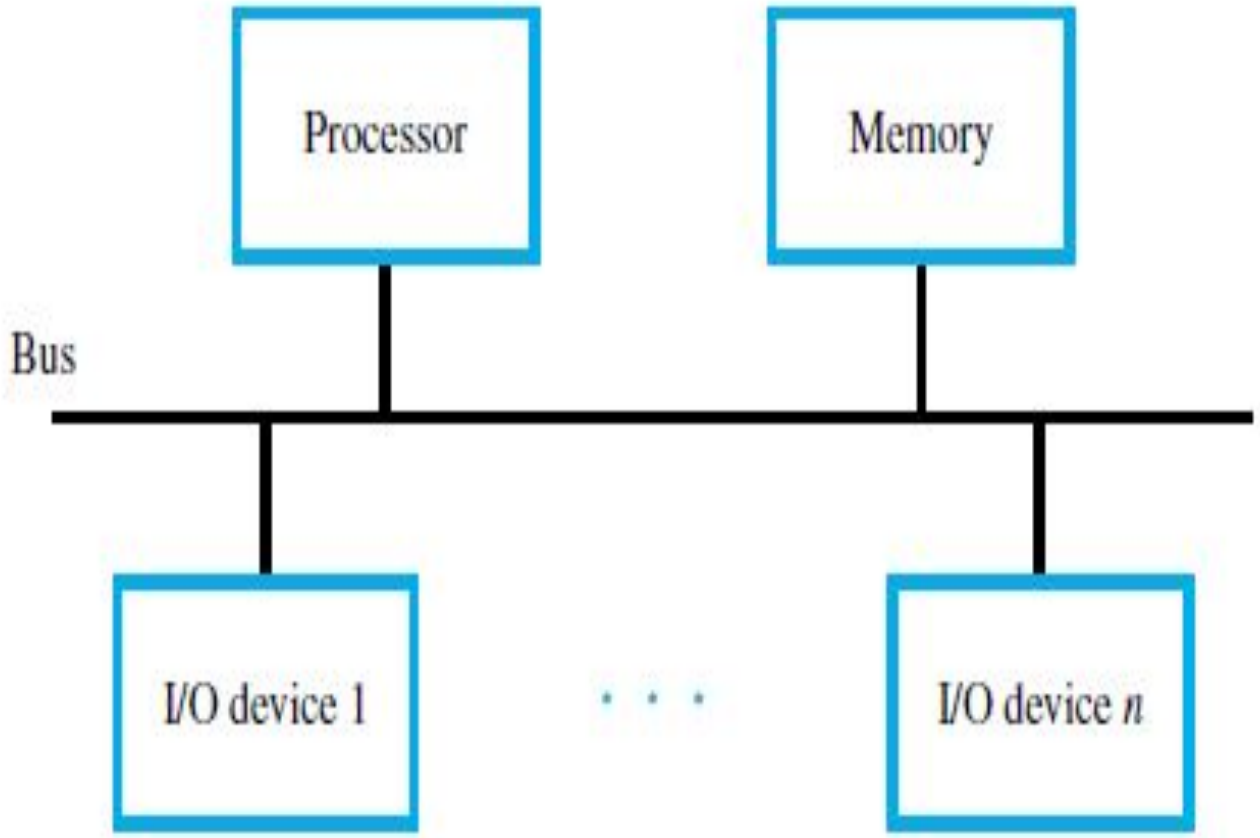
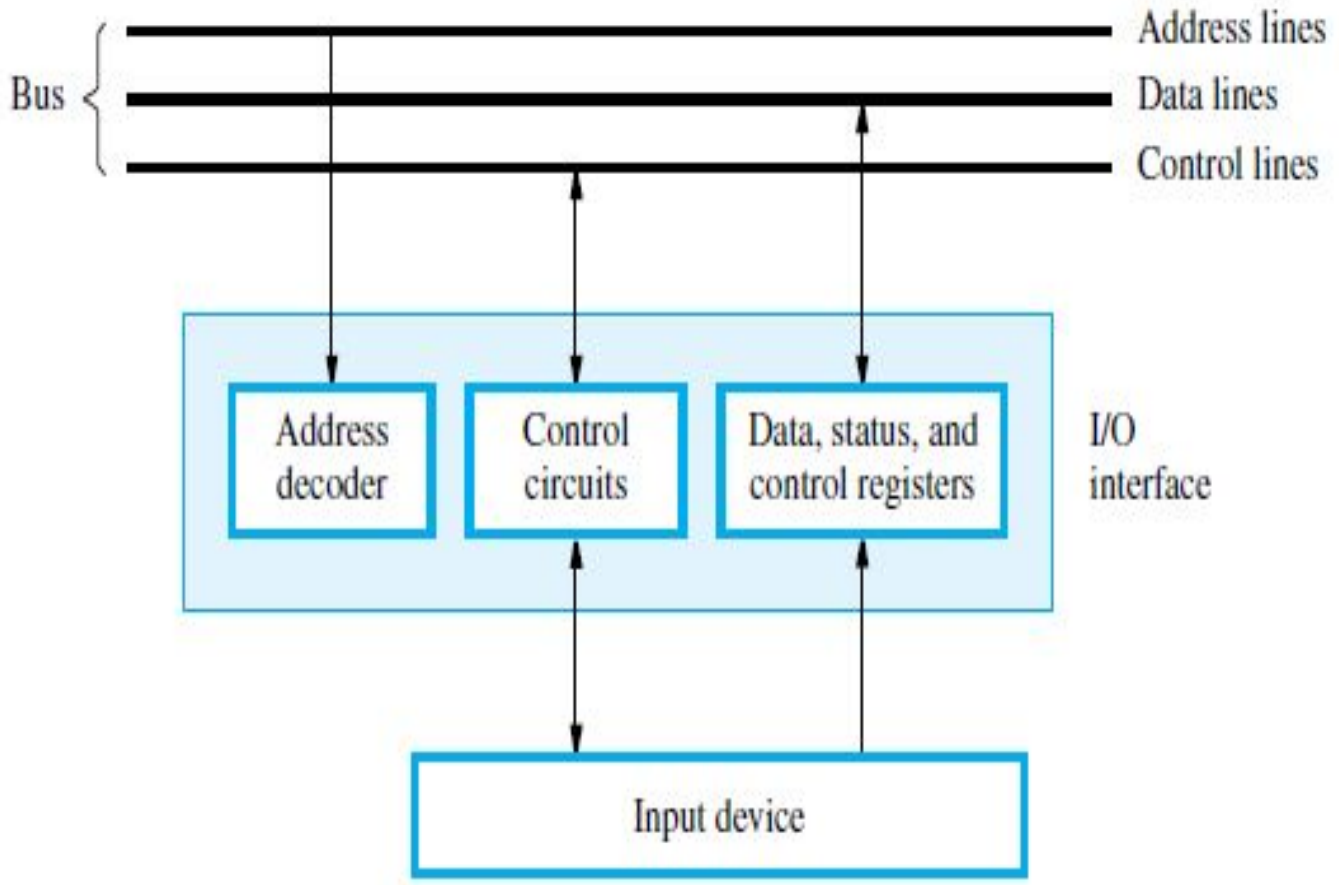
Input / Output Organization: Bus Structure - Bus Operation – Arbitration - Interface Circuits - Interconnection Standards

Bus Structure

- The bus consists of three sets of lines used to carry address, data, and control signals. I/O device interfaces are connected to these lines for an input device.
- Each I/O device is assigned a unique set of addresses for the registers in its interface.
- When the processor places a particular address on the address lines, it is examined by the address decoders of all devices on the bus.
- The device that recognizes this address responds to the commands issued on the control lines.
- The processor uses the control lines to request either a Read or a Write operation, and the requested data are transferred over the data lines.



A single-bus structure.



I/O interface for an input device.

For example,
if the input device is a keyboard and if DATAIN is its data register, the instruction

Load R2, DATAIN

reads the data from DATAIN and stores them into processor register R2. Similarly, the instruction

Store R2, DATAOUT

sends the contents of register R2 to location DATAOUT, which may be the data register of a display device interface.

The status and control registers contain information relevant to the operation of the I/O device.

The address decoder, the data and status registers, and the control circuitry required to coordinate I/O transfers constitute the device's interface circuit.

Bus Operation

- The bus protocol determines when a device may place information on the bus, when it may load the data on the bus into one of its registers, and so on.

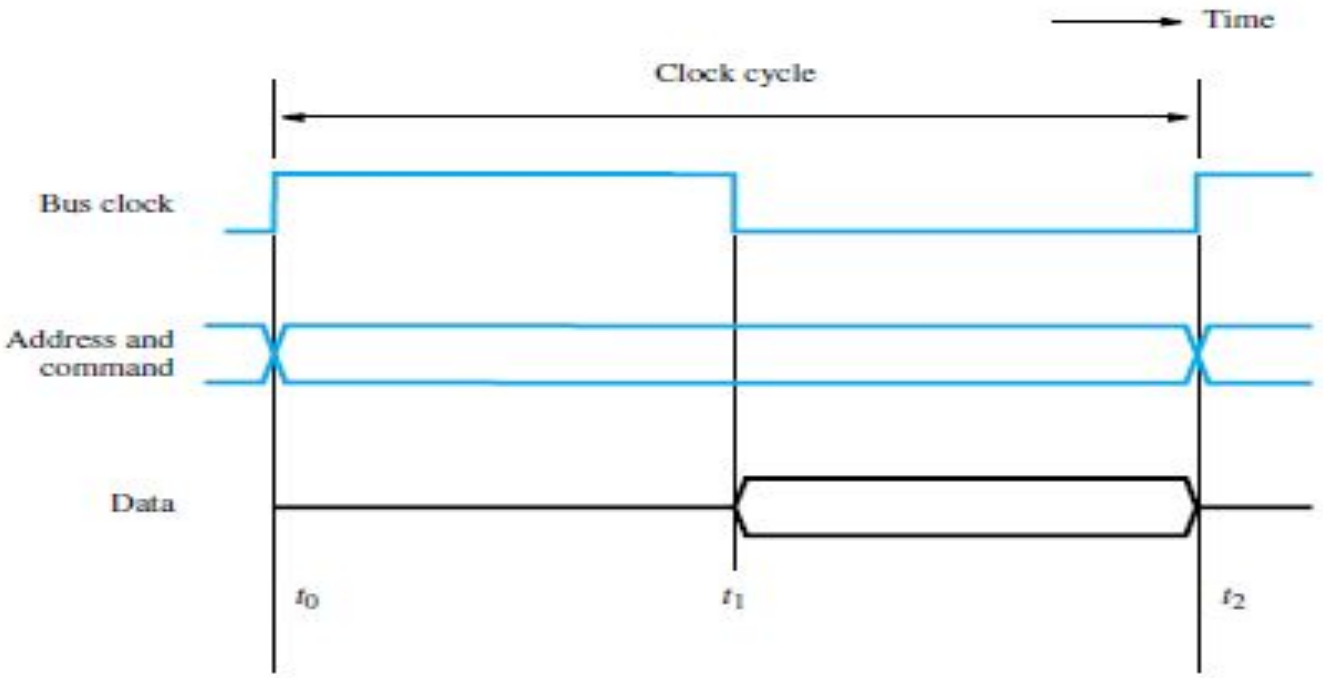
- These rules are implemented by control signals that indicate what and when actions are to be taken.

- Master or initiator- Device which initiate data transfers

- Slave or target- Device addressed by master

Synchronous Bus

- All devices derive timing information from a control line called the bus clock
- The timing diagram shows an idealized representation of the actions that take place on the bus lines.



Timing of an input transfer on a synchronous bus.

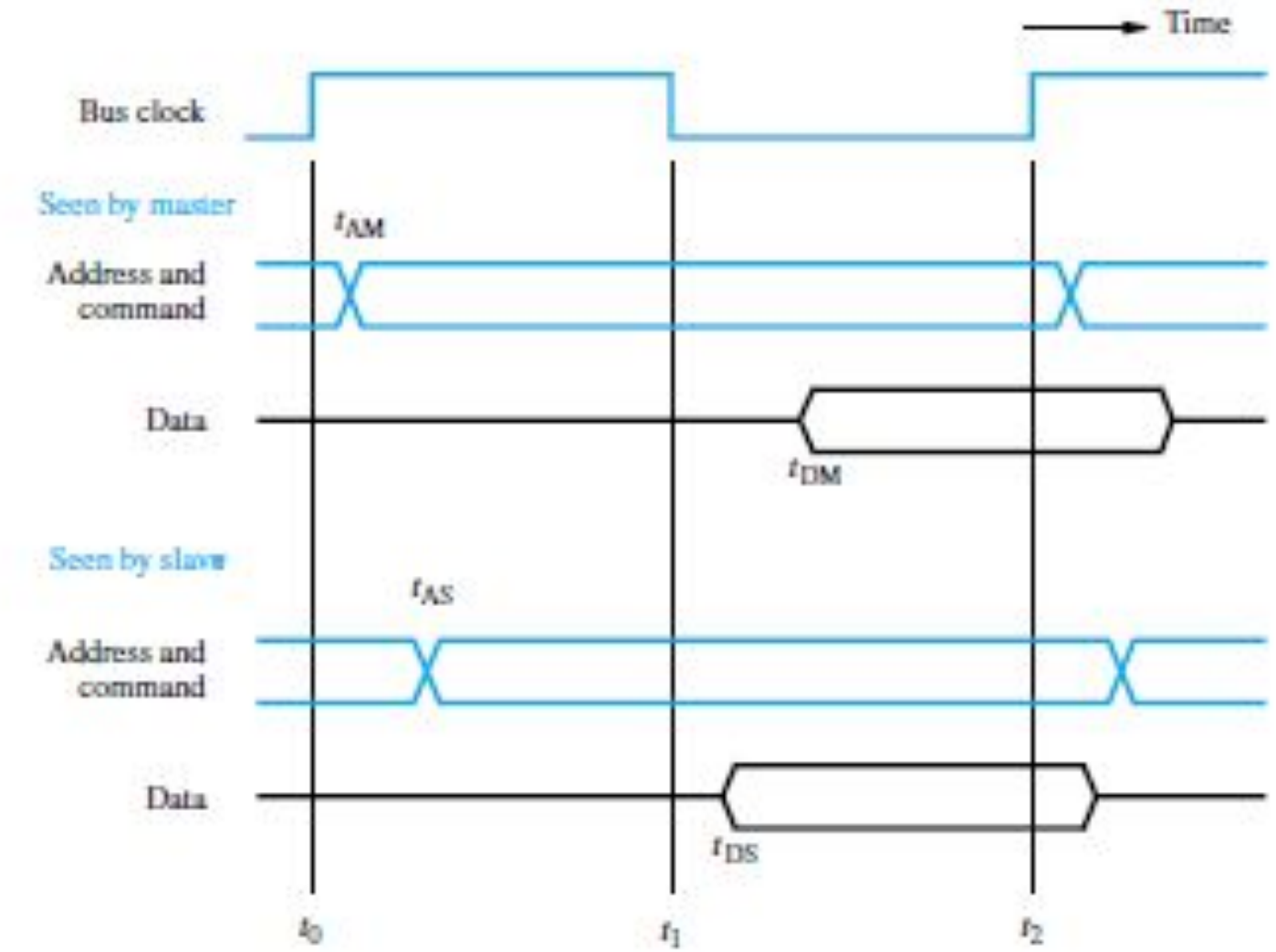
Synchronous Bus

- The sequence of signal events during an input (Read) operation. At time t0, the master places the device address on the address lines and sends a command on the control lines indicating a Read operation.

