SCSI

- **Parallel SCSI**
- **Serial SCSI = SAS (Serial Attached SCSI)**

Parallel SCSI

- Parallel 16 bit disk interface
- High Performance
- BUS based interfaces-up to 16 devices per bus
- Disconnect / reconnect capability

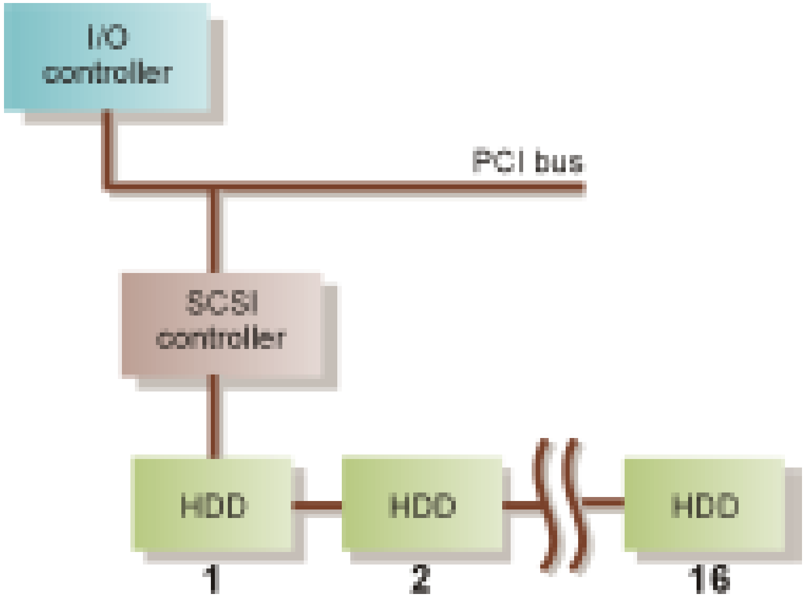
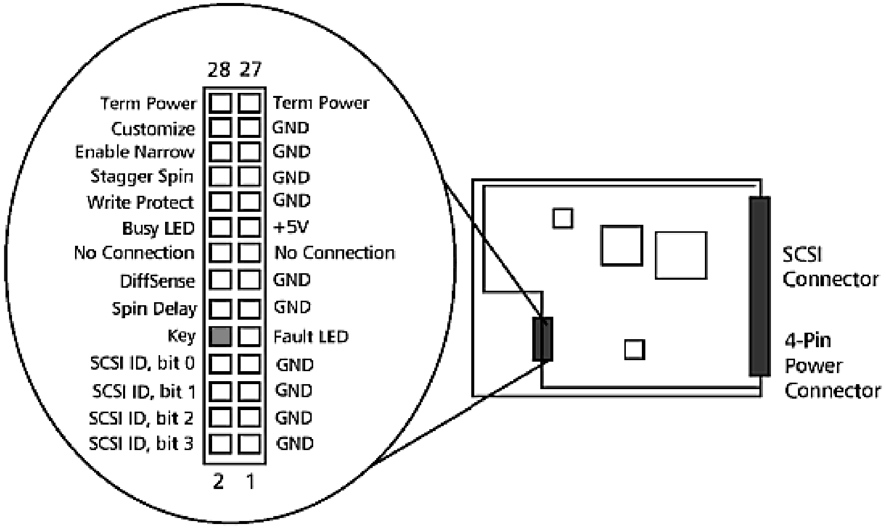


Figure 4. Ultra2 SCSI With up to 16 Peripheral Devices



SCSI BUS

- **Arbitration phase** — Initiator gains control of the SCSI bus.
- **Selection phase** — Initiator selects a device for communication.
- **Message phase** — Messages are exchanged that identify specific devices and specify parameters for data-transfer transactions.
- **Command phase** — Initiator sends out a request for action.
- **Data phase** — Data is transferred.
- **Status phase** — The status of the last operation is posted.

