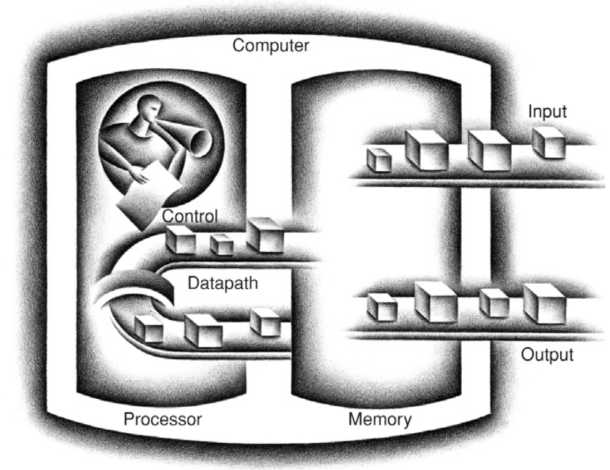
Computer Architecture
CSL3020

Deepak Mishra

http://home.iitj.ac.in/∼dmishra/
Department of Computer Science and Engineering
Indian Institute of Technology Jodhpur

I/O Devices

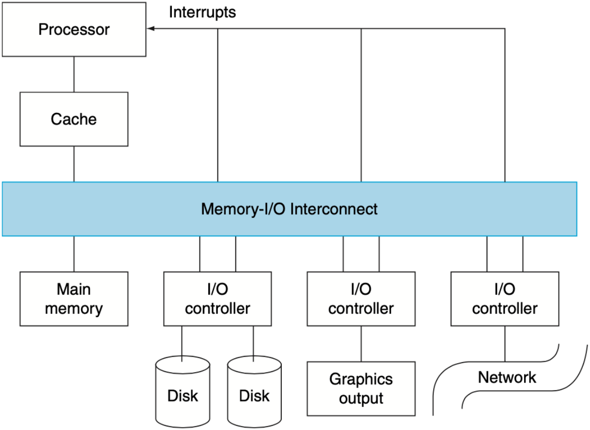


Peripheral devices are auxiliary devices that are connected to a computer to enhance its functionality.

I/O Devices

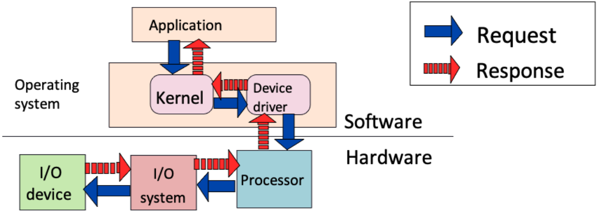
Device	Behavior
Keyboard	Input
Mouse	Input
Voice input	Input
Sound input	Input
Scanner	Input
Voice output	Output
Sound output	Output
Laser printer	Output
Graphics display	Output
Cable modem	Input or output
Network/LAN	Input or output
Network/wireless LAN	Input or output
Optical disk	Storage
Magnetic tape	Storage
Flash memory	Storage
Magnetic disk	Storage

I/O Devices



The connections between the I/O devices, processor, and memory are historically called **buses**.

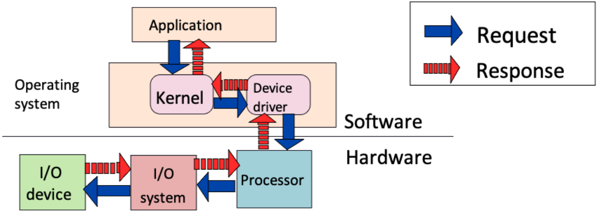
I/O Devices



A request goes through the kernel, device driver, processor, and I/O system.

Source: Basic Computer Architecture by SR Sarangi

I/O Devices

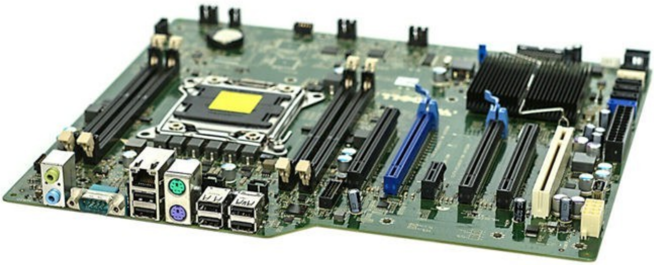


A request goes through the kernel, device driver, processor, and I/O system.