- Information travels over the bus. The clock pulse width, t1 − t0, must be longer than the maximum propagation delay over the bus.

- Also, it must be long enough to allow all devices to decode the address and control signals, so that the addressed device (the slave) can respond at time t1 by placing the requested input data on the data lines. At the end of the clock cycle, at time t2, the master loads the data on the data lines into one of its registers.

- To be loaded correctly into a register, data must be available for a period greater than the setup time of the register. Hence, the period t2 − t1 must be greater than the maximum propagation time on the bus plus the setup time of the master's register.



A detailed timing diagram for the input transfer

Multiple-Cycle Data Transfer

However, it has some limitations.

- Because a transfer has to be completed within one clock cycle, the clock period, t2 − t0, must be chosen to accommodate the longest delays on the bus and the slowest device interface.

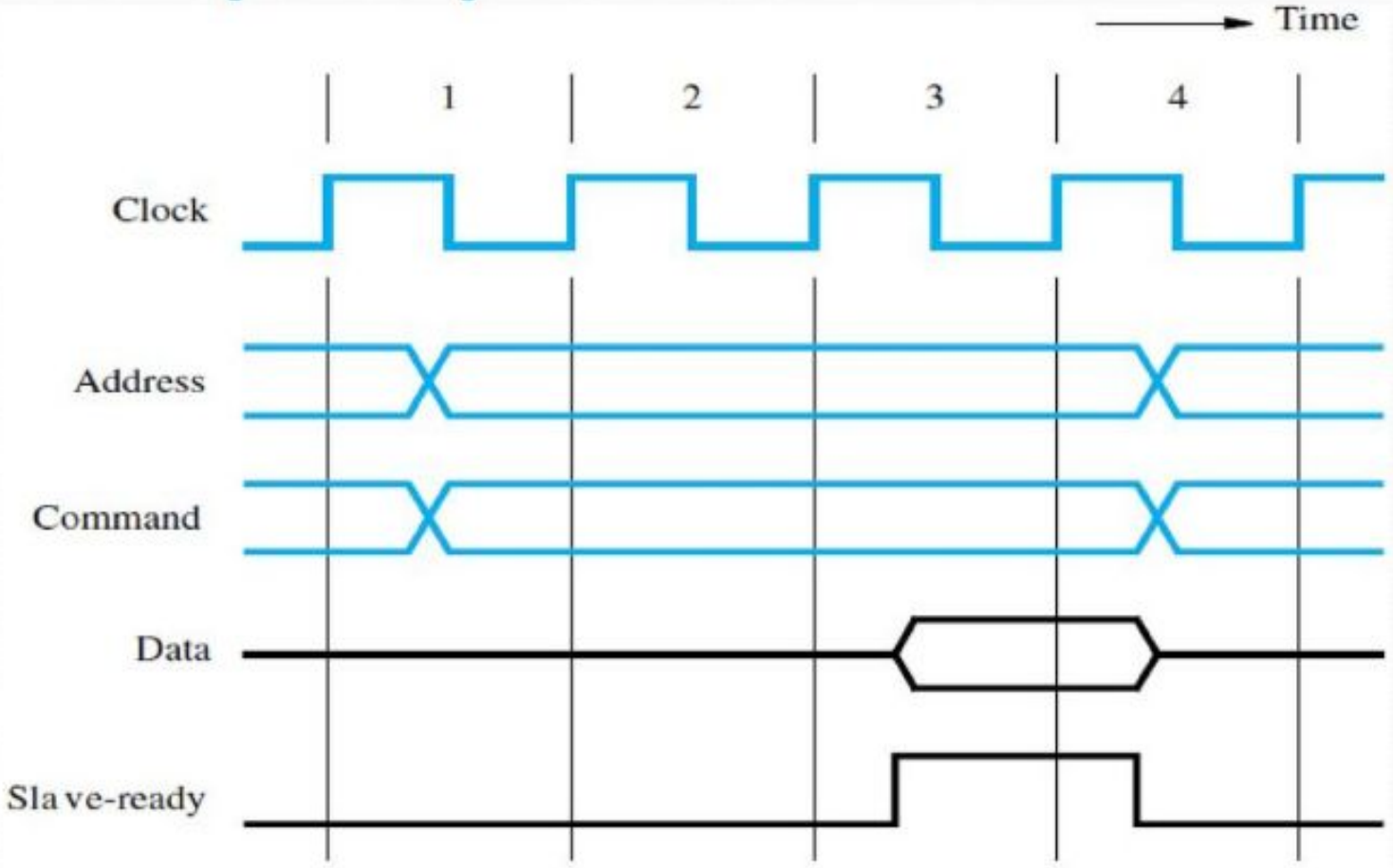
- This forces all devices to operate at the speed of the slowest device. To overcome these limitations, most buses incorporate control signals that represent a response from the device.

- These signals inform the master that the slave has recognized its address and that it is ready to participate in a data transfer operation.

- They also make it possible to adjust the duration of the data transfer period to match the response speeds of different devices.

- This is often accomplished by allowing a complete data transfer operation to span several clock cycles.

Multiple-Cycle Data Transfer



An input transfer using multiple clock cycles.

Asynchronous Bus

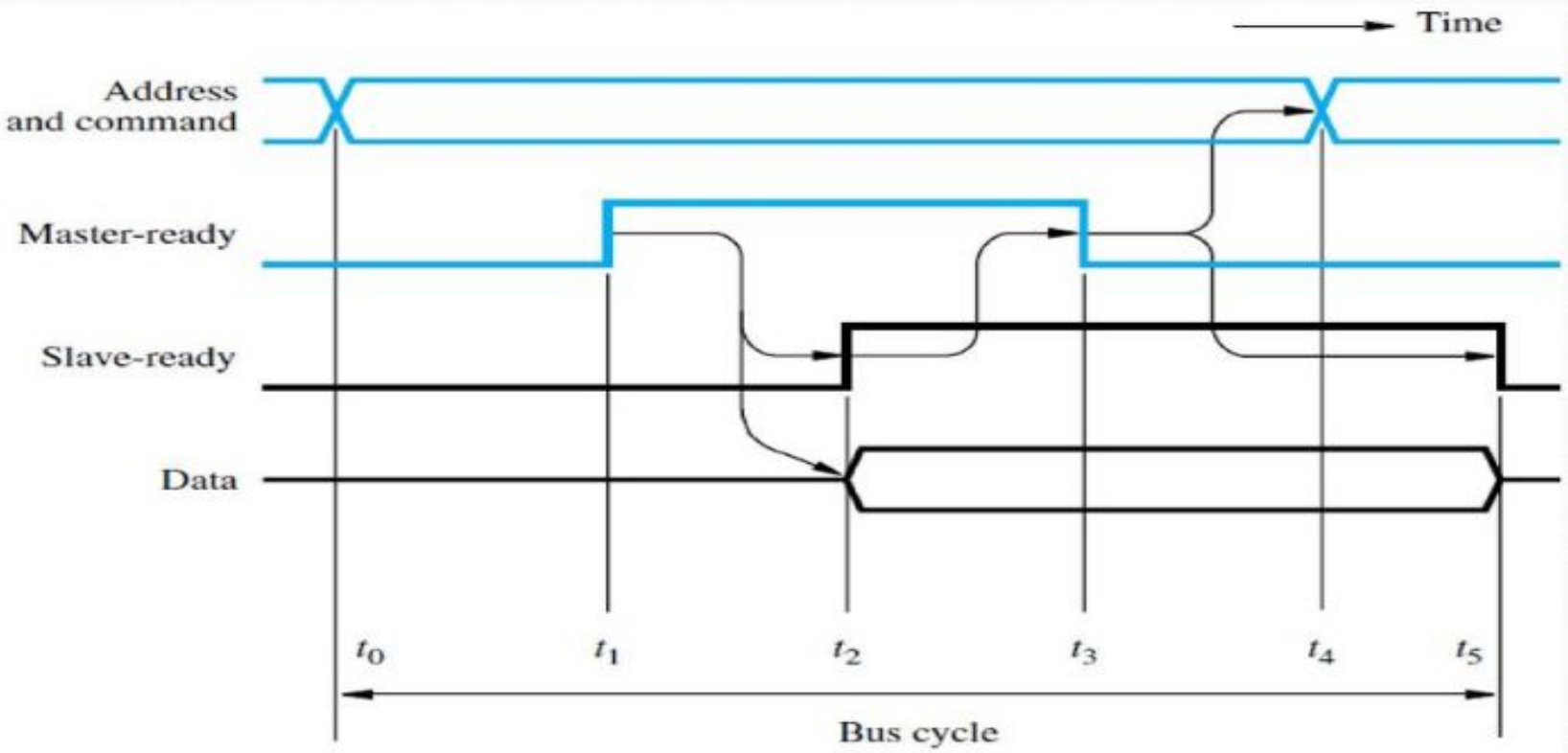
- An alternative scheme for controlling data transfers on a bus is based on the use of a handshake protocol between the master and the slave. A handshake is an exchange of command and response signals between the master and the slave.

- The master places the address and command information on the bus. Then it indicates to all devices that it has done so by activating the Master-ready line.

- This causes all devices to decode the address. The selected slave performs the required operation and informs the processor that it has done so by activating the Slave-ready line.

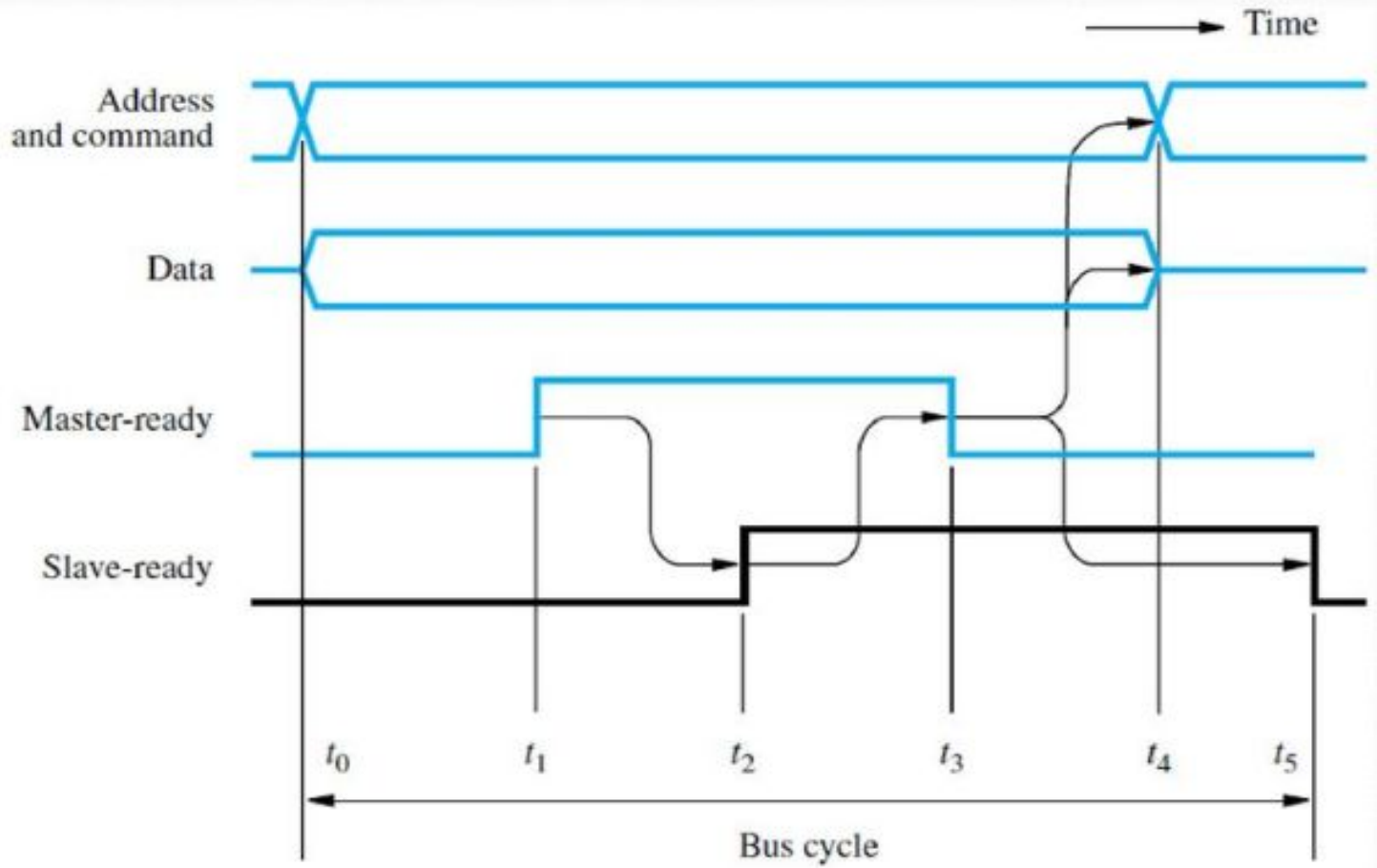
- The master waits for Slave-ready to become asserted before it removes its signals from the bus. In the case of a Read operation, it also loads the data into one of its registers.

Asynchronous Bus



Handshake control of data transfer during an input operation.