History of parallel SCSI

Interconnect	Standard	Year	Speed	Key features
SASI		1979		Shugart Associates
SCSI-1	SCSI-1	1986	~ 2 MB/sec	Asynchronous, narrow
SCSI-2	SCSI-2	1989	10 MB/sec	Synchronous, wide
SCSI-3	Split command sets, transport protocols, and physical interfaces into separate standards			
Fast-Wide	SPI/SIP	1992	20 MB/sec	
Ultra	Fast-20 annex	1995	40 MB/sec	
Ultra 2	SPI-2	1997	80 MB/sec	LVD
Ultra 3	SPI-3	1999	160 MB/sec	DT, CRC
Ultra 320	SPI-4	2001	320 MB/sec	Paced, Packetized, QAS

Overview of SCSI-3 Standards

Protocol	Description	Document	Abbreviation and Generation	Status	Standard or Project
Interlocked (Parallel Bus)	Defines the protocol for "regular" parallel SCSI	*SCSI-3 Interlocked Protocol*	*SIP*	Withdrawn; now incorporated into later versions of the SCSI-3 Parallel Interface	--
Fibre Channel	Defines the protocol for running SCSI on the Fibre Channel interface	*SCSI-3 Fibre Channel Protocol*	*FCP*	Published	X3.269-1996
			FCP-2	Pending Publication	T10 1144-D
Serial Bus	Defines the protocol for transporting commands over the IEEE-1394 (serial) interface	*Serial Bus Protocol*	*SBP*	Withdrawn	--
			SBP-2	Published	NCITS.325-1998
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*	Published	NCITS.309-1998
		Serial Storage Architecture Transport Layer	*SSA-TL2*	Published (replaced SSA-TL1)	NCITS.308-1998

SCSI Bus Speed

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

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.

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.

Negotiation and Domain Validation

- 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. (If this sounds similar to the way that two regular analog modems determine a communications speed, that's because it is!)
- **Domain validation** is one of the five "optional" features of Ultra3 SCSI, and is a required feature for hardware meeting the Ultra160 or Ultra160+ 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
or idle. If active, the bus is busy transmitting dat ,
it is available for a device to begin sending a co
decides it wants to use the bus, it "bids" for cont t
other devices on the bus will want to use it at th
for control. A specific method is used to resolve
device gets to use the bus first; this is based to
respective priority levels. This process is called