I/O devices are connected to the motherboard via add-on cards, or directly

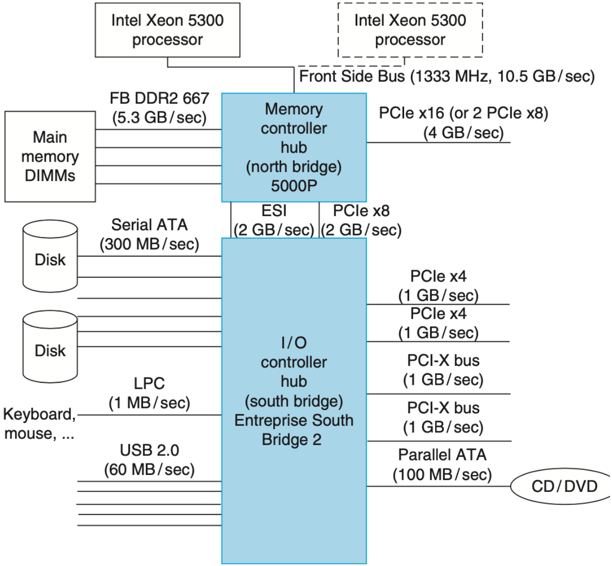
Source: Basic Computer Architecture by SR Sarangi

I/O Devices

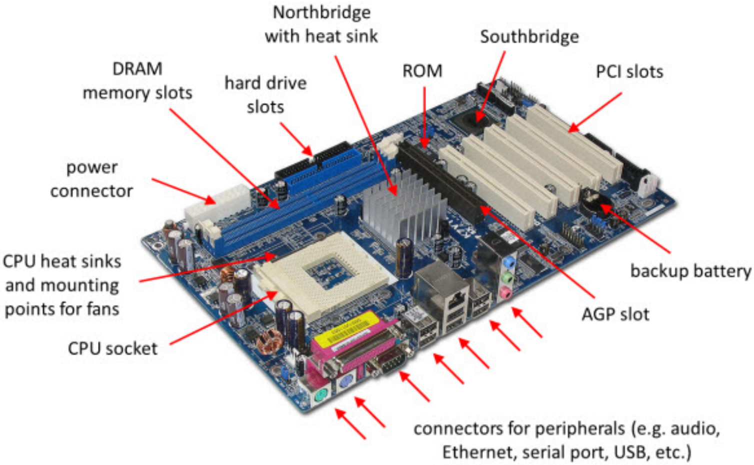


Dell Precision T3600 System Motherboard

Architecture of Motherboard



Architecture of Motherboard



Source: blogger.googleusercontent

Bus

- A bus is a shared communication link, which uses one set of wires to connect multiple subsystems.

Bus

- A bus is a shared communication link, which uses one set of wires to connect multiple subsystems.
- Versatile and low-cost

Bus

- A bus is a shared communication link, which uses one set of wires to connect multiple subsystems.
- Versatile and low-cost
- However creates a communication bottleneck.

Bus

- A bus is a shared communication link, which uses one set of wires to connect multiple subsystems.
- Versatile and low-cost
- However creates a communication bottleneck.
- Processor-memory buses, I/O buses, Backplane bus etc.

Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.

Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.
- Two disadvantages

Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.
- Two disadvantages
  - Every device on the bus must run at the same clock rate.
  - Clock skew problem

Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.
- Two disadvantages
  - Every device on the bus must run at the same clock rate.
  - Clock skew problem
- These problems led to asynchronous interconnects.

Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.
- Two disadvantages
  - Every device on the bus must run at the same clock rate.
  - Clock skew problem
- These problems led to asynchronous interconnects.
  - can accommodate a wide variety of devices of differing speeds.

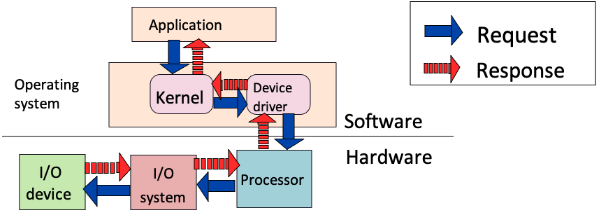
Bus

- Traditional buses are synchronous – A bus that includes a clock in the control lines and a fixed protocol for communicating that is relative to the clock.
- Two disadvantages
  - Every device on the bus must run at the same clock rate.
  - Clock skew problem
- These problems led to asynchronous interconnects.
  - can accommodate a wide variety of devices of differing speeds.
- To coordinate the transmission of data between sender and receiver, an asynchronous bus uses a *handshaking* protocol.