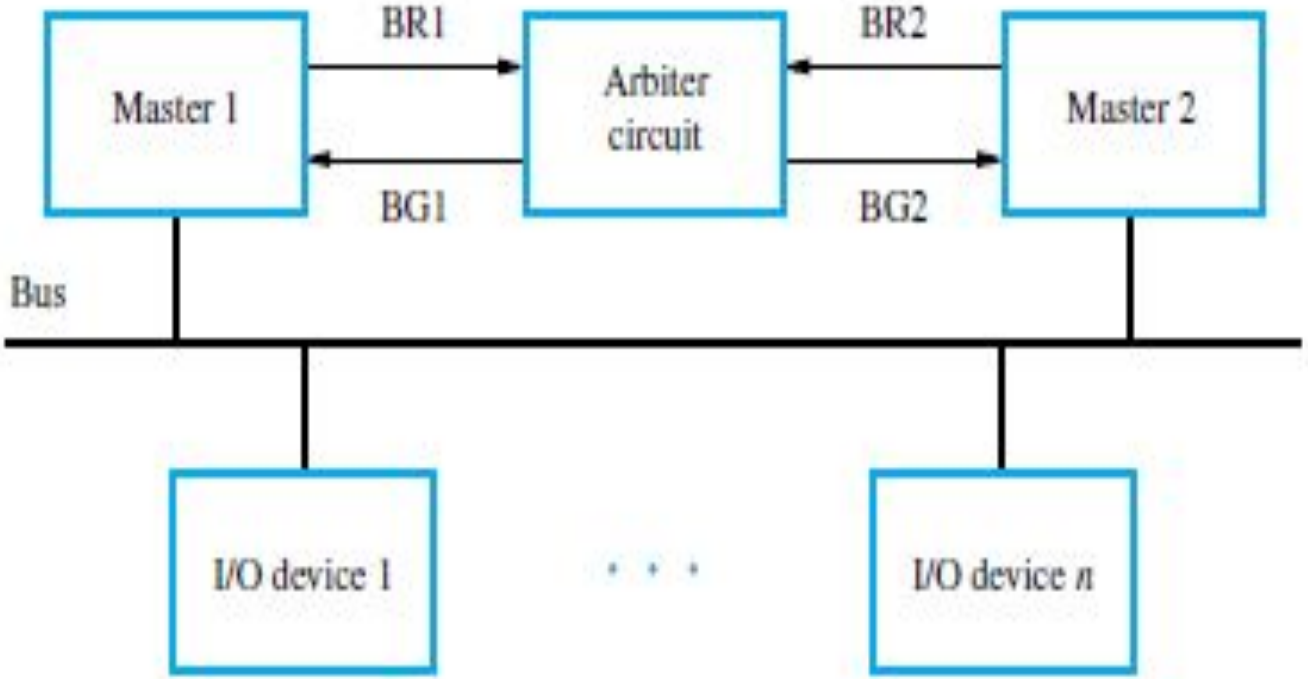
Asynchronous Bus



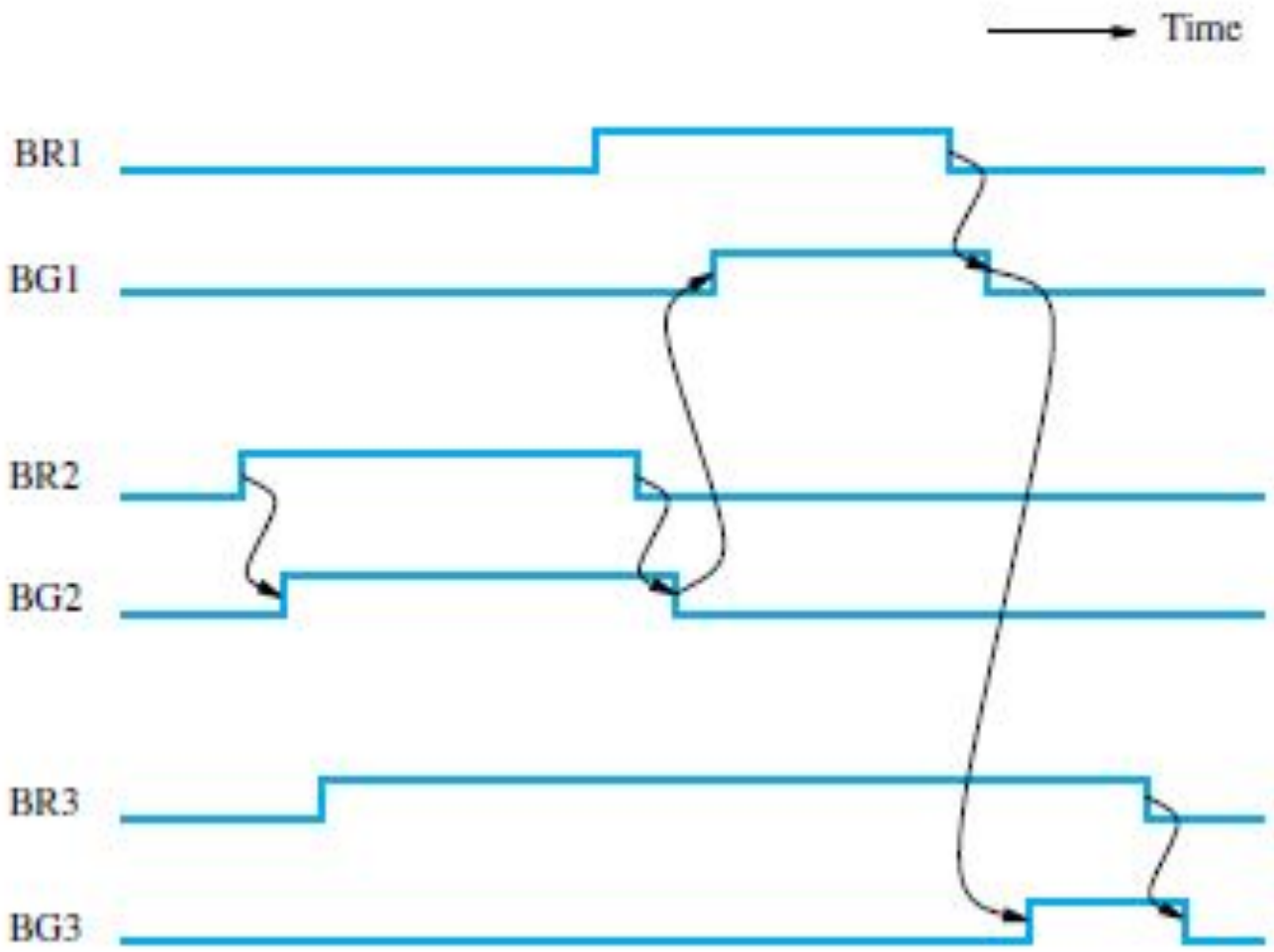
Handshake control of data transfer during an output operation.

Arbitration

- There are occasions when two or more entities contend for the use of a single resource in a computer system.
- For example, two devices may need to access a given slave at the same time. In such cases, it is necessary to decide which device will access the slave first.
- The decision is usually made in an arbitration process performed by an arbiter circuit.
- The arbitration process starts by each device sending a request to use the shared resource.
- The arbiter associates priorities with individual requests. If it receives two requests at the same time, it grants the use of the slave to the device having the higher priority first.



Bus arbitration.



Granting use of the bus based on priorities.

Interface circuits

Interface circuits

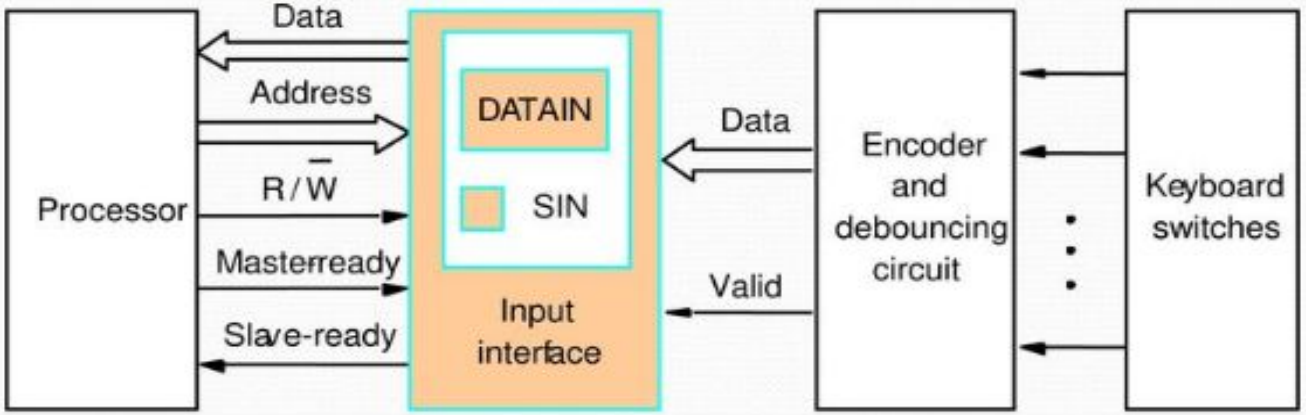
- I/O interface consists of the circuitry required to connect an I/O device to a computer bus.

- Side of the interface which connects to the computer has bus signals for:
  - Address,
  - Data
  - Control

- Side of the interface which connects to the I/O device has:
  - Datapath and associated controls to transfer data between the interface and the I/O device.
  - This side is called as a “port”.

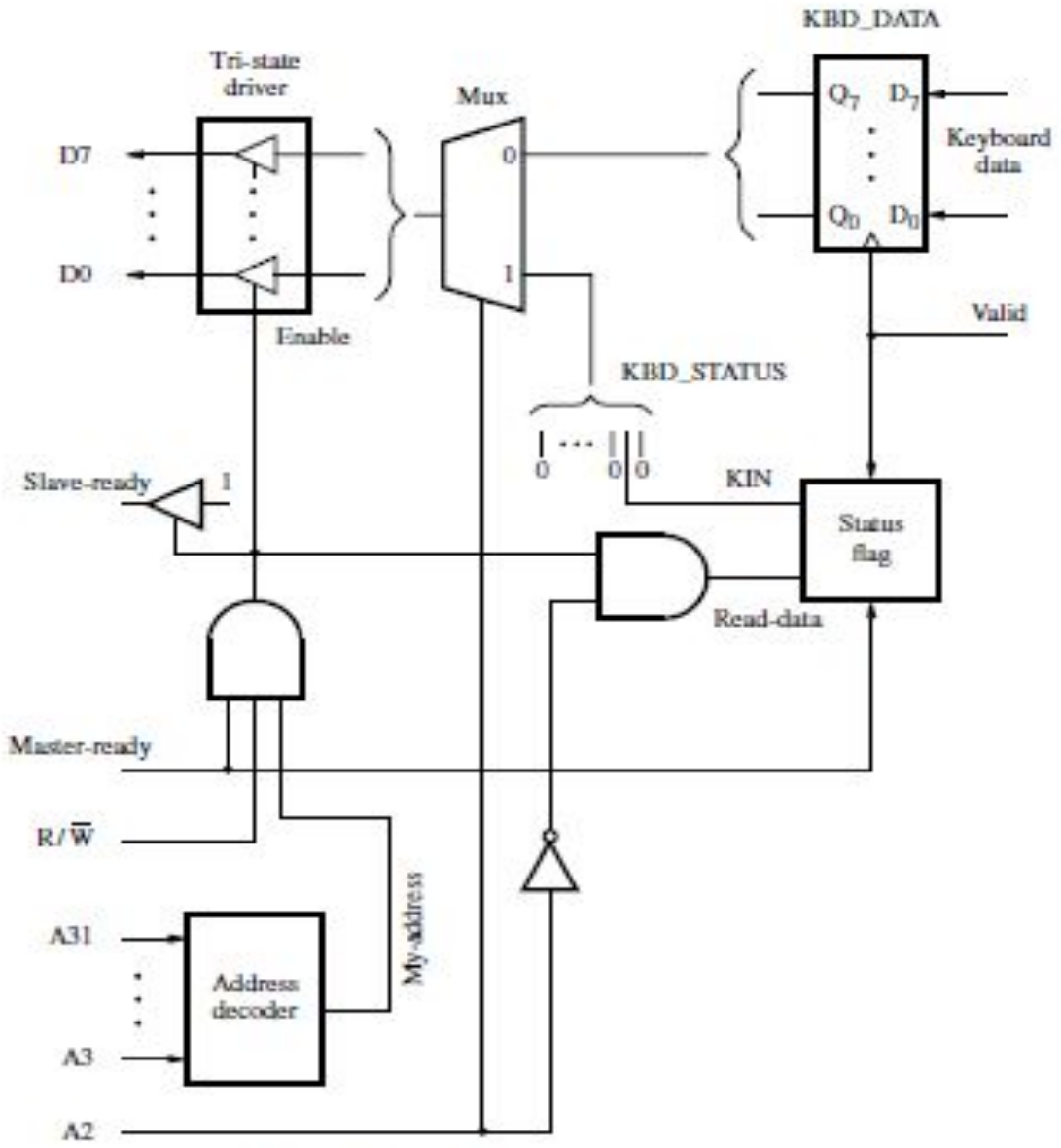
- Ports can be classified into two:
  - Parallel port,
  - Serial port.

Parallel port



- *Keyboard is connected to a processor using a parallel port.*
- *Processor is 32-bits and uses memory-mapped I/O and the asynchronous bus protocol.*
- *On the processor side of the interface we have:*
  - *Data lines.*
  - *Address lines*
  - *Control or R/W line.*
  - *Master-ready signal and*
  - *Slave-ready signal.*

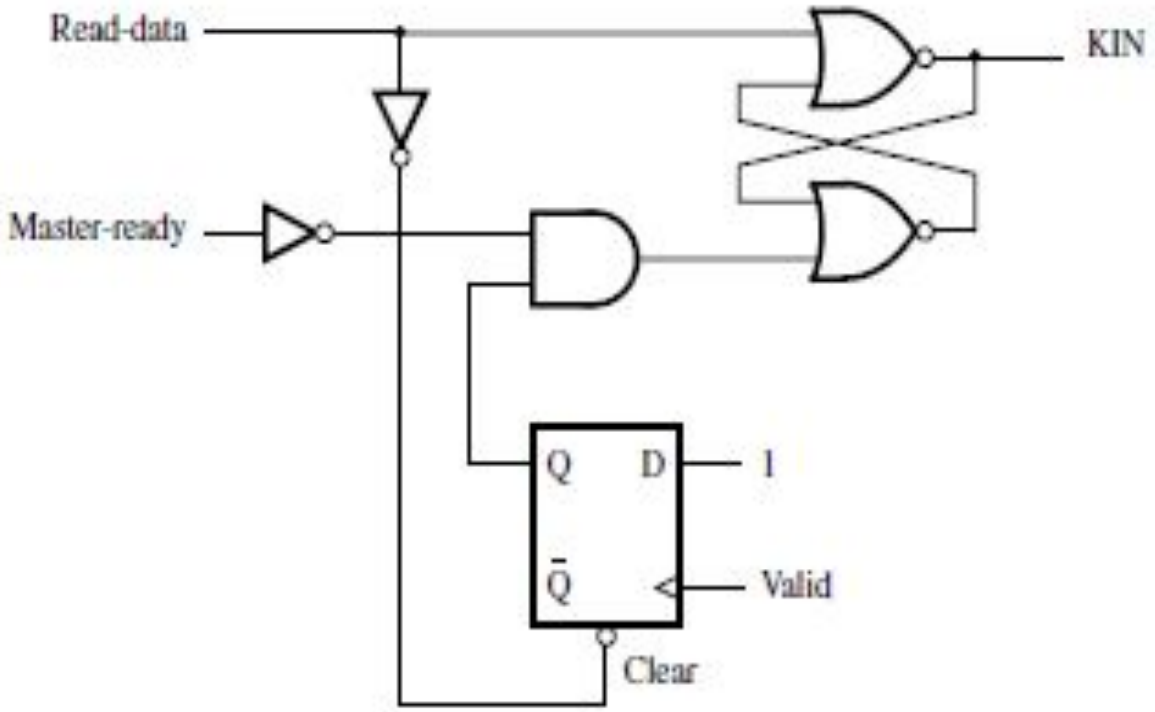
Input interface circuit



An input interface circuit.

