[Chapter- 8]

[Input Output Systems]

Input Output Devices → To make a computer system interactive some input (Keyboard, mouse scanner etc) and output (printer, plotter) devices are attached with the system.

They can be broadly grouped into following three classes:

(a) Human readable → used to communicate with the user eg. mouse, printer etc.

(b) Machine readable → used to communicate with electronic equipment eg. sensors, controller and actuators.

(c) Communication → used to communicate with remote devices eg: modems.

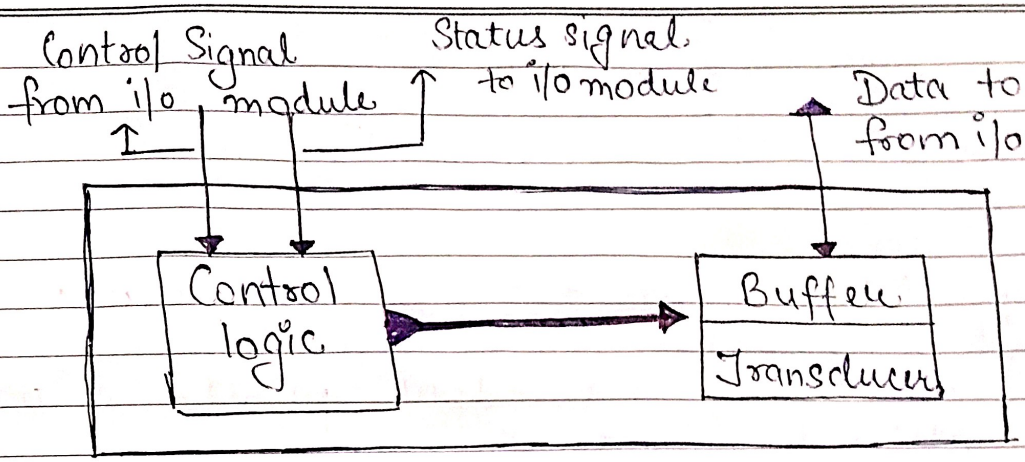
Hardware Support for I/O →

→ With a computer, huge variety of i/o devices can be used.

→ The data transfer rate of these device is often much slower than that of memory or CPU.

→ I/o devices often use different data formats and word length than the computer to which they are attached. Because of these reasons one does not connect i/o devices directly to the system bus.

→ An interface called i/o module or controller is used between memory / CPU and i/o device.



[OVERVIEW OF I/O DEVICE]

→ I/O Devices can send and receive data to and from i/o module.

→ Control signal sent by the i/o modules decides whether to read from the devices or write to the devices

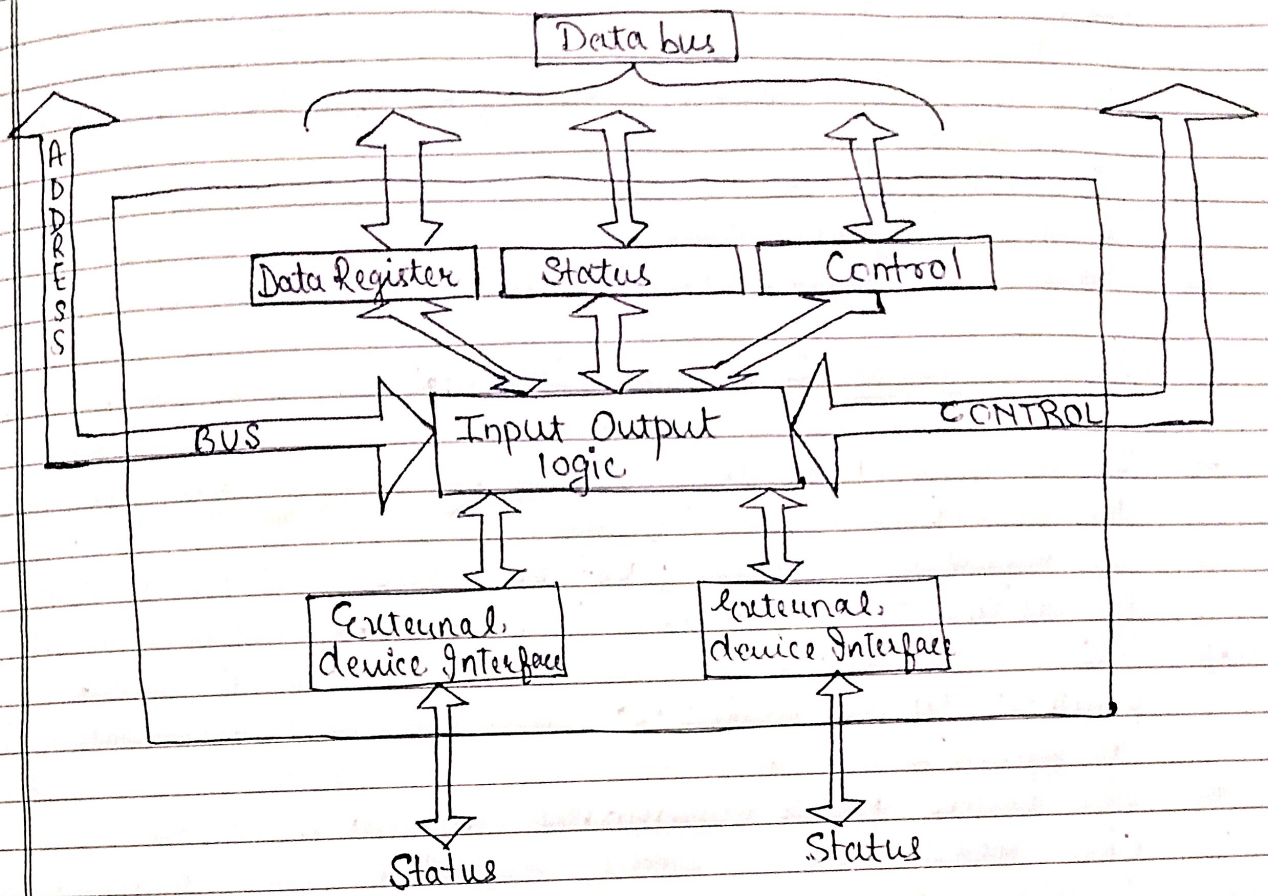
→ The Status signal indicates whether the device is ready for data transfer or busy in work.

