-pin HD (high density) connectors use either jackscrews (2-56) or spring locks. Most critical are the 68-pin HD "P" connectors, mainly because the 68 wire-pairs make up a rather non-flexible cable. The SCSI specification calls for 2-56 jackscrews, but 4-40 size jackscrews are used too.

CABLING, TERMINATION, AND "TERMPWR"

The cable placement in the system is very critical. Cables should be laid out in a daisy-chained fashion, with no stubs, or stubs shorter than 10 cm, with exactly two terminators located on physical ends of the cable. The terminating circuits protect from reflecting the signals back into the cable. Only the DIFFSENS line is not terminated. Cable termination depends on the electrical connection used. The following illustrations show examples of terminators used on a Single-Ended, HVD, and LVD SCSI bus.

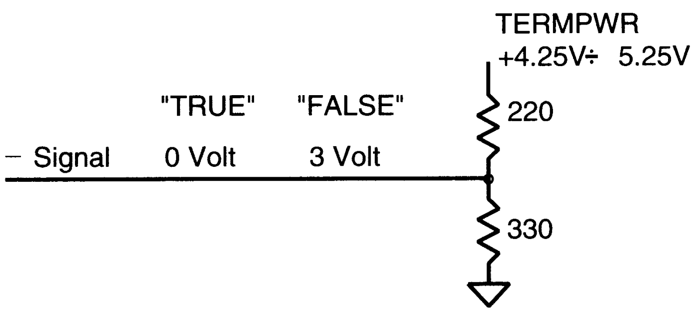

Figure 18. Single-Ended (SE) passive termination

Active termination for Single-Ended is strongly recommended for its high noise immunity. SCSI-3 allows only active termination for the Single-Ended interface. It is permissible to mix the active and passive terminators on the same cable.

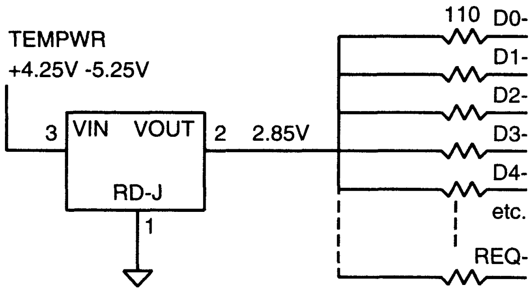

Figure 19. Single-Ended (SE) active termination

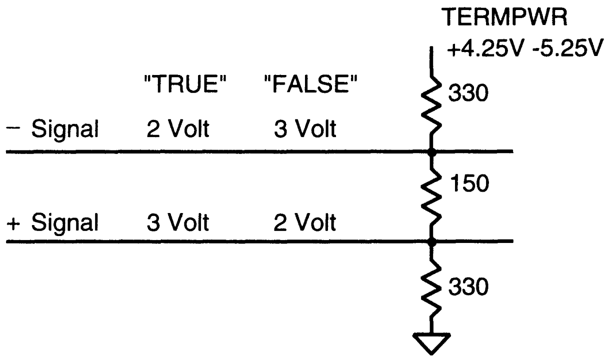

Figure 20. High Voltage Differential (HVD) passive termination

The HVD interface is inherently more immune to noise, so the resistor type termination is generally used. The LVD interface specification calls for active terminators.

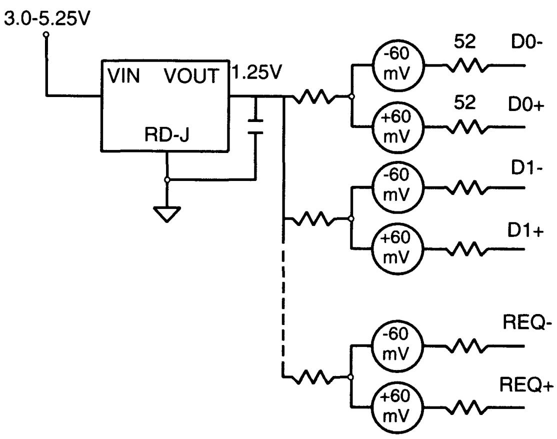

Figure 21. Low Voltage Differential (LVD) active termination

The TERMPWR line is used for providing the power for the terminators. Most often, the devices provide power for their own terminators, but if they don't, the TERMPWR provided by the host adapter will be used. The TERMPWR is specified to be at least 4.25 volts (SCSI-2 specification), but theory and experiments show that the higher the level, the better the noise immunity. Also, it is not sufficient to measure the voltage level at the source. If the cable is longer then you have to consider the voltage drop across the full length of the cable. You may discover that the TERMPWR at the terminator, where it is finally used, is lower than the specified limit.

SCSI CABLE LENGTH & NUMBER OF DEVICES

The maximum cable lengths recommended by the SCSI-2 specification are 6 meters for the S-E, 12 meters for the LVD, and 25 meters for the HVD bus. These lengths are recommended for configurations where you need to achieve the maximum specified performance. However, in many situations where longer lengths are required and maximum performance is not the issue, it is possible to use much longer cables. Especially in low noise environments, or if slower transfer rates are used, and if all design standards are respected.

When connecting several physical cables together, as in daisy-chaining, use cables with the same impedance values, otherwise you may get reflections from the junctions. It is always wise to build up good margin in the system using high quality short cables wherever possible, much shorter than the SCSI Spec allows.

Cable length is a function of speed used and physical properties of the wiring itself. Number of devices and their location (spacing) affects the maximum length for maximum performance. See the table below.

SCSI Version	Peak Burst Rate MBps	S-E Maximum Length	S-E Number of devices	LVD Maximum Length	LVD Number of devices	HVD Maximum Length	HVD Number of devices
SCSI-1	5	6	8			25	8
SCSI-2 Narrow	10	3	8			25	8
Wide	20	22	16			25	16
SCSI-3 Ultra1							
Narrow	20	1.5	8	25	8	25	8
		3	4				
Wide	40	1.5	8	12	16	25	16
		3	4				
SCSI-3 Ultra2							
Narrow	40	*	*	25	2	25	2
Wide	80			12	16	12	16
SCSI-3 Ultra3							
Narrow	80	*	*	25	2	*	*
Wide	160			12	16		

Note: Devices are spaced at least 8" apart
* Not specified
Lengths in meters

Figure 22. Cable length versus number of devices

EXTENDING THE SCSI BUS

Total length of the cable, for maximum performance at the maximum transfer rates of 5 MHz, as recommended by the SCSI-2 specification, is 6 meters for Single-Ended or 25 meters for differential configurations. It is possible to extend this length considerably by using Single-Ended to differential converters.

Several schemes are possible, depending on how much distance is needed and which interface is used on the SCSI devices. See the following examples.