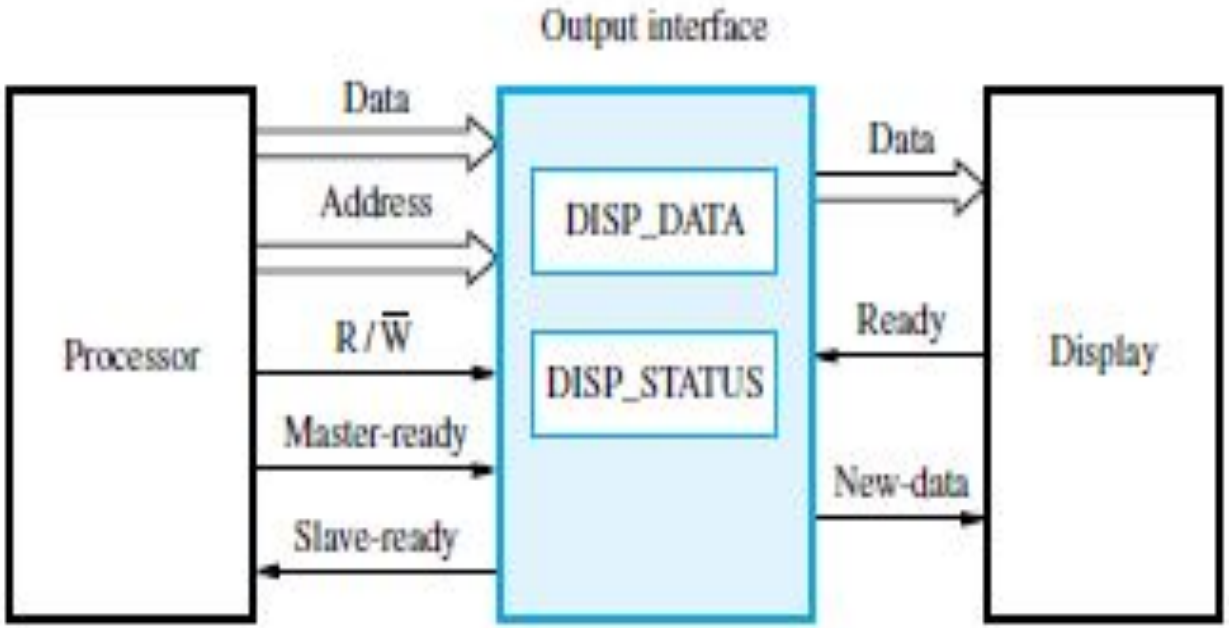
Input Interface Circuit

- Output lines of DATAIN are connected to the data lines of the bus by means of 3 state drivers
- Drivers are turned on when the processor issues a read signal and the address selects this register.
- SIN signal is generated using a status flag circuit. It is connected to line D0 of the processor bus using a three-state driver. Address decoder selects the input interface based on bits A1 through A31.
- Bit A0 determines whether the status or data register is to be read, when Master-ready is active.
- In response, the processor activates the Slave-ready signal, when either the Read-status or Read-data is equal to 1, which depends on line A0.



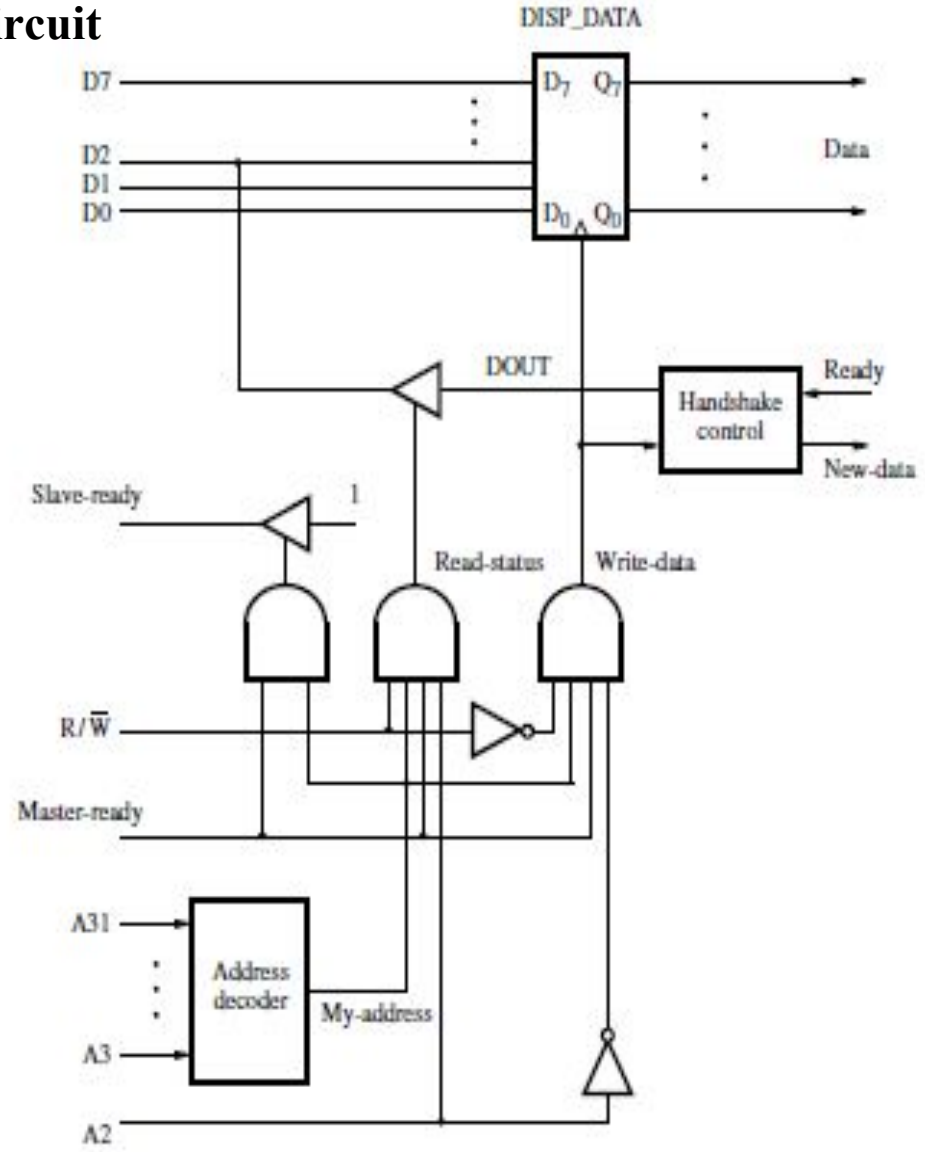
Circuit for the status flag block

Output interface circuit



Display to processor connection.

Output interface circuit

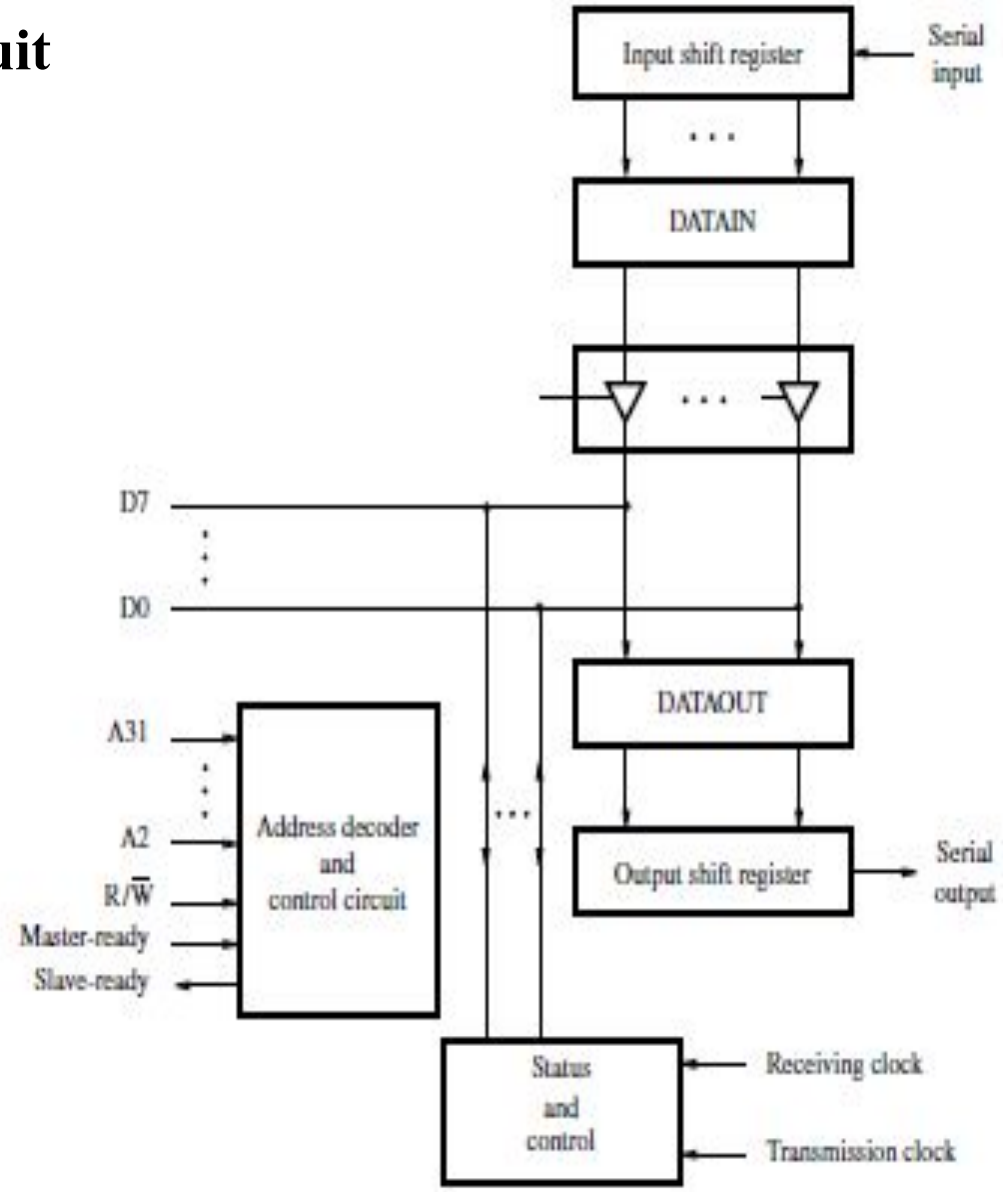


An output interface circuit.

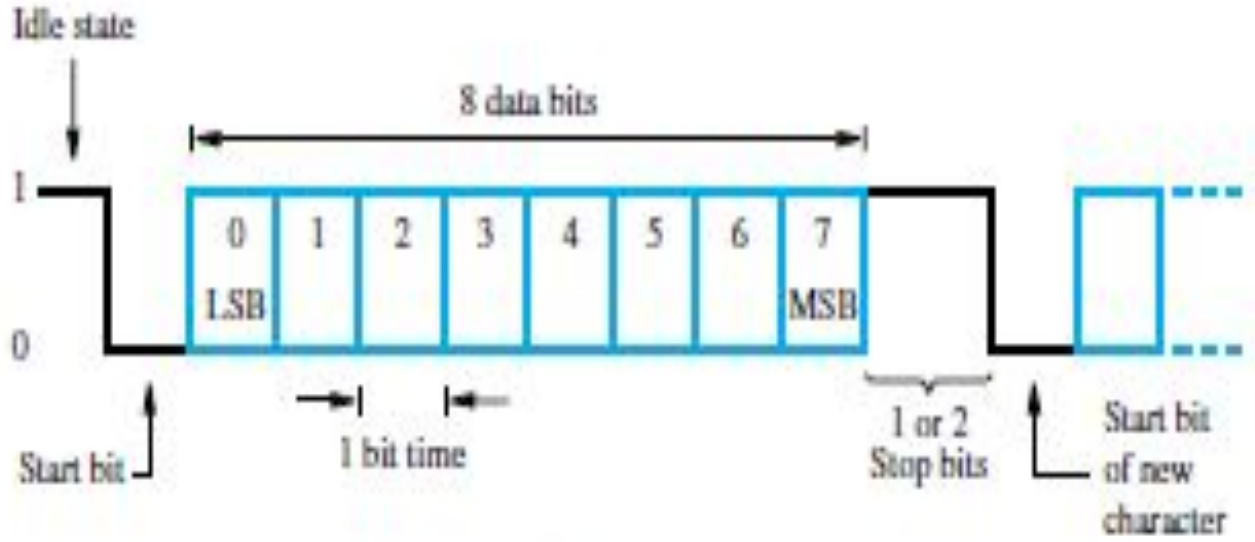
Serial port

- Serial port is used to connect the processor to I/O devices that require transmission of data one bit at a time.
- Serial port communicates in a bit-serial fashion on the device side and bit parallel fashion on the bus side.
- Transformation between the parallel and serial formats is achieved with shift registers that have parallel access capability.

Serial interface circuit



Synchronous vs Asynchronous Transmission	
Synchronous Transmission is a transmission method that uses synchronized clocks to ensure both the sender and receiver are synchronized to transmit data.	Asynchronous Transmission is a transmission method that sends data using flow control to transmit data between the source and the destination.
Efficiency	
More efficient	Less efficient
Method of Sending Data	
Sends blocks or frames of data at a time	Sends one byte or character at a time
Cost	
Comparatively, high	Cost is low
Time Interval	
Uses fixed time intervals	Uses arbitrary time intervals
Examples	
Some examples of synchronous transmission are chat rooms, video conferencing, telephone conversations, etc.	Emails, television, and radios are few examples for asynchronous transmission.



Asynchronous serial character transmission.

Interconnection Standards

- I/O device is connected to a computer using an interface

- A personal computer has:
  - A motherboard which houses the processor chip, main memory and some I/O interfaces.
  - A few connectors into which additional interfaces can be plugged.
- Processor bus is defined by the signals on the processor chip.
  - Devices which require high-speed connection to the processor are connected directly to this bus.

- Motherboard usually provides another bus that can support more devices.
  - Processor bus and the other bus (called as expansion bus) are interconnected by a circuit called “bridge”.
  - Devices connected to the expansion bus experience a small delay in data transfers.

Standard I/O interfaces

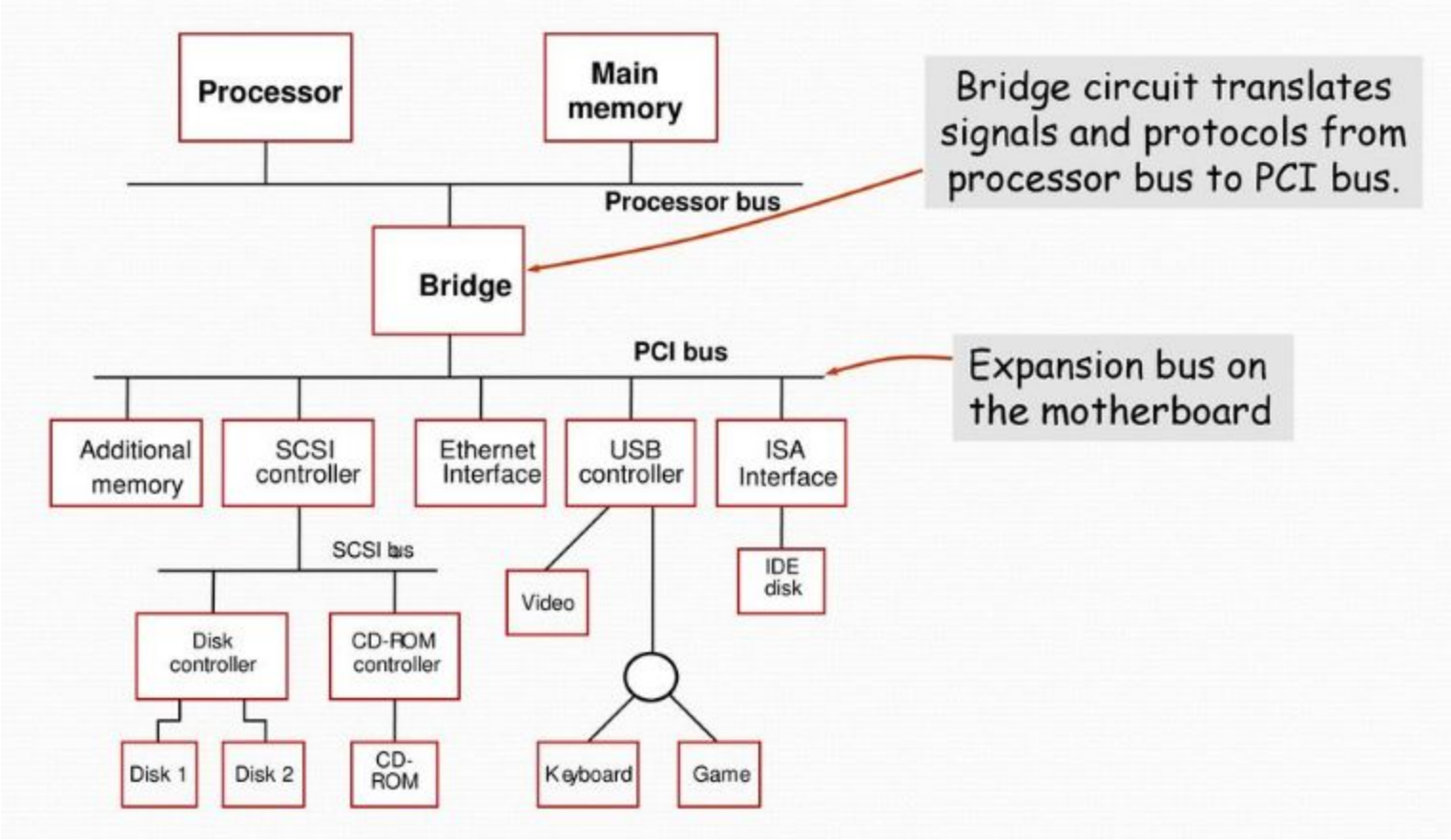
A number of standards have been developed for the expansion bus.