- A bus defines how data should be communicated on a set of wires.

Interfacing I/O Devices

- A bus defines how data should be communicated on a set of wires.
- There are several other tasks that must be performed to actually cause data to be transferred



The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.

Interfacing I/O Devices

The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.
- handle the interrupts generated by I/O devices.

Interfacing I/O Devices

The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.
- handle the interrupts generated by I/O devices.
- be able to give commands to the I/O devices.

The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.

Interfacing I/O Devices

The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.
- handle the interrupts generated by I/O devices.

Interfacing I/O Devices

The OS must:

- guarantee that a user's program accesses only the portions of an I/O device to which the user has rights.
- handle the interrupts generated by I/O devices.
- be able to give commands to the I/O devices.

Memory Mapped I/O:

- An I/O interfacing scheme in which portions of memory address space are assigned to I/O devices.

Memory Mapped I/O:

- An I/O interfacing scheme in which portions of memory address space are assigned to I/O devices.
- OS can use the paging mechanism to map I/O ports to memory addresses.

Interfacing I/O Devices

Memory Mapped I/O:

- An I/O interfacing scheme in which portions of memory address space are assigned to I/O devices.
- OS can use the paging mechanism to map I/O ports to memory addresses.
- Reads and writes to those addresses are interpreted as commands to the I/O device.

Memory Mapped I/O:

- An I/O interfacing scheme in which portions of memory address space are assigned to I/O devices.
- OS can use the paging mechanism to map I/O ports to memory addresses.
- Reads and writes to those addresses are interpreted as commands to the I/O device.
- Reading and writing to I/O devices requires normal load/store instructions.

Memory Mapped I/O:

- An I/O interfacing scheme in which portions of memory address space are assigned to I/O devices.
- OS can use the paging mechanism to map I/O ports to memory addresses.
- Reads and writes to those addresses are interpreted as commands to the I/O device.
- Reading and writing to I/O devices requires normal load/store instructions.
- The same I/O program can run on multiple machines.

Processor – I/O device communication:

- Fulfilling an I/O request for a program requires multiple operations.

Processor – I/O device communication:

- Fulfilling an I/O request for a program requires multiple operations.
- Processor needs to check device availability, error, task completion status etc.

Processor – I/O device communication:

- Fulfilling an I/O request for a program requires multiple operations.
- Processor needs to check device availability, error, task completion status etc.
- There are three methods of communication

Processor – I/O device communication:

- Fulfilling an I/O request for a program requires multiple operations.
- Processor needs to check device availability, error, task completion status etc.
- There are three methods of communication
  - Polling
  - Interrupts
  - Direct Memory Access (DMA)

Interfacing I/O Devices – Polling

Polling:

- The simplest method.

Polling:

- The simplest method.
- The I/O device simply puts the information in a Status register.

Interfacing I/O Devices – Polling

Polling:

- The simplest method.
- The I/O device simply puts the information in a Status register.
- The processor need to periodically check the status register.

Interfacing I/O Devices – Polling

Polling:

- The simplest method.
- The I/O device simply puts the information in a Status register.
- The processor need to periodically check the status register.
- It can waste a lot of processor time.

Interrupts:

- *Interrupt-driven I/O:* I/O interrupts indicate to the processor that an I/O device needs attention.

Interrupts:

- *Interrupt-driven I/O:* I/O interrupts indicate to the processor that an I/O device needs attention.
- Interrupts are similar to exceptions, often called as externally generated exceptions.

Exception vs. Interrupt

Exceptions and interrupts are events other than branches or jumps that change the normal flow of instruction execution.

Exception vs. Interrupt

Exceptions and interrupts are events other than branches or jumps that change the normal flow of instruction execution.

Type of event	From where?	MIPS terminology
I/O device request	External	Interrupt
Invoke the operating system from user program	Internal	Exception
Arithmetic overflow	Internal	Exception
Using an undefined instruction	Internal	Exception
Hardware malfunctions	Either	Exception or interrupt

Exception vs. Interrupt

When an exception occurs the processor (MIPS) performs following steps.

