# SCSI Interface

## EXPLORE SUMMER 2006

Submitted By

**GROUP 04**


| | |
|---|---|
| Nabeel ur Rehman | 2003-NUST-BICSE-090 |
| Asad Asif | 2004-NUST-BICSE-116 |
| Junaid Iqbal | 2003-NUST-BICSE-076 |


Submitted To

Mr. Aamir Jelani

# Contents


| Title | Page |
|---|---|
| **List of Figures** | 4 |
| **List of Contents** | 5 |
| **Abstract** | 6 |
| **Technical Documentation** | 7 |
| 1. Overview and history | 8 |
| 2. SCSI Standards | 10 |
| • SCSI 1 | 11 |
| • SCSI 2 | 12 |
| • SCSI 3 | 13 |
| o SCSI-3 Architecture | 14 |
| o SCSI-3 Parallel Interface (SPI) | 17 |
| o SCSI(-3) Parallel Interface - 2 (SPI-2) | 18 |
| o SCSI(-3) Parallel Interface - 3 (SPI-3) | 19 |
| o SCSI(-3) Parallel Interface - 4 (SPI-4) | 20 |
| o SCSI(-3) Parallel Interface - 5 (SPI-5) | 20 |
| o iSCSI | 20 |
| o Serial SCSI | 21 |
| 3. SCSI Transfer Mode and Feature Set Compatibility | 22 |
| 4. SCSI Protocols and Interface Features | 23 |
| • Single-Ended (SE) and Differential (High Voltage Differential, HVD) Signaling | 23 |
| • Low-Voltage Differential (LVD) Signaling | 25 |
| • SCSI Bus Width | 27 |
| • SCSI Bus Speed | 27 |
| • Bus Parity and Cyclic Redundancy Checking (CRC) | 29 |
| • Command Queuing and Reordering | 29 |
| • Negotiation and Domain Validation | 30 |
| • Quick Arbitration and Selection (QAS) | 31 |
| • Packetization | 32 |
| • SCSI Protocol Map | 32 |
| • Summary of SCSI Protocols and Transfer Modes | 33 |
| 5. SCSI Host Adapter | 35 |
| • Adapter Types and PC Bus Connections | 36 |
| • Protocol Support | 37 |
| • Signaling Type Support (SE,HVD,LVD) | 37 |
| • Connectors | 38 |
| • Caching | 38 |
| • Multiple Segment and Channel Support | 39 |
| • RAID Support | 40 |
| • Drivers and Compatibility Issues | 40 |
| • Manual vs. Automatic Configuration | 40 |
| • Resource Usage | 41 |
| 6. SCSI Cables, Connectors and Bus Termination | 42 |

- General Cable and Connector Issues 42
- SCSI Cable Types 43
- SCSI Connector Types 45
- SCSI Bus Termination 48
- Summary of SCSI Cables and Connectors 50

7. SCSI command protocol 52
8. Glossary 54

**Special Interest Areas** 55
**Ongoing Research** 56
**References** 57
**Conclusion** 59

# List of Figures


| Figure No | Figure Name | Page |
|---|---|---|
| 01 | T10's approved SCSI Project Family | 14 |
| 02 | Icons for SE and HVD SCSI | 25 |
| 03 | Icons for LVD and LVD/SE SCSI | 26 |
| 04 | A PCI-based Wide Ultra2 SCSI host adapter | 35 |
| 05 | Internal and external Wide Ultra2 SCSI host adapter connectors | 38 |
| 06 | Close-up view of an external SCSI cable | 44 |
| 07 | Various internal ribbon cables | 45 |
| 08 | A male DD-50 SCSI connector | 46 |
| 09 | Male and female 50-pin Centronics connectors | 46 |
| 10 | 50-pin and 68-pin external high density connectors | 47 |
| 11 | 68-pin VHDCI connector | 47 |
| 12 | Male and female 50-pin regular density internal connector | 47 |
| 13 | Internal male high density 68-pin connector | 48 |
| 14 | Female 80-pin SCA connector | 48 |
| 15 | Internal and external active terminators | 49 |

# List of Tables


| Table no | Table title | Page |
|---|---|---|
| 01 | SCSI command standards | 15 |
| 02 | SCSI Protocols | 16 |
| 03 | SCSI Interconnects | 17 |
| 04 | SE and differential SCSI cable lengths | 24 |
| 05 | Parallel SCSI bus speeds | 28 |
| 06 | Bus width and speeds for various SCSI transfer modes and feature sets | 32 |
| 07 | SCSI transfer modes and feature sets | 33 |
| 08 | Connector types for internal and external SCSI cables | 51 |
| 09 | Features and benefits of SAS devices | 55 |

# Abstract

In today's world, the demand for an increase in computing performance has reached new heights. Various technologies have evolved from the time of the birth of computing and some of them still exist and are in use today while others have been lost to time and technologically better successors. SCSI is one such technology which has developed over time and is quite an important part of the computing world today. Nearly all highly demanding applications utilize the SCSI interface to accomplish their task and increase their efficiency. SCSI is most often used for data storage applications. However, it can virtually be used for any I/O application since it is an interface first of all.

The technical documentation covers various aspects of the SCSI interface but focuses specifically on the SCSI Parallel Interface. Serial SCSI has occasionally been mentioned but has not been discussed in depth.

# Technical Documentation

# Overview and History

What we currently know of as the *Small Computer System Interface* (SCSI) had its beginnings back in 1979. Shugart Associates, led by storage industry pioneer Alan Shugart (who was a leader in the development of the floppy disk, and later founded Seagate Technology) created the *Shugart Associates Systems Interface* (SASI). This very early predecessor of SCSI was very rudimentary in terms of its capabilities, supporting only a limited set of commands compared to even fairly early "true" SCSI, and rather slow signaling speeds of 1.5 Mbytes/second. For its time, SASI was a great idea, since it was the first attempt to define an intelligent storage interface for small computers. The limitations must be considered in light of the era; we are talking about a time when *8" floppy drives were still being commonly used*.

Shugart wanted to get SASI made into an ANSI standard, presumably to make it more widely-accepted in the industry. In the same time period, NCR Corporation's Peripherals division (now Engenio), was developing a more sophisticated product called BYSE (for Byte Serial), and was developing an ASIC to implement it. In late 1981, Shugart Associates teamed up with NCR Corporation, whose management decided it could not support a proprietary interface so it abandoned BYSE in favor an improved SASI and together, they convinced ANSI to set up a committee to standardize the interface. In 1982, the X3T9.2 technical committee was formed to work on standardizing SASI. A number of changes were made to the interface to widen the command set and improve performance. The name was also changed to SCSI, probably because having Shugart Associates' name on the interface would have implied that it was proprietary and not an industry standard. The first "true" SCSI interface standard approved by ANSI was published in 1986 (X3.131-1986), and evolutionary changes to the interface have been occurring since that time.

It's important to remember that *SCSI is, at its heart, a system interface*, as the name suggests. It was first developed for hard disks, is still used most for hard disks, and is often compared to IDE/ATA, which is also used primarily for hard disks. For those reasons, SCSI is sometimes thought of as a hard disk interface. However, SCSI is not an interface tied specifically to hard disks. Any type of device can be present on the bus, and the very design of SCSI means that these are "peers" of sorts - though the *host adapter* is sort of a "first among equals". SCSI was designed from the ground up to be a high-level, expandable, high-performance interface. For this reason, it is frequently the choice of high-end computer users. It includes many commands and special features, and also supports the highest-performance storage devices used in the high performance server market.

Of course, these features don't come for free. Most PC systems do not provide native, "built in" support for SCSI the way they do for IDE/ATA, which is one of the key reasons why SCSI isn't nearly as common as IDE/ATA in the PC world. Implementing SCSI on a PC typically involves the purchase of a storage device and a special card called a host adapter. Special cables and *terminators* may also be required. All of this means that deciding between SCSI and IDE/ATA is an exercise in tradeoffs.

SCSI began as a parallel interface, allowing the connection of devices to a PC or other systems with data being transmitted across multiple data lines. Today, parallel or "regular" SCSI is still the focus of most SCSI users, especially in the PC world. SCSI itself, however, has been broadened greatly in terms of its scope, and now includes a wide variety of related technologies and standards, as defined in the SCSI-3 standard.

# Standards

There was a time that SCSI standards were relatively few and not that difficult to understand. That time is now long past. In some ways, the best way to describe the current situation regarding SCSI standards, feature sets and marketing terms is that it makes the standards and terms associated with IDE/ATA seem simple by comparison. Understanding all of the documents and labels associated with SCSI can be very baffling at times.

It's not that the standards are poorly written, or that the technology is all that hard to understand. The main issue with SCSI today is that it has become so broad, and includes so many different protocols and methods, that it's hard to get a handle on all of it. The confusion surrounding SCSI standards has increased since the creation of SCSI-3, which is really a collection of different standards, some of them rather different from each other. The situation has been made worse by manufacturers who like to create funky new unofficial names and terms for transfer modes or feature sets, or apply overly-broad labels to specific hardware.

As stated before, the first organization that was charged with developing the first SCSI standard was ANSI technical committee X3T9.2. Today, SCSI standards are developed, maintained and approved by a number of related organizations, each playing a particular role. Here's how they all fit together:

- **American National Standards Institute:** ANSI is usually thought of as an organization that develops and maintains standards, but in fact they do neither. They are an oversight and accrediting organization that facilitates and manages the standards development process. As such, they are the "high level management" of the standards world. They qualify other organizations as *Standards Developing Organizations* (SDO). ANSI also publishes standards once they have been developed and approved.
- **Information Technology Industry Council:** ITIC is a group of several dozen companies in the information technology industry. ITIC is the SDO approved by ANSI to develop and process standards related to many computer-related topics.
- **National Committee for Information Technology:** NCITS is a committee established by ITIC to develop and maintain standards related to the information technology world. NCITS was formerly known under the name "Accredited Standards Committee X3, Information Technology", or more commonly, just "X3". It maintains several sub-committees that develop and maintain standards for various technical subjects.
- **T10 Technical Committee:** T10 is the actual technical standards committee responsible for the SCSI interface.

Just to avoid any confusion, T13 technical committee develops ATA standards and T10 and T13 are sibling committees.

What all of this boils down to is that T10 is the group that actually does the work of developing new SCSI standards. The other organizations support T10s activities. The T10 group is comprised primarily of technical people from various hard disk and other technology companies, but the group (and the development process itself) is open to all interested parties. Comments and opinions on standards under development are welcomed from anyone, not just T10 members. The standards development process is intended to create consensus, to ensure that everyone who will be developing hardware and software agrees on how to implement new technology.

Once the T10 group is done with a particular version of a standard, they submit it to NCITS and ANSI for approval. This approval process can take some time; which is why the official standards are usually published several years after the technology they describe is actually implemented. While approval of the standard is underway, companies develop products using technology described in the standard, confident that agreement has already been reached. Meanwhile, the T10 group starts work on the next version of the standard. With SCSI-3 now including a number of different "sub-standards", it is in some ways constantly "under development".

There are also other organizations that are involved in the creation and maintenance of SCSI-related standards. Since SCSI-3 has a broad scope, it defines and structures certain standards that are in fact "owned" by other groups. In particular, the documents describing the physical layer for Fibre Channel are developed by the *T11 technical committee*, and the *IEEE-1394* (Firewire) interface is of course an IEEE standard.

In this section we describe the three main standards that define SCSI. They are listed in chronological order, and SCSI-3 is expanded into its own full section, reflecting its new status as an "umbrella" standard containing several others.

Just a word of warning before we continue; you may occasionally see a hardware device being sold based on the name of a standard; for example, a "SCSI-3 drive". Be aware that this is a meaningless label, because it is very vague. With the possible exception of SCSI-1, the standards define several different transfer speeds and signaling methods, so just giving the name of a standard is insufficient information to properly describe a SCSI device. With SCSI-3 especially, the label could mean just about anything--always look for the specifics.

## SCSI-1

As we have mentioned before, SCSI evolved from the Shugart Associates Systems Interface or SASI, which was originally created in 1979. The first SCSI standard was approved by ANSI in 1986 as standard X3.131-1986. To avoid confusion when subsequent SCSI standards came out, the original specification was later renamed "SCSI-1".

SCSI-1 defines the basics of the first SCSI buses, including cable length, signaling characteristics, commands and transfer modes. It was quite limited, especially by today's standards, and defined only the most fundamental of SCSI features and transfer modes. Devices corresponding to the SCSI-1 standard use only a narrow (8-bit) bus, with a 5 MB/s maximum transfer rate. Only single-ended transmission was supported, with passive termination. There were also difficulties associated with the standard gaining universal acceptance, due to the fact that many manufacturers implemented different subsets of its features. The standard did not call for all devices to implement support for the same commands, so *there was no guarantee that any given device would work with any other!*

SCSI-1 is now obsolete, and the standard has in fact been withdrawn by ANSI. Devices that adhere to the SCSI-1 standard can in most cases be used with host adapters and other devices that use the higher transfer rates of the more advanced SCSI-2 protocols, but they will still function at their original slow speed.

Since all SCSI-1 devices are single-ended, they may cause performance degradation if placed onto a multimode LVD SCSI bus. If you want to run LVD devices to their full potential, you will want to avoid mixing them with single-ended devices.

## SCSI-2

In 1985, a year before the SCSI-1 standard was formally approved, work began on the SCSI-2 specification. Important goals of this evolution of the SCSI standard were to improve performance, enhance reliability, and add features to the interface. However, the most important objective was to formalize and properly standardize SCSI commands. After the confusion that arose from the non-standardized implementations of original SCSI, a working paper was created to define a set of standard commands for SCSI hard disks, called the *common command set* or *CCS*. This paper eventually formed the basis for the new SCSI-2 standard. SCSI-2 was approved by ANSI in 1994 and released as document X3.131-1994.

You might see references to X3.131-1990 while reading and researching SCSI-2. The SCSI-2 standard was originally released in 1990 as X3.131-1990, but it was retracted for further changes and didn't actually get formally approved until four years later. Due to this, you may see reference to the 1990 version of the standard on occasion; there are actually few differences between it and the 1994 version.