→ Control logic control the operation of device in response to the direction from the i/o module.

→ Transducer is responsible to convert data from electrical to mechanical signals in case of output devices and mechanical to electrical in case of input device.

→ Each device also has a piece of memory called buffer to temporary hold the data being transferred

→ Some of i/o modules are attached within the mother board and some may be attached with the i/o devices itself.



[STRUCTURE OF I/O MODULES]

→ The I/o module connects to the rest of the computer through a system bus.

→ Data Register → Data transferred to and from the module are buffered in one or more data registers

→ Status Register → Status Register provide status information.

→ Control Register → control register can accept detailed control information from the CPU.

→ I/O Logic → interacts with the CPU via Control bus.

→ Address bus → i/o module uses address bus to recognize and generate address associated with the devices it control.

I/O Communication Techniques →

The following three techniques are possible for i/o operations :-

- Programmed i/o
- Interrupt driven i/o
- Direct Memory Access (DMA)

① **Programmed I/O** → CPU's task is to execute the instruction of a program.

→ CPU execute that instruction by issuing the command to the appropriate i/o module.

→ I/o module in turn perform the requested i/o operation and set the bits in the status register.

→ It is the responsibility of the CPU to check periodically the status register of the i/o module to determine whether the given task is completed.

(a) CPU issues Read command to i/o module.
(b) CPU read reads status register of i/o module.
(c) If i/o module is not ready repeat the step ②
(d) otherwise, Read from i/o module
(e) write data into main memory

In this technique CPU wants to read from the i/o device. The disadvantage of this technique is that CPU remain unnecessary busy in checking status register to determine whether i/o module is free or not. This technique is known as **polling**.

(2) Interrupt - Driven I/O → In programmed i/o, CPU is unnecessary busy in waiting i/o module to be free. To overcome this problem CPU issues an i/o command to a module and start doing some other work.
The i/o module will interrupt the CPU whenever it become free to accomplish next task.

(a) CPU issues Read command to i/o module.
(b) CPU start executing instruction of some other other program.
(c) I/o module when complete its prior assignment issue an interrupt signal to CPU.
(d) CPU reads status bits from status register of i/o module.
(e) If i/o module is still busy, CPU issues error message.
(f) Otherwise CPU reads data from i/o module.
(g) CPU writes data into memory.

This solution prevents the CPU from busy waiting situation but it requires context switching.

(3) Direct Memory Access (DMA) →

→ Although interrupt-driven i/o removes the problem of busy waiting of CPU but still CPU is busy in doing i/o.
→ CPU act as intermediary between memory and i/o module.
→ But if a large data transfer are required, it seems wasteful to use a general purpose processor to watch status bit and to feed data into i/o

module registers.

→ Instead of using general purpose processor a dedicated special purpose processor called DMA Controller can be used.