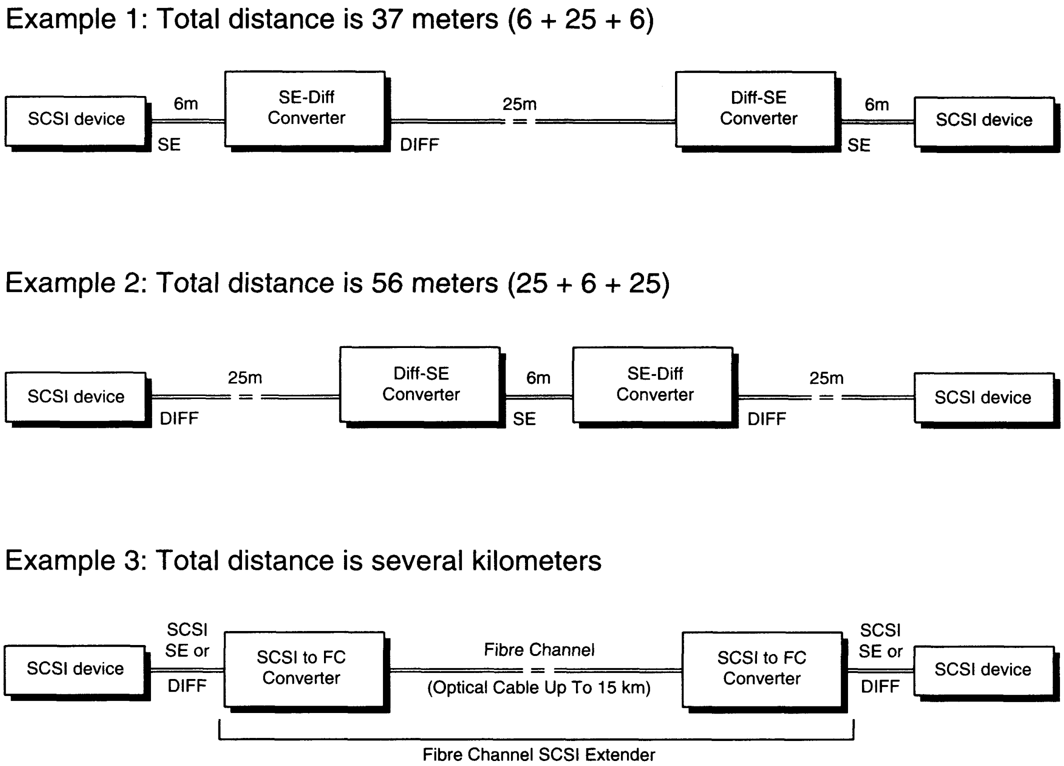

Figure 23. Examples of Extending the SCSI bus

It is possible to use more than two converters in series and thus achieve even greater length. However, delays in each converter accumulate and eventually may affect the Arbitration and Selection phases. You may have no problem in systems which use a single SCSI host on one end and a single SCSI target on the other. The situation becomes more difficult in systems using several targets, especially if these targets use the disconnect feature. The specified timeout is based on maximum delay between devices, and the arbitration phase may fail.

CONNECTING 8-BIT & 16-BIT DEVICES ON THE SAME SCSI BUS

It is possible to mix different size devices on the same SCSI bus cable. Signal routing of the 68-pin (WIDE) "P" and the 50-pin HD "A" connectors is such that the second byte on the 68-pin connector is straddled around the 50-pin layout. For pinouts, see Figures 14 and 15. In order to connect the single-byte devices, you simply connect the 50-pin cable to the middle 50 pins of the 68-pin cable, as shown in the following illustration.

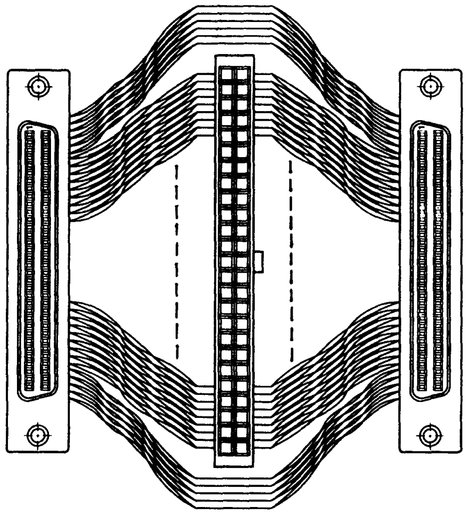

Figure 24. 8-bit and 16-bit devices on the same cable

Be aware of several potential pitfalls, though. Make sure that you maintain correct termination, especially if the single-byte devices are at the end of the cable; with the 50-pin terminators, you are terminating only the 50-pin part and you must not forget to terminate the upper byte for the WIDE devices too. Also remember that capacitance and reflections of the NARROW and WIDE sections of the cable are different due to different geometry, and may cause problems for the 16-bit devices at higher speeds.

Select the ID assignment so that all devices that should work together will "see" each other during arbitration phase, and therefore position all Initiators on the high priority lower byte. Make sure that selection and reselection after disconnect still work; when the single-byte devices are selecting, they will not drive the upper byte and this will result in parity error on this byte.

MIXING SE & DIFF SCSI DEVICES

The Single-Ended, LVD, and HVD interfaces are mutually exclusive. The SCSI bus must use either all Single-Ended, all LVD, or all HVD components. If your host adapter is HVD, then all devices must also be HVD. In situations when you need to connect Single-Ended and HVD devices together on the same SCSI bus, use a Single-Ended to HVD converter.

THE SCAM PROTOCOL

The SCAM (SCSI Configured AutoMatically) protocol, defined in the SCSI-3 parallel standard, enables assignment of SCSI IDs to individual devices automatically, totally under software control. The SCAM protocol has been designed to tolerate and to allow coexistence of legacy devices with hardwired IDs Oumpers) as well as the new SCAM-capable devices on the same SCSI cable.

The SCAM protocol is defined in two levels, or versions:

l . Level-1 SCAM master devices use hard IDs. Only a single level-1 SCAM master is allowed on the bus. The SCAM slave devices must power up before or at the same time with the master. Legacy SCSI devices are allowed. This level protocol can be implemented using most of the currently available SCSI controller chips, starting in 1995.

2. Level-2 SCAM master devices may use hard or soft IDs. There may be multiple level-2 SCAM masters on the bus. The SCAM masters and slaves may power up independently. Legacy SCSI devices are not allowed.

The SCAM ID assignment sequence executes immediately after power-up, or after SCSI reset. The sequence is as follows:

First, the SCAM master (the host adapter) scans the bus for hardwired IDs (the legacy devices), and builds a table of IDs already assigned. The SCAM devices are designed to ignore these short-duration selections.

In the next step, the SCAM master assigns the soft IDs to the SCAM capable devices, using a modified Selection phase (with MSG line asserted by the master), and with no data bits being driven on the data bus. The BSY and SEL stay asserted to hold off the legacy devices. All SCAM devices now "wake up" and participate in the protocol.

The devices drive certain control lines (C/D and I/O); they execute certain sequences on the D5, D6, and D7 lines, and use the lower five data bits for transferring information about ID which they (the slaves) prefer. The slaves then receive ID confirmation from the SCAM master via these low-order bits. The SCAM protocol will assign IDs 0 through 7 on a NARROW bus, and IDs 0 through F hex on a WIDE bus. The soft IDs come from the pool of IDs available following the hard ID scan.