SCSI-2 is an extensive enhancement of the very limited original SCSI. The command set used for SCSI devices was standardized and enhanced, and several confusing "options" removed. In addition, the standard defines the following significant new features as additions to the original SCSI-1 specification:

- **Fast SCSI:** This higher-speed transfer protocol doubles the speed of the bus to 10 MHz, meaning 10 MB/s transfer rate with 8-bit regular SCSI cabling or even higher when used with Wide SCSI.
- **Wide SCSI:** The width of the original SCSI bus was increased to 16 (or even 32) bits. This permits more data throughput at a given signaling speed. Wide SCSI eventually replaced original "narrow" SCSI buses for the fastest drives.
- **More Devices per Bus:** On buses that are running with Wide SCSI, 16 devices are supported (as opposed to 8 with regular SCSI).
- **Improved Cables and Connectors:** As will be discussed later, SCSI uses a large number of different cable and connectors. SCSI-2 defined new higher-density connections, extending the basic 50-pin connectors defined in SCSI-1.

- **Active Termination:** Termination is an important technical consideration in setting up a SCSI bus. SCSI-2 defined the use of active termination, which provides more reliable termination of the bus.
- **Differential Signaling:** To allow longer cable lengths, differential signaling was introduced. (This was later renamed "high-voltage differential" to distinguish it from low voltage differential (LVD) signaling.)
- **Command Queuing:** One of SCSI's strengths is its ability to allow multiple outstanding requests between devices on the bus, simultaneously. Command queuing was introduced in SCSI-2.
- **Additional Command Sets:** SCSI-2 added new command sets to support the use of more devices such as CD-ROMs, scanners and removable media. The older command set focused more on hard disks.

There were also several other minor changes to the standard, mostly low-level technical changes that we don't really need to get into. It is important to note that one of the major design criteria in the creation of SCSI-2 was backward compatibility with SCSI-1. SCSI-2 devices will in most cases work with older SCSI-1 devices on a bus. This is not always done, however, because the older devices have no ability to support the SCSI-2 enhancements and faster transfer protocols.

Also, keep in mind that SCSI-2 is not the same as Ultra2 SCSI, which is a much newer and higher-performance feature set. We will discuss Ultra2 SCSI later.

## SCSI-3

Work on the next version of the SCSI standard, called *SCSI-3* of course, began in 1993. At the time, a large number of different technologies, command sets and features were being considered for SCSI. At the time SCSI-3 work started, the SCSI-2 standard was eight years old and had not yet been formally released; SCSI-2 was also a *much* bigger document than SCSI-1 had been. Considering how much more technology was vying for inclusion in SCSI-3, and how many different parties would have been involved in defining it, trying to do everything in one large document as had been done with SCSI-1 and SCSI-2 would have been a bad idea. The standard would have been much too long, would have required too many people to work on it, and development of hardware would have been held up during discussions of the standard. Standards that take too long to develop, or that are too cumbersome, get ignored by impatient companies that start developing their own proprietary extensions. If this happens, it causes a lot of confusion in the industry.

Recognizing the potential for problems here, the decision was made to make SCSI-3 not one huge standards document, but rather a collection of different, but related standards. Splitting up SCSI into these different sub-documents has allowed for a "divide and conquer" strategy that enables multiple standards to be worked on at once by different groups. In addition, it lets popular technologies advance at a faster rate than ones where changes are needed less frequently. Keeping all the standards together under one "banner" helps to ensure that the commands and other common attributes used between technologies remain synchronized.

Unfortunately, SCSI-3 tries to bite off a lot; some would say, more than it can chew. Within this umbrella standard are over a dozen other standards that define command sets, protocols and signaling methods related to SCSI and "SCSI-like" interfaces such as IEEE-1394 and Fibre Channel. Each of these documents has its own standards name and is revised independently of the others. For example, the most implemented form of SCSI, which was formerly known as just "SCSI" in the earlier standards, became the *SCSI-3 Parallel Interface* (*SPI*) under SCSI-3. There are now several versions of SPI, each defining new features and transfer speeds for conventional, parallel SCSI devices.

We shall discuss SCSI-3 in a fair bit of detail since this is the current standard in use in the industry.

## SCSI-3 Architecture

Since SCSI-3 defines a number of different standards, each covering different aspects of SCSI, it is necessary to organize these into a format that defines how they relate to each other, and the goals of the interface as a whole. This structure is called the *architecture* of SCSI-3. SCSI-3 architecture is defined by a document called the *SCSI-3 Architecture Model* or *SAM*, which has been approved as ANSI standard X3.270-1996.

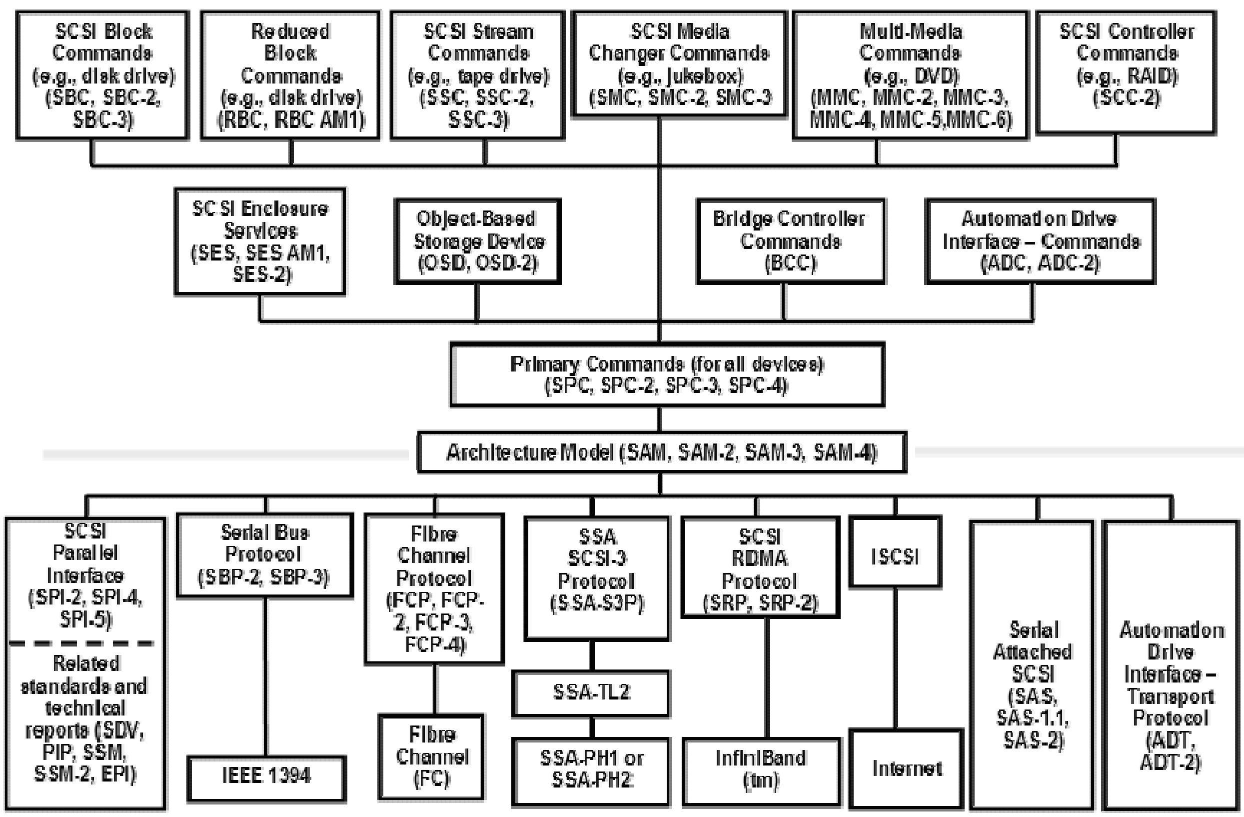


**Figure 1** - [1]This chart reflects the currently approved SCSI project family. T10 is responsible for all projects, except: IEEE is responsible for IEEE 1394; T11 responsible for Fibre Channel; Internet Engineering Task Force (IETF) is responsible for iSCSI and the Internet; and InfiniBand Trade Association is responsible for InfiniBand (tm).

The SCSI-3 Architecture Model has several functions. An important one is to organize and categorize the various other standards that fall under SCSI-3. This serves to structure these standards in a way that makes sense to SCSI standards developers, hardware designers and users. The structure defines broad, generic requirements at a high level, which are refined to more specific low-level requirements through the use of particular implementation standards. Most of the different SCSI-3 documents fall into the following three general categories:

**Commands:** These are standards that define specific command sets for either all SCSI devices, or for particular types of SCSI devices.

| **Command Set** | **Description** | **Document** | **Abbreviation and Generation** |
|---|---|---|---|
| **Shared** | Commands defined for all SCSI devices | *SCSI-3 Primary Commands* | *SPC* |
| | | | *SPC-2* |
| | | | *SPC-3* |
| **Block** | Commands defined for random-access devices that transfer data in blocks, such as hard disks | *SCSI-3 Block Commands* | *SBC* |
| | | | *SBC-2* |
| **Block (Reduced)** | A "simplified" version of the block command set | *SCSI-3 Reduced Block Commands* | *RBC* |
| **Stream** | Commands for streaming, sequential-access devices such as tape drives | *SCSI-3 Stream Commands* | *SSC* |
| | | | *SSC-2* |
| **Medium Changer** | Commands for medium-changing devices such as tape or disk "jukeboxes" | *SCSI-3 Medium Changer Commands* | *SMC* |
| | | | *SMC-2* |
| **Multimedia** | Commands for "multimedia devices" (typically, optical drives) | *SCSI-3 Multimedia Commands* | *MMC* |
| | | | *MMC-2* |
| | | | *MMC-3* |
| **Multimedia (Reduced)** | A "simplified" version of the multimedia command set | *SCSI-3 Reduced Multimedia Commands* | *RMC* |

[1] SCSI architecture family tree taken from the t10 website.

| | | | |
|---|---|---|---|
| **Controller** | Commands for RAID controllers | *SCSI-3 Controller Commands* | *SCC* |
| | | | *SCC-2* |
| **Enclosure Services** | Commands for SCSI device enclosures | *SCSI-3 Enclosure Services* | *SES* |
| **Object Based Storage Devices** | Defines an object-oriented command set for accessing data | *Object Based Storage Device Commands* | *OSD* |

**Table 1** - SCSI command standards

- **Protocols:** These standards formalize the rules by which various devices communicate and share information, allowing different devices to work together. These standards are sometimes said to describe the *transport layer* of the interface.

| Protocol | Description | Document | Abbreviation and Generation |
|---|---|---|---|
| **Interlocked (Parallel Bus)** | Defines the protocol for "regular" parallel SCSI | *SCSI-3 Interlocked Protocol* | *SIP* |
| **Fibre Channel** | Defines the protocol for running SCSI on the Fibre Channel interface | *SCSI-3 Fibre Channel Protocol* | *FCP* |
| | | | *FCP-2* |
| **Serial Bus** | Defines the protocol for transporting commands over the IEEE-1394 (serial) interface | *Serial Bus Protocol* | *SBP* |
| | | | *SBP-2* |
| **Serial Storage Architecture** | Defines the transport layer for Serial Storage Architecture, an advanced interface used in servers and enterprise hardware; there are two documents that specify the protocol | *Serial Storage Architecture SCSI-3 Protocol* | *SSA-S3P* |
| | | *Serial Storage Architecture Transport Layer* | *SSA-TL2* |

**Table 2** - SCSI protocols

- **Interconnects:** These are standards that define specific interface details, such as electrical signaling methods and transfer modes. They are sometimes called *physical layer* standards as well.

| Interconnect | Description | Document | Abbreviation and Generation |
|---|---|---|---|
| **Parallel Bus** | Describes the electrical signaling, connectors and related issues associated with "regular" parallel SCSI; starting with SPI-2 these include the formerly separate SIP protocol document | *SCSI-3 Parallel Interface* | *SPI* |
| | | | *Fast-20* (addendum to SPI) |
| | | | *SPI-2* |
| | | | *SPI-3* |
| | | | *SPI-4* |
| **Serial Storage Architecture** | Defines the physical connections for the Serial Storage Architecture interface | *Serial Storage Architecture Physical Layer* | *SSA-PH* |
| | | | *SSA-PH2* |
| **Fibre Channel** | Several documents define alternative physical layer standards for Fibre Channel; these are maintained by the *T11 technical committee* and include Fibre Channel Arbitrated Loop (FC-AL) and several revisions of the Fibre Channel Physical Interface (FC-PHx) | | |
| **Serial Bus** | The physical layer standards for the serial bus (IEEE-1394) are developed by the *IEEE High Performance Serial Bus Bridges Working Group* (P1394) | | |

**Table 3** - SCSI Interconnects

## <u>SCSI-3 Parallel Interface (SPI)</u>

“Regular SCSI" uses a parallel bus (many wires transferring data in parallel). Therefore, this technology became known as the *SCSI-3 Parallel Interface* or *SPI*. The first description of the parallel interface was accomplished in a rather confusing way, through the use of three different documents:

- **Protocol:** The protocol for parallel SCSI was defined in a document entitled *SCSI-3 Interlocked Protocol* (*SIP*).
- **Physical Layer:** The physical layer was defined in the *SCSI-3 Parallel Interface* or *SPI* document, ANSI standard X3.253-1995. This specification only called for bus speeds of up to 10 MHz, which is so-called "Fast SCSI", first defined in SCSI-2.
- **Fast-20:** This is an addendum to the original SPI document, published as ANSI standard X3.277-1996. It defined faster 20 MHz bus signaling, increasing maximum throughput to as much as 40 MB/s on the SCSI bus.

Taken collectively, these are sometimes called Ultra SCSI or Wide Ultra SCSI, which are really informal or marketing terms; sometimes, Ultra SCSI refers specifically to the faster signaling rates themselves. Aside from the faster signaling, which allows for speeds of up to 20 MB/s on narrow (8-bit) SCSI buses or 40 MB/s on wide (16-bit) buses, the other main change associated with SPI is the creation of new cabling. Wide buses previously required two cables, a cumbersome solution that was never widely accepted. SPI introduced the high-density, 68-pin "P" cable and connectors now widely used for faster SCSI buses.