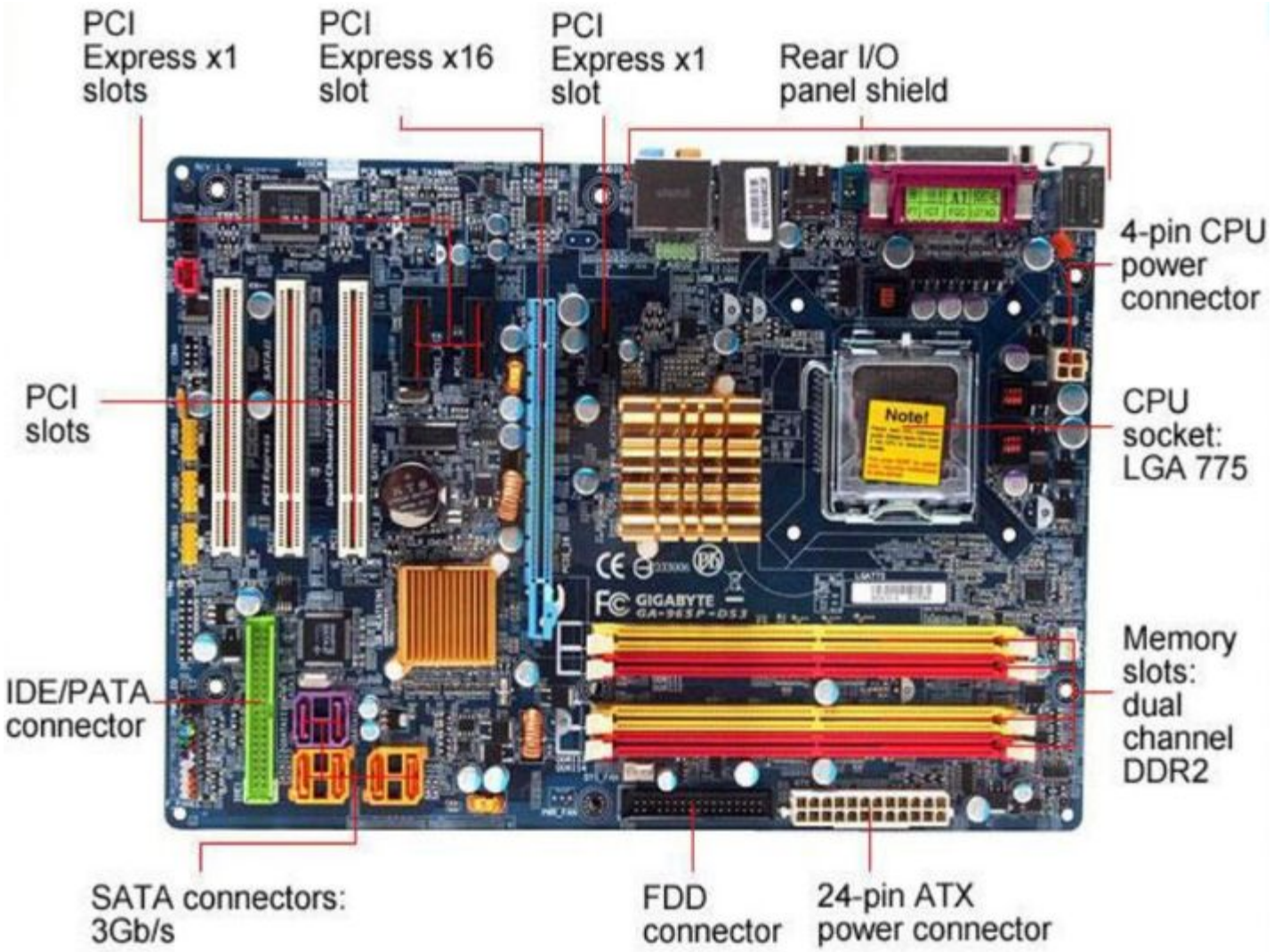
- Some have evolved by default.
- For example, IBM's Industry Standard Architecture.

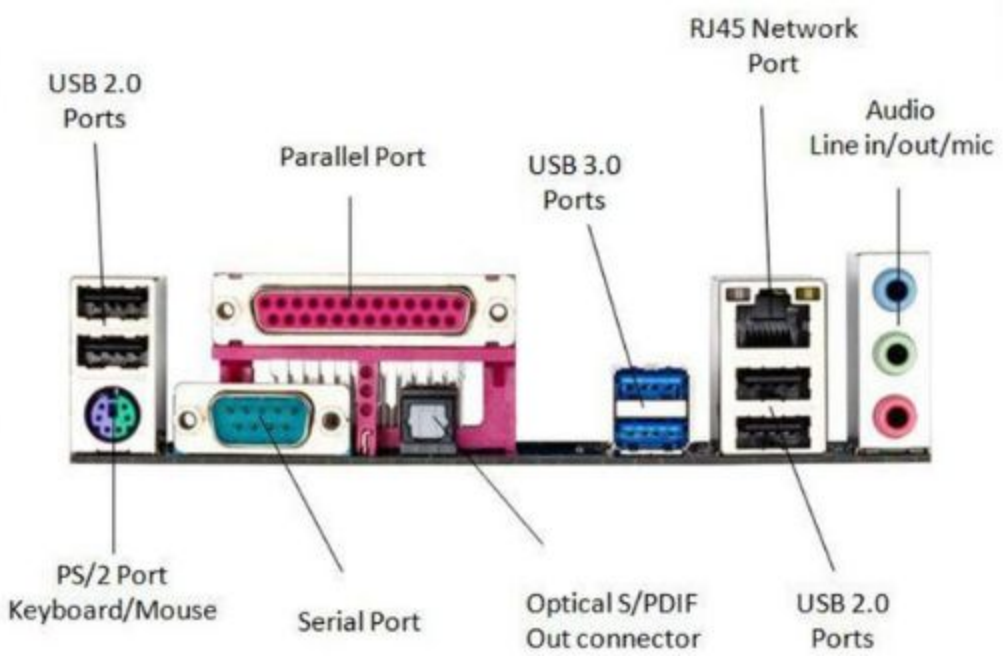
Three widely used bus standards:

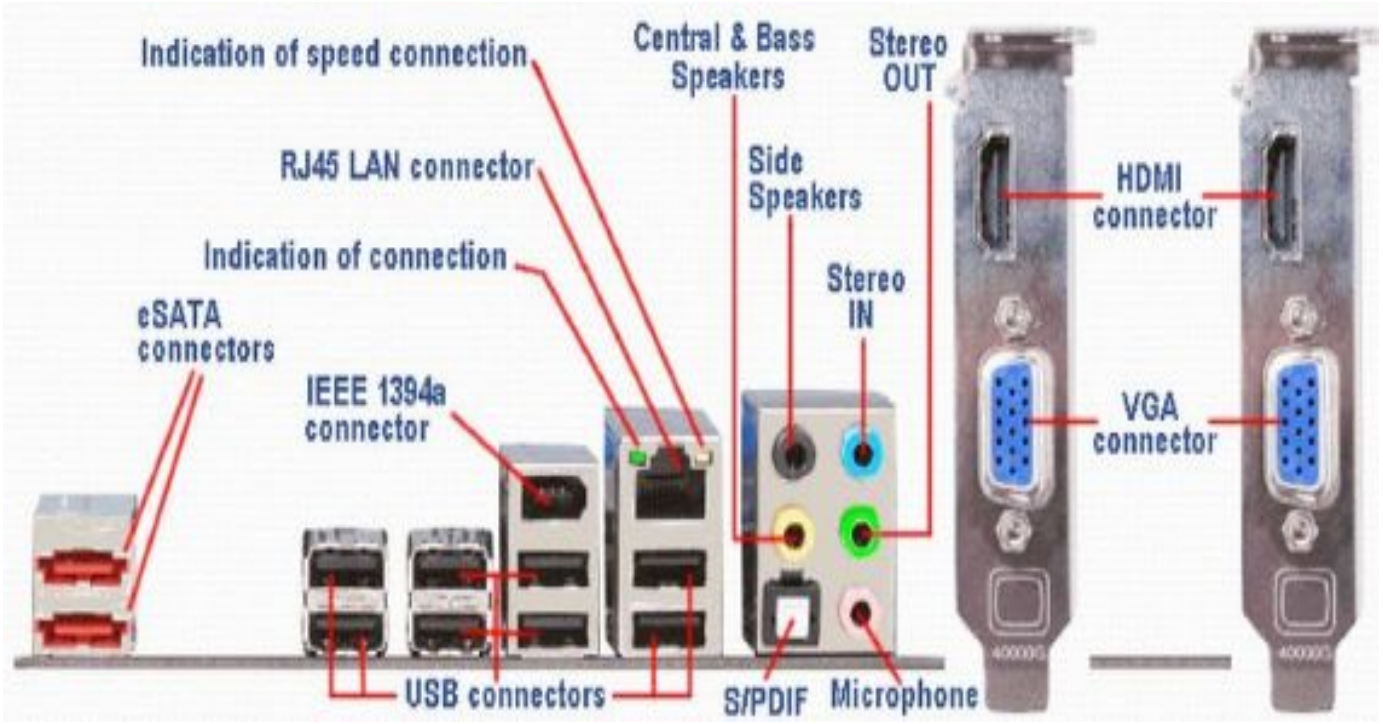
- PCI (Peripheral Component Interconnect)
- SCSI (Small Computer System Interface)
- USB (Universal Serial Bus)

Standard I/O interfaces



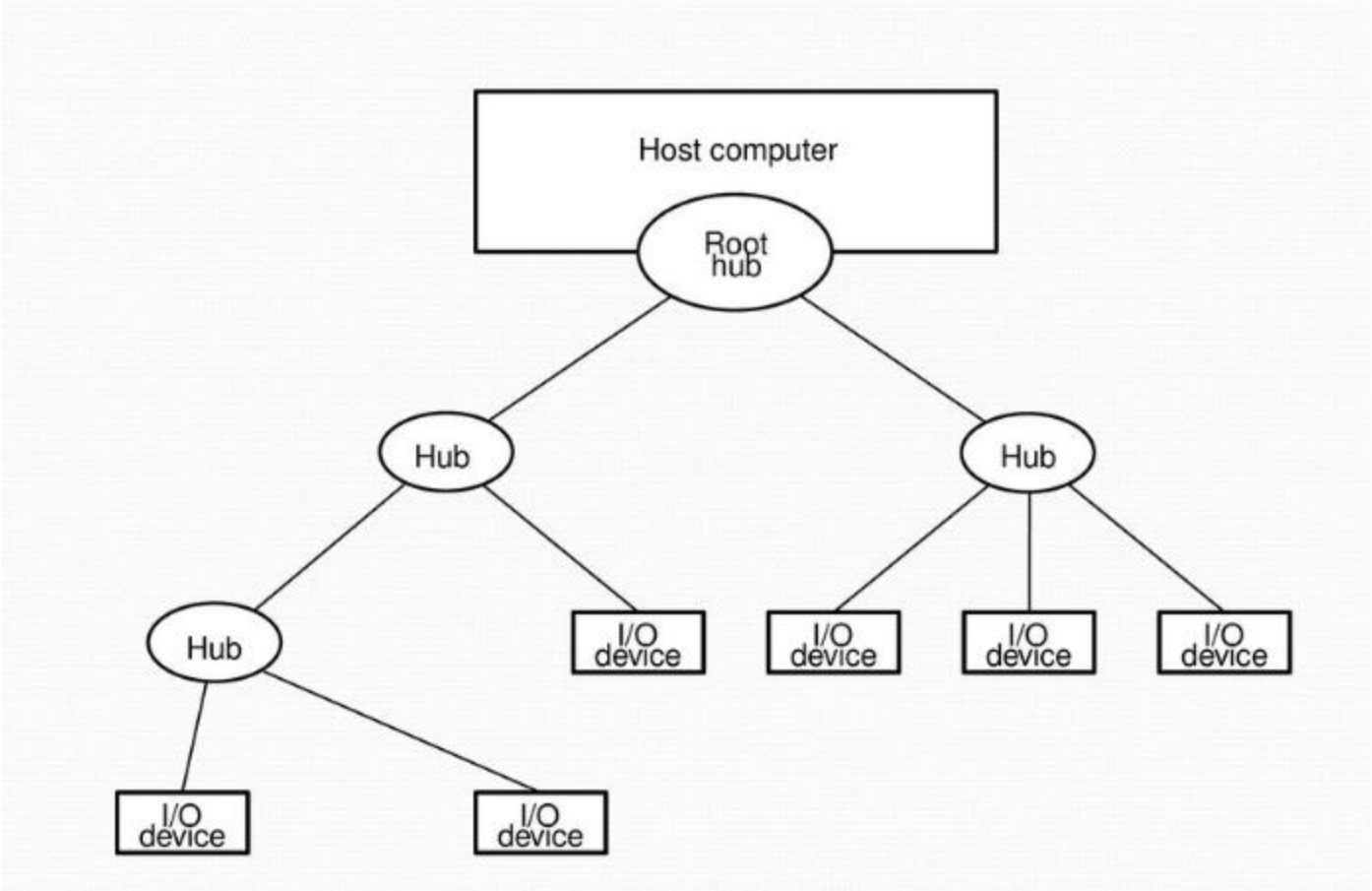

PCI Express x1 slots
PCI Express x16 slot
PCI Express x1 slot
Rear I/O panel shield
4-pin CPU power connector
PCI slots
CPU socket: LGA 775
IDE/PATA connector
Memory slots: dual channel DDR2
SATA connectors: 3Gb/s
FDD connector
24-pin ATX power connector


USB 2.0
Ports
Parallel Port
USB 3.0
Ports
RJ45 Network
Port
Audio
Line in/out/mic
PS/2 Port
Keyboard/Mouse
Serial Port
Optical S/PDIF
Out connector
USB 2.0
Ports


Indication of speed connection
Central & Bass
Speakers
Stereo
OUT
RJ45 LAN connector
Side
Speakers
HDMI
connector
Indication of connection
Stereo
IN
eSATA
connectors
IEEE 1394a
connector
VGA
connector
USB connectors
S/PDIF
Microphone

USB

- Universal Serial Bus (USB) is an industry standard developed through a collaborative effort of several computer and communication companies, including Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, Nortel Networks, and Philips.
- Speed
  - Low-speed(1.5 Mb/s)
  - Full-speed(12 Mb/s)
  - High-speed(480 Mb/s)
- Port Limitation
- Device Characteristics
- Plug-and-play



Universal Serial Bus tree structure

Universal Serial Bus tree structure

- To accommodate a large number of devices that can be added or removed at any time, the USB has the tree structure. Each node of the tree has a device called a hub, which acts as an intermediate control point between the host and the I/O devices.