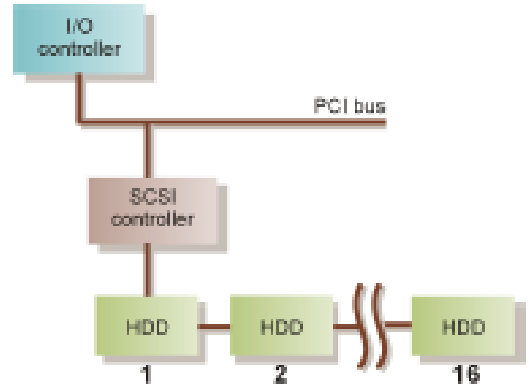


Figure 4. Ultra2 SCSI With up to 16 Peripheral Devices

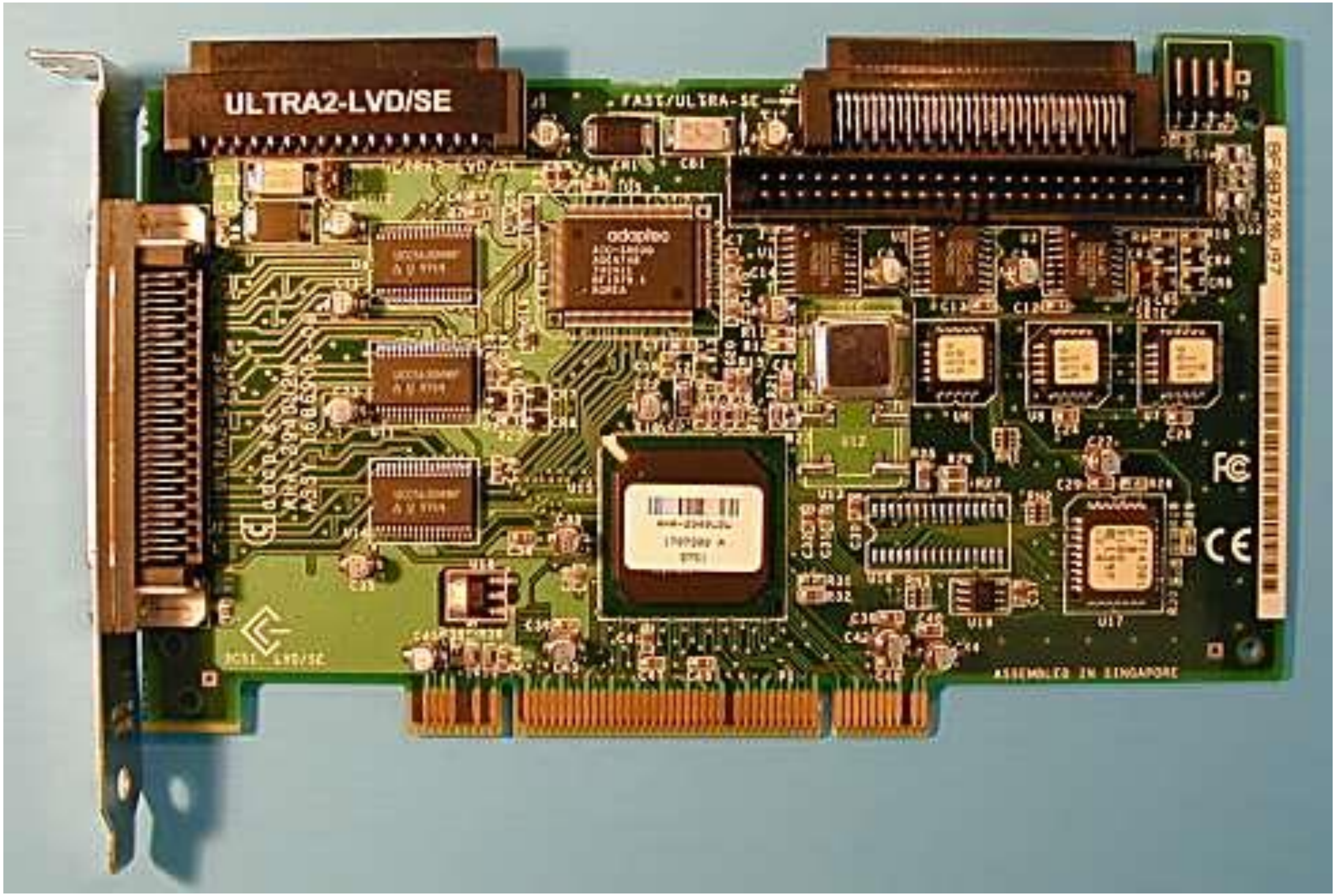
- While arbitration works fine in regular SCSI con
During the time that arbitration is going on, no d
so it makes sense that doing this faster will allo e
SCSI subsystem. To this end, the SPI-3 standa defined a feature that reduces the overhead required for arbitration. This feature is called *quick 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+ devices. It may also be part of the requirements for Ultra320 SCSI when that specification is complete.

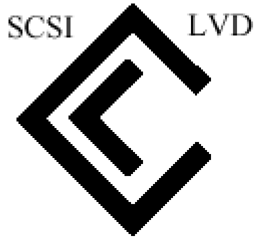
SCSI kontroleri

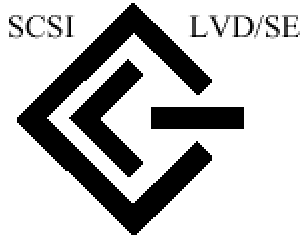
SCSI Host Adapters



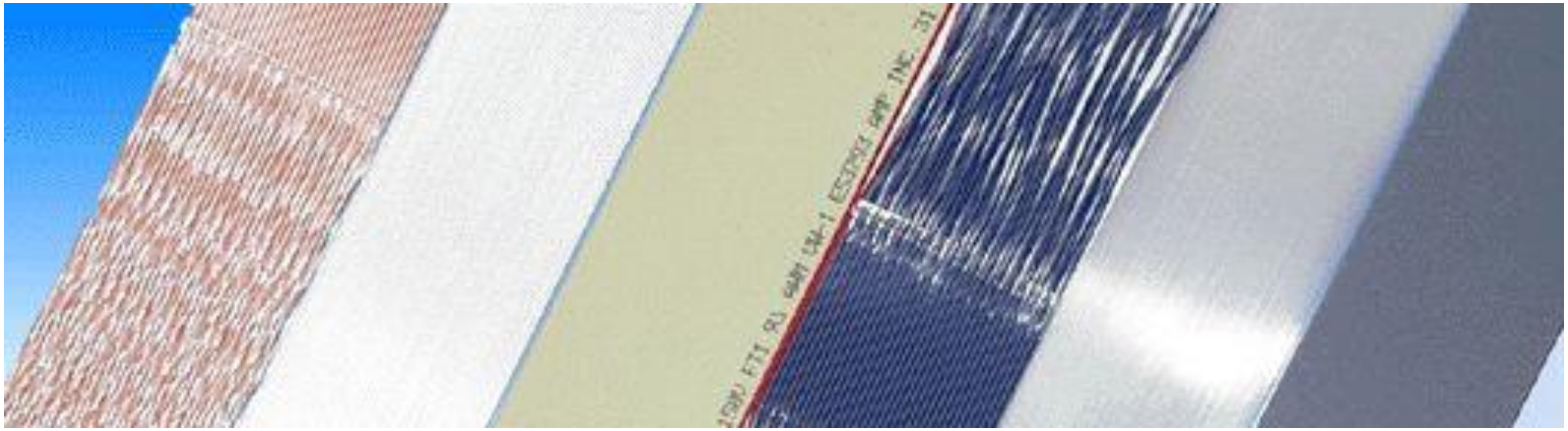
Low-Voltage Differential (LVD) Signaling