- Save the address of the offending instruction in the exception program counter (EPC).

Exception vs. Interrupt

When an exception occurs the processor (MIPS) performs following steps.

- Save the address of the offending instruction in the exception program counter (EPC).
- Then, transfer control to the operating system at some specified address e.g. $8000\ 0000_{hex}$.

Exception vs. Interrupt

When an exception occurs the processor (MIPS) performs following steps.

- Save the address of the offending instruction in the exception program counter (EPC).
- Then, transfer control to the operating system at some specified address e.g. $8000\ 0000_{hex}$.

MIPS includes a status register (called the Cause register), which holds a field that indicates the reason for the exception.

Exception vs. Interrupt

Interrupts have important distinctions from exceptions.

- An I/O interrupt is asynchronous with respect to the instruction execution.

Exception vs. Interrupt

Interrupts have important distinctions from exceptions.

- An I/O interrupt is asynchronous with respect to the instruction execution.
  - An interrupt does not prevent the instruction completion.

Exception vs. Interrupt

Interrupts have important distinctions from exceptions.

- An I/O interrupt is asynchronous with respect to the instruction execution.
    - An interrupt does not prevent the instruction completion.
- Identity of the device generating the interrupt needs to be communicated to the processor.

Exception vs. Interrupt

Interrupts have important distinctions from exceptions.

- An I/O interrupt is asynchronous with respect to the instruction execution.
  - An interrupt does not prevent the instruction completion.
- Identity of the device generating the interrupt needs to be communicated to the processor.
  - Cause register is used for this purpose.

Interfacing I/O Devices – DMA

- Interrupt-driven I/O method is not suitable for tasks like transferring of large blocks of data from hard disk.

Interfacing I/O Devices – DMA

- Interrupt-driven I/O method is not suitable for tasks like transferring of large blocks of data from hard disk.
- The entire operation will require a large amount of CPU time.

Interfacing I/O Devices – DMA

- Interrupt-driven I/O method is not suitable for tasks like transferring of large blocks of data from hard disk.
- The entire operation will require a large amount of CPU time.
- DMA enable transfer of data between an I/O device and memory without involving the processor.

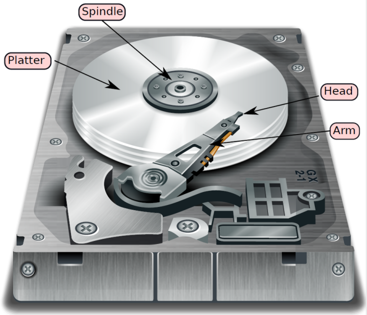
Interfacing I/O Devices – DMA

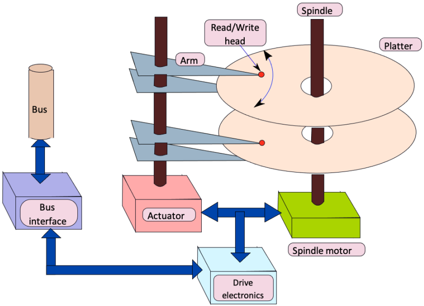
- Interrupt-driven I/O method is not suitable for tasks like transferring of large blocks of data from hard disk.
- The entire operation will require a large amount of CPU time.
- DMA enable transfer of data between an I/O device and memory without involving the processor.
  - The processor supplies the addresses in memory, size of data, and I/O address locations to the DMA controller.
  - The DMA starts the operation on the device and arbitrates for the interconnect.
  - Once the DMA transfer is complete, the controller interrupts the processor.

Interfacing I/O Devices – DMA

- Interrupt-driven I/O method is not suitable for tasks like transferring of large blocks of data from hard disk.
- The entire operation will require a large amount of CPU time.
- DMA enable transfer of data between an I/O device and memory without involving the processor.
  - The processor supplies the addresses in memory, size of data, and I/O address locations to the DMA controller.
  - The DMA starts the operation on the device and arbitrates for the interconnect.
  - Once the DMA transfer is complete, the controller interrupts the processor.
- Unlike polling or interrupt-driven I/O, DMA can be used to interface a hard disk without consuming all the processor cycles for a single I/O.