The SCAM protocol executes during the boot process. It may take a few seconds or less, depending on how many devices are on the bus.

For detailed description of the SCAM protocol, we recommend reading the SCSI-3 specification, document: SCSI-3 Parallel Interface "SPI", Annex B.

PLUG AND PLAY

Configuring the SCSI system correctly requires knowledge of certain rules. There are many technicians that do not have access to the important information or simply do not have the patience to study SCSI specifications. To simplify installation for the less technically interested, a set of system level rules and recommendations has been developed. It is called Plug-and-Play design profile, or simply "PnP".

PnP is meant to be a complement to the SCSI standard, which enables mostly automatic configuration of SCSI systems. It describes rules for behavior of SCSI host adapters, peripherals, subsystems, and their connections.

In short, the following are the characteristics of PnP: SCSI disks, tapes and CD ROMs are included. A limited number of cable connector types is used. Automatic (self) termination and the automatic TERMPWR scheme is used. SCSI IDs are assigned automatically, using the SCAM protocol. Rules for executing Synchronous Data Transfer (SDTR message) negotiation are outlined.

SYSTEM ASPECTS

TRANSFER RATES AND THROUGHPUT

What are the maximum transfer rates, or what throughput can we expect from a SCSI system? Let's start with some history to put SCSI in perspective.

Before SCSI started, the ST506 interface, introduced by Seagate Technology on their Winchester disk drives, was considered "high performance". The ST506 could achieve 600 KBps (KiloBytes per second) as a maximum, but in practical applications was mostly interleaved 2 to 1, therefore it would practically do only 300 KBps. In those days, the early 1980s, the closest competitor was the floppy disk interface running about 60 KBps.

SCSI does not dictate the maximum speed: the asynchronous protocol allows dynamic speed adjustments and speed matching to the slowest device. SCSI-1 allowed data transfer burst rates up to 5 MBps synchronous, but almost all implementations used asynchronous protocol at rates around 1 MBps, with some up to 2 MBps. It was not until the late 1980s that SCSI-2 became popular, and more systems began to use synchronous protocol at up to 5 MBps. By 1990, practically all magnetic disk drives and SCSI host adapters sold supported the synchronous protocol at 4 to 5 MBps rates. However about the same time, designers "discovered" the FAST mode (up to 10 MHz or 10 MBps on the Narrow bus), and finally the new SCSI chips started supporting it. As a result, early in 1992 most magnetic disk drives supported the 10 MHz FAST mode. About the same time, yet another performance enhancement option started appearing - the SCSI-2 two-byte WIDE option. If you combine the FAST (10MHz) and WIDE (2-bytes), the transfer rate can be up to 20 MBps. By the end of 1992 we started seeing magnetic disk drives with 20 MBps transfer rates. In late 1995 the SCSI protocol chips came out supporting FAST/20 and WIDE with transfer rates up to 40 Mbps. And finally in late 1997 there were several SCSI protocol chips and several SCSI disks supporting FAST/40 WIDE allowing transfer rates at 80MBps and now we are starting with FAST/80 WIDE allowing transfer rates of 160MBps. SCSI-2 allows up to a four-byte WIDE bus, but so far we did not see such an implementation.

Remember that in this section we discussed the data transfer burst rates; however, there is more to performance than that. Other factors that affect performance include the fact that the FAST and WIDE features do not apply to the non-data phases within the command, that there is overhead associated with arbitration and selection, and that there is overhead in the host computer itself (the operating system and the hardware). The only meaningful measure for system evaluation is not the data burst rate, although it looks impressive on the data sheets; rather it's the sustained real throughput, which includes all overhead components, from the SCSI device itself, the host bus adapter, host hardware, the I/O driver and the operating system. The negative impact of overhead can, to a degree, be reduced by transferring longer data blocks in one command.

MORE ABOUT SPEED AND PERFORMANCE

As mentioned earlier, performance on new computer systems is increasing all the time, and with it, the required speed of data transfers constantly increases. This latter increase is an attempt to keep up with the faster processors and their insatiable demand for data.

It is important to realize that the raw speed of transfer during the Data phase is not the whole story, as mentioned in the preceding section. The Data phase is a small part of a SCSI command, and therefore a SCSI system which is a little slower in the data phase, but capable of speeding up the other phases, can be much faster than a system which blazes through the data phase, but is slow in the other phases. To illustrate this, we may consider the non-data phases as overhead in relation to the data phase. In real implementations we may be surprised to find out that SCSI systems have an overhead of 30% to 50% or even more.

When looking for a SCSI device, the burst data transfer rate is important, but more important is the sustained data transfer rate, which includes all the overhead for single-block reads and writes. The newest SCSI protocol controllers are reducing overhead by putting more and more intelligence in the controller, adding cache buffers, and relieving the central processor of most of the responsibility for smooth command execution. At the same time, the dedicated controllers can be optimized for the SCSI protocol, anticipating phase changes and therefore minimizing the overhead. The result is a significant reduction in overhead, and an order-of- magnitude improvement in sustained data transfer rate.

When designing an efficient computer system, it is important to use comparable components; it does not help to have a very fast device (Initiator) on one end, if it communicates with a slow device at the other end (Target). The slower member of the pair determines the overall speed of the whole. For optimum performance, the devices attached to the bus should be approximately the same level of speed.

The sustained throughput is one of the factors to consider. In computer systems the question of performance is more complex. Depending on characteristics of a peripheral device and loading (traffic) on the system bus (CPU bus) it may be important to insure that one device does not hog the system bus, and lock other peripherals out. For example, a relatively slow tape drive with slow SCSI data transfer rate should use a fast cache memory in order to minimize its time on the SCSI bus and allow the rest of the system to operate smoothly.

When considering overall system performance, an important feature in SCSI is the ability to DISCONNECT and RECONNECT. This feature allows slow operations, like seeks in disk drives, to be executed offline. Using disconnect and reconnect, several operations in a system may progress concurrently.

SCSI-2 added another feature to enhance performance: COMMAND QUEUING. This again may play an important role in fast systems using relatively slow peripherals. If the peripheral device uses command queuing, it allows the host to download several commands, and execute them later in an optimized sequence controlled by the Target itself. The execution sequence is optimized from the peripheral device standpoint. For example, several Reads may be reordered to execute in a sequence to minimize the seek time.

EVOLUTION OF SCSI

The introduction described briefly how SCSI started. This section discusses how SCSI evolved, and the differences between SCSI-1, SCSI-2, and SCSI-3.