## SCSI(-3) Parallel Interface - 2 (SPI-2)

The second generation of the SCSI-3 parallel interface standard is called the *SCSI(-3) Parallel Interface - 2*, or *SPI-2*. (The "-3" was dropped from "SCSI-3" to reduce confusion.) This ANSI standard, document X3.302-1999, replaced the older SPI standard, and also incorporated the SCSI-3 Interlocked Protocol (SIP) document. Thus, SPI-2 included everything from the earlier SCSI-2, SPI, SIP and Fast-20 documents--as well as adding some new features of course. SPI-2 formally replaced the earlier collection of SPI documents in 1999, and in doing so simplified matters significantly, since at least now everything associated with parallel SCSI was back in one document.

Several important new technologies and features were defined as part of SPI-2; the most important changes are the following:

- **Fast-40 Data Transfer:** SPI-2 defines another doubling of the maximum speed of the SCSI bus, from 20 MHz to 40 MHz, allowing maximum throughput of 40 MB/s on a narrow (8-bit) channel or 80 MB/s on a wide (16-bit) channel. The document also defines several restrictions associated with these faster signaling speeds, such as the use of differential signaling.
- **Low Voltage Differential Signaling:** A new type of signaling for the SCSI bus, called *low voltage differential* or *LVD* signaling, was specified as part of SPI-2. LVD is an attempt to blend the best attributes of conventional single-ended (SE) signaling and the older type of differential signaling that is now called high voltage differential (HVD). LVD (or the older HVD) is required to run the SCSI bus at Fast-40 speeds.
- **Multimode Operation:** Specification is provided for a way to create devices that will automatically work on both LVD and regular single-ended buses; such units are called *multimode devices*. They are also discussed in the section on LVD.
- **SCA-2 Single Connector Attachment Connectors:** An improvement to the original SCA connectors, called SCA-2, was defined.
- **Very High Density Connectors:** SPI-2 defined a smaller version of the older high-density 68-pin connectors. This new standard is called *Very High Density Cable Interconnect*, abbreviated *VHDCI*.

The features defined as part of SPI-2 are sometimes referred to by the informal (marketing) terms Ultra2 SCSI and Wide Ultra2 SCSI.

## SCSI(-3) Parallel Interface - 3 (SPI-3)

The third generation of the SCSI parallel interface is called the *SCSI(-3) Parallel Interface - 3* or *SPI-3*. This document builds upon the physical and protocol definitions of the SPI-2 document.

Five main features were added to parallel SCSI in the SPI-3 standard:

- **Fast-80(DT) Data Transfer:** Reflecting the continuing appetite for speed on the SCSI bus, data transfer rates were again doubled, this time to 160 MB/s on a wide bus. This was accomplished not by increasing the speed of the bus from 40 MHz to 80 MHz, but rather through the use of double transition clocking; thus the "DT" sometimes found in the name for this signaling speed.
- **Cyclic Redundancy Check (CRC):** This is a common error checking protocol used to ensure data integrity. It was added as a safety measure since transfer speeds were being increased, leading to the possibility of data corruption.
- **Domain Validation:** This feature improves the robustness of the process by which different SCSI devices determine an optimal data transfer rate.
- **Quick Arbitration and Selection (QAS):** This feature represents a change in the way devices determine which has control of the SCSI bus, providing a small improvement in performance.
- **Packetization:** Another small change to improve performance, packetization reduces the overhead associated with each data transfer.

Other, smaller changes were also made. SPI-3 also does some "cleanup" of the parallel SCSI standard, by making obsolete several older features that either never caught on in the industry, or were replaced with superior ways of accomplishing the same tasks:

- **High Voltage Differential:** With the widespread adoption of low voltage differential, the older "high voltage" differential became unnecessary. Since it was never very popular, it was removed from the standard.
- **32-Bit Bus Width:** Introduced in SCSI-2, the 32-bit parallel SCSI option never caught on in the industry and was finally removed from the specification in SPI-3.
- **SCAM:** SPI-3 removed the "SCSI Configured AutoMatically" (SCAM) feature, which was a good idea but never was universally adopted and sometimes led to configuration problems. In doing so, the SCSI world was mercifully rid of one of the worst acronyms in the history of the computer industry. ☺
- **Narrow High-Speed Transfers:** Narrow (8-bit) SCSI hasn't been technically "made obsolete", but 8-bit transfers are not defined for Fast-80 transfers. (Considering that faster transfer modes are used to get more throughput, increasing data transfer speeds while staying on an 8-bit bus doesn’t makes much sense.)

Unfortunately, despite the lessons that should have been learned in the past regarding what happens when standards aren't kept universal, the SCSI industry managed to create another mess out of the SPI-3 standard. The *SCSI Trade Association* defined the marketing term "Ultra3 SCSI" to correspond to the features introduced in SPI-3. However, they allowed a device that implemented *any* sub-set of the five main new features to be called "Ultra3 SCSI"; this "optionality" meant that there was no guarantee that any two devices labeled "Ultra3 SCSI" had the same features! Hardware manufacturers didn't like this, so they decided to market alternative names for more concrete subsets of the Ultra3 features, and we were off to the competing standards races yet again. The results were Ultra160 and Ultra160/m SCSI, and Ultra160+ SCSI.

## SCSI(-3) Parallel Interface - 4 (SPI-4)

SPI-4 is the name given by the SCSI Trade Association to hardware being developed to the next generation of the SCSI parallel interface, as defined in drafts of the *SCSI-3 Parallel Interface - 4* (*SPI-4*) standard. As the name implies, this standard support a maximum throughput of 320 MB/s using 80 MHz bus signaling and double transition clocking. This technology was at one point known by the name "Ultra4 SCSI", but it appears that this terminology will not be used going forward.

The main features are:

- **Special Features:** Fast-160(DT) data transfer.
- **Bus Width:** Wide (16-bit) only.
- **Signaling Method:** LVD only. (Multimode drives may optionally run in SE mode, but throughput will drop to Fast-20 (Ultra) levels.)
- **Signaling Speed and Bus Throughput:** 80 MHz bus speed; 320 MB/s.
- **Number of Devices Supported:** 16 for cables up to 12m in length; 2 for cables over 12m.
- **Termination:** LVD termination.
- **Cabling and Maximum Cable Length:** "P" cable (68 pins). Maximum of 25m if no more than 2 devices are used, otherwise 12m.

## SCSI(-3) Parallel Interface - 5 (SPI-5)

Ultra-640 (otherwise known as Fast-320) was promulgated as a standard (NCITS 367-2003 or SPI-5) in early 2003. Ultra-640 doubles the interface speed yet again, this time to 640 MB/s. Ultra-640 pushes the limits of LVD signaling; the speed limits cable lengths drastically, making it impractical for more than one or two devices. Because of this, most manufacturers have skipped over Ultra640 and are developing for Serial Attached SCSI instead.

## iSCSI[2]

iSCSI stands for “internet SCSI” and preserves the basic SCSI paradigm, especially the command set, almost unchanged. iSCSI advocates project the iSCSI standard, an embedding of SCSI-3 over TCP/IP, as displacing Fibre Channel in the long run, arguing that Ethernet data rates are currently increasing faster than data rates for Fibre Channel and similar disk-attachment technologies. iSCSI could thus address both the low-end and high-end markets with a single commodity-based technology.

[2] Source: Wikipedia

## Serial SCSI[3]

Four recent versions of SCSI; SSA, FC-AL, IEEE1394, and Serial Attached SCSI (SAS) break from the traditional parallel SCSI standards and perform data transfer via serial communications. Although much of the documentation of SCSI talks about the parallel interface, most contemporary development effort is on serial SCSI. Serial SCSI has number of advantages over parallel SCSI - faster data rates, hot swapping, and improved fault isolation. Serial SCSI devices are more expensive than the equivalent parallel SCSI devices but this is likely to change as the technology is commercialized on a larger scale.

[3] Source: Wikipedia

# SCSI Transfer Mode and Feature Set Compatibility

The sheer numbers of different kinds of SCSI make the interface seem overwhelming. How does a SCSI user make it all of this hardware work together? Fortunately, while the standards and feature sets can be quite confusing, the hardware is actually well-engineered, and the standards are designed to allow different hardware types to work together fairly readily.

It's important to remember that a key design goal of all SCSI standards is *backwards compatibility*. Few people want to buy new hardware that won't work with their older hardware. Therefore, in most cases, at least in theory, you can mix older, slower hardware with newer, faster hardware. You can, again *in theory*, put a brand-new Ultra160 SCSI hard disk on the same SCSI bus with a decade-old SCSI-1 host adapter (albeit with added hardware and suboptimal results.) This is generally true, but note the important qualifier: *in theory*. Since changes are always being made to the signaling and other aspects of the interface, there is no guarantee that any two very different pieces of SCSI hardware will work together.

There are no hard and fast rules regarding the compatibility of different SCSI transfer modes and feature sets, especially if they are very different in terms of key attributes. Here are some issues that you should keep in mind as you consider device compatibility:

- **Age:** The greater the difference in age between two devices, the greater the difficulties associated with getting them to work together. The example given above would probably not be much fun. However, mixing Ultra160 and Ultra2 devices is fairly straightforward.
- **Drive and Host Speed Negotiation:** You can use faster drives on slower host adapters or vice versa, but communication will only occur as fast as the slowest device can handle. For example, you can connect a Wide Ultra SCSI drive to an Ultra160 host adapter, but the drive will only run at a maximum of 40 MB/s throughput, not 160 MB/s.
- **Signaling:** Mixing different types of signaling on the same bus can lead to problems ranging from slowdowns to disaster. The older (high voltage) differential signaling is not electrically compatible with either single-ended or LVD devices, and should never be mixed with those types, or you risk *disaster* such as smoked hardware. Multimode LVD devices can be mixed with SE devices, but they won't function at Ultra2 or higher speeds if you do so.
- **Bus Width:** You can mix wide and narrow devices on the same SCSI bus, but there are specific requirements in doing this, to ensure that the bus functions properly.
- **Packages:** If you want to be sure that a particular SCSI implementation will work, buy a complete system or SCSI "package" (including a host adapter, drives, cables and terminators) from a reputable dealer.

# SCSI Protocols and Interface Features

In detailing the various standards, transfers modes and feature sets associated with the SCSI interface, several important concepts have been introduced that define the attributes of various types of SCSI buses. New technologies are often introduced specifically to change these traits, to improve the interface. It's important to understand how the various aspects of SCSI combine to create different specific SCSI varieties.

In this section I will describe the most important characteristics of the SCSI interface. This includes a discussion of the three most important defining characteristics of any SCSI bus: signaling, bus speed and bus width. I then discuss several important SCSI bus features, many of which have been introduced to improve performance or reliability on the newest, highest-performance SCSI implementations. This includes a discussion of bus integrity protection, and advanced features such as command queuing and reordering, domain validation, quick arbitration and selection, and packetization.

## Single-Ended (SE) and Differential (High Voltage Differential, HVD) Signaling

Conventional SCSI signaling is very similar to that used for most other interfaces and buses within the PC. Conventional logic is used: a positive voltage is a "one", and a zero voltage (ground) is a "zero". This is called *single-ended* signaling, abbreviated *SE*. Up until recently, single-ended SCSI had been by far the most popular signaling type in the PC world, for a simple reason: it is relatively simple and inexpensive to implement.

There's an important problem with SE signaling, however. SCSI is a high-speed bus capable of supporting multiple devices, including devices connected both inside and outside the PC. As with all high-speed parallel buses, there is always a concern about signal integrity on the bus; problems can arise due to bouncing signals, interference, degradation over distance and cross-talk from adjacent signals. The faster the bus runs, the more these problems manifest themselves; the longer the cable, the more the problems exist for any given interface speed. As a result, the length of a single-ended SCSI cable is rather limited, and the faster the bus runs, the shorter the maximum allowable cable length.

To get around this problem, a different signaling method was also defined for SCSI, which uses two wires for each signal that are mirror images of each other. For a logical "zero", zero voltage is sent on both wires. For a logical "one", the first wire of each signal pair contains a positive voltage, similar to the signal on an SE bus, but not necessarily at the same voltage. The second wire contains the electrical opposite of the first wire. The circuitry at the receiving device takes the *difference* between the two signals sent, and thus sees a relatively high voltage for a one, and a zero voltage for a zero. This method is *much* more resilient to signaling problems than regular SE signaling. It is called *differential* signaling, after the technique used to determine the value of each signal by the recipient. The two signals in each pair are usually named with "+" and "-" signs; for example, the signal carrying data bit 0 would use "+DB(0)" and "-DB(0)".

This table shows the great difference in cable length that exists between SE and differential devices, particularly as bus speed increases:

| Signaling Speed | Bus Speed (MHz) | Single-Ended SCSI Maximum Cable Length (m) | Differential SCSI Maximum Cable Length (m) |
|---|---|---|---|
| **Slow** | 5 | 6 | 25 |
| **Fast** | 10 | 3 | 25 |
| **Fast-20** | 20 | 1.5 | 25 |

**Table 4** - SE and differential SCSI cable lengths

As you can see, each doubling of the bus speed results in a halving of the maximum cable length for single-ended SCSI, but differential SCSI allows long (25m) cables for all three speeds. (Fast-20 buses allow a cable length of 3m if no more than four devices are used, but this is really a kludge of sorts to get around the limitations associated with a 1.5m cable restriction.)

Differential SCSI is a great idea in theory, and one might have thought it would become very popular. In fact, this never happened in the PC world, largely due to cost. The circuits needed to drive differential signals are more expensive and use more power than those for single-ended SCSI. For many years, single-ended SCSI was "good enough", and allowed cable lengths sufficient for the needs of most users, so little impetus was seen to move to the more expensive differential signaling. From there, "chicken-and-egg" syndrome kicked in: since differential was less popular, it was not produced in volume and so never saw its costs come down due to economies of scale.

The end result of all of this is that the older type of differential signaling is rarely seen in the PC world. The concept of differential signaling, however, did not die out. As the SCSI bus was pushed to faster and faster speeds, the cable limits of SE were finally too great to be worked around. However, the cost of regular differential was unappealing, so a new type of differential signaling was created, called *low voltage differential* or *LVD*. With the creation of LVD, the old name of "differential" for the higher-voltage version became vague, so the older style was renamed *high voltage differential* or *HVD*.

High voltage differential signaling has been around since the earliest SCSI-1 standard, so devices have been theoretically available as either SE or HVD since the start of SCSI use on the PC. With the creation of the SPI-3 specification and the standardization of LVD, HVD had no more *raison d'etre*, and has been removed from the SCSI standard entirely, leaving only LVD. There are no new drives being produced that use HVD.