- At the root of the tree, a root hub connects the entire tree to the host computer. The leaves of the tree are the I/O devices being served (for example, keyboard, Internet connection, speaker, or digital TV)In normal operation, a hub copies a message that it receives from its upstream connection to all its downstream ports.

- As a result, a message sent by the host computer is broadcast to all I/O devices, but only the addressed device will respond to that message.

- However, a message from an I/O device is sent only upstream towards the root of the tree and is not seen by other devices.

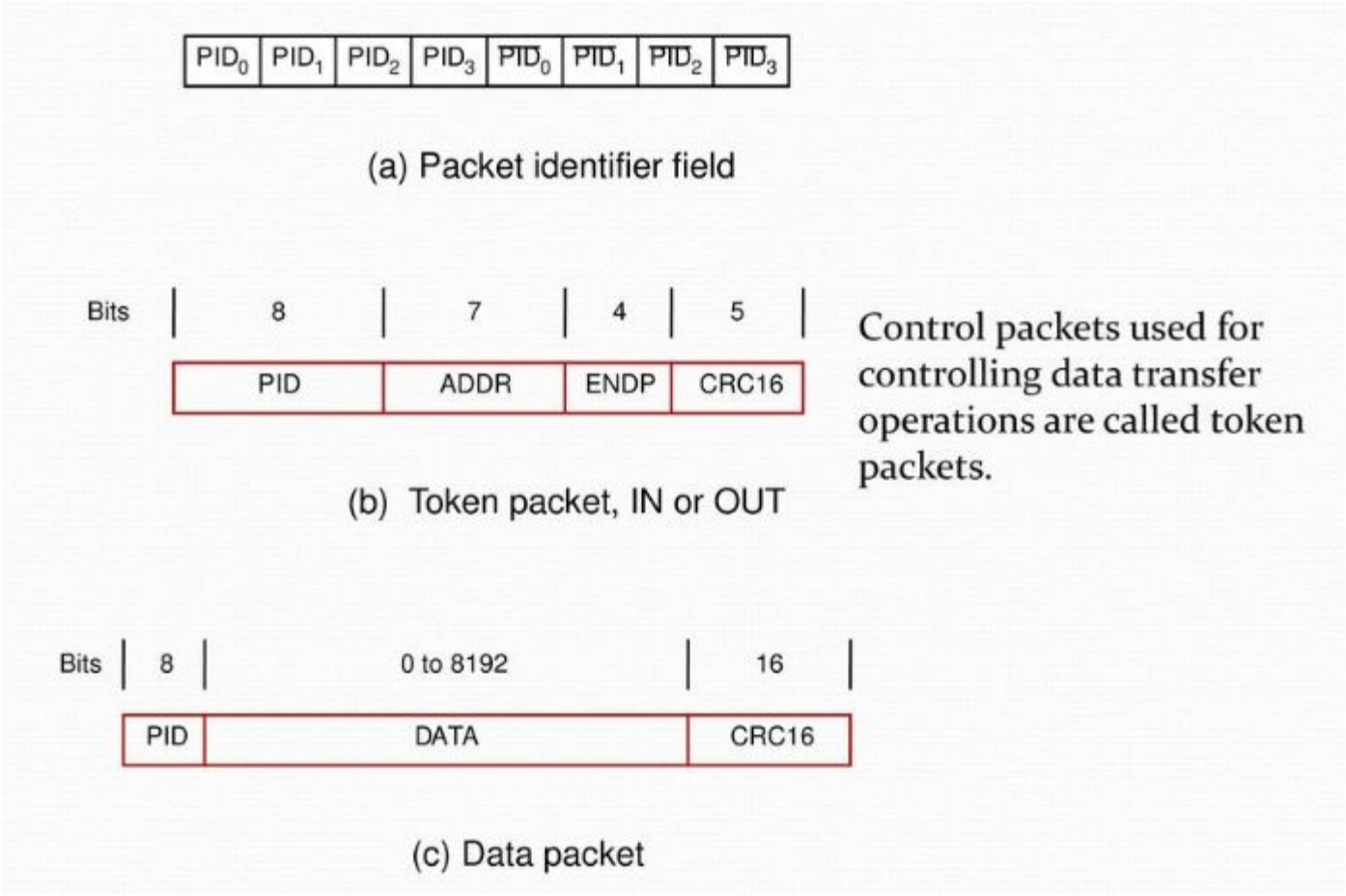
USB Protocols

- All information transferred over the USB is organized in packets, where a packet consists of one or more bytes of information. There are many types of packets that perform a variety of control functions.

- The information transferred on the USB can be divided into two broad categories: control and data. Control packets perform such tasks as addressing a device to initiate data transfer, acknowledging that data have been received correctly, or indicating an error.

- Data packets carry information that is delivered to a device. A packet consists of one or more fields containing different kinds of information. The first field of any packet is called the packet identifier, PID, which identifies the type of that packet.

- They are transmitted twice. The first time they are sent with their true values, and the second time with each bit complemented The four PID bits identify one of 16 different packet types.



Control packets used for controlling data transfer operations are called token packets.

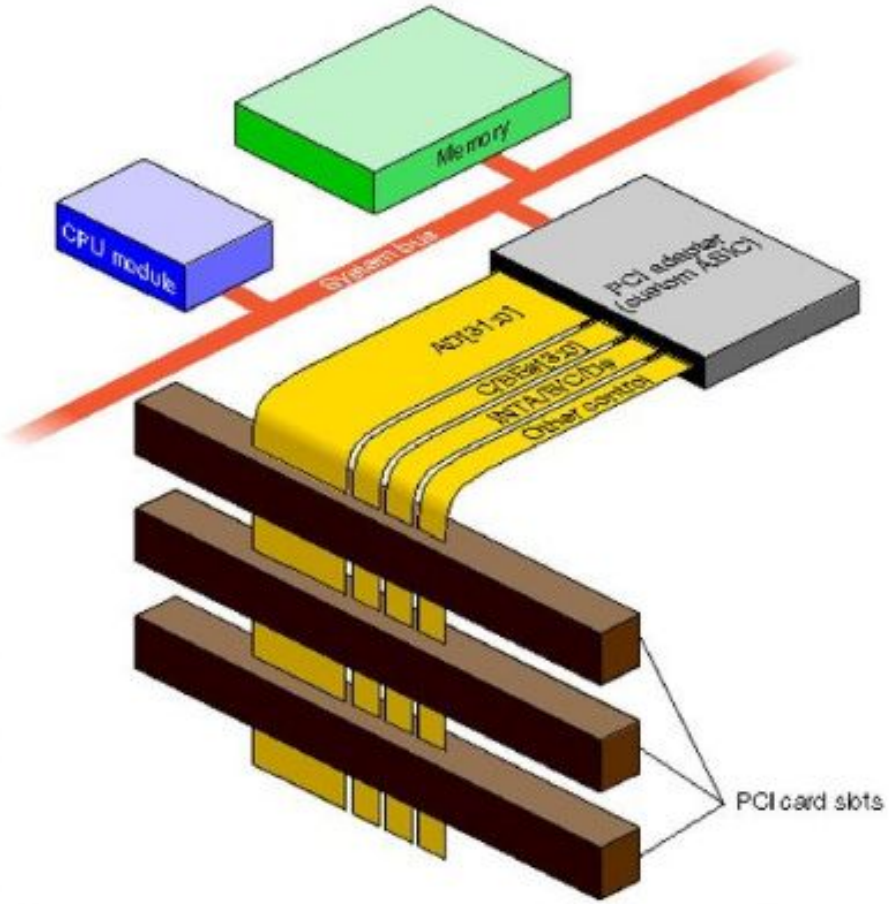
USB packet format

PCI Bus (Peripheral Component Interconnect)

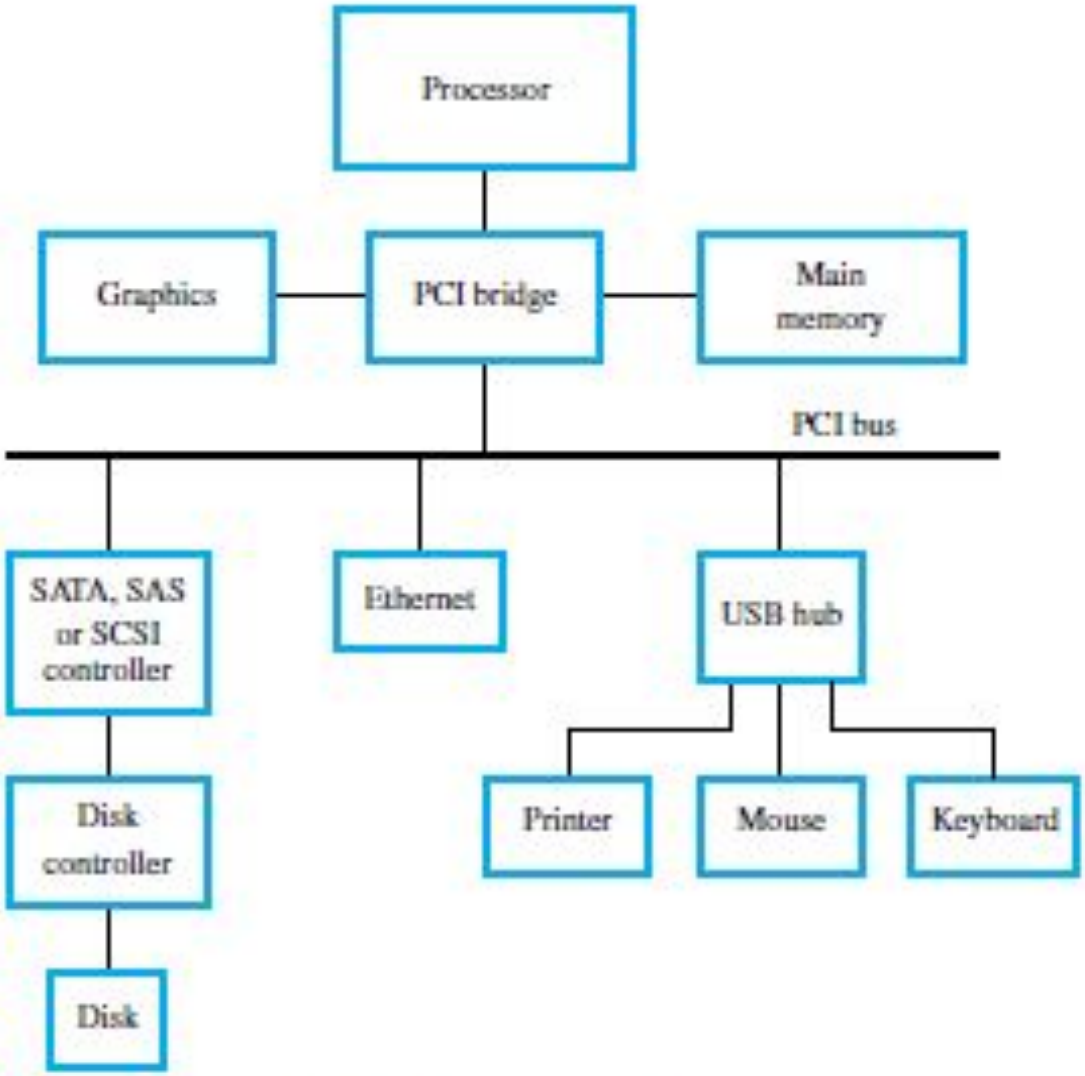
- Low-cost bus
- Processor independent
- Plug-and-play capability
- In today's computers, most memory transfers involve a burst of data rather than just one word. The PCI is designed primarily to support this mode of operation.
- The bus supports three independent address spaces: memory, I/O, and configuration.
- We assumed that the master maintains the address information on the bus until data transfer is completed. But, the address is needed only long enough for the slave to be selected.
- Thus, the address is needed on the bus for one clock cycle only, freeing the address lines to be used for sending data in subsequent clock cycles. The result is a significant cost reduction.
- A master is called an initiator in PCI terminology. The addressed device that responds to read and write commands is called a target.

Data transfer signals on the PCI bus.

Name	Function
CLK	A 33-MHz or 6
FRAME#	Sent by the initi transaction.
AD	32 address/data increased to 64
C/BE#	4 command/byte-
IRDY#, TRDY#	Initiator-ready a
DEVSEL#	A response from recognized its a transfer transacti
IDSEL#	Initialization D



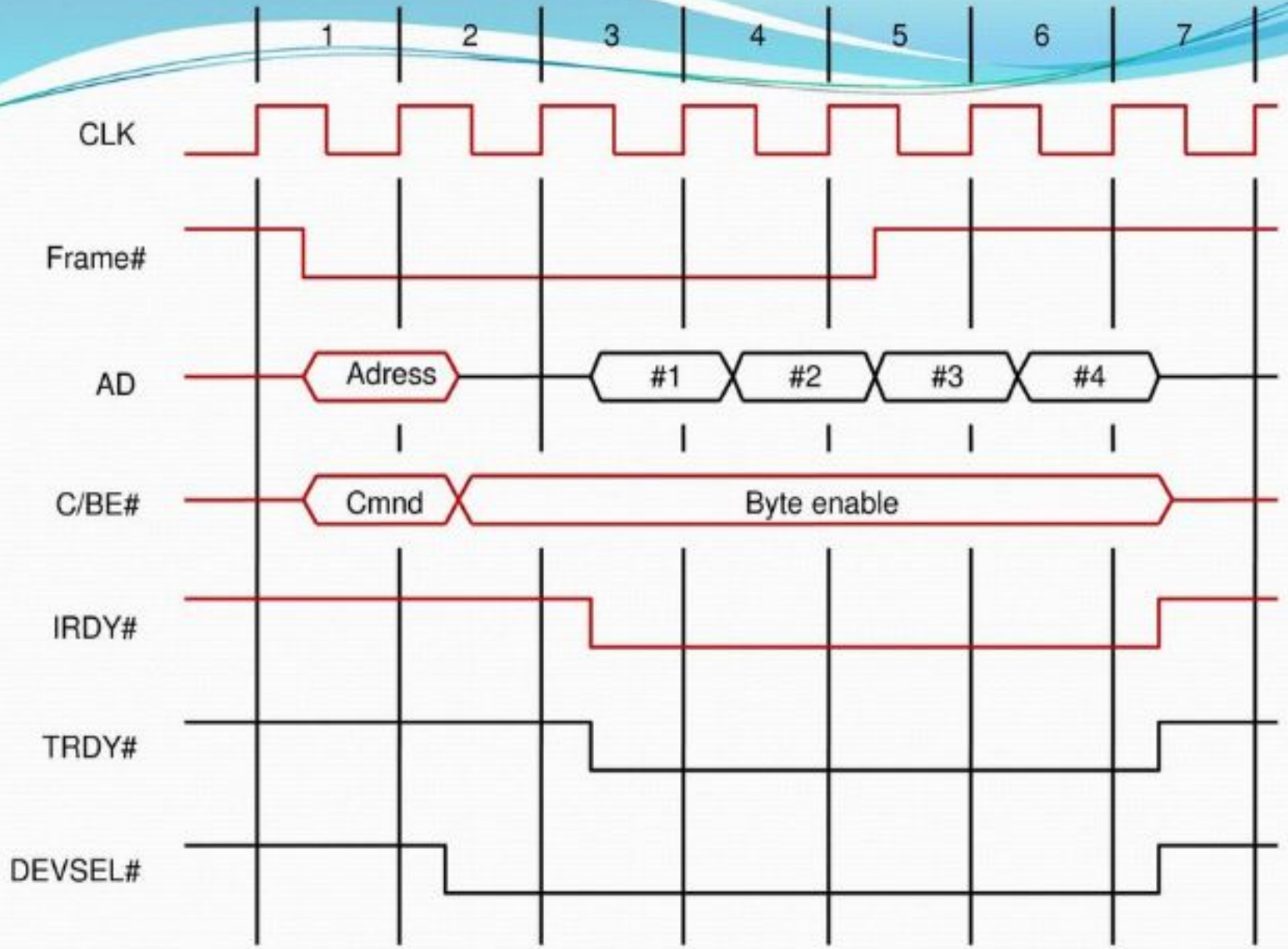
A signal whose name ends with the symbol # is asserted when in the low-voltage state.