Before the 1980s, there were mainly two categories of computers: one included big mainframes and mini-computers, and the other was made up of microcomputers, starting with the introduction of the microprocessor in 1974. The capabilities of microcomputers of that day were limited; they used an 8-bit microprocessor, 64K of main memory, and two floppy disks for external storage. After the first small system compilers, spreadsheets and other applications were developed, and with them came the need for higher performance processing and for mass storage. The first Winchester disk drive was introduced with a maximum transfer rate of 600 KBps and up to 10-MByte capacity. The Seagate ST506 scheme, used as a dedicated interface, was not satisfactory for long. With the need for higher performance, a universal interface like SCSI became a requirement.

SCSI-1

The first work on SCSI started around 1980. At that time expectations for microcomputers were very limited, and therefore only magnetic disks were considered. The SCSI bus was originally used with only two devices: the host bus adapter (SCSI Initiator) and the disk itself (SCSI Target). Arbitration was not needed or used, since there were no other devices to arbitrate against. The computer hardware was much smaller, and the disk was built into the main enclosure together with the processor. The cable was very short and there were no EM radiation problems. Parity protection was not needed and was not used. When the first standard was approved, it was named SCSI, but I ater it became known also as SCSI-1 to distinguish it from the future versions.

MAIN CHARACTERISTICS OF A TYPICAL SCSI-1 SYSTEM:

- very short bus (1 meter or less) with only two SCSI connectors/devices
- single Initiator and single Target
- transfer rates of 1 MBps or less
- asynchronous mode only
- no arbitration, no Parity, no Disconnect
- Single-Ended interface
- 6-byte CDBs
- very high overhead (80% or even higher)

From today's standpoint SCSI-1 may appear as low performance. It is important to remember that SCSI-1 was competing against the ST506 interface, typically running at 300 KBps or slower, often interleaved 2:1 or 3:1. SCSI represented a significant improvement at that time.

One problem with SCSI-1 became obvious very early; the standard was too permissive, and allowed too many "vendor specific" options. Implementations varied and there was a serious interoperability problem between products from different vendors. When the SCSI was finally approved in 1986, the Common Command Set (CCS) was proposed as an attempt to resolve the compatibility problems. This was a subset of the standard which did not allow exceptions. The first version of SCSI was already out however, so now we had SCSI-1 and CCS.

SCSI-2

Work on the new standard, SCSI-2, started in 1986. The original idea was to merge the CCS and SCSI-1, and add a "few improvements". It turned out that the SCSI-2 specifications document grew to more than double the size of SCSI-1. The new draft was submitted to ANSI for approval in 1990, and was finally released in January 1994, as revision "10L" with almost 500 pages.

WHAT ARE THE MAIN IMPROVEMENTS IN SCSI-2?

There are several, and all are important. Here is the list:

- much higher performance than SCSI-1
- transfer rate increased to 10MHz, routinely using 5MHz
- lower overhead, often below 30%
- synchronous mode
- Single-Ended (SE) and differential (HVD) interfaces
- the SCSI bus can be up to 4 bytes wide (SCSI-1 was only a single byte). Although the 4-byte width did not gain popularity (the cabling is so awkward that it is almost unusable), WIDE systems today use the 2-byte wide bus and are achieving 20 MBps burst transfer rates. diversity of peripheral device types (10 types)
- greatly improved compatibility
- possibility of building more complex configurations using a mix of peripherals with differing functions and performances, all on the same SCSI cable
- improved functionality: execution of higher intelligence functions, and executing them concurrently (Disconnect, Command Queuing)
- improved reliability (Arbitration, Parity, Error reporting & classification, new commands and extended message system added)

In the early nineties, SCSI-2 graduated to a rather powerful interface, and it is now being used not only on small systems; its main application is actually in the arena of workstations and big systems. At the same time, SCSI-1 is becoming outmoded.

The transition from SCSI-1 to SCSI-2 was gradual and relatively painless, since compatibility was preserved.

even higher performance. Burst rates must be increased and overhead must be lowered.

For all these reasons, the ANSI standards committee started work on a new version of SCSI, which would not only satisfy today's requirements, but would also allow fast and extensive growth in the future. This new standard is called SCSI-3. It defines two kinds:

1. SCSI-3 Parallel interface
2. SCSI-3 Serial interface

SCSI-3 Parallel is an extension to SCSI-2 (at one time a proposal was made to call it SCSI-2.5). The parallel version allows upward migration of the SCSI-2 designs.

SCSI-3 Serial represents the new generation. It is being designed to achieve all the requirements listed below, including high performance and compatibility with networks.

WHAT ARE THE MAIN CHARACTERISTICS OF SCSI-3 ?

SCSI-3 PARALLEL:

- much higher performance in respect to burst transfer rates (FAST/20, FAST/40,FAST/80,..)

SCSI-3 SERIAL:

- much higher performance
- network compatibility
- distance in kilometers, rather than just meters
- expanded addressability, allowing an almost unlimited number of devices
- physical layer is scalable over media types, transfer rates, distance, other protocols besides SCSI
- higher reliability and robustness of the system

HOW CAN SCSI-3 ACHIEVE ALL THIS ?

- it uses serial interconnect as an alternative to the traditional parallel interconnect. The serial interface is the very high performance version, which will be used almost exclusively. The parallel version allows back compatibility with the large SCSI-2 installed base.
- SCSI-3 Serial is scalable over various media types. The media can be optical fiber, copper coax, twisted pair, or wireless.
- packetized protocol with very low overhead.
- transfer rates will not be limited by the protocol. SCSI-3 Serial will be able to utilize faster media as they become available.
- SCSI-3 Serial is scalable over distance. The protocol does not change with distance, and there is not a strictly defined timeout.
- SCSI-3 Serial is using a more efficient and reliable new communication protocol named FIBRE CHANNEL in place of the traditional SCSI parallel physical layer.

The SCSI command repertoire remains; it is being expanded, improved, and extended. The SCSI-3 Serial hardware is different, but software requires only small changes. The main investment in software is being carried forward.

In 1993, three proposals were considered for SCSI-3 Serial as the carrier

- the Fibre Channel protocol
- the Serial Storage Architecture (SSA)
- the 1394 protocol (also called "FireWire")

Three alternatives were proposed because SCSI became a universal interconnect, and it is trying to serve diverse interests. For example, there are conflicting demands on computer systems from applications like data processing and multimedia. Data processing requires error-free data delivery. Multimedia requires delivery of data on time, and the error-free requirement is secondary.

Out of the three candidates, Fibre Channel became the winner for its superior performance and versatility. It is a network protocol that can be used for channel applications with equal efficiency. The SSA has been limited to lower performance levels only and will not be expanded further. The 1394

Will SCSI-2 be replaced by SCSI-3 Serial, or even become obsolete? It appears now that SCSI-3 Serial will have its own market, mainly in high performance workstations and other bigger and more expensive computer configurations. There is a good chance that the SCSI-2 and SCSI-3 parallel will keep their current market for some time without any radical changes. According to the latest market research studies, the SCSI-3 parallel will continue to grow through the early 2000s. In any case, we do not expect that SCSI-3 Serial would suddenly replace the SCSI-2 market, as it happened between SCSI-2 and SCSI-1 not so long ago.