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



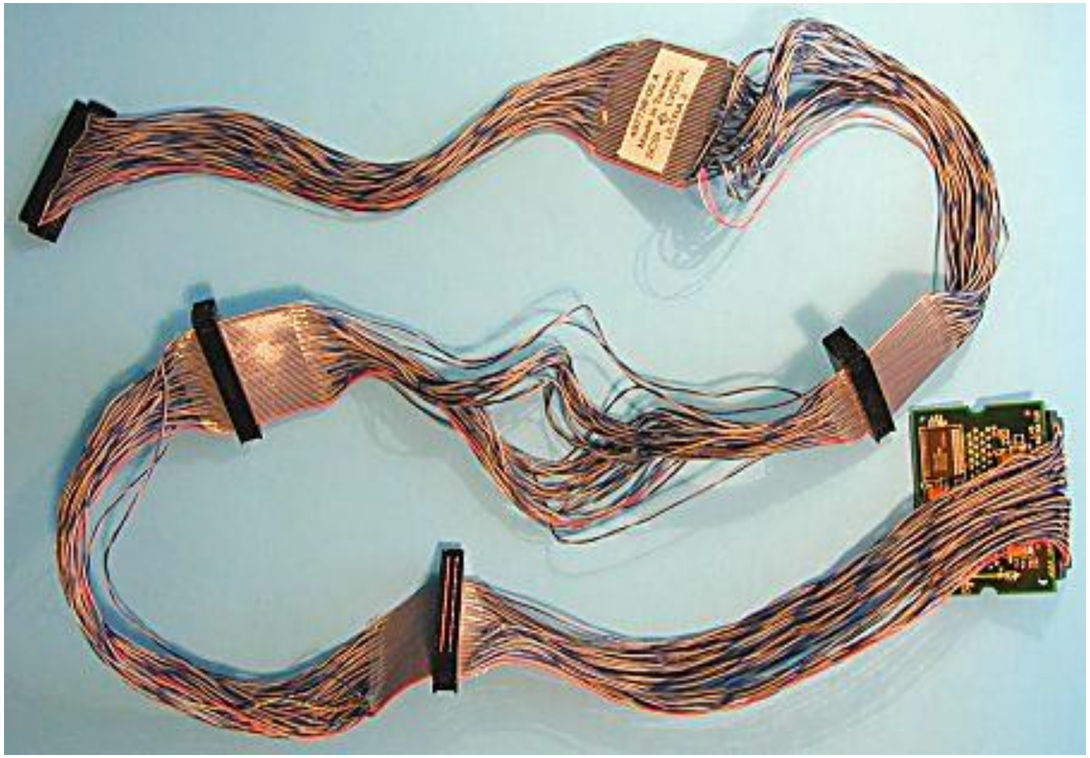


SCSI Cable Types



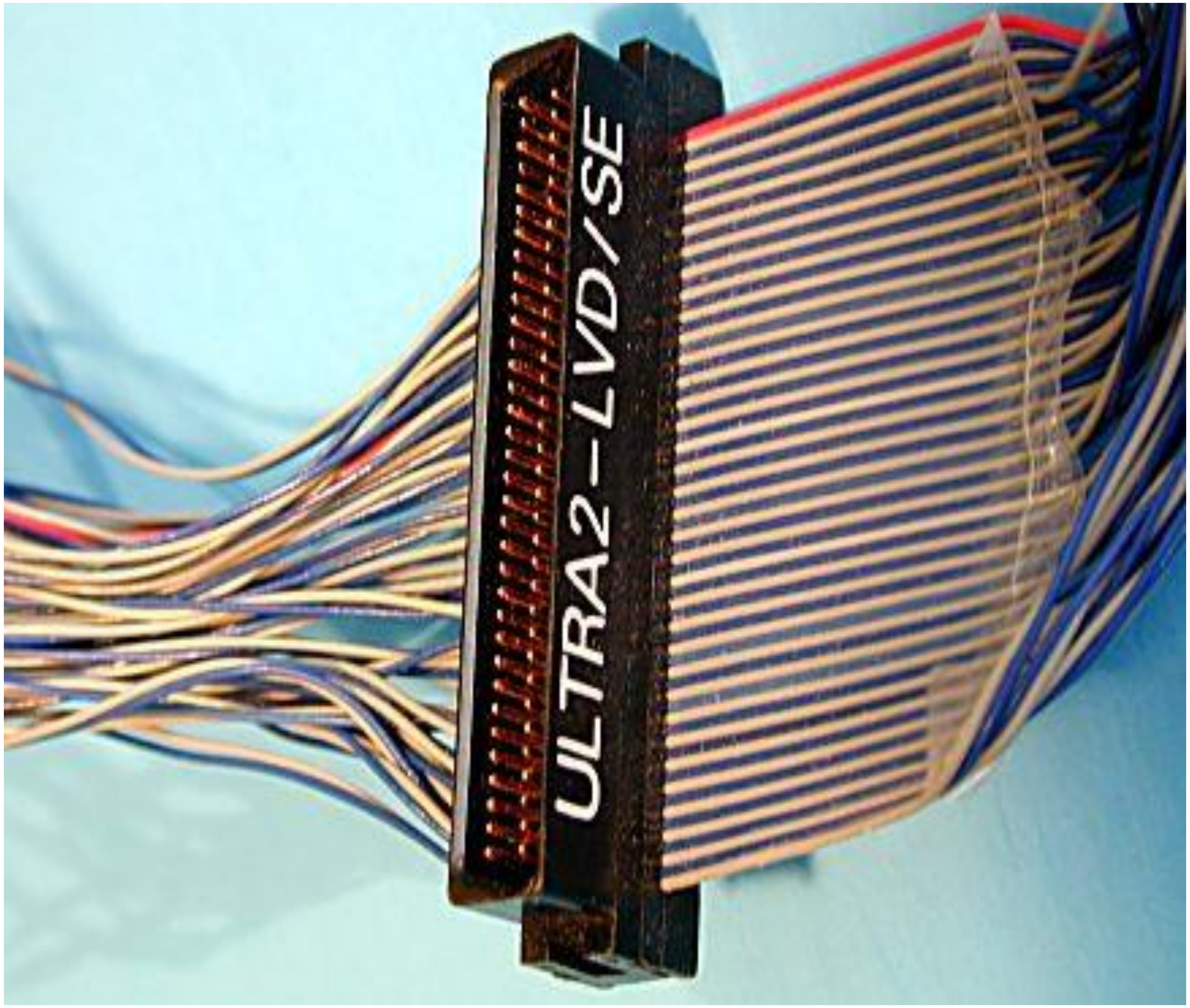
- An **assortment** of different internal ribbon cables used for connecting SCSI hardware. Note that some are strictly flat cables, but the one on the far left and the one third from the right are partially flat and partially twisted pair cable.

Low Voltage Differential (LVD) Cables, Connectors and Signals



- An internal, **68-wire**, 5-connector LVD cable. Note the distinctive "loose" twisted pair wiring between the connectors. The circuit board at bottom right is an integrated LVD/SE terminator. (Incidentally, one of the connectors is hidden behind the terminator and hard to see.)

SCSI kablovi



Single Connector Attachment (SCA)

- **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.

- A female 80-pin SCA connector. This is the connector that would be found on a backplane designed for SCA SCSI

Tipičan SCSI disk

Quantum Atlas 10K II	
maximum Capacity	73.0 GB
average seek time	4.7ms
rotational speed	10.000 rpm
media speed	54MB/s
disk interface	ultra 160
maximum disk buffer throughput	160 MB/sec
disk buffer	8 MB

Primeri za razumevanje

- PRIMER ZA VEŽBU BR. 1
  - SCSI diskovi
- PRIMER ZA VEŽBU BR. 2
  - SCSI taktovi i brzine prenosa

PRIMER BR. 1
< SCSI diskovi >

- **Date su karakteristike jednog realnog SCSI diska**.