Since single-ended and HVD SCSI use very different voltage levels, they are incompatible at the electrical level. You should not mix single-ended (or LVD) devices with high voltage differential

SCSI devices on the same bus. If you do, actual physical damage could result--this is one of those cases where actually smoking your hardware is a distinct possibility, because of the high voltages that might be sent to the single-ended or LVD devices! To compound the matter, the cables and connectors used for single-ended and differential SCSI look the same.

To help reduce the chances that similar-looking SE and HVD hardware will be interconnected, special *icons* are imprinted on SCSI hardware that indicates the signaling method used by the device. Make sure you know what you have before putting together your SCSI bus, and look for these identifying symbols on devices to be sure they are electrically compatible. Note that slightly different symbols are used for LVD and LVD/SE devices, as will be described in the next section.

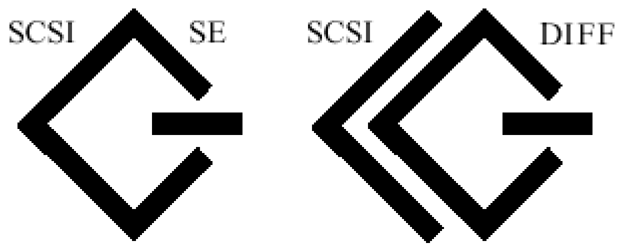



**Figure 2** - Icons for hardware using single-ended SCSI (left) and regular (high voltage) differential SCSI (right).

## Low-Voltage Differential (LVD) Signaling

A third option was created as an alternative to single-ended and high voltage differential signaling, with the intention of marrying the best attributes of both SE and HVD signaling. This is a differential signaling method that was designed to use the advantages of differential signaling to allow long cable lengths, while reducing implementation cost and allowing for electrical compatibility with single-ended devices. This technology is called *low voltage differential* or *LVD* signaling. It was first defined in the SPI-2 standard and is rapidly becoming the signaling method of choice in the SCSI world. In fact, LVD signaling is required for Ultra2 or Wide Ultra2 SCSI, and LVD is the exclusive signaling method for all SCSI modes faster than Ultra2. Even the fastest LVD SCSI chains can be up to 12m in length, or 25m if only two devices are used on the chain (this is called *point-to-point* operation; remember that one of these must be the interface card, the host adapter.)

The concept behind LVD is relatively straight-forward: continue using two wires for each signal, but use lower voltage to create the complementary signal pairs. Using lower voltage allows cost to be reduced and power requirements to be kept under control. It also means that the dangers associated with mixing SE and differential devices is eliminated. In fact, single-ended devices are not just electrically compatible with LVD devices, some types of LVD devices can even function on single-ended SCSI buses.

A particular type of LVD device was defined when LVD was created; drives that correspond to this variant of LVD are called *multimode LVD device*. These are usually abbreviated *LVD/SE* or *LVD/MSE* (the "M" is for "multimode"). A multimode LVD device will automatically switch between LVD and single-ended operation by detecting whether the other devices on the chain are running in SE or LVD mode. However, only one or the other can be used at a time; the device won't use both simultaneously.

In addition to the usual SCSI rules, such as unique IDs for each device and proper topology and termination, LVD operation requires the following:

1. All devices on the chain must be LVD-capable; if even one device is only SE, all devices "drop down" and run as single-ended.
2. All devices must not be set to run in SE mode; some multimode devices have a jumper to "force" SE operation, which will cause the entire SCSI chain to not work in LVD.
3. LVD (or multimode LVD/SE) terminators must be used.

Remember that bus speeds over 20 MHz are not supported under single-ended operation. This means that a multimode LVD/MSE Ultra160 device will run at only a maximum of 40 MB/s if it is connected to a SCSI chain with single-ended devices.

As soon as multimode LVD devices begin running as single-ended, all the rules and restrictions of single-ended operation apply, including cable length. For example, suppose you have a 4m cable connecting an LVD Ultra160 host adapter to a multimode LVD Ultra160 device; this is perfectly fine. Now, let's say you decide to add to this cable a Wide Ultra single-ended device. As soon as this happens, the other devices will drop down to single-ended operation, and probably will try to run at Ultra speeds (Fast-20). Communication problems will then result due to the fact that a 4m cable is not supported at Ultra speeds in single-ended operation.

LVD signaling is rapidly taking over the SCSI world. Single-ended operation is not supported for bus speeds faster than 20 MHz ("Ultra" bus speeds), so to use Ultra2 or faster SCSI, differential must be used. HVD for its part was made obsolete in the SPI-3 standard, so for Ultra3, Ultra160, Ultra160+ and faster speeds, LVD is the only option.

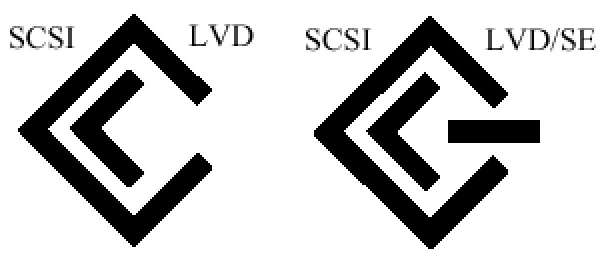



**Figure 3** - Icons for hardware using LVD SCSI (left) and multimode LVD/SE SCSI (right).

## SCSI Bus Width

There are two commonly used SCSI bus widths: narrow and wide. Narrow SCSI uses a data pathway that is 8 bits wide. Wide SCSI uses a data pathway 16 bits wide. Since its introduction, wide SCSI has been steadily increasing in popularity, since it allows a doubling of bus bandwidth for any given signaling speed. It also allows the use of 16 devices on the SCSI bus, compared to the standard 8 devices for narrow SCSI.

Wide SCSI originally required the use of two cables: a 68-pin "B" cable in addition to the regular 50-conductor "A" cable used for narrow SCSI. This use of two cables was expensive and cumbersome, and the "A+B" configuration was eventually replaced by a single 68-pin "P" cable.

Today, narrow SCSI is actually being left behind, as the need for extra performance has led to the dominance of wide forms of SCSI, especially for hard disks. Transfer modes faster than Ultra2 have done away with narrow buses altogether. Presumably, if one is designing a device that needs throughput enough to justify going to speeds faster than Ultra2, it would be silly to "give away" half the throughput by going narrow instead of wide. Fast-80(DT) and Fast-160(DT) signaling, as defined in the SPI-3 and SPI-4 standards respectively, are only for wide implementation. As a result, all relevant marketing terms such as Ultra3, Ultra160, Ultra160+ and Ultra320 have *wide bus operation implied*, reversing the way it was with the earlier SCSI flavors.

It is possible to mix narrow and wide SCSI on the same bus, but there are issues that must be overcome to do so. These typically revolve around cabling, which is different for narrow and wide SCSI, and also with termination. Adapters may be required to convert between the narrow and wide cables. Note that there are also host adapters available that are specifically designed to support both wide and narrow devices.

A "very wide" 32-bit form of SCSI was defined as part of the SCSI-2 standard, but was never accepted by the industry due to cost. It required the use of two 68-conductor cables, for one thing. It was also non-standard; with the extra costs involved, there was little interest in it. It was finally withdrawn from the SCSI standard in SPI-3.

## SCSI Bus Speed

SCSI buses run at a variety of different speeds. Generally, newer buses run faster than older ones, reflecting the increased performance of newer hardware. These are the three ways that SCSI bus speeds are commonly quoted:

- **Clock Speed:** This refers strictly to the frequency of the clock (strobe) used to control synchronous transfers of data on the SCSI bus. With current technology this can be 5, 10, 20, 40 or 80 MHz. For more on clocks and how clock speeds work, see this fundamentals page.
- **Transfer Rate:** This refers to the number of times per second that data is transferred across the interface. This is only the same as the clock speed of the bus if single transition

(conventional) clocking is used. Faster SCSI implementations now use double transition clocking, and this means the transfer rate (in millions of transfers per second) will be double the clock speed in MHz.

- **Throughput:** This number represents the theoretical maximum amount of data that can be moved across the SCSI bus, and is measured in millions of bytes per second (MB/s). On a narrow bus, throughput and transfer rate are the same, because each transfer is of 8 bits (one byte). But for a wide bus, throughput is double transfer rate, because each transfer is of 16 bits--two bytes.

Now that the terminology is clear, we can look at the various bus speeds used in the SCSI world and understand what they mean. The table below shows all of the bus speeds used for parallel SCSI. (You may also find that looking at the table makes more clear the relationship between clock speed, transfer rate and throughput):

| **Standard-Defined Bus Speed** | **Common Signaling Speed Name** | **Clock Speed (MHz)** | **Clocking** | **Transfer Rate (Mtransfers/s)** | **Throughput (MB/s)** | |
|---|---|---|---|---|---|---|
| | | | | | **Narrow (8-bit)** | **Wide (16-bit)** |
| **SCSI-1** | Regular | 5 | Single | 5 | 5 | -- |
| **Fast** | Fast | 10 | Single | 10 | 10 | 20 |
| **Fast-20** | Ultra | 20 | Single | 20 | 20 | 40 |
| **Fast-40** | Ultra2 | 40 | Single | 40 | 40 | 80 |
| **Fast-80(DT)** | Ultra3 or Ultra160 | 40 | *Double* | 80 | -- | 160 |
| **Fast-160(DT)** | Ultra320 | 80 | *Double* | 160 | -- | 320 |
| **Fast-320(DT)** | Ultra640 | 80 | *Double* | 320 | -- | 640 |

**Table 5** - Parallel SCSI bus speeds

The "(DT)" in "Fast-80(DT)", "Fast-160(DT)" and “Fast-320(DT)” represents the fact that this suffix is sometimes attached to represent the use of double transition clocking for those interfaces.

As you can see, the use of double transition clocking and wide buses means that the numbers in the latest transfer modes do not refer to the actual speed of the bus at all. The "160" in "Ultra160" represents the maximum throughput of such devices, but the clock speed is "only" 40 MHz.

## Bus Parity and Cyclic Redundancy Checking (CRC)

As mentioned in a few places in this discussion of SCSI, parallel buses can have signal integrity problems, especially if used on long cables or at high signaling speeds. To help ensure that the data sent from one device arrives intact at its destination, various SCSI buses use two different data protection methods.

The first technique is *SCSI bus parity*. The parity method uses an extra bit for each eight bits of data, which is computed by the sending device so that the sum of all the "ones" in the nine bits taken together is either *odd* or *even*--one is chosen as the standard for the interface, and for SCSI odd parity is used. At the receiving device, the data is checked to see if the total is still odd; if an even number of "ones" is seen, this means there was a data corruption problem (because one bit is the wrong value somewhere) and the sender is signaled to retransmit. This simple data protection method is not unique to SCSI; it has also been used for years for serial communications and in memory circuits.

SCSI parity is useful, but is limited in its effectiveness, especially for very high transfer rates. It cannot detect if two bits in a given byte of data flip, for example. To further safeguard data, the SPI-3 standard introduced *cyclic redundancy checking* or *CRC* to the SCSI world when double transition clocking was introduced, allowing 160 MB/s data throughput on the bus. CRC is another technique that is not new, just "new to SCSI". It has been used in a variety of places in the computing world for decades, for example, in modems and for data integrity checking.

CRC is a more robust way of checking for data corruption that can occur anywhere in a transmitted data message. A special algorithm is used that calculates a binary code as a result of arithmetic operations on the data; this is called a *cyclic redundancy code* (also abbreviated *CRC*). This code is sent along with the data over the bus. The recipient runs the same computation on the data and checks to see if it gets the same value that the sender computed; if there is any difference then an error occurred. In fact, modern CRC implementations don't actually run the computations using formulas, but rather use pre-created tables to speed up the process. This is sort of like using "multiplication tables" as you might have memorized as a child, instead of doing the computation from scratch each time.

At any rate, the bottom line is that CRC does a much better job of protecting data transmitted on the bus, especially at high signaling speeds. It can in theory be used in conjunction with parity, as they are independent. However, once CRC is being used, parity is somewhat unnecessary, except perhaps for compatibility with older hardware. CRC is one of the "optional" features of Ultra3 SCSI, and is a required feature for hardware meeting the Ultra160 or Ultra160+ specifications.

## Command Queuing and Reordering

SCSI is often described as being "advanced", or is called an "intelligent interface". One of the reasons for these descriptions is that SCSI hardware incorporates features that improve overall system performance, where simpler interfaces such as IDE/ATA do not. One of these techniques is a special feature that allows for concurrent, multiple requests to devices on the SCSI bus. This feature is called *command queuing and reordering*; sometimes the name is given as *tagged command queuing*. It was first introduced in the SCSI-2 standard.

Traditionally, a simple interface like SCSI-1 or IDE/ATA will allow only a single command to be outstanding at a time to any device. This means that once a particular command is sent to a device, any other commands must wait for the first one to be completed, which slows down performance. *Command queuing* allows a device to accept as many as 64 or even 256 concurrent commands. The commands can also come from different originating devices. *Command reordering* allows a device that has multiple commands outstanding to fill them "out of order", meaning, not necessarily in the order that they were received.

For a very simple SCSI bus, such as a single hard disk on a host adapter in a desktop PC, command queuing and reordering may not make a particularly huge difference in performance. The reason is simply that there aren't that many concurrent processes running, and not a great deal of activity on the bus. This feature really comes into its own in a multiple-device, multitasking environment, such as that experienced by a shared server. In that environment, command queuing and reordering will improve performance significantly, by allowing devices to accept multiple simultaneous requests from different users, and fill them in the most efficient manner.

This is very important for devices like hard disks, which are very slow compared to the rest of the system. If commands are processed only as they are received, a great deal of time may be wasted while the hard disk's mechanical components move past a physically close piece of data that will be needed one or two requests "down the road". For a more thorough explanation of how drives can improve performance by reordering commands, see this discussion. In newer storage devices, this feature is also sometimes called Native Command Queuing (NCQ).

## Negotiation and Domain Validation

SCSI hardware supports many different speeds, and newer, faster hardware is generally backwards-compatible with older, slower devices. You can use a host adapter capable of 160 MB/s throughput with drives that can only support 20 MB/s transfers, or vice-versa. This leaves an obvious question: how does each device determine what speeds the others on the bus are capable of? Without knowing this, senders can't figure out how fast receivers can handle data being sent.