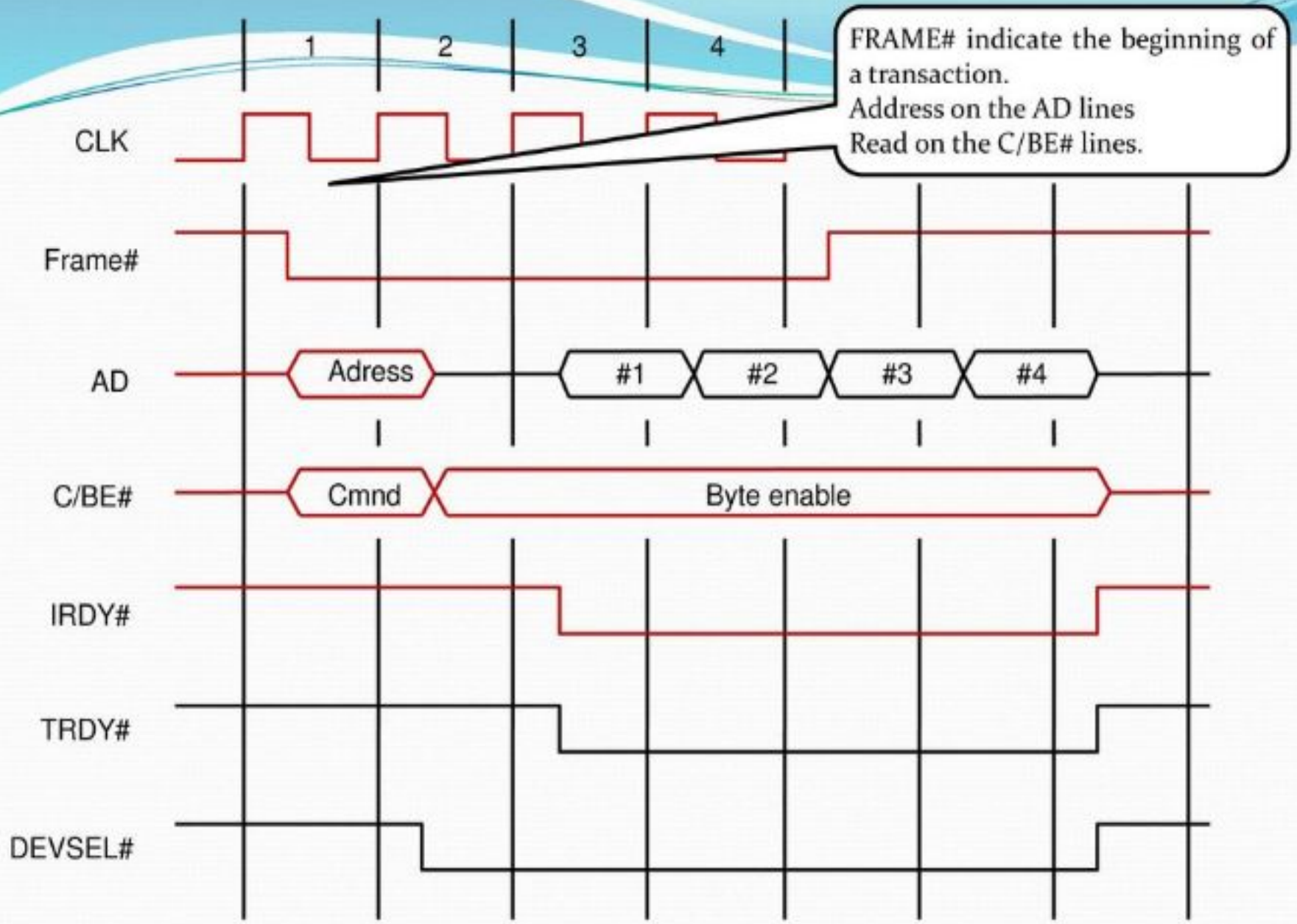
Use of a PCI bus in a computer system.

Table 7.1 Data transfer signals on the PCI bus.

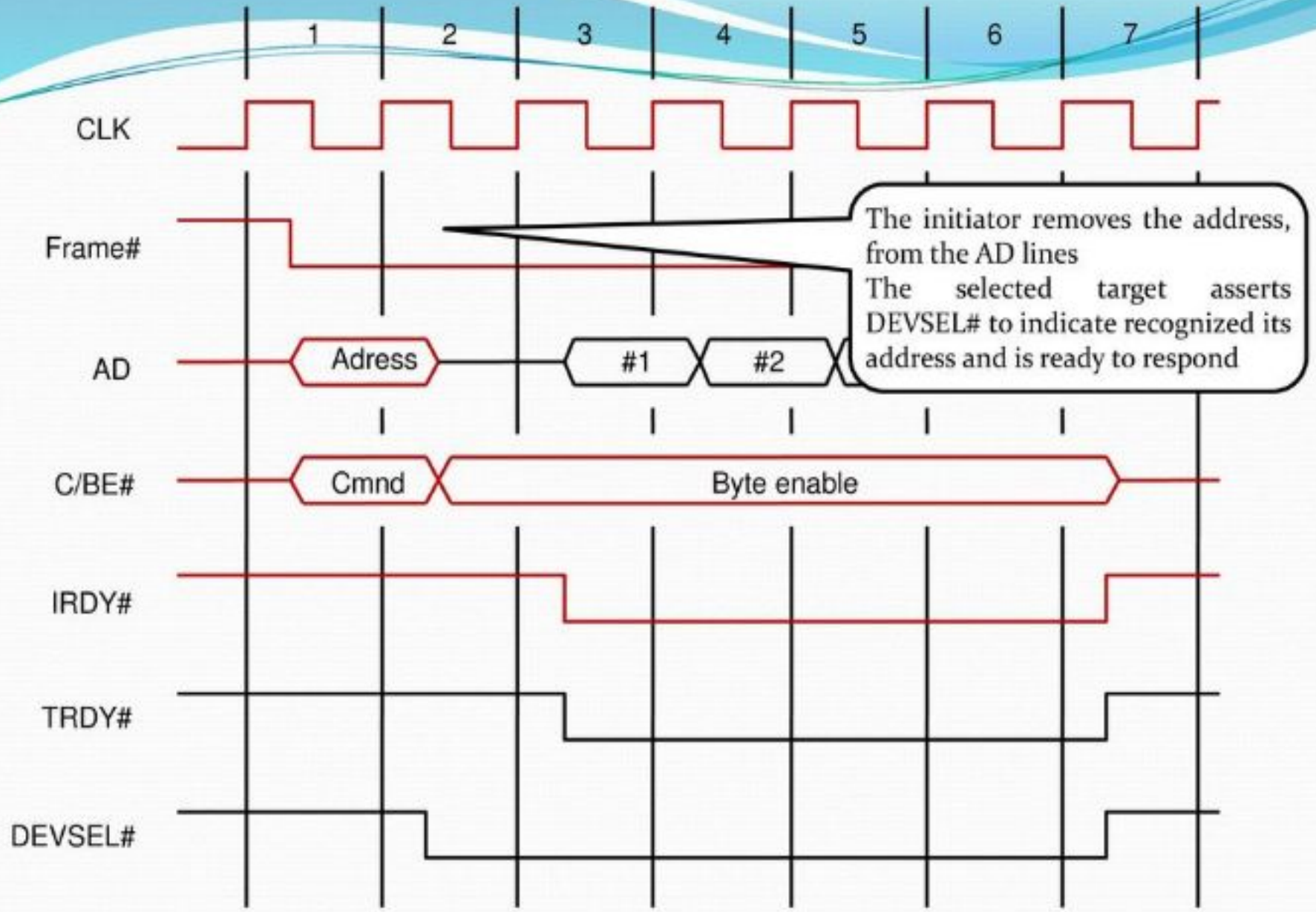
Name	Function
CLK	A 33-MHz or 66-MHz clock
FRAME#	Sent by the initiator to indicate the duration of a transmission
AD	32 address/data lines, which may be optionally increased to 64
C/BE#	4 command/byte-enable lines (8 for a 64-bit bus)
IRDY#, TRDY#	Initiator-ready and Target-ready signals
DEVSEL#	A response from the device indicating that it has recognized its address and is ready for a data transfer transaction
IDSEL#	Initialization Device Select