→ When a processor wishes to read or write a block of data, it issues a command to the DMA module by sending following information to the DMA module:
(a) what is required Read or write.
(b) Address of the i/o device.
(c) Starting location of memory from where read or write will take place.
(d) Number of bytes to be read or written.

→ CPU handed over the i/o operation to DMA module and continue with other work. The DMA module transfer the entire block of data directly to or form memory without involving the CPU.

→ Only one overhead of this technique is that since control of the bus is handed over to DMA, there may be many times when the CPU needs the bus and must wait for it.

I/O SOFTWARE LAYER →

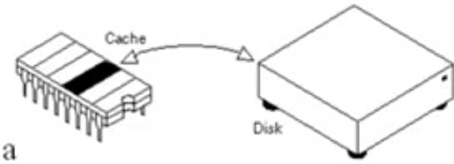
User-Level i/o S/W
Device independent OS S/W
Device Driver
Interrupt Handler
Hardware

Disk Caching

❑**Definition:** A disk cache (cache memory) is a temporary holding area in the hard disk or random access memory (RAM) where the computer stores information that used repeatedly.

❑A disk cache is a **software utility** that works by reserving a section of memory where it keeps a copy of information that previously read from our disks.

❑The next time our computer needs that same information, the data can be accessed directly from the cache, bypassing the slower disk.



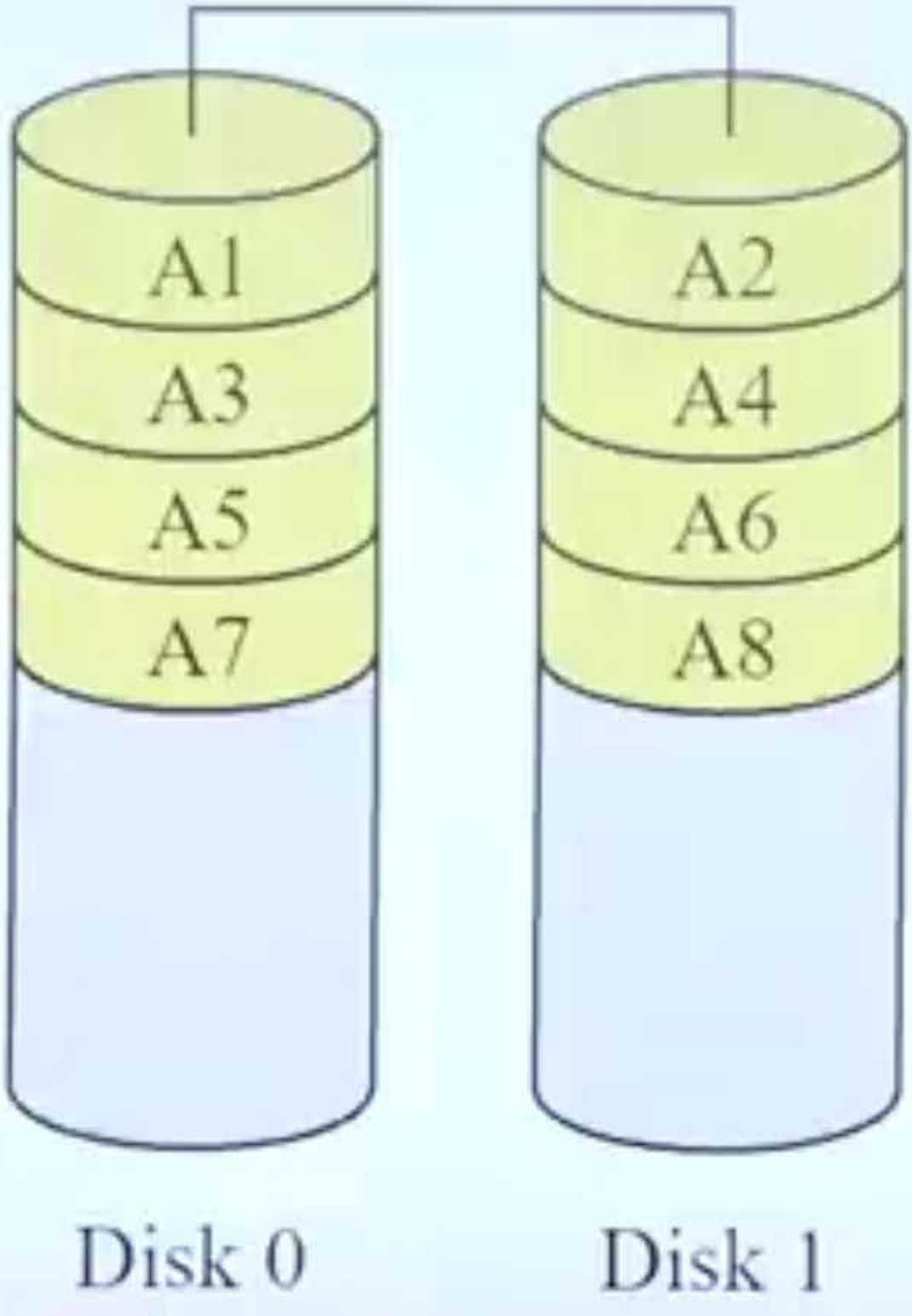
❑A disk cache is a **mechanism** for improve the time it takes to read from or write to a hard disk. [DC is portion of RAM used to speed up access to data on disk]