Which applications will be using the SCSI-3 Serial over Fibre Channel? In 1995, the main use was for channel interfacing, i.e., host to peripherals rather than for networks. There is a simple explanation for this. Fibre Channel was proposed and developed by the channel developers. Recently we are starting to see interest developing in the network industry in using Fibre Channel as a backbone interface for LAN, carrying IP protocol. We expect that the coming years will be exciting times for the acceptance of Fibre Channel as the protocol of choice for both, channels and networks, carrying SCSI-3 Serial, IP, and possibly other protocols.

FIBRE CHANNEL PROTOCOL

Fibre Channel is a broadband communications protocol developed for interfacing within computer systems (host-to-peripherals), recently called Storage Area Networks (SAN or STAN), as well as interfacing between computer systems, such as supercomputers, workstations, or desktop PCs, called networks. It is a carrier protocol that can transport channel and network traffic equally efficiently.

MAIN CHARACTERISTICS

Transfer rates:	Specified from 133 Mbps up to 16 Gbps (gigabits per second). As of today, the 1 Gbps rate is being used almost exclusively.
Efficiency:	6% in protocol overhead and 20% in encoding overhead. The 1.063 Gbps yields 100 Mbps sustained net data throughput.
Addressing:	Over 16 million (switched), or 126 (Arbitrated Loop)
Distance:	500m (50-micron multimode), or 10km (single-mode) between hops at 1 Gbps.
Topologies:	Switched (fabric) Arbitrated Loop Point-to-Point
Media:	Optical fiber, copper coax, twisted pair
Classes of service:	Circuit switched or frame switched, datagram, isochronous service (voice & video)
Versatility:	FC is a carrier for both Channels & Networks
Frame size:	Variable, up to 2148 bytes, including a payload of 0 to 2112 bytes.
Protocol type:	“Lossless” also called “reliable” carrier type (called classes of service).
Cost:	Lower cost if sued with Arbitrated Loop topology with copper; higher cost if used with switched topology for very high performance.
Other:	8B/10B encoding, (running) disparity, error detection, buffer-to-buffer and end-to-end credit flow control.

SOURCES OF INFORMATION

PUBLICATIONS

1. Basics of SCSI - fourth edition by Ancot Corporation (1998)
A gentle introduction to SCSI and an overview. Available directly in Germany from: Manhattan Skyline GmbH, Wiesbadenerstr. 5a, D-65232 Taunusstein

2. The Book of SCSI by Peter M.Ridge (1995)
Co-authors: D.A. Deming, S.Groll, J.Heim, G.Islinger, J.Lohmeyer Publisher: No Starch Press, Daly City, California 94014 A general introduction to SCSI operation, its hardware, installation in PCs, and troubleshooting available at many bookstores

3. SCSI-1 Specifications
Doc.# X3.131-1986
Title: American National Standard for Information Systems
Small Computer System Interface (SCSI)

4. SCSI-2 Specifications
Doc.# X3.131-1994 Rev: 10L, approved Jan 31, 1994
Title: American National Standard for Information Systems,
Small Computer System Interface-2

5. SCSI-3 Documents:
SCSI-3 Architecture Model (SAM)
SCSI-3 Primary Commands (SPQ
SCSI-3 Block Commands (SBC)
SCSI-3 Stream Commands (SSC)
SCSI-3 Enclosure Services
SCSI-3 Graphic Commands (SGC)
SCSI-3 Medium Changer Commands (SMC)
SCSI-3 Controller Commands
SCSI-3 Multimedia Commands
SCSI-3 Interlocked Protocol (SIP)
SCSI-3 Parallel Interface (SPI)
SCSI-3 Serial Bus Protocol (SBP)
SCSI-3 Fibre Channel Protocol (FCP)
Fibre Channel Physical and Signaling Interface (FC-PH)

6. COMMON ACCESS METHOD (CAM) SPEC:
SCSI-2 Common Access Method Transport and SCSI Interface Module (CAM) Doc.# X3T9.2/90-186 Rev 3.0

7. The SCSI Encyclopedia, by Jeffrey D. Stai, published by ENDL (1991)
A detailed description of basic and advanced operations of SCSI-1 and SCSI-2. Available at many bookstores, or directly from: ENDL Publications, 14426 Black Walnut Court, Saratoga, California 95070 Tel: (408)867-6630

8. SCSI Interconnection Guide Book, by AMP Inc. Catalog #65237.
A good description of all types of SCSI connectors, pinouts, cabling schemes, and cable adapters, with many photos, illustrations and drawings. Available directly from AMP Inc., Harrisburg, PA 17105. Tel: (800)522-6752

9. What is Fibre Channel?, by Ancot Corporation (1997)
Introduction and a brief overview of the protocol. Available directly from: Ancot Corporation, 115 Constitution Drive, Menlo Park, California 94025 Tel: (650)322-5322 Fax: (650)322-0455 Internet: www.ancot.com

10. Fibre Channel - The Basics, by G. Stephens & J. Dedek

Available directly from: Ancot Corporation, 115 Constitution Drive, Menlo Park, California 94025 Tel: (650)322-5322 Fax: (650)322-0455 Internet: www.ancot.com

11. Fibre Channel Documents:

FC-PH	Fibre Channel Physical Interface
FC-IG	Implementation Guide
FC-FP	Mapping to HIPPI-FP
FC-LE	Link Encapsulation
FC-SB	Single Byte Command Code Sets
FC-FCP	SCSI-3 Fibre Channel Protocol
FC-FG	Generic Requirements
FC-SW	Switched Fabric
FC-AL	Arbitrated Loop
FC-GS	Generic Services
FC-AE	Avionics
FC-AV	Audio-Visual

SCSI Specifications, items 3, 4, 5, 6, and 11, are available from:
AMERICAN NATIONAL STANDARDS INSTITUTE, Inc.
I I West 42nd Street, New York, NY 10036 (212)642-4900
or from:
GLOBAL ENGINEERING DOCUMENTS
3130 South Harbor Blvd, Suite 330, Santa Ana, California 92704
Tel: (800)854-7179

ANSI STANDARDS

The American National Standards Institute is a non-government organization which publishes standards in the USA. Its subcommittee T10 has a charter to develop Lower Level Interfaces, mainly the Small Computer System Interface.

For information on joining a committee, contact:

NCITS Secretariat, National Committe for Information Technology Standards 1250 Eye Street NW, Suite 200, Washington DC 20005-3922 Tel: (202)737-8888 (press I twice) or (202)626-5741, Fax: (202)638-2829

SCSI Bulletin Board: Tel: (719) 574-0424 ftp site: ncinfo.ncr.com

SCSI TRADE ASSOCIATION (STA)

The STA, founded in February 1996, is an association comprised of companies with the following objectives:

- promote the understanding and use of the SCSI Parallel Interface technology
- provide a focal point for communicating SCSI benefits to the market
- direct the evolution and growth of SCSI Parallel Interface technology into the twenty-first century.