I/O Devices – Hard Disks

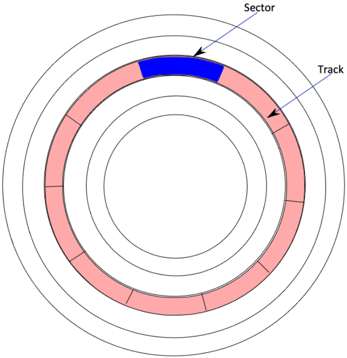




Hard Disk Drives (HDDs) commonly referred to magnetic disks that rely on a rotating platter coated with a magnetic surface (sequence of tiny magnets) and use a movable read/write head to access the disk.

Source: Basic Computer Architecture by SR Sarangi

I/O Devices – Hard Disks



- Each disk surface is divided into concentric circles, called *tracks*.
- Each track is in turn divided into *sectors* that contain the information; each track may have 100 to 500 sectors.
- Sectors are typically 512 bytes in size.

To access data, the OS must direct the disk through a three-stage process.

To access data, the OS must direct the disk through a three-stage process.

- *Seek:* The process of positioning a read/write head over the proper track on a disk.

I/O Devices – Hard Disks

To access data, the OS must direct the disk through a three-stage process.

- *Seek:* The process of positioning a read/write head over the proper track on a disk.
- Wait for the desired sector to come under the head. This time is called the rotational latency or rotational delay.

I/O Devices – Hard Disks

To access data, the OS must direct the disk through a three-stage process.

- *Seek:* The process of positioning a read/write head over the proper track on a disk.
- Wait for the desired sector to come under the head. This time is called the rotational latency or rotational delay.
- Read or Write data. This time is called as transfer time.

I/O Devices – Hard Disks

To access data, the OS must direct the disk through a three-stage process.

- *Seek:* The process of positioning a read/write head over the proper track on a disk.
- Wait for the desired sector to come under the head. This time is called the rotational latency or rotational delay.
- Read or Write data. This time is called as transfer time.

A disk controller usually handles the detailed control of the disk and the transfer between the disk and the memory.

To access data, the OS must direct the disk through a three-stage process.

- *Seek:* The process of positioning a read/write head over the proper track on a disk.
- Wait for the desired sector to come under the head. This time is called the rotational latency or rotational delay.
- Read or Write data. This time is called as transfer time.

A disk controller usually handles the detailed control of the disk and the transfer between the disk and the memory.

Disk access time = Seek time + Rotational delay + Transfer time + Controller overhead

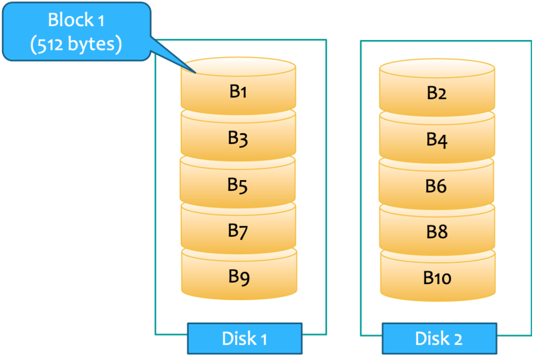
I/O Devices – RAID

RAID (redundant array of independent disks) employs the techniques of striping, mirroring, or parity to create large reliable data stores from multiple HDDs.

Source: Basic Computer Architecture by SR Sarangi

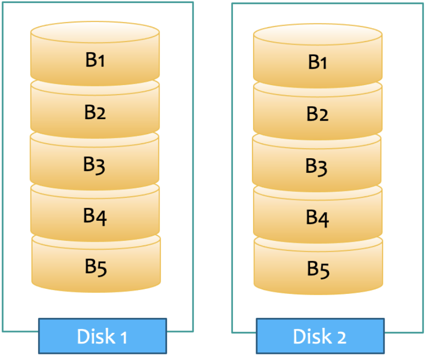
RAID (redundant array of independent disks) employs the techniques of striping, mirroring, or parity to create large reliable data stores from multiple HDDs.

RAID 0: No redundancy, only data stripping.



Source: Basic Computer Architecture by SR Sarangi

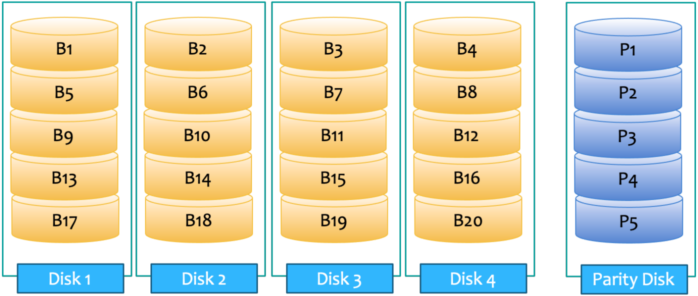
RAID 1: Mirroring, immune to one disk failure, 100% overhead in storage.