❑The basic idea behind a disk cache is that working with information stored in memory (RAM) is much faster than working with information stored on disk..

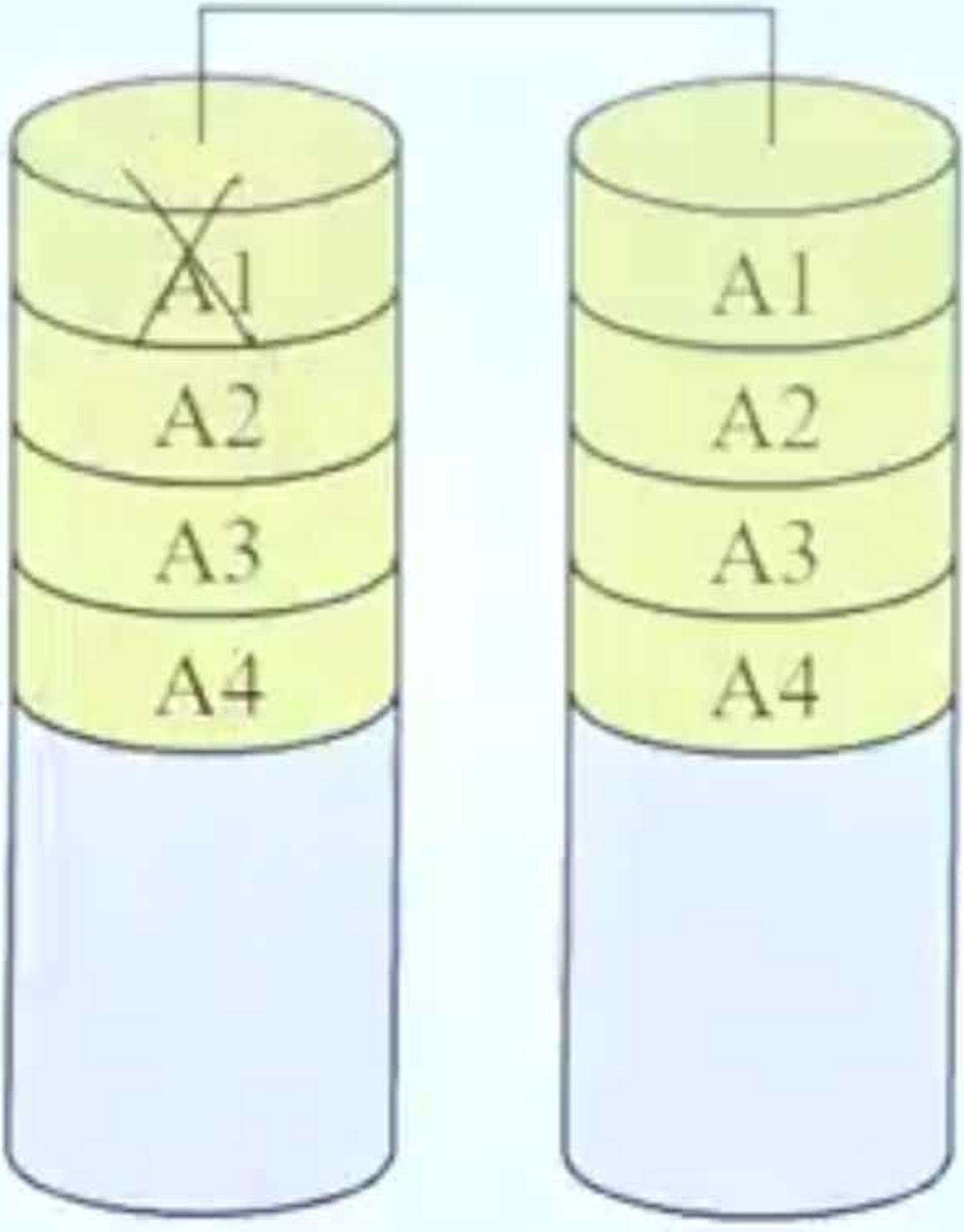
RAID