THE SCSI TRADE ASSOCIATION (STA) 404 Balboa Street, San Francisco, California 94118 Tel: (650)750-8351 Fax: (650)751-4829 Info@scsita.org www.scsita.org

SCSI Glossary

ANSI

American National Standards Institute. A standards-setting, non-government organization, which develops and publishes standards for voluntary use in the USA. ANSI is located at 1430 Broadway, New York, NY 10018 (212)642-4900.

arbitration

Process of selecting one respondent from a collection of several candidates that request use of the SCSI bus concurrently.

Architecture

Refers to the way a system is designed and how the components are connected with each other. There are computer architectures, network architectures, and software architectures.

ASCII

American Standard Code for Information Interchange. Pronounced *"as-kee."* The most popular coding method used by small computers for converting letters, numbers, punctuation and control codes into digital form. Once defined, ASCII characters can be recognized and understood by other computers and by communications devices. ASCII represents characters, numbers, punctuation marks or signals in seven on-off bits. For example: capital "C" is 1000011, number "3" is 0110011, etc.

asynchronous transmission

Transmission in which each byte of the information is synchronized individually, using interlocking the REQ and ACK signals.

bit

Binary digit, with a value of 0 or 1. It is the smallest unit of data a computer can process. Bits are arranged into groups of eight called bytes. A byte is the equivalent of one character.

block

An amount of data moved or addressed as a single unit; the smallest amount of data that can be read or written at a time. Blocks are separated by physical gaps, or identified by their track/sector addresses or logical addresses.

buffer

Device or allocated memory space used for temporary storage. Printers commonly use buffers, for example, to hold incoming text because the text arrives at a much faster rate that the printer can output.

bus

1. a collection of unbroken signal lines that interconnect computer modules.
2. a facility for transferring data between several devices located between two end points, only one device being able to transmit at a given time.
3. a signal path or line shared by many circuits or devices. Information is often sent to all devices throughout the same bus; only the device to which it is addressed will accept it.

bus free phase

The phase when no SCSI device is actively using the SCSI bus, and the bus is available for use.

Byte

An 8-bit (octet) construct, a common unit of computer storage. Generally, eight bits, which equal one character.

cache

A small portion of high speed memory used for temporary storage of frequently used data. Reduces the time it would take to access that data, since it no longer has to be retrieved from the physical medium. channel a general term for a path on which electronic signals travel.

CCITT

Consultative **C**ommittee **I**nternational **T**elegraph and **T**elephone. An international association which sets worldwide communications standards, recently renamed International Telecommunications Union (ITU).

CCS

Common Command Set. CCS is a collection of 18 commands, which is a subset of SCSI-1. The SCSI-1 specification allowed too many vendor specific features; the CCS was designed to limit choices and thus improve compatibility between SCSI devices from different vendors. CCS became a part of SCSI-2.

CDB

Command Descriptor Block. The 6-byte, 10-byte, or 12-byte structure used to communicate commands from a SCSI initiator to target.

CD or Compact Disk

A standard medium for storage of digital data in machinereadable form, accessible with a laser-based reader. CDs are 4-3/4" in diameter, and are faster and more accurate than magnetic tape for data storage. Faster, because even though data is generally written on a CD contiguously within each track, the tracks themselves are directly accessible.

CD-ROM

Compact Disk Read-Only Memory. A data storage system using Compact Disk as the medium. CD-ROMs generally hold more than 600 megabytes. Character a single letter, digit or punctuation symbol, equivalent to one byte.

character device

A printer or other peripheral device that sends or receives data character by character, rather than in bursts of data.

command

An instruction transferred from SCSI Initiator to SCSI Target, typically containing function codes, an address, flags, and possibly other information. Command queue refers to the queue used to store queued I/O processes.

connect

Function that occurs when a SCSI Initiator selects a SCSI Target to start an operation, or SCSI Target reselects an SCSI Initiator to continue an operation.

contingent allegiance

A condition typically indicated by a CHECK CONDITION status during which a SCSI target preserves sense data (information about failure or a specific condition) to be received by the corresponding SCSI initiator.

control signals

Set of 9 lines used to put the SCSI bus into its different phases. A combination of asserted and negated control signals define the SCSI phases.

CRC

Cyclic Redundancy Check. An error detection scheme in which the block check character is generated by dividing all the data in a transmission block by a predetermined number.

crosstalk

Unwanted transmission of energy from one circuit to another adjacent circuit.

data

According to AT&T Bell Labs: data is arepresentation of facts, concepts or instructions in a formalized manner, suitable for communication, interpretation or processing."

de-skewing

the adjustment made to a group of related signals to compensate for various delays inherent in the system.

device

a single unit on the SCSI bus, identifiable by an unique SCSI address. A SCSI device can act as an Initiator or as a Target.

device

I/O driver a small program that tells the computer's operating system how to communicate with a particular type of peripheral device.

differential interface

An electrical signal configuration using a pair of lines for transfer. On SCSI bus "TRUE" (logical "1") is defined as -SIGNAL, higher than the +SIGNAL line, and opposite polarity for "FALSE." The advantage of differential configuration, compared to single-ended, is in relative higher tolerance for common-mode noise, and little crosstalk when used with twisted pair cables. There are two variants: HVD (High Voltage Differential) and LVD (Low Voltage Differential). HVD allows connections up to 25 meters; LVD up to 12 meters. The disadvantage of HVD is the higher cost of components.

digital

pertaining to data that consists of digits. The use of binary code to record information. "Information" can be text in a binary code like ASCII, or scanned images in a bit-mapped form, or sound in a sampled digital form, etc. Recording information digitally has many advantages over its analog counterpart, mainly ease in manipulation and accuracy in transmission.

disconnect

Action that occurs when a SCSI Target releases control of the SCSI bus, allowing the bus to go to the Bus Free phase.

disk

A round, flat, magnetic or other medium with one or more layers deposited on on the surface where data can be recorded.

disk array / disc array

Combining redundant disk or disc drives for more capacity, speed, or for disaster recovery. Also see RAID.

disk drive

A device containing motors, electronics and other gadgetry for storing (writing) and retrieving (reading) data on a disk. A hard disk drive is one which is generally not removable from the machine. A floppy disk drive accepts removable diskettes.

disk sector

Magnetic disks are typically divided into tracks, each which contains a number of sectors. A sector, also called a block, typically contains a fixed amount of data, such as 512 bytes.

driver

When used in the context of electrical configuration, "driver" is the circuitry that creates a signal on a line. When used in the context of software, "driver" is more often called "device I/O driver".

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-level code, like ASCII, developed by IBM. Supported by some OCR programs.

ECC

Error Correction Code. In the event of read-write error, ECC is a method of recovering a block of data.

EDAC

Error Detection And Correction. An operation that includes all phases of identifying and dealing with data errors, including direct-read-after-write and error-correction codes.

EIA

Electronic Industries Association. A standards organization in the USA specializing in the electrical and functional characteristics of interface equipment.