Quantum Atlas 10K II	
maximum Capacity	73.0 GB
average seek time	4.7ms
rotational speed	10.000 rpm
media speed	54MB/s
disk interface	ultra 160
maximum disk buffer throughput	160 MB/sec
disk buffer	8 MB

- U konfiguraciji računara imate 2 takva diska.
- Pristigla su 2 zahteva, za svaki disk po jedan
- **disk #1:** sa adrese (cilindar 15, head 0, sektor 13), pročitati 10 sektora
- **disk #2:** sa adrese (cilindar 940, head 2, sektor 1), pročitati 40 sektora
- odrediti ukupno vreme trajanja oba disk pristupa bez preklapanja media i interface faze

PRIMER BR. 1
< SCSI diskovi >

- Rešenje:
- Podjimo od šeme SCSI magistrale

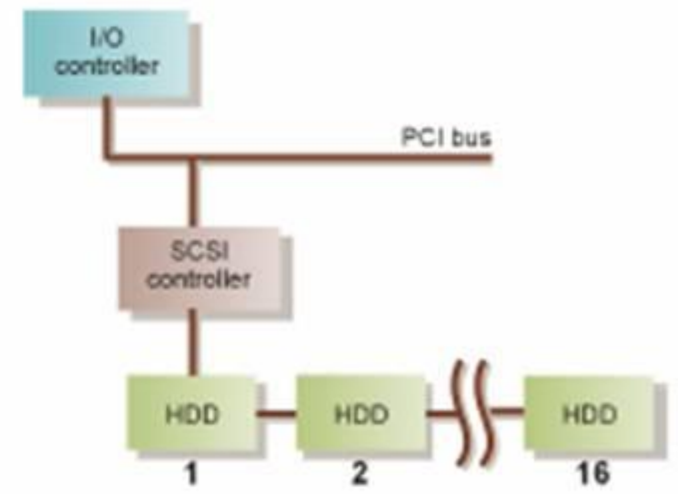


Figure 4. Ultra2 SCSI With up to 16 Peripheral Devices

- SCSI magistrala je multi-task orijentisana, dozvoljava više od jedne disk operaciju u jednom trenutku.
- U našem slučaju, 2 SCSI disk su multi-tasksing orijentisana, svaki radi nezavisno, tako da se 2 disk pristupa mogu istovremeno odvijati.

PRIMER BR. 1
< SATA diskovi >

- Oba diska su SCSI
- i ovde se disk operacije preklapaju, obe se rade istovremeno
- **Ttotal=Max(Taccess1, Taccess2)**
- disk #1: sa adrese (cilindar 15, head 0, sektor 13), pročitati 10 sektora
- Taccess1 = Tseek1 + Trotate1 + Tmedia1 + Tinterface1
- Tseek1 = 1 x average seek time = 4.7 msec
- Trotate1 = Trotate_avg(10000 rpm)   = 3 msec
- Tmedia1 = Q/Vmedia=10x0.5KB/54MB/s= 0.093 msec
- Tinteface1=Q/Vinterface=10x0.5KB/160MB/s= 0.03125msec
- Taccess1= 7.824 msec

PRIMER BR. 1
< SCSI diskovi >

- **disk #2: sa adrese (cilindar 940, head 2, sektor 1), pročitati 40 sektora**

- Taccess2 = Tseek2 + Trotate2 + Tmedia2 + Tinterface2
- Tseek2 = 1 x average seek time = 4.7 msec
- Trotate1 = Trotate_avg(10000 rpm) = 3 msec
- Tmedia2 = Q/Vmedia=40x0.5KB/54MB/s= 0.37msec
- Tinteface2=Q/Vinterface=40x0.5KB/160MB/s= 0.125msec
- Taccess2= 8.195msec

- **Ttotal= Max(7.824, 8.195) = 8.195msec**

PRIMER BR. 2
<SCSI taktovi i brzine prenosa >

- **Zadatak 1.**
- Odrediti osnovni UDMA takt za SCSI ultra 320?

- Rešenje:
- Prvo ćemo poći od formule:
- V=(Frequency x 2 x 16)/8
- Ultra 320 transfer rate=
- X MHz strobe
- x 2 for double data rate clocking
- x 16 for bits per edge
- / 8 bits per byte
- = 320 Mbytes/sec

- **Frequency= 8xV / (2x16) = 8x320/32=80MHz**