Since this is so important, the SCSI protocols build in support for a method by which the host adapter can interrogate all devices on the bus to find out what speeds they support. This process is called *negotiation*, and is one of the first tasks performed by the SCSI host adapter when the system power is applied. Under conventional SCSI rules, this negotiation is done with each device; the host adapter records the maximum transfer speed that each device claims to support, and then uses that information when the device is accessed.

This works great in theory, but there's a problem with it: theory doesn't always translate into practice, especially when the technology "pushes the envelope" with high-speed signaling. For example, even if the host adapter can support Ultra160 transfers and the device says it can as well, this doesn't mean that 160 MB/s signaling is actually possible on the bus. Perhaps the cabling being used is inferior or too long, or there's a problem with a terminator, or the system is in a particularly electrically noisy environment. Regular negotiation just "trusts" that everything will work at the speed the hardware decides is possible, but it may not actually work. If there are difficulties, they may manifest themselves in the form of data errors or reliability problems.

To improve negotiation, the SPI-3 standard introduced a new feature called *domain validation*, sometimes abbreviated *DV*. This feature basically adds a verification step to the normal negotiation procedure (note that "domain" is another word for a SCSI channel or bus). After a device tells the host adapter that it is capable of transfers at a particular speed, the host adapter tests the device by sending write requests to the device's internal buffer at that speed. The data just written is then read back and compared. If the data is different, or if parity or CRC errors occur during either the read or the write, the host adapter knows that communication at that speed is not reliable. It will then retry at the next lower speed, and continue until reliable operation is established. This is quite similar to the way that two regular analog modems determine and negotiate the communications speed.

Domain validation is one of the five "optional" features of Ultra3 SCSI, and is a required feature for hardware meeting the Ultra160 or Ultra160+ and higher specifications. This feature may be expanded in the future to include more frequent validation during the operation of the system, since over time errors may occur on a channel that worked fine when the system was first powered up.

## Quick Arbitration and Selection (QAS)

During the time when the system is running, the SCSI bus is generally either active or idle. If active, the bus is busy transmitting data from one device to another; if idle, it is available for a device to begin sending a command or data. When a device decides it wants to use the bus, it "bids" for control of the bus. It is also possible that other devices on the bus will want to use it at the same time, so they too may "bid" for control. A specific method is used to resolve these requests and decide which device gets to use the bus first; this is based to some extent on the devices' respective priority levels. This process is called *arbitration*.

While arbitration works fine in regular SCSI configurations, it introduces overhead. During the time that arbitration is going on, no data is being transferred on the bus, so it makes sense that doing this faster will allow improved performance of the entire SCSI subsystem. To this end, the SPI-3 standard defined a feature that reduces the overhead required for arbitration. This feature is called *quick arbitration and selection* or *QAS*. You may also see it called by the name it carried during development, *quick arbitration and select*; IBM calls it *quick arbitration select* and Adaptec, simply *quick arbitrate*. These are all different names for the same feature.

In a nutshell, QAS works by reducing the number of times arbitration must occur on the bus. When the feature is used, a device waiting for the bus can grab it more quickly after the last device on the bus sends the signal that it is done, without having to begin a new arbitration process. Provision is made in the specification to ensure that one device does not "dominate" the bus by "unfairly" blocking out other devices that may be of a lower priority or may not implement QAS.

Quick arbitration and selection is one of the five "optional" features of Ultra3 SCSI. It was not included as one of the required features for hardware meeting the Ultra160 specification, but is present in Ultra160+ devices.

## Packetization

While the SCSI interface is widely implemented on high-end hardware due to its flexibility and high performance, its complexity does mean that some of its potential performance is lost to overhead. In an effort to improve SCSI bus performance by reducing overhead, the SPI-3 SCSI standard describes a new feature that is generally called *packetization* or *packetized SCSI*.

Packetization is a technique whereby some of the phases that are involved in setting up a command request and data transfer are combined. For example, under traditional SCSI interfacing, several different types of information are sent over the bus separately: commands, data, status messages and so on. With packetization, these are grouped together into *packets* (also called *information units*) and sent as a single entity. This reduces some of the wasted bus cycles normally sent on managing all the individual transfers in regular SCSI.

Packetization is one of the five "optional" features of Ultra3 SCSI. It was not included as one of the required features for hardware meeting the Ultra160 specification, but is present in Ultra160+ and higher devices.

## SCSI Protocol Map

The table below shows how the two most important protocol characteristics; bus width and bus speed, map into the different names for SCSI transfer modes and feature sets that are commonly seen in the SCSI world. The throughput for each SCSI variety is also shown:

| **Signaling Speed** | **Narrow** | | **Wide** | |
|---|---|---|---|---|
| | **Mode** | **Throughput (MB/s)** | **Mode** | **Throughput (MB/s)** |
| **SCSI-1** | SCSI-1 | 5 | Wide SCSI | 10 |
| **Fast** | Fast SCSI | 10 | Fast Wide SCSI | 20 |
| **Fast-20** | Ultra SCSI | 20 | Wide Ultra SCSI | 40 |
| **Fast-40** | Ultra2 SCSI | 40 | Wide Ultra2 SCSI | 80 |
| **Fast-80(DT)** | -- | | Ultra3 SCSI, Ultra160(/m) SCSI, | 160 |

| | | | |
|---|---|---|---|
| | | Ultra160+ SCSI | |
| **Fast-160(DT)** | -- | Ultra320 SCSI | 320 |
| **Fast-320(DT)** | -- | Ultra640 SCSI | 640 |

**Table 6** - Bus width and speeds for various SCSI transfer modes and feature sets

# Summary of SCSI Protocols and Transfer Modes

For easier comparison, the chart below shows all of the different SCSI transfer modes and feature sets, along with their key characteristics.

| Transfer Mode | Defining Standard | Bus Width (bits) | Bus Speed (MHz) | Through-put (MB/s) | Special Features | Cabling | Signaling Method | Maximum Devices Per Bus | Maximum Cable Length (m) |
|---|---|---|---|---|---|---|---|---|---|
| **"Regular" SCSI (SCSI-1)** | SCSI-1 | 8 | 5 | 5 | -- | 50-pin | SE | 8 | 6 |
| | | | | | | | HVD | 8 | 25 |
| **Wide SCSI** | SCSI-2 | 16 | 5 | 10 | -- | 68-pin | SE | 16 | 6 |
| | | | | | | | HVD | 16 | 25 |
| **Fast SCSI** | SCSI-2 | 8 | 10 | 10 | -- | 50-pin | SE | 8 | 3 |
| | | | | | | | HVD | 8 | 25 |
| **Fast Wide SCSI** | SCSI-2 | 16 | 10 | 20 | -- | 68-pin | SE | 16 | 3 |
| | | | | | | | HVD | 16 | 25 |
| **Ultra SCSI** | SCSI-3 / SPI | 8 | 20 | 20 | -- | 50-pin | SE | 8 | 1.5 |
| | | | | | | | | 4 | 3 |
| | | | | | | | HVD | 8 | 25 |
| **Wide Ultra SCSI** | SCSI-3 / SPI | 16 | 20 | 40 | -- | 68-pin | SE | 8 | 1.5 |
| | | | | | | | | 4 | 3 |
| | | | | | | | HVD | 16 | 25 |
| **Ultra2 SCSI** | SCSI-3 / SPI-2 | 8 | 40 | 40 | -- | 50-pin | LVD | 8 | 12 |
| | | | | | | | | 2 | 25 |
| | | | | | | | HVD | 8 | 25 |
| **Wide Ultra2 SCSI** | SCSI-3 / SPI-2 | 16 | 40 | 80 | -- | 68-pin | LVD | 16 | 12 |
| | | | | | | | | 2 | 25 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | HVD | 16 | 25 |
| **Ultra3 SCSI** | SCSI-3 / SPI-3 | 16 | 40 (DT) | 160 | At least one of Fast-80, CRC, DV, QAS, Packet | 68-pin | LVD | 16 | 12 |
| | | | | | | | | 2 | 25 |
| **Ultra160(/m) SCSI** | SCSI-3 / SPI-3 | 16 | 40 (DT) | 160 | Fast-80, CRC, DV | 68-pin | LVD | 16 | 12 |
| | | | | | | | | 2 | 25 |
| **Ultra160+ SCSI** | SCSI-3 / SPI-3 | 16 | 40 (DT) | 160 | Fast-80, CRC, DV, QAS, Packet | 68-pin | LVD | 16 | 12 |
| | | | | | | | | 2 | 25 |
| **Ultra320 SCSI** | SCSI-3 / SPI-4 | 16 | 80 (DT) | 320 | Fast-160 | 68-pin | LVD | 16 | 12 |
| | | | | | | | | 2 | 25 |

**Table 7** - SCSI transfer modes and feature sets

Some notes on this table:

- "(DT)" means that transfer mode uses double transition clocking.
- Throughput numbers are in decimal megabytes per second.
- Only current cabling is mentioned, not obsolete cables.
- The number of devices includes the host adapter.
- For Ultra and faster speeds, the maximum length of cable for some signaling types depends on the number of devices on the chain; thus, the multiple rows.
- The number of devices (8 or 16) includes the host adapter.

# SCSI Host Adapter

Most IDE/ATA hard disks are controlled today by integrated IDE controllers that are built into the chipset on the motherboard. The SCSI interface is not, for the most part, controlled by built-in motherboard SCSI controllers, although some are and this is growing in popularity. Most systems require the addition of a special card which serves as the interface between the SCSI bus and the PC.

This device is called a *SCSI host adapter*, or alternately a *host bus adapter* (sometimes abbreviated *HBA*). It is sometimes called a *SCSI controller* or even just a *SCSI card*, though these are technically incorrect names. They are not accurate because SCSI is a systems-level interface, and every device on the bus has its own controller. Logically, the host adapter is just a SCSI device like any other. Its job is to act as the gateway between the SCSI bus and the internal PC's I/O bus. It sends and responds to commands and transfers data to and from devices on the bus and inside the computer itself. Since it is inside the PC, the host adapter really *isn't* the same as the other devices on the bus--it's sort of a "first among equals", if you want to think about it that way.

Since SCSI is a very "intelligent" interface--meaning it has a lot of capabilities and the devices on it are able to interact in advanced ways--many SCSI host adapters have evolved rather exceptional capabilities, and can act in many ways to improve performance. In some ways, the host adapter is the key to good SCSI implementation in the PC, since no matter how advanced the peripherals are that you attach to the bus, everything goes through that host adapter.

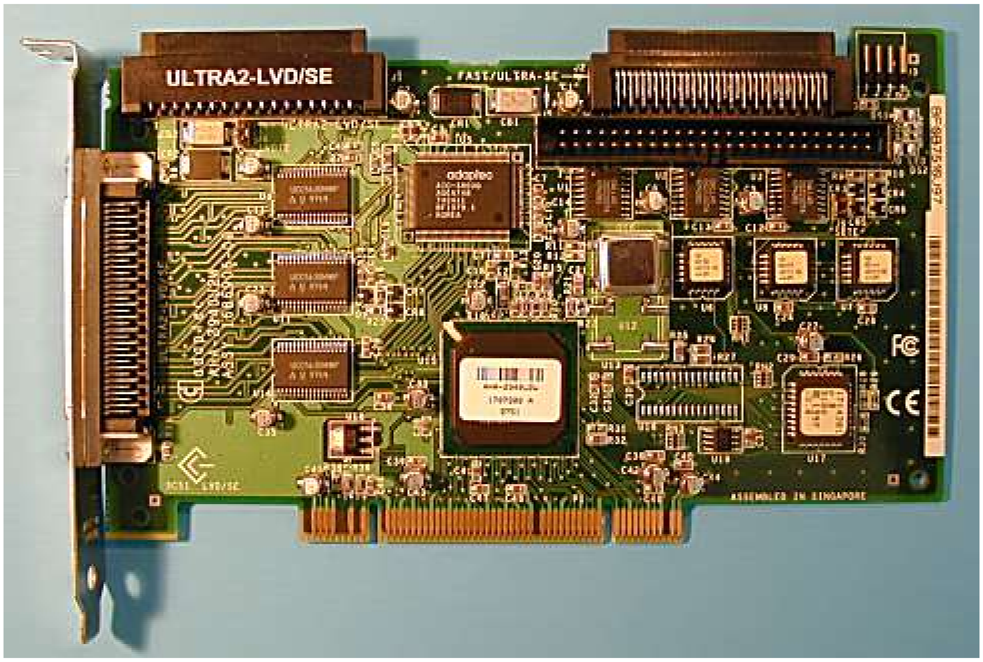



**Figure 4** - A PCI-based Wide Ultra2 SCSI host adapter.

In the picture of a Wide Ultra2 SCSI host adapter above, note the numerous connectors, which allow several chains of internal and external devices to be attached to the host adapter.

In this section we take a reasonably detailed look at host adapters. I focus primarily on the most important attributes and features that someone considering a SCSI setup might look for. This includes a discussion of device support, interfacing, connectors, resources and compatibility.

## Adapter Types and PC Bus Connections

There are many different kinds of SCSI host adapters on the market, and they vary in cost and capabilities dramatically. Many lower-end adapters are designed specifically are designed to keep costs down to allow easy, inexpensive access to SCSI devices like scanners or CD-RW drives. Higher-end devices provide more capabilities and performance for users who require a full-featured implementation for hard disks and other performance drives.

A key distinguishing characteristic between various host adapter models is the type of system bus the card is designed for. SCSI host adapters have been made for all of the common PC I/O buses, including ISA, EISA, VLB, MCA and PCI. You obviously need to choose a host adapter that matches the system bus(es) in your machine. Until recently, motherboards that featured both PCI and ISA slots were common, giving you a choice. Now, PCI bus is more commonly found in motherboards. An ISA-based card is generally a bad idea since this will greatly limit the performance of the bus because ISA cannot handle more than about 8 MB/s of data throughput.

After 20 years, the ISA bus is finally being phased out; Intel and Microsoft are hard at work trying to kill off this old system bus interface, and many systems come only with PCI slots, making a PCI host adapter the only real option.