EPROM

Erasable Programmable Read Only Memory.

erasable optical disk

A type of read/write optical disk that permits the deletion of information and the re-use of previously recorded disk areas.

Ethernet

A local-area network protocol that uses high-speed communications at 10 megabits per second.

FASTSCSI

Also called FAST/10, with asynchronous transmission rate defined in SCSI-2 to be between 5 MHZ and 10 MHZ. It is used in data-in/out phases only. Higher rates are also proposed: FAST/20 (at 20 MHZ), FAST/40 (at 40 MHZ) and up.

FCC

Federal Communications Commission.

fiber optic cable

Cable made from thin strands of glass through which data is transmitted in the form of light pulses. Used for high-speed transmission over medium to long distances.

Fibre Channel

A communications network protocol also used as a carrier for SCSI-3 Serial.

field

The smallest logically distinguished unit of data in a record. For example, "name" is a field, an entire mailing address is a record.

FIFO

First In First Out. Queue handling method that operates on a first-come, first-served basis. Contrast with LIFO.

File

All the data that describes one document or image, and is maintained under a single naming code and stored in a computer or storage medium.

firmware

A computer program or software stored permanently in EPROM, PROM, or ROM.

full duplex

A data communication scheme that permits simultaneous transmission in both directions.

gigabyte

1,073,741,824 bytes. Broadly, one billion bytes, or one thousand megabytes.

half duplex

A data communications mode which permits transmission in both directions, but in only one direction at a time.

hard disk

A storage device that uses a magnetic recording material. Generally, hard disks are fixed inside a PC, but there are removable cartridge versions too. Hard disks store up to hundreds of megabytes.

hard error

An error (in data communications or other) that cannot be repaired by correction schemes.

hard wired

Originally used to indicate a fixed, permanent hardware connection. Now expanded to include a selected option, in hardware or software, which cannot be easily rewired or re-programmed by the user.

head

The device which comes in contact with or comes very close to the magnetic storage device (disk, diskette, drum, tape) and reads and/or writes to the medium, in computer devices.

hex

Short for hexadecimal. A number system with a base of 16, designated by 10 digits and six letters. In hexadecimal notation, the decimal numbers 0 through 15 are represented by the decimal digits 0 through 9 and the letters A through F. A = decimal 10, B = decimal 11, and so forth.

host

A processor, usually consisting of a CPU and memory. Typically, a host communicates with other devices, such as peripherals and other hosts. On the SCSI bus, a host has a SCSI address (SCSI ID).

host adapter

Circuitry that translates between a processor's internal bus and a different bus, such as SCSI. On the SCSI bus, a host acts as an Initiator and a peripheral acts as a Target.

Hz

Abbreviation for Hertz; cycles per second.

HVD

High Voltage Differential interface, in the past referred to simply as Differential.

initialization

To start the process by which a device or system is prepared or prepares itself for normal operation. Usually all parameters are returned to their default values.

initiator

SCSI device (usually a host system) that requests an operation to be performed by another SCSI device, the target.

I/O

Input/ Output

I/O process

Consists of one initial connection and zero or more reconnections, all pertaining to a single command or a group of linked commands.

JBOD

Just-**B**unch-**o**f-**D**isks, also called a disk farm.

jukebox

A device that holds multiple optical discs or magnetic tapes, and one or more disk or tape drives, and which can swap the media in and out of the drive as needed. Same as an autochanger. Also called medium changer, or disc or tape libraries.

Kbyte

Kilobyte, or 1024 bytes. Broadly, one thousand bytes. 16 Kbytes, or 16K, is 16,384 bytes; 64K is 65,536 bytes, etc.

LAN

Local Area Network. High-speed transmissions over twisted pair, coax, or fiber optic cables that connect terminals, personal computers, mainframe computers, and peripherals at distances of about one mile or less.

laser

Light Amplification by Stimulated Emission of Radiation. A device for generating coherent radiation I the visible, ultraviolet, or infrared portions of the electromagnetic spectrum.

laser disc

A disk storage device using laser technology for storing and retrieving data.

laser printer

Printer that uses a beam of light to charge a drum so that it attracts toner, which is transferred to heated paper. LBA Logical Block Address.

LIFO Last In First Out

A queuing scheme whereby the most recent element to be entered is acted on first. Contrast with FIFO.

Low Voltage Differential Signaling (LVDS)

A differential interface that uses a 3.3 volt supply voltage rather than a 5 volt one.

logical

A feature not physically present, but applied by software. Sectors on a hard disk are physically arranged contiguously; logically, sectors may be placed anywhere on a hard disk, requiring a software program to arrange them in the correct order.

logical unit (LU)

A physical or virtual device addressable through a target. A physical device can have more than one logical unit.

LSB

Least **S**ignificant **B**yte or **B**it.

LUN

Logical **U**nit **N**umber, used to identify a logical unit.

LVD

Low **V**oltage **D**ifferential interface

magnetic disk

A flexible or hard-disk medium used to store data in the form of minute local variations in magnetization of the disk surface.

magnetic tape

Storage medium that uses a thin plastic ribbon coated with an iron oxide compound to record data with electrical pulses. Magnetic tape is a sequential storage medium; the next block of data is recorded after the last block. In order to locate a specific block of data, you have to look through the whole tape to find it.

magneto-optic

a high-density, erasable recording method. Similar to magnetic disk and tape recording, but the grains of iron oxide matter are much smaller. A laser heats the grain, which makes it susceptible to magnetic influence. The write head passes over the grain while it is still susceptible. The data can then be read by another laser, whose light is not hot enough to change the grain's polarity. megabyte 1,048,576 bytes. Broadly, one million bytes, or one thousand kilobytes.

MHZ

Megahertz is a million cycles per second. Used as a measurement of data transfer rate.

mil

A unit of linear measure equaling a thousandth of an inch, or 0.0254 mm. For example, 5 mils is 0.005 inches.

modem

Short for modulator-demodulator. Device that allows digital signals to be transmitted and received over analog telephone lines.

mouse

Hand-driven computer input and pointing device.

MSB

Most Significant Byte or Bit.

multimedia

Combining more than one medium for dissemination of information, such as using text, audio, graphics, animation and full-motion video all together. Requires large amounts of bandwidth and processing power.

multi-mode

In SCSI, an interface that can switch between S-E and LVD automatically.

nexus

A connection between a SCSI initiator and target. A relationship that begins with the establishment of an initial connection and ends with the completion of the I/O process.

nibble

Informal term for half a byte; the first four (0, 1,2,3) or last four (4,5,6,7) bits of a byte, are the low nibble and high nibble respectively.

offline

Not controlled by a system, nor communicating with it.

online

Controlled by a system, or available to it.

optical disc

A direct access storage device that is written and read by laser light. Certain optical discs are considered Write Once Read Many, or WORM, because data is permanently engraved in the disc's surface, either by gouging pits (ablation), or by causing the non-image area to bubble, reflecting light away from the reading head.