Source: Basic Computer Architecture by SR Sarangi

RAID 2: Rarely used in practice, stripes data at the bit (rather than block) level, and uses a Hamming code for error correction.

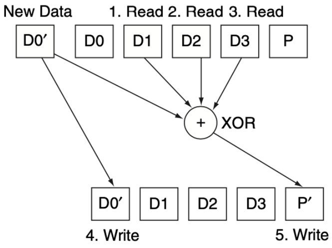
RAID 3: uses striping and dedicates one drive for storing *parity* information.



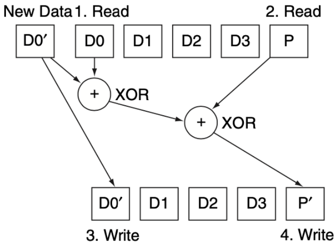
$$P1 = B1 \oplus B2 \oplus B3 \oplus B4$$

Source: Basic Computer Architecture by SR Sarangi

RAID 4: Improvised version of RAID 3.

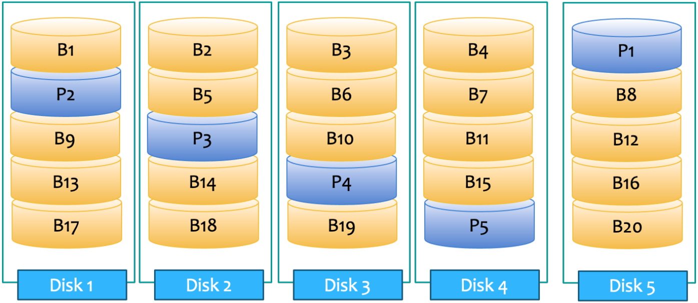


RAID 3 Write



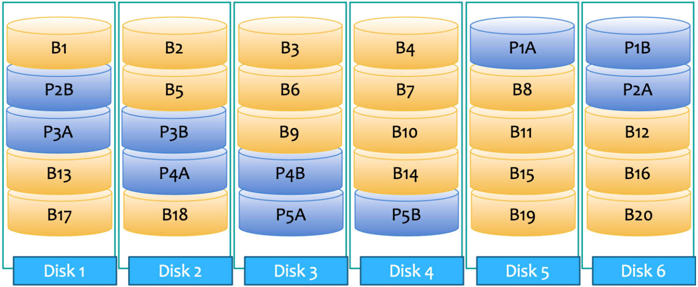
RAID 4 Write

RAID 5: Distributes the parity blocks across the disks. High bandwidth.



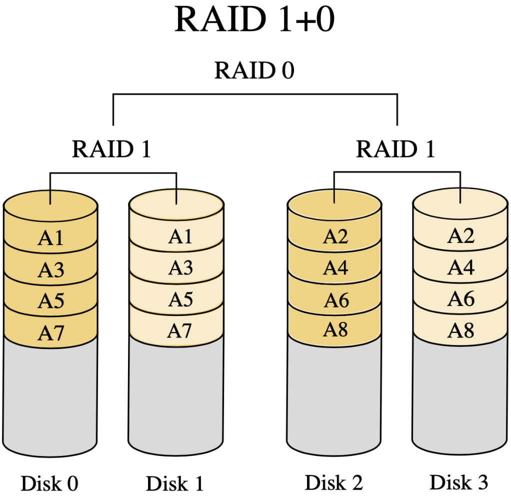
Source: Basic Computer Architecture by SR Sarangi

RAID 6: Immune to two disk failures. High reliability at the cost of increased storage overhead.



Source: Basic Computer Architecture by SR Sarangi

RAID 10: Nested RAID level, also called RAID 1+0. RAID 0 array of mirrors.



Source: Wikipedia

I/O Devices – SSD

- A solid-state drive (SSD) uses integrated circuits to store data.
- SSDs rely on non-volatile memory, typically NAND flash.
- Latency is 100–1000 times faster than disk.
- It is comparatively smaller, more power efficient, and more shock resistant.