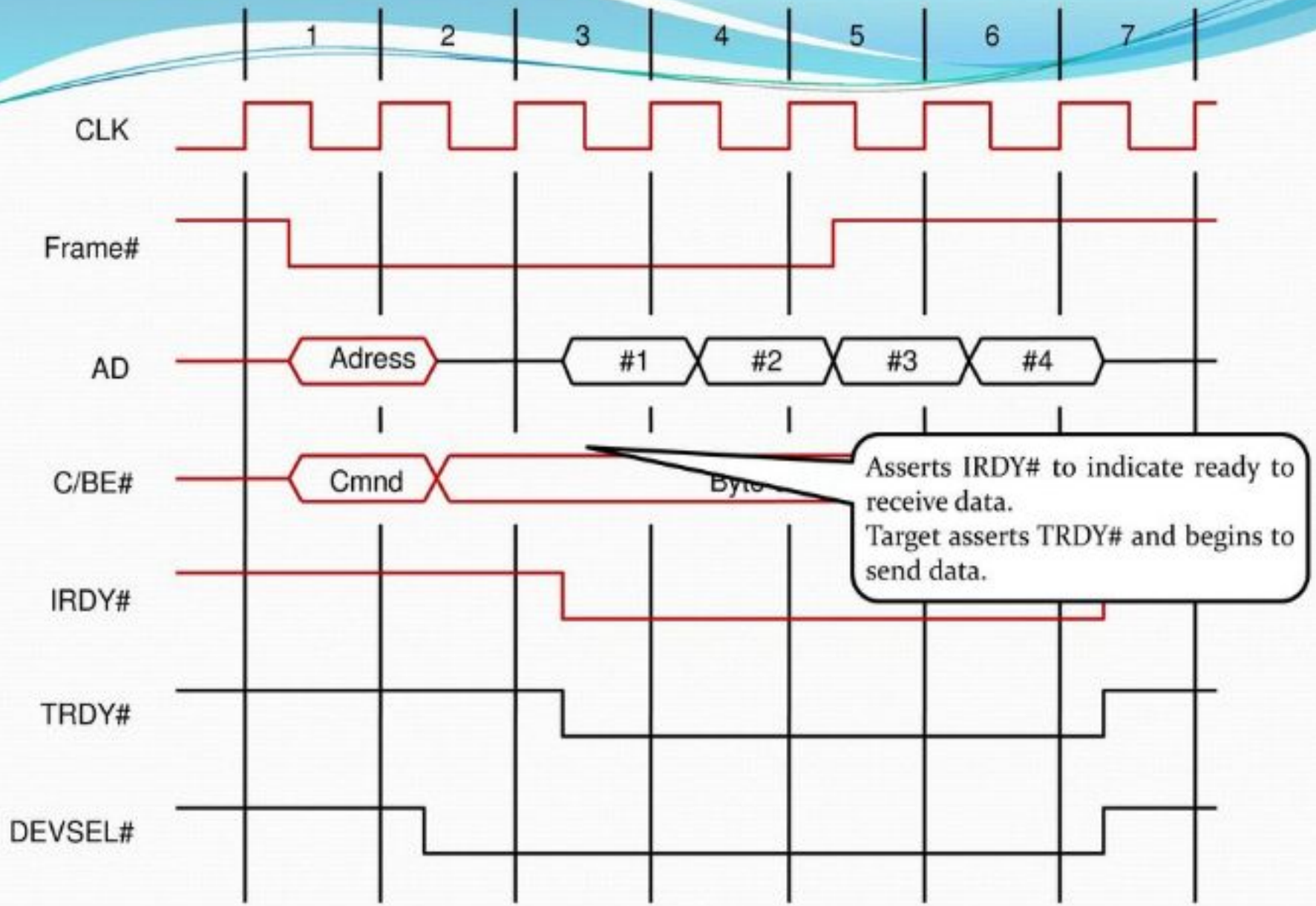
A read operation on the PCI bus



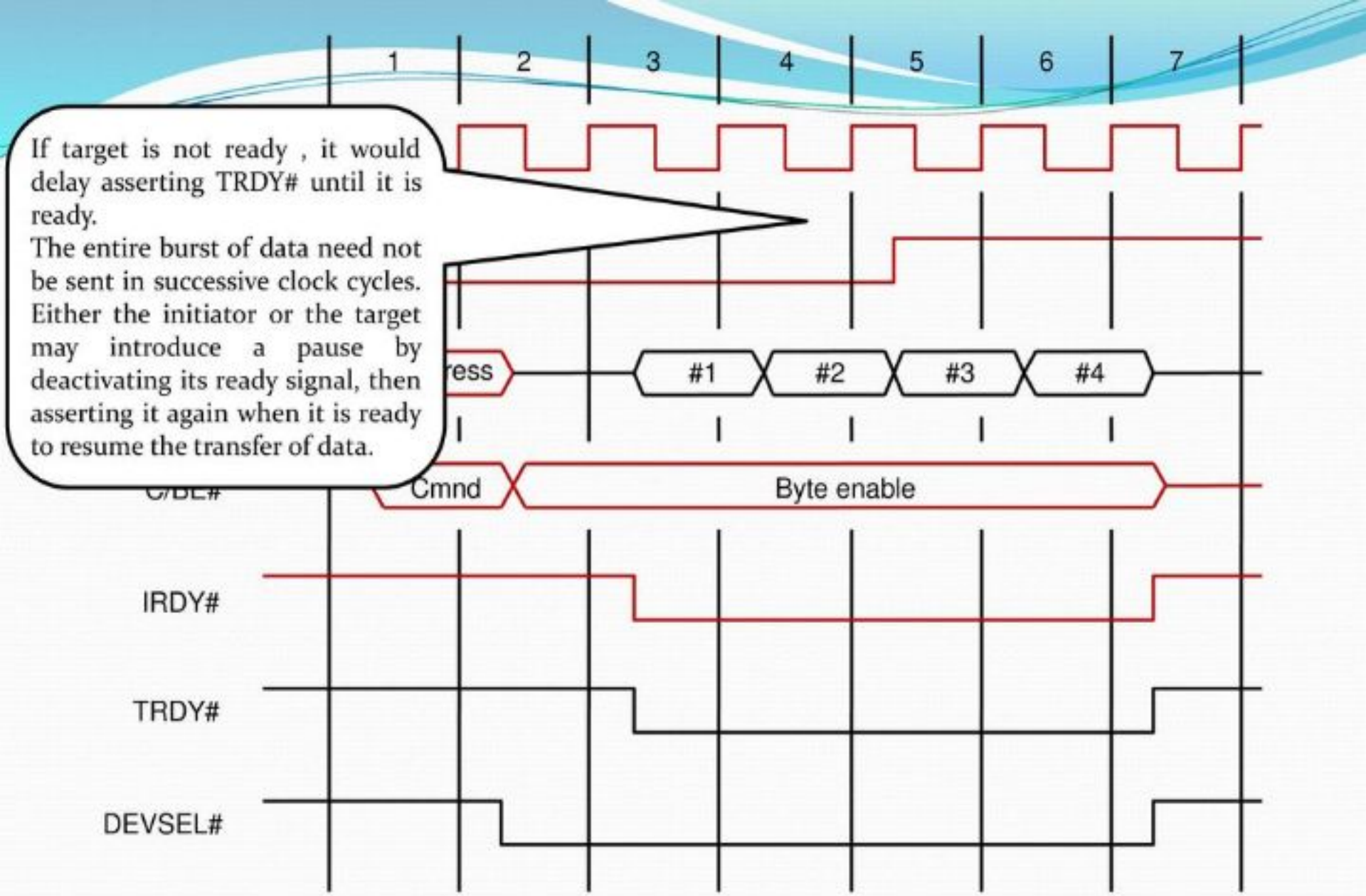
A read operation on the PCI bus



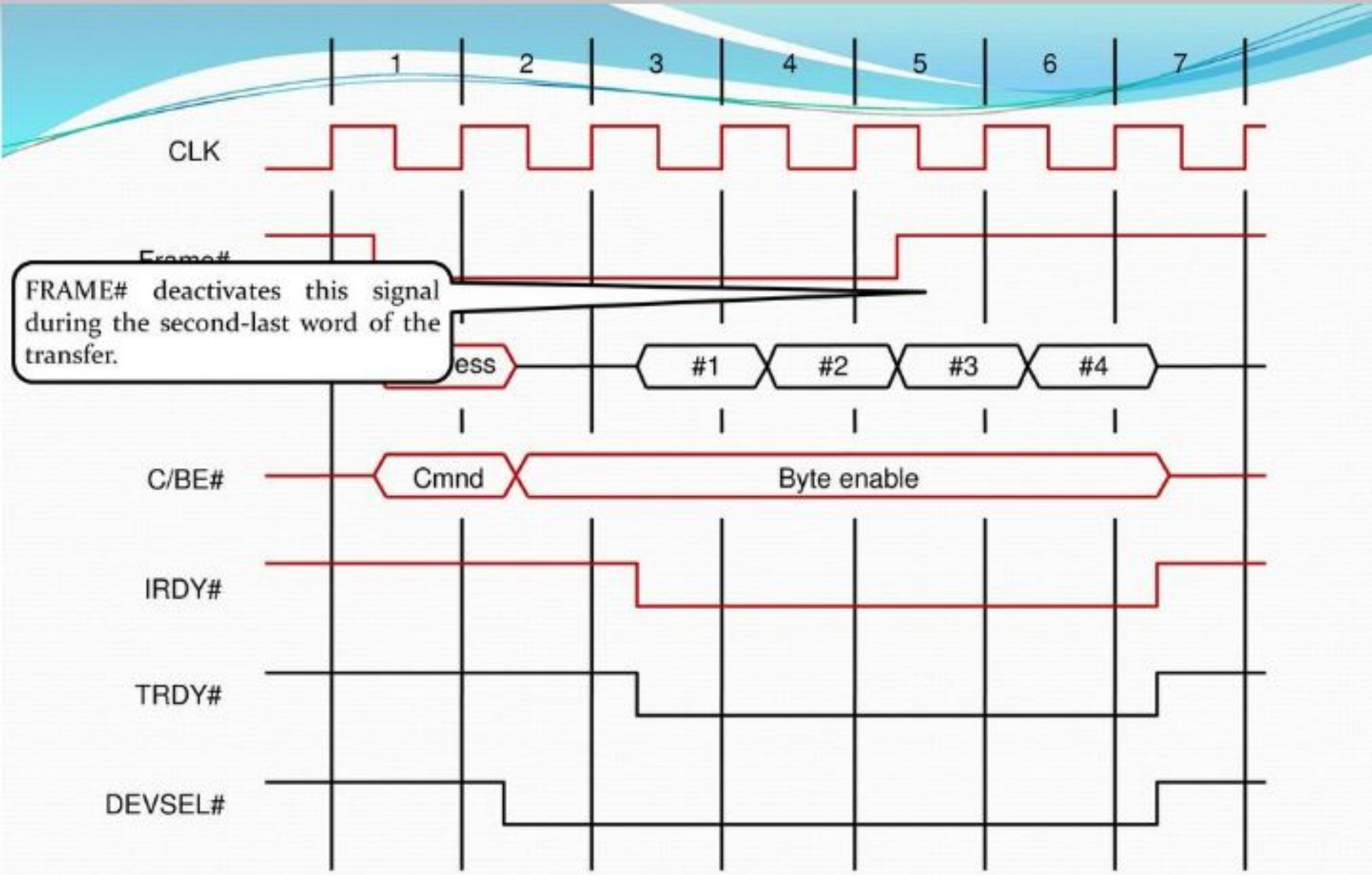
A read operation on the PCI bus



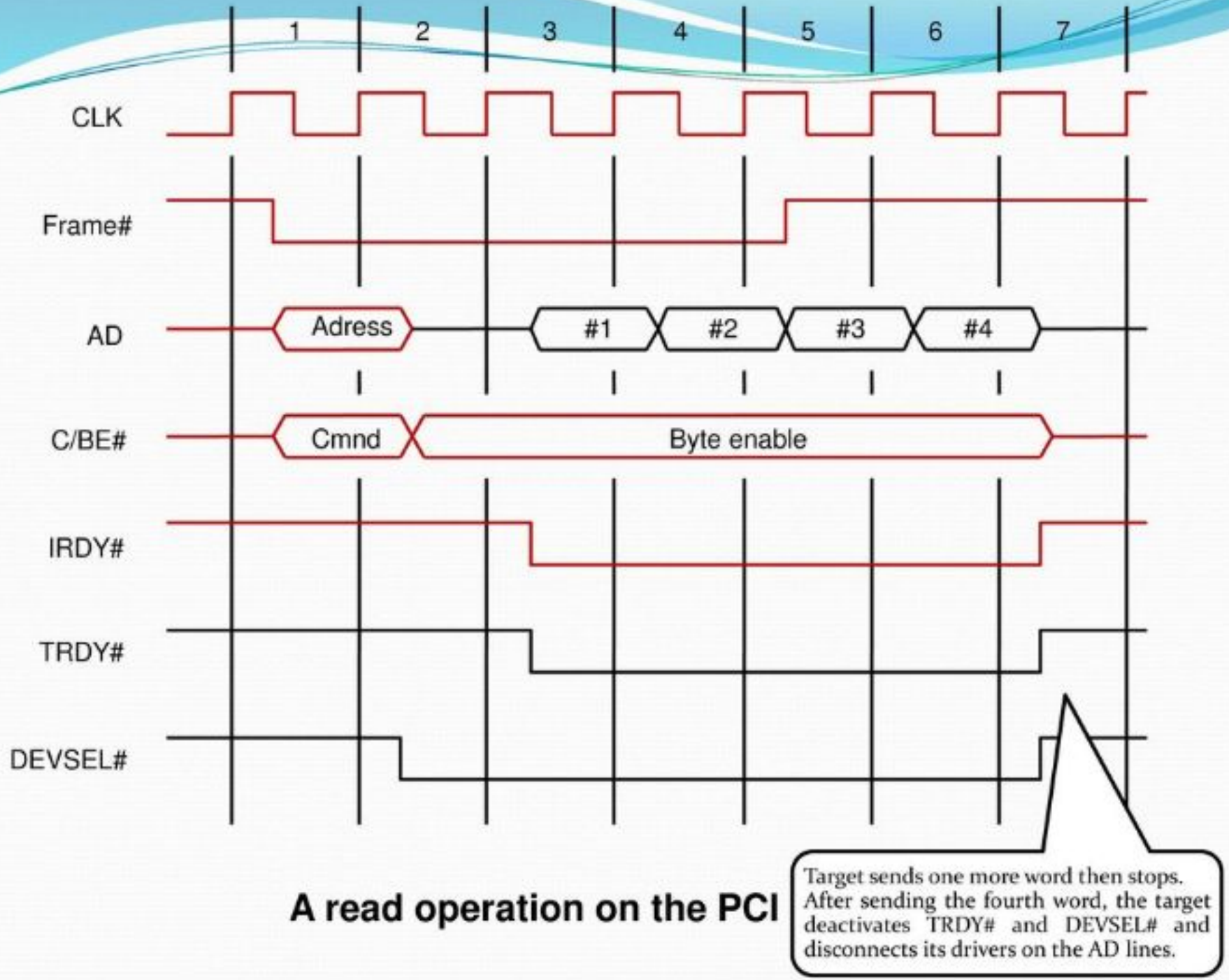
A read operation on the PCI bus



A read operation on the PCI bus



A read operation on the PCI bus



A read operation on the PCI

SCSI Bus (Small Computer System Interface)

- It refers to a standard bus defined by the American National Standards Institute (ANSI) under the designation X3.131

- In the original specifications of the standard, devices such as disks are connected to a computer via a 50-wire cable, which can be up to 25 meters in length and can transfer data at rates up to 5 megabytes/s.

- The SCSI bus standard has undergone many revisions, and its data transfer capability has increased very rapidly, almost doubling every two years.

- SCSI-2 and SCSI-3 have been defined, and each has several options.

- Because of various options SCSI connector may have 50, 68 or 80 pins.

Category	Name	Function
Data	– DB(0) to – DB(7)	Data lines: Carry one byte of information during the information transfer phase and identify device during arbitration, selection and reselection phases
	– DB(P)	Parity bit for the data bus
Phase	– BSY	Busy: Asserted when the bus is not free
	–SEL	Selection: Asserted during selection and reselection
Information type	– C/D	Control/Data: Asserted during transfer of control information (command, status or message)
	– MSG	Message: indicates that the information being transferred is a message

Table 4. The SCSI bus signals.

Table 4. The SCSI bus signals.(*cont.*)

Category	Name	Function
Handshake	– REQ	Request: Asserted by a target to request a data transfer cycle
	– ACK	Acknowledge: Asserted by the initiator when it has completed a data transfer operation
Direction of transfer	– I/O	Input/Output: Asserted to indicate an input operation (relative to the initiator)
Other	– ATN	Attention: Asserted by an initiator when it wishes to send a message to a target
	– RST	Reset: Causes all device controls to disconnect from the bus and assume their start-up state

The processor sends a command to the SCSI controller, which causes the following sequence of events to take place:

1. The SCSI controller contends for control of the SCSI bus.

2. When it wins the arbitration process, the SCSI controller sends a command to the disk controller, specifying the required Read operation.

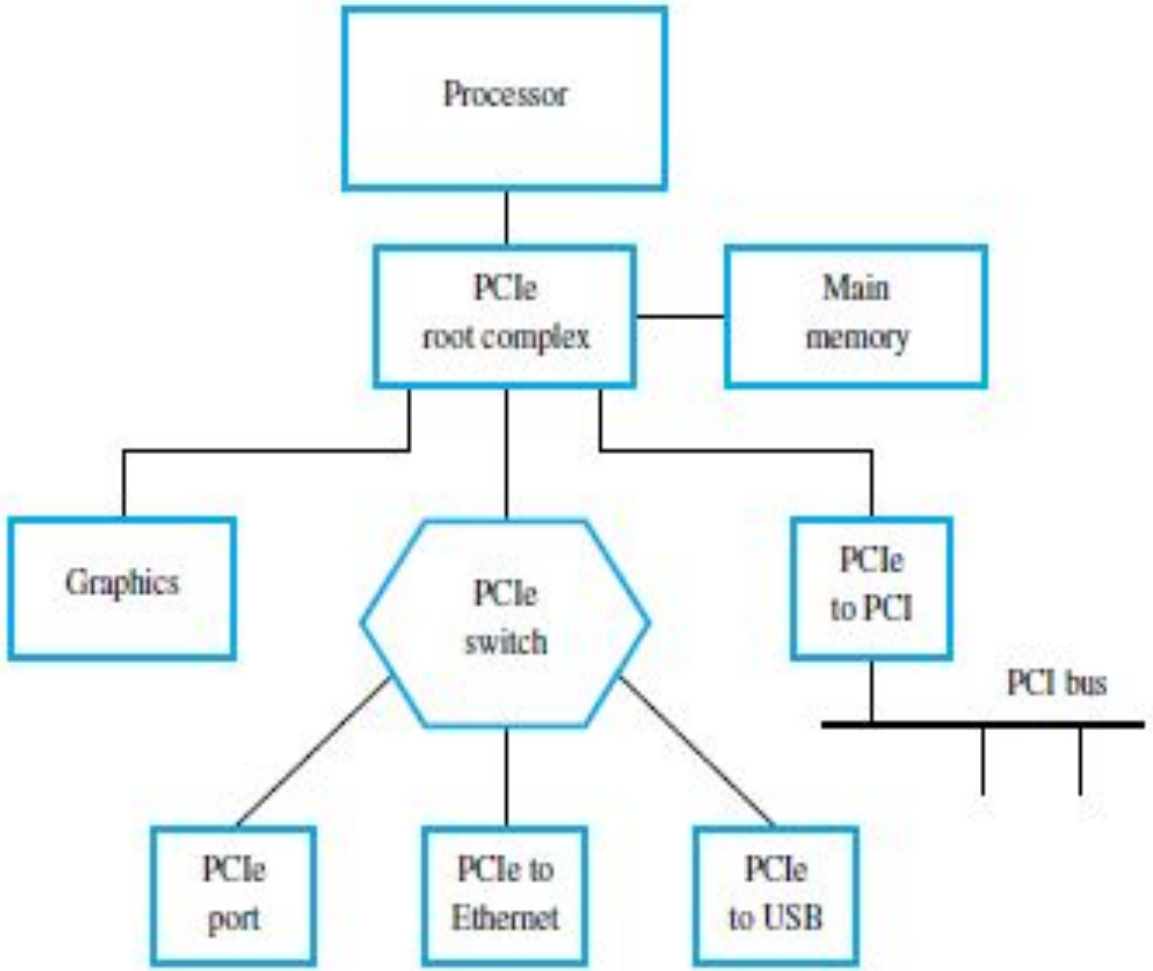
3. The disk controller cannot start to transfer data immediately. It must first move the read head of the disk to the required sector. Hence, it sends a message to the SCSI controller indicating that it will temporarily suspend the connection between them. The SCSI bus is now free to be used by other devices.

4.The disk controller sends a command to the disk drive to move the read head to the first sector involved in the requested Read operation. It reads the data stored in that sector and stores them in a data buffer. When it is ready to begin transferring data, it requests control of the bus. After it wins arbitration, it re-establishes the connection with the SCSI controller, sends the contents of the data buffer, then suspends the connection again.

5. The process is repeated to read and transfer the contents of the second disk sector.

6. The SCSI controller transfers the requested data to the main memory and sends an interrupt to the processor indicating that the data are now available.

PCI Express


Processor
PCIe
root complex
Main
memory
Graphics
PCIe
switch
PCIe
to PCI
PCI bus
PCIe
port
PCIe to
Ethernet
PCIe
to USB