Erasable optical drives use technologies such as the magneto-optic technique, which electrically alters the bias of grains of material after they have been heated by a laser.

Compact discs (CDs) and laser or video discs are optical discs. Their storage capacities are far greater than those for magnetic media.

optical scanner

Input device that translates human-readable or microform images to bit-mapped or rastered machine-readable data.

output device

Any device by which a computer transforms its information to the "outside world." In general, an output device is a machine that translates machine-readable data into human readable information. Examples: video screens, printers, microform devices.

overhead

In data communications, it is the transmitted data which is not part of the information sent. Overhead deals with control, addressing, and error checking.

packet

A group of bits, packaged together, for transmission purposes. Three principal elements are included in the packet: (1) control information -destination, origin, length of packet, etc.; (2) the data to be transmitted; and (3) error detection and correction bits. Sending data in packets rather than continuous streams offers more efficient use of transmission lines.

parallel

The transmission of bits over multiple wires at one time. Accomplished by devoting a wire for each bit of a byte. Parallel data transmission is very fast, but usually practical only for short distances, typically under 500 feet, because of the need for heavy cable. Most often used in computer-to-printer, scanner-to-computer, and other host-to-peripheral applications.

parity

An extra bit, added to a number, used for checking the accuracy of binary numbers. Odd parity means that the sum of all I's in the number, including its corresponding parity bit, is odd.

peripheral device

A device that can be attached to a host computer, using a SCSI bus for example. Typical SCSI peripherals are disk drives, tape drives, printers, and communication devices.

plotter

An output device for drawing maps or engineering drawings.

port

A connection to a bus. An access point for data entry or exit. A connector on a device to which cables other devices are attached. Also referred to as an adapter. The single-byte SCSI bus (SCSI-I) allows eight ports; the 16-bit SCSI-2 WIDE bus allows sixteen ports.

priority

the ranking of the devices on the SCSI bus during arbitration. The higher numbered IDs have higher priority.

PROM

Programmable **R**ead **O**nly **M**emory

protocol

A convention for data transmission that defines timing, control, formatting, and data representation.

RAID

Redundant Array of Inexpensive, or Independent, Discs. A storage device that uses several magnetic or optical discs working in tandem to increase bandwidth output and to provide redundant backup.

receiver

The circuitry that receives electrical signals on a line.

reconnect

The function that occurs when a SCSI Target reselects a SCSI Initiator to continue an operation. A reconnect operation would typically be preceded by a disconnect. See also "reselect".

record

A record is a group of related data items treated as one unit of information. For example, policyholders's name, address, social security number, etc. Each item in the record is a field.

reselect

Action by a target on an initiator to reestablish a connection, after a disconnect to perform a time-consuming task, such as a disk seek.

reserved

Bits, fields and code values that are set aside for future standardization. In SCSI, reserved bits should equal zero.

ROM

Read Only Memory. Data stored in a medium that allows it to be accessed but not erased or altered.

SAN

Storage **A**rea **N**etwork. Also referred to as STAN, to differentiate it from System Storage Area Network.

SASI

Shugart **A**ssociates **S**tandard **I**nterface**.** Designed by Shugart Associates in 1980, it was the predecessor of SCSI.

SCA

Single **C**onnector **A**ttachment, which uses an 80-pin connector to connect SCSI disk drives to a system. It has contacts for SCSI signals, auxiliary signals and power.

SCAM

SCSI **C**onfigured **A**uto**M**atically, a function in the SCSI-3 parallel standard that enables assignment of SCSI IDs to individual devices automatically, under software control.

scanner

A device that optically senses a human-readable image, and contains software to convert the image to machine-readable code.

SCSI

Small **C**omputer **S**ystem **I**nterface. Pronounced "skuzzy." An industry standard for connecting peripheral devices and their controllers to a microprocessor. The SCSI defines both hardware and software standards for communication between a host computer and a peripheral. Computers and peripheral devices designed to meet SCSI specifications should work together. It can be either SCSI-I, SCSI-2, or SCSI-3.

SCSI-1

The first version defined by ANSI - X3.131-1986. SCSI-1 is largely a subset of SCSI-2.

SCSI-2

Successor to SCSI-1. It is defined by ANSI - X3.131-1992. SCSI-2i s upward compatible from SCSI-1.

SCSI-3

Successor to SCSI-2. SCSI-3 differs from SCSI-1 and SCSI-2 in being designed primarily for communication over a serial media (fiber, copper,..) using a packetized protocol.

SCSI address

Also called SCSI ID#, an octal representation of the unique address assigned to a SCSI device. SCSI ID see SCSI address

Sector

Also called a block. The smallest addressable unit of the track of a magnetic, optical, or other disc. Often contains 512 bytes.

serial interface

Data communications mode in which bits are sent in sequence. Contrast with parallel interface.

signal assertion

The act of driving a signal to the TRUE state.

signal negation

Act of driving a signal to the FALSE state (active negation), or placing the driver in the high impedance condition, and allowing the cable terminators to bias the signal to the FALSE state.

signal release

Act of allowing the cable terminators to bias the signal to the FALSE state, by placing the driver in the high impedance condition.

single-ended interface

An electrical signal configuration using a single line for each signal, referenced to a ground path common to the other signal lines. The advantage of single-ended configuration, compared to differential, is in using half the number of pins, chips, and PC board area. Its disadvantage is higher vulnerability to common mode noise, and limited cable distance, up to 6 meters.

shielding

Protective covering that eliminates electromagnetic and radio frequency interference.

skew

To slant a selected item or delay a certain signal in any direction relative to another reference item signal.

status

In SCSI, one byte of information sent from a target to an initiator upon completion of each command, indicating the failure mode, if any, of the command.

synchronous transmission

Transmission in which the sending and receiving devices operate continuously at the same frequency, and are held in a desired phase relationship by a correction device. For buses, synchronous transmission is a timing protocol that uses a master clock with a clock period and an allowable offset.

tape drive

The machine, actually a collection of devices, that transports, reads from and writes to a magnetic tape.

target

The SCSI device, usually a peripheral, that performs an operation requested by a SCSI initiator.

terabyte

1,099,511,627,776 bytes, from the prefix "tera", which means trillion. Broadly, one trillion bytes, or one thousand megabytes.

termination

Electrical connection at each end of the SCSI bus, composed of a set of resistors, or possibly other components. Its function is to provide a pull-up for open collector drivers on the bus, and also an impedance match, to prevent signal reflections at the ends of the cable. Depending on the internal circuitry, a terminator can use simple resistors ("passive termination"), or a combination of a voltage regulator and resistors ("active terminator"). Terminators for single-ended and differential interfaces are different and are not interchangeable.

twisted pair

Type of cable consisting of two wires twisted together. It is excellent and inexpensive for high-speed transmissions over short distances.

UltraSCSI

Another name for SCSI-3/FAST-20. See also FAST-SCSI.

WORM

Write **O**nce **R**ead **M**any. Optical storage device on which data is permanently recorded. Data can be erased, but not altered, and no additional data can be added.