A very important performance issue concerning SCSI host adapters and the system bus they use is the throughput of the bus. If the throughput of the system bus is less than the maximum throughput of the SCSI channel, SCSI performance will be limited to whatever the bus's maximum rate is. Until recently, this wasn't much of a concern as long as a PCI host adapter was used, because PCI had more than enough "overhead" to handle any SCSI bus. In the last few years, however, with SCSI channels continuing to increase in speed like Ultra640 SCSI, even the performance of regular PCI is now reaching a limiting point. The maximum practical bandwidth of regular (32-bit, 33 MHz) PCI is a little over 100 MB/s, and the newest SCSI devices use Ultra160 SCSI, capable of well over 100 MB/s of throughput. To get maximum performance from such an interface, regular PCI is not sufficient.

To this end, some higher-end Ultra160 and faster SCSI host adapters are designed to use new enhancements to traditional PCI. These include 64-bit PCI, capable of throughput of over 200 MB/s, and also the new PCI-E bus, which promises performance of up to 1 GB/s. Cards using 64-bit PCI are readily available and are backwards-compatible with regular 32-bit PCI, so they can be used on both newer systems with 64-bit PCI slots, or systems that have only 32-bit slots.

Another reason to use PCI is that most newer host adapters that run on the PCI local bus support the use of bus mastering. This can be a very important feature, as it allows for more efficient transfer of data from the host adapter to the system memory. Full performance from high-end SCSI chains requires bus mastering support.

## Protocol Support

Obviously, a key issue in selecting a host adapter is support for the particular transfer modes and feature sets that you want to use. The host adapter must be able to support all of the protocols that you want to use on the bus. If you want to use wide SCSI, the adapter must support 16-bit operation; if you want to use Ultra320 SCSI hard disks, your host adapter must be capable of supporting this speed.

Typically, host adapters are backwards compatible with older devices, so you can run a newer drive on an older, slower host adapter--you just give up some of the performance of the drive in doing so. Similarly, newer host adapters will support older, slower devices, providing that they are properly configured.

## Signaling Type Support (SE, HVD, LVD)

With the arrival of Ultra2 SCSI, the SCSI world now uses three different types of electrical signaling: conventional, single-ended SCSI, high-voltage differential (HVD) SCSI, and low voltage differential (LVD) SCSI. The host adapter used in a system must be electrically compatible with the drives that are to be used.

High voltage differential was never very popular in the market because of its expense and the fact that it is not compatible at the electrical level with the more popular single-ended devices. To use high voltage differential drives, you need to use a host adapter specifically designed to support HVD signaling. Adapters do exist to allow HVD drives to run on SE SCSI chains, but this is an expensive solution.

Since low voltage differential signaling is required for the newest, fastest devices, host adapters supporting LVD are taking over the market. Host adapters that support only SE signaling are still available, but these are severely restricted in terms of the performance and cable lengths they will allow, so they are generally used only for devices other than hard disks.

LVD also creates an issue for people who want to use both high-speed hard disks and also slower devices on the same host adapter. LVD and SE are electrically compatible--assuming that the LVD devices are multimode capable (LVD/SE or LVD/MSE)--so it isn't the same issue as with mixing SE and HVD. The problem is that LVD only works if *all* the devices on the SCSI bus are running in LVD mode. If you put a single-ended drive on a SCSI chain with an Ultra160 hard disk, the hard disk will run at no greater than Ultra speeds, knocking down performance. You will also see your maximum cable length reduced from 12m to either 3m or 1.5m.

To get around this problem, many SCSI host adapters that support LVD modes (Ultra2 SCSI and higher) generally include separate support for running devices at Ultra or slower speeds in single-ended mode. This is implemented either through two distinct segments on the same SCSI channel--using translator chips that allow the SE and LVD devices to share data without interfering with each other electrically--or separate, independent SCSI channels within the same card. Be aware that if your SCSI host adapter does not support this feature, you will not be able to run Ultra speed or slower devices and Ultra2 or faster devices on the same machine. Of course, you can choose to use two SCSI host adapters, but this causes other complications.

## Connectors

SCSI is a bus that supports both internal and external devices. To support these two types of devices, most SCSI host adapters come with both internal and external connectors. Internal connectors are usually mounted along the top edge of the SCSI host adapter, and are used for the ribbon cables employed for internal SCSI devices. External connectors are mounted along the outside edge of the host adapter.

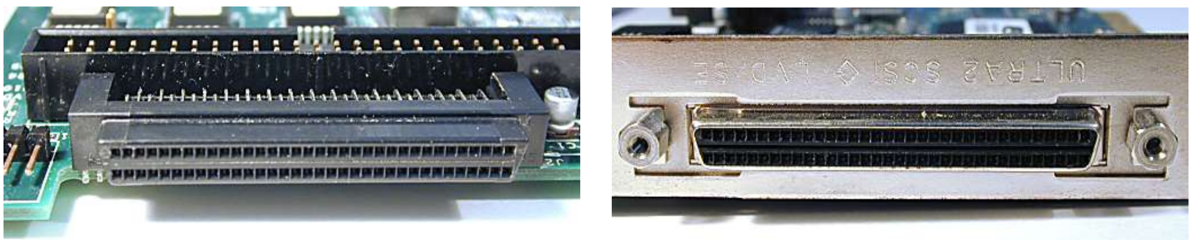


**Figure 5** - Internal (left) and external (right) connectors on a Wide Ultra2 SCSI host adapter. In the upper photo you can see two connectors; facing you is a 68-pin (wide) high-density connector, and facing up is a 50-pin (narrow) "regular density" connector. In the right side photo is a 68-pin (wide) high-density connector.

The exact type of connectors provided on any given card depends on its design, and more specifically, the type of SCSI it is intended to support. A card that is designed to support narrow devices will have narrow (50-pin) connectors, while cards that are built to run wide devices will have 68-pin connectors. There are also different types of each of these two sizes of connector; for example, an older or lower-end host adapter may use the older high-density 68-pin connectors while high-end Ultra160 card may use the smaller very-high-density (VHDCI) connectors.

## Caching

Many higher-end SCSI controllers have built-in caches. The idea behind a cache is to use high-speed memory to hold recently retrieved results, to save time if the results are needed again in the near future. This improves performance because most SCSI devices are drives, much slower than memory in relative terms. Caching is a concept that is used extensively within the PC world; it is found in CPUs, hard disks, optical drives and a variety of other devices

Caching that is done by the SCSI controller adds an additional caching level that exists, logically, in front of the cache that resides within SCSI hard disks or other components. When data is requested from a device on the SCSI bus, the host adapter sees if it is already in its internal cache and if so, returns the results much more quickly, saving transaction bandwidth on the SCSI bus at the same time. This of course improves performance over a system that does not have this type of cache.

The amount of cache memory on a host adapter depends on the model; some are user-upgradeable. Note that there are diminishing returns associated with cache memory; each megabyte you add helps performance less than the previous megabyte did.

## Multiple Segment and Channel Support

In its simplest form, a host adapter provides support for a single SCSI chain: that is, a single set of devices that are all connected together on the same SCSI bus. This is the way that many older and low-end SCSI host adapters work. They are fine for simple implementations, but are too limiting for complex SCSI setups. Especially with modern systems that need to use both LVD and single-ended devices, an adapter with support for just a single segment is insufficient for maximum performance. To expand capabilities, host adapter manufacturers make cards that support multiple *segments*, multiple *channels*, or both.

A *segment* is an electrically-isolated "piece" of a SCSI bus; a single bus can be made up of one or more segments. Cards that implement multiple segments allow for more flexibility because the segments are electrically separate. Each segment can have a cable as long as the normal maximum cable length allowed for that particular type of SCSI, for example. One segment can use an internal cable within the PC and another an external cable. It's important to remember though that two segments on a single channel are *logically* considered to be part of the same SCSI bus even if they are *electrically* separate. This means all devices on all segments must have unique IDs, and that maximum bandwidth is shared between all devices on all segments that make up the bus.

The most expensive host adapters go beyond multiple segment support and actually have multiple *channels*. These are similar in concept to the way an IDE/ATA controller typically has two channels. Each channel is completely independent of the other, both electrically *and* logically. This means the two run in parallel with each other: you get support for twice as many devices, and twice as much throughput. In essence, a card with two channels is two host adapters in the same package. For example, an Ultra160 host adapter with dual channels will support 30 drives (16 per channel less one each per channel for the host adapter) and theoretical throughput of up to 320 MB/s (160 MB/s per channel). Note that each channel can itself have more than one electrical segment.

Host adapters that support multiple channels are not really needed for most applications, especially if already using high-performance SCSI like Ultra160; they are more common in servers than desktop PCs. Multiple segments, however, are commonly found even in desktop SCSI cards. One common use for multiple segments is to allow independent use of LVD and SE devices on the same host adapter without causing the LVD devices to degrade to SE operation. Some cards use multiple channels to isolate LVD and SE, which is probably even better (though it may be more expensive.)

## RAID Support

SCSI is the interface of choice for servers and high-end workstations, where both performance and reliability are critical. One of the most important ways that performance, data integrity and reliability are improved in modern PC systems is through the use of *redundant arrays of inexpensive disks,* or *RAID*. This term simply refers to the use of multiple hard disks in an array, with data spread across the disks. Accessing multiple disks simultaneously allows for faster performance; the optional use of redundancy allows for protection against hardware faults.

Most higher-end RAID solutions use SCSI, so support for RAID is commonly found in SCSI host adapters. In practice, these are not usually sold as "SCSI host adapters with RAID support" but rather are considered as a separate product line: "SCSI RAID controllers”. RAID cards vary widely in terms of features and implementation requirements.

## Drivers and Compatibility Issues

It's important to ensure that your operating system will support whatever host adapter you decide to use. You have to make sure that the card comes with configuration software and drivers that will support whatever operating system and applications you are using. Most well-known cards will either be supported natively in Windows (and other operating systems) or will be provided with good drivers.

Compatibility difficulties may result from the mixing of devices using different types of SCSI transfer modes or feature sets.

One thing that is important to know is that different SCSI host adapters may use different addressing or translating methods to access data on hard disks to which they are connected. This means that switching host adapters can render the contents of the hard disk inaccessible. In some cases the disk must be reformatted after the new host adapter is installed.

## Manual vs. Automatic Configuration

Host adapters vary in terms of the methods that are used for configuring them. Older and cheaper cards, particularly ones that use the ISA bus to connect to the PC, typically require the use of hardware jumpers for configuration tasks such as setting the SCSI device ID for the host adapter, enabling or disabling termination, and so on. These are relatively inconvenient because making changes requires opening up the PC, and in some cases, pulling out the card to tinker with it.

Newer cards, especially those that use the PCI bus, are generally configured through software. This is done either using a separate configuration utility, or the built-in SCSI BIOS, a hardware program that resides on a chip within the host adapter (much like the system BIOS in concept, but dedicated to the SCSI card, not the system as a whole.) Some cards may use both. Some better cards may also automatically configure certain options, such as termination, by detecting which connectors are in use, for example.

## Resource Usage

From the perspective of the PC as a whole, SCSI host adapters are *expansion devices*, since they plug into a system bus and represent a peripheral device on the system bus. Host adapters typically require several different system resources, depending on the system bus that the host adapter is designed for, and the method it is using for transferring data over the system bus.

The following resource types are typically used on various adapters:

- **Interrupt Request Line (IRQ):** All SCSI host adapters use an interrupt request line or IRQ. The most commonly used ones are 9, 10, 11 or 12. IRQs 14 or 15 can usually be used as well if one or the other of the IDE/ATA channels in the system are not being used. It should be noted that PCI-based host adapters will not require an explicit IRQ assignment. Rather, they will use one the system IRQ mapped to whatever PCI slot they are placed into. PCI also supports IRQ sharing.
- **DMA Channel:** Many older host adapters based on the ISA or VLB buses use DMA channels to permit the transfer of data directly from SCSI devices to system memory. Usually DMA channels 1, 3 or 5 are used. PCI-based host adapters typically make use of PCI bus mastering to improve performance, which is a separate type of DMA that does not use regular ISA system DMA channels.
- **I/O Address:** The I/O address is used as the place through which data is transferred to the system. There are several ranges that are used by some SCSI cards.
- **BIOS ROM Memory:** The SCSI BIOS that contains the commands for controlling the host adapter typically takes up a 4000h address location in the upper memory area. This is usually one of the five 4000h spaces in the address range of CC000h to DDFFFh.

Most newer SCSI host adapters, especially those using the PC bus, support the Plug and Play initiative. Plug and Play allows the system to configure resources for the host adapter automatically in many cases, reducing configuration difficulties. However, this is different from so-called Plug and Play SCSI, which is similar in concept but is applied to dynamically allocating SCSI device IDs on the SCSI bus, not PC system resources.

# SCSI Cables, Connectors and Bus Termination

There are many different aspects about SCSI that can be confusing to someone new to the technology--and even someone *not* new to it. Of all of the aspects of SCSI that sometimes cause a bit of difficulty, cables and connector issues are probably the worst. Unlike the IDE/ATA world, where there are a handful of different cable types, with SCSI there are literally *dozens* of different types of cables! It is difficult to even describe all of the options available. This is a result of the flexibility of the SCSI interface--more choice means more options, and hence, more decisions.

In this section, we shall look at the cable and connector hardware most commonly seen on the SCSI interface. There are many different types of cables, and many types of connectors, and to some extent they can be "mixed and matched"--meaning that you may find different types of cables for each connector type and vice-versa.

## General Cable and Connector Issues

The main reason why there are so many types of SCSI cables is simply that there are so many types of SCSI--and so many different ways of implementing them. This great flexibility is actually one of the key strengths of the interface. The design of any SCSI cable is based on a combination of different attributes chosen to implement a particular kind of SCSI bus.

Each SCSI cable must meet the specific electrical requirements associated with the SCSI signaling speeds and methods it supports. This refers not just to obvious matters--such as how many pins are on a particular connector type, or which signals are carried on which wires--but the more complex factors that are the domain of electrical engineering professionals. For example, the thickness of each wire in a cable, the characteristic impedance of the cable, materials used for the wires, connectors and covers, and so on.

The following are other factors that have an impact on the design of SCSI cables, as well as on the selection of cables to meet a particular application:

- **Cable Type:** SCSI is different from most PC interfaces in that it supports both internal and external devices. These use drastically different types of cabling, because the environment inside the PC is very different from that outside it. Both internal and external cables come in a variety of styles themselves.
- **Connector Type:** Different types of connectors are used for different kinds of SCSI. These are only partially dependent on the type of physical cable used; to some extent, connector types are "mixed and matched" with cable technologies to make particular cables.

- **Cable Length:** The maximum length of a SCSI cable is dictated by the signaling type and signaling speed of the interface. However, not all cables are built to the maximum length. Cables of all different lengths are made to suit different needs and budgets (most people don't need 12-meter-long cables for LVD devices, for example, even though they are legal.)
- **Number of Connectors:** Cables vary in terms of the number of connectors they include. Generally speaking, longer cables have more connectors, allowing more devices to be attached to the same SCSI bus segment. Specialized cables may have fewer connectors; for example, LVD cables can be 25m in length instead of the usual 12m if they are used "point to point"--just two devices on the cable.
- **Connector Spacing:** Some types of SCSI have limits regarding how closely two connectors can appear on the cable. If the cable has many connectors you may have to leave some of the connectors unused for maximum performance. In all cases, it is recommended that devices be evenly spaced across the cable.
- **Termination:** Some cables have a built-in terminator at the end of the cable while others require the addition of a separate terminator.
- **General Quality:** The overall quality of a SCSI cable is very important, but is not something tangible that can be easily measured or quantified. Remember that not all cables are created equal. SCSI cables are often the culprits in problematic SCSI buses, so don't compromise on the quality of your cables.

Some companies sell cables with labels such as "SCSI-1 cable", "SCSI-2 cable" or "SCSI-3 cable". With the possible exception of "SCSI-1", these are extremely vague terms that do not tell you nearly enough about the cable to decide if it is the one you want. These terms refer to SCSI standards, which define SCSI *families*, not specific types.

## SCSI Cable Types

The term "SCSI cable" usually refers to a complete cable, including the wire, connectors and possibly a terminator as well. There are a number of different types of cables available; these are combined with various connector types to create specific cable implementations.

SCSI cables come in two distinct varieties: *external* and *internal*. External cables are used to connect SCSI devices that do not reside inside the PC, but rather have their own enclosures and power supplies; internal cables connect SCSI devices installed within the PC system box. These cables are totally different in construction, primarily because the external environment represents much more of a risk to data corruption. This means external cables must be designed to protect the data traveling on the cable. Internal cables don't have this problem because the metal case of the PC shields the components inside from most of the electromagnetic and radio frequency noise and interference from the "outside world". Thus, internal cables can be made more simply and cheaply than external ones.

Let's start by looking at external cables. These are commonly called *shielded cables* because they are made specifically to protect the data they carry from outside interference. They have a very specific design in order to ensure that data traveling on the cable is secured, including the following properties:

- **Twisted Pair Wiring:** All the wires in the cable are formed into pairs, consisting of a data signal paired with its complement. For single-ended signaling, each signal is paired with a ground wire. For differential signaling, each "positive" signal is paired with its corresponding "negative" signal. The two wires in each pair are then twisted together. This twisting improves signal integrity compared to running all the wires in parallel to each other. So an external narrow cable with 50 wires actually contains 25 pairs; a 68-wire cable 34 pairs. This sort of wiring is also commonly used in other applications, such as network cabling, for the same reason.
- **Shielding:** The entire cable is wrapped with a metallic shield, such as aluminum or copper foil or braid, to block out noise and interference.
- **Layered Structure:** The pairs of wires aren't all just tossed into the cable at random; instead, a structure of layers is used. The "core layer" of the cable contains the pairs carrying the most important control signals: REQ and ACK (request and acknowledge). Around that core, pairs of other control signals are arranged in a "middle layer". The outer layer of the cable contains the data and other signals. The purpose of this three-layer structure is to further insulate the most important signals to improve data integrity.

External cables have a round cross-section, reflecting the circular layers mentioned just above. Needless to say, these cables aren't simple to manufacture. All this precise engineering doesn't come without a cost: external SCSI cables are generally quite expensive. For internal cables all these special steps are not required to protect the data in the wires from external interference. Therefore, instead of special shielded, multiple-layer construction, internal devices use *unshielded cables*, which are flat ribbon cables similar to those used for floppy drives and IDE/ATA devices. These are much cheaper than external cables to make.


**Figure 6** - Close-up view of an external SCSI cable

Even with internal cables, there are differences in construction (beyond the width issue, 50 wires for narrow SCSI or 68 wires for wide SCSI). One issue is the thickness of the wires used; another is the insulation that goes over the wires. Better cables generally use Teflon as a wire insulation material, while cheaper ones may use PVC (polyvinyl chloride; vinyl). Regular flat cables are typically used for single-ended SCSI applications up to Ultra speeds (20 MHz).

For Ultra2 or faster internal cables using LVD signaling, the poor electrical characteristics of cheap flat ribbon cables begin to become an issue in terms of signal integrity even within the PC. Therefore, a new type of internal ribbon cable was created for these cables, which actually combines some of the characteristics of regular internal and external cables. With these ribbon cables, pairs are twisted between the connectors on the cable--just like in external cables--but the ribbon remains flat near where the connectors go, for easier attachment. The return to pair twisting improves performance for high-speed SCSI applications, while increasing cost somewhat, though not as much as if external cables are used. This technology is sometimes called "Twist-N-Flat" cable, since it is partially flat and partially twisted-pair.

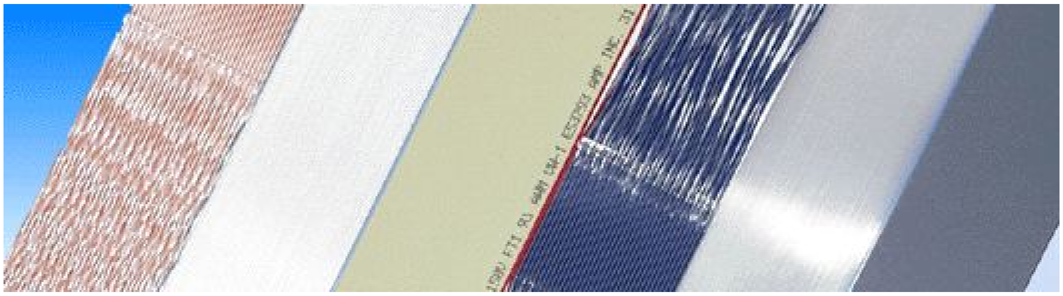


**Figure 7** - An assortment of different internal ribbon cables used for connecting SCSI hardware

## SCSI Connector Types

Connectors are of course the physical devices that are used to attach a SCSI cable to a SCSI device. Several different types of SCSI connectors are used to construct SCSI cables. This is in itself unfortunate in a way; whenever there are multiple types of connectors for an interface, this means the potential exists for mismatched connectors between devices. Different connector types have evolved over the years as the SCSI interface has matured. In particular, the desire for *miniaturization* has been a driving force in the creation of new connector types--the oldest SCSI connectors were large, and creating smaller connectors improves the usability of SCSI cables and devices.

Below are the connector types most commonly seen used with SCSI cables in the PC world. Note that this list is not exhaustive, in part because there are several obscure variations used for some proprietary SCSI implementations. However, most of the cables you will find in the SCSI world use one of these connector types. The SCSI standards call different connector types "alternatives" (not really a good name since the "alternatives" describe different devices types and not really "choices" as that word implies). Since external and internal cables generally use different connectors, each has four different "alternatives". We will begin with external connector types:

- **D-Shell (D-Sub, DD):** The earliest SCSI standard, SCSI-1, defined a 50-pin *D-shell* connector for narrow SCSI implementations. The name of this connector comes from the "D-shaped" metal shell that goes around the pins on the male half of the connector. The design is identical to the 25-pin and 9-pin D-shell connectors used for parallel and serial connections on PCs, but bigger. This connector type was very large and cumbersome and never really caught on. However, an alternative 25-pin version of the D-shell was widely used in the Apple hardware world. (Apple "stripped out" the 25 signal return and ground wires that normally would be paired with the true SCSI signals, to save cost). This also never became a standard in the PC world and is not generally seen unless you go looking for it.

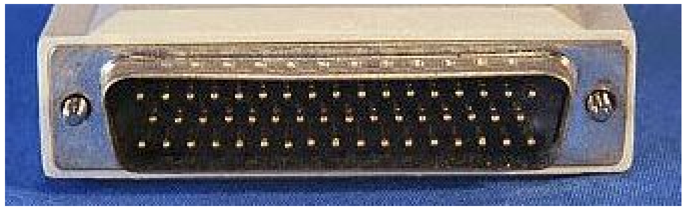


**Figure 8** - A male DD-50 SCSI connector

- **Centronics:** The other external connector type defined by the SCSI-1 standard is a 50-pin connector that is commonly called a *Centronics* connector, after a formerly-popular printer that first used this type of connector. In Centronics connectors, instead of thin pins, two rows of flat contacts are used. Two latches on either side are used to hold the connector in place. Centronics connectors are still used for PC printer cables, on the end that attaches to the printer; SCSI Centronics connectors are the same, just with a different number of pins. These 50-pin connectors are still present in the current SCSI specification and are called "Alternative 2" external connectors.

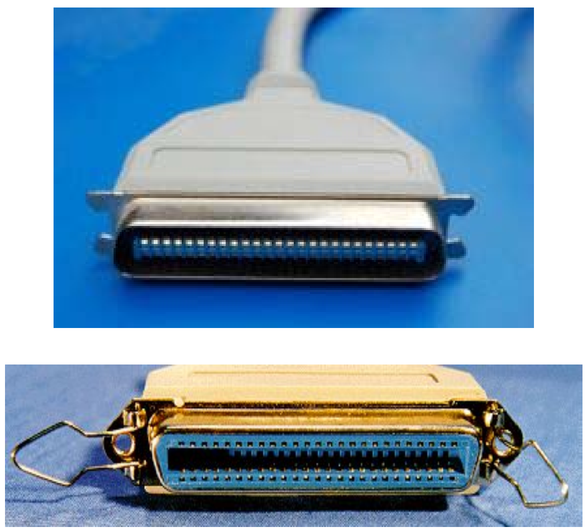
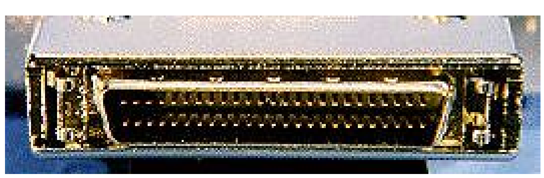


**Figure 9** - Male (above) and female (below) 50-pin Centronics connectors

- **High-Density (HD):** The D-shell connectors defined in the SCSI-1 standard were replaced by newer, *high-density* shielded connectors in SCSI-2. These are really not all that different from the older D-shell connectors, but the space between pins was reduced, making the connectors smaller, cheaper to make and easier to use. The narrow, 50-pin version is called "Alternative 1", and the wide, 68-pin version "Alternative 3". These connectors use a "squeeze to release" latching mechanism instead of Centronics-style latches, and are still used by hardware devices today.



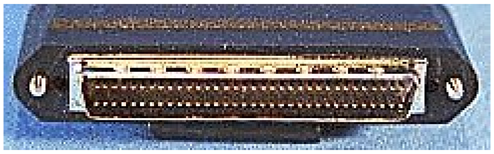


**Figure 10** - Male 50-pin (above) and 68-pin external high density connectors

- **Very High Density Cable Interconnect (VHDCI):** To further improve the flexibility of SCSI hardware, a new type of external connector was defined as part of the SPI-2 standard. This connector is wide only (68 pins) and is sometimes called a "micro-Centronics" connector, because it uses the same design as the Centronics connectors, only with the contacts *much* smaller and closer together. This is "Alternative 4" for external connectors and is growing in popularity because of its small size. One way that VHDCI is useful; for example, is that two of these connectors can be squeezed side-by-side within the width of a single SCSI host adapter's back edge (expansion slot insert). This doubles the number of external connectors that can be crammed onto a high-end SCSI host adapter.

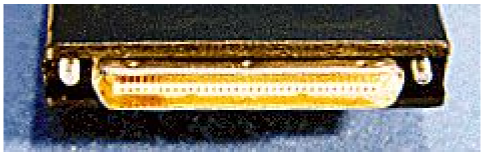


**Figure 11** - A male 68-pin VHDCI connector

Now let's look at internal (unshielded) connectors:

- **Regular Density:** The SCSI-1 standard defined a single connector type for internal narrow (8-bit) devices. This is a rectangular connector with two rows of 25 pins. This connector type is very similar to that used for IDE/ATA devices, except that there are five extra pins in each row. It is most often seen in older devices and also some newer, slower drives. It is called unshielded "Alternative 2" in the current SCSI standards.

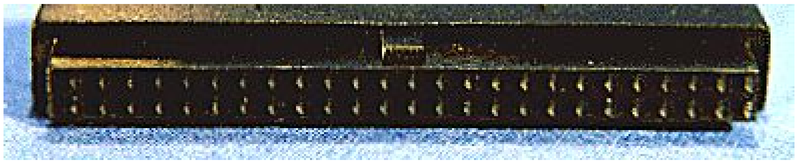


**Figure 12** - Male (above) and female (below) 50-pin regular density internal connectors

- **High Density:** SCSI-2 defined two new connector types, which are both called *high density* because their pin spacing is half that of the older SCSI-1 connectors, making them much smaller. These are the most common SCSI connectors used today within the PC box. The narrow, 50-pin version is unshielded connector "Alternative 1" and the 68-pin version is "Alternative 3".

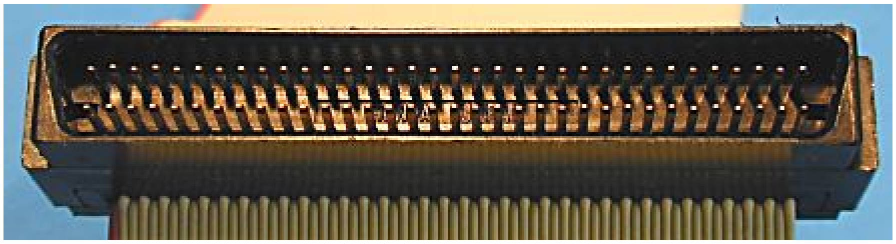


**Figure 13** - A male, internal, high-density 68-pin connector

- **Single Connector Attachment (SCA):** "Alternative 4" in the SCSI standards for unshielded connectors doesn't actually refer to cable connectors, but the connector used for the *single connector attachment* system for backplane-connection of SCSI drives.

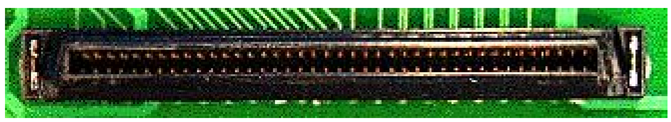


**Figure 14** - A female 80-pin SCA connector

## SCSI Bus Termination

You can do an experiment (either physically or mentally) to illustrate why termination is required on a SCSI bus. Hold one end of a piece of rope about six feet long and have someone else hold the other end. Stretch the string so it is reasonably taut, but not tight, and then snap down on one end sharply. You will form a wave that travels down the string. When it reaches the end of the string it will "reflect" off the end and travel back again toward you, and then reflect again. It will go back and forth across the string, decreasing in amplitude each time until it eventually dies out.

Electrical signals travel across wires in much the same way as physical waves travel across a string. When they reach the end of the wire, they will reflect and travel back across the wire. The problem is that if this is allowed to happen, the reflected signals will interfere with the "real" data on the bus and cause signal loss and data corruption. To ensure that this does not happen, each end of the SCSI bus is *terminated*. Special components are used that make the bus appear *electrically* as if it is infinite in length. Any signals sent along the bus appear to go to all devices and then disappear, with no reflections.

There are several different kinds of termination used on SCSI buses. They differ in the electrical circuitry that is used to terminate the bus. Better forms of termination make for more reliable SCSI chains; the better the termination, the fewer problems (all else being equal) with the bus, though cost is generally higher as well. In general terms, slower buses are less particular about the kind of termination used, while faster ones have more demanding requirements. In addition, buses using differential signaling (either HVD or LVD) require special termination.

The different types of SCSI termination are:

- **Passive Termination:** This is the oldest, simplest and least reliable type of termination. It uses simple resistors to terminate the bus, similar to the way terminators are used on coaxial Ethernet networks. Passive termination is fine for short, low-speed single-ended SCSI-1 buses but is not suitable for any modern SCSI speeds; it is rarely used today.
- **Active Termination:** Adding voltage regulators to the resistors used in passive termination allows for more reliable and consistent termination of the bus. Active termination is the minimum required for any of the faster-speed single-ended SCSI buses.
- **Forced Perfect Termination (FPT):** This is a more advanced form of active termination, where diode clamps are added to the circuitry to force the termination to the correct voltage. This virtually eliminates any signal reflections or other problems and provides for the best form of termination of a single-ended SCSI bus.
- **High Voltage Differential (HVD):** Buses using high voltage differential signaling require the use of special HVD terminators.
- **Low Voltage Differential (LVD):** Newer buses using low voltage differential signaling also require their own special type of terminators. In addition, there are special LVD/SE terminators designed for use with multimode LVD devices that can function in either LVD or SE modes; when the bus is running single-ended these behave like active terminators.


**Figure 15** - Internal (above) and external (below) active terminators

Terminators must be at the very ends of the bus, after all of the actual devices on the chain. This includes any devices that may be powered off or temporarily disconnected. Therefore, there are always exactly two terminators per bus or bus segment. Many devices contain internal terminators that can be used if the device is at one of the ends of the SCSI bus. However, differential drives typically do not include the ability to terminate the bus, so newer LVD applications require explicit terminator hardware. Sometimes terminators are built in to the end of the SCSI cable. In addition, systems using the Single Connector Attachment (SCA) system have a different termination arrangement because the connection system is different. SCA drives do not have termination on them.

Host adapters usually do include the ability to terminate the SCSI bus. In fact, many host adapters include multiple segments, and so have the ability to terminate each segment they support. Termination should only be enabled on a host adapter if the host adapter is the last device on any segment. If you are using both internal and external devices on a host adapter that has only one logical segment being shared by both internal and external drives, the host adapter is going to typically be in the middle of the chain between them, and its internal termination should be disabled.

Termination is a rather straight-forward affair when all of the devices on the SCSI bus are the same width: either narrow (regular, 8 bit) or wide (16 bit) SCSI. When you mix narrow and wide SCSI on the same bus, you must be more careful about termination. The issue that arises is that if part of the device is running in wide mode, but not all devices are wide, half of the data lines (the "high byte") may end somewhere on the bus; they need to be terminated, and that termination may occur in a different place than where the "low byte" data signals are terminated.

Normally these issues are handled using special adapters or cables that only extend the extra width to the devices that are using the wide portion of the bus. However, the extra signals on the wide part of the bus must *also* be terminated properly. Problems can result with wide devices when these extra signals are not terminated and are left "dangling".

## Summary of SCSI Cables and Connectors

The table below provides a quick reference summary of the different connector types used for both internal and external SCSI cables:

| Cable Type | Connector "Alternative" | Connector Type | Contacts | Cable Name |
|---|---|---|---|---|
| **External** | **1** | High Density | 50 | External High Density "A" Cable |
| | **2** | Centronics | 50 | External Centronics "A" Cable |
| | **3** | High Density | 68 | External High Density "P" Cable |
| | **4** | VHDCI | 68 | External Very High Density |

| | | | | |
|---|---|---|---|---|
| | | | | "P" Cable |
| **Internal** | **1** | High Density | 50 | Internal High Density "A" Cable |
| | **2** | Regular Density | 50 | Internal Regular Density "A" Cable |
| | **3** | High Density | 68 | Internal High Density "P" Cable |
| | **4** | SCA | 80 | (n/a) |

**Table 8** - Connector types for internal and external SCSI cables

# SCSI command protocol

In addition to many different hardware implementations, the SCSI standards also include a complex set of command protocol definitions. The SCSI command architecture was originally defined for parallel SCSI buses but has been carried forward with minimal change for use with iSCSI and serial SCSI.

In SCSI terminology, communication takes place between an initiator and a target. The initiator sends a command to the target which then responds. SCSI commands are sent in a Command Descriptor Block (CDB). The CDB consists of a one byte operation code followed by five or more bytes containing command-specific parameters.

At the end of the command sequence the target returns a Sense Code byte which is usually 00h for success, 02h for an error (called a Check Condition), or 08h for busy. When the target returns a Check Condition in response to a command, the initiator usually then issues a SCSI Request Sense command in order to obtain a Key Code Qualifier (KCQ) from the target. The Check Condition and Request Sense sequence involves a special SCSI protocol called a Contingent Allegiance Condition.

There are 4 categories of SCSI commands: N (non-data), W (writing data from initiator to target), R (reading data), and B (bidirectional). There are about 60 different SCSI commands in total, with the most common being:

- Test unit ready - "ping" the device to see if it responds
- Inquiry - return basic device information
- Request sense - give any error codes from the previous command
- Send diagnostic and Receive diagnostic results - run a simple self-test, or a specialised test defined in a diagnostic page
- Start/Stop unit
- Read capacity - return storage capacity
- Format unit
- Read (4 variants)
- Write (4 variants)
- Log sense - return current information from log pages
- Mode sense - return current device parameters from mode pages
- Mode select - set device parameters in a mode page

Each device on the SCSI bus is assigned at least one logical unit number (LUN). Simple devices have just one LUN, more complex devices may have multiple LUNs. A storage device consists of a number of logical blocks, usually referred to by the term Logical Block Address (LBA). A typical LBA equates to 512 bytes of storage.

The usage of LBAs has evolved over time and so four different command variants are provided for reading and writing data. The Read(6) and Write(6) commands contain a 21-bit LBA address. The Read(10), Read(12), Read Long, Write(10), Write(12), and Write Long commands all contain a 32-bit LBA address plus various other parameter options.

# Glossary

## A

ANSI - American National Standards Institute
ATA - AT Attachment

## B

BIOS – Basic Input Output System

## C

CAC - Contingent Allegiance Condition
CCS - Common Command Set
CDB - Command Descriptor Block
CRC - Cyclic Redundancy Check

## D

DB - Data Bit
DT - Double Transition
DV - Domain Validation

## F

FC-AL - Fibre Channel Arbitrated Loop
FPT - Forced Perfect Termination

## H

HBA - Host Bus Adapter
HD - High Density
HVD - High Voltage Differential

## I

I/O - Input/Output
IDE - Integrated Drive Electronics
IEEE - Institute of Electrical and Electronics Engineers
IRQ - Interrupt ReQuest line
iSCSI - Internet SCSI
ITIC - Information Technology Industry Council

## K

KCQ - Key Code Qualifier

## L

LBA - Logical Block Address
LUN - Logical Unit Number
LVD - Low Voltage Differential

## N

NCITS - National Committee for Information Technology Standards
NCQ - Native Command Queuing
NCR - National Cash Register

## P

PC - Personal Computer
PCI - Peripheral Component Interconnect

## Q

QAS - Quick Arbitration and Selection

## R

RAID - Redundant Arrays of Inexpensive Disks

## S

SAM - SCSI-3 Architecture Model
SAS - Serial Attached SCSI
SASI - Shugart Associates Systems Interface
SCA - Single Connector Attachment
SCAM - SCSI Configured AutoMatically
SCSI - Small Computer System Interface
SDO - Standards Developing Organizations
SE - Single Ended
SIP - SCSI-3 Interlocked Protocol
SPI - SCSI-3 Parallel Interface
SSA - Serial Storage Architecture
STA - SCSI Trade Association

## V

VHDCI - Very High Density Cable Interconnect

# Special Interest Areas

Storage interfaces have been a necessary component of computer systems since computing's inception. At a basic level, a storage interface functions like any generic interface, defining the boundary between two dissimilar surfaces or systems. In a computer system, a storage interface defines both the boundaries between storage devices—such as hard drives, tape drives, or similar media—and how those dissimilar computing resources engage one another to work as a coherent system.

Today's storage interface arena consists of diverse industry standards combined with R&D investments from major industry players who continue to aid in the evolution of these numerous technologies. Although SCSI—the Small Computer System Interface—is probably the most pivotal standard in use today, other crucial storage protocols include Fibre Channel, IEEE 1394, Serial ATA, and iSCSI.

The table below lists the features and benefits of Serial Attached SCSI devices:

| **Features** | **Benefits** |
|---|---|
| Leverages industry standards | Improved interoperability<br>Performance roadmap to 6.0 Gbps |
| Allows coexistence with SATA disk drives | Flexible price-performance points |
| Point-to-point architecture supports more than 128 devices | Ease of scalability<br>Flexible topologies |
| Thinner cables and fewer signals<br>Smaller connectors | Improved cable routing, airflow, and cooling |

**Table 9** - Features and benefits of SAS devices

# Ongoing Research

Data storage plays an essential role in today's fast growing data-intensive network services. SCSI is one of the most recent standards that allow SCSI protocols to be carried out over IP networks.

The SCSI interface is an area where constant research is being conducted. The parallel SCSI interface has reached its physical limitations and now the direction of ongoing research is toward serial SCSI. However, the magnitude and speed of the research is not very fast as judged from the fact that few research papers have been published about this topic in the last few years in international journals. While this research report mainly deals with parallel SCSI, a small amount of info about serial SCSI has also been included as an interest area.

Serial SCSI currently offers the highest data rates in the whole SCSI family. Various instances of serial SCSI are under research like iSCSI which deals with data transfer using SCSI devices over the Ethernet and the internet in general. Another area is and FC-AL SCSI. Serial Attached SCSI is like traditional SCSI, offering the benefits of Serial SCSI under normal usage.

Some of the topics from already published research papers which piqued my interest during this research relating to SCSI are:

- A Caching Strategy to Improve iSCSI Performance
- Performance Analysis of the iSCSI Protocol
- Design of the iSCSI Protocol

Details of these research papers have been provided at the end in "References" section for those who might be interested in reading them.

Furthermore, topics like Storage Area Networks have been covered in IEEE Communications Magazine's April 2004 edition, which provide an interesting read into the technologies involved. IEEE Communications Magazine's August 2003 edition covers iSCSI and provides good info about it.

# References

Various sources of information were consulted during this research. They are listed as under:

- **Books**

  Unfortunately, the NIIT library, at the time of this research, did not possess any book relating to SCSI interface. Hopefully, students who will research this topic in future will have resources pertaining to SCSI available from the library as I have advised the librarian on suitable books for this topic which are in the process of being acquired..

- **Magazines**

  IEEE Communications Magazine, August 2003

  IEEE Communications Magazine, April 2004

- **Internet**

  The various online sites used/consulted for info on this topic are listed as under:


  - http://www.google.com
  - http://www.paralan.com
  - http://www.pcguide.com
  - http://www.webopedia.com
  - http://www.wikipedia.org
  - Images courtesy of http://www.cablemakers.com
  - http://www.ieee.org
  - http://www.acm.org
  - http://www.scsita.org
  - http://www.njp.org
  - http://www.gutenberg.org
  - http://www.sciencedirect.com
  - http://scitation.aip.org

- **Research papers**
  - ***"A Caching Strategy to Improve iSCSI Performance"*** by Xubin He of *Tennessee Technological University* and Qing Yang, and Ming Zhang of *University of Rhode Island* at the proceedings of the *27th Annual IEEE Conference on Local Computer Networks* in 2002.

  - "**A Performance Analysis of the iSCSI Protocol"** by Stephen Aiken, Dirk Grunwald, Andrew R. Pleszkun and Jesse Willeke of *University of Colorado, Boulder* at the proceedings of *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies* in 2003.

  - *"**Design of the iSCSI Protocol**"* by Kalman Z. Meth, Julian Satran of *IBM Haifa Research Laboratory* at the proceedings of the *20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies* in 2003.

# Conclusion

SCSI is a very versatile interface. From its inception to the present time, it has passed the test of time and has been developed accordingly to compete with other similar technologies.

Starting from SCSI Parallel Interface and now progressing to SCSI serial interface, the increase in data transfer rates is phenomenal. Merging SCSI and IP has resulted in iSCSI and has provided new avenues of research for people who are interested in this particular topic of study. FC-AL is another flavor of serial SCSI which has been a cause of additional interest in SCSI.

Admittedly, SCSI is a rather confusing topic, in major part due to the large number of standards and flavors of SCSI. The plethora of cabling, connector and terminating varieties add to the confusion. However, once a person masters it, there are a lot of opportunities in the field. The ongoing research provides lots of opportunities for the enterprise market who are the biggest users of SCSI related products. Indeed, as time passes, SCSI will provide an interesting avenue for those who want to join the research effort and even those who are just interested in computing technology and follow it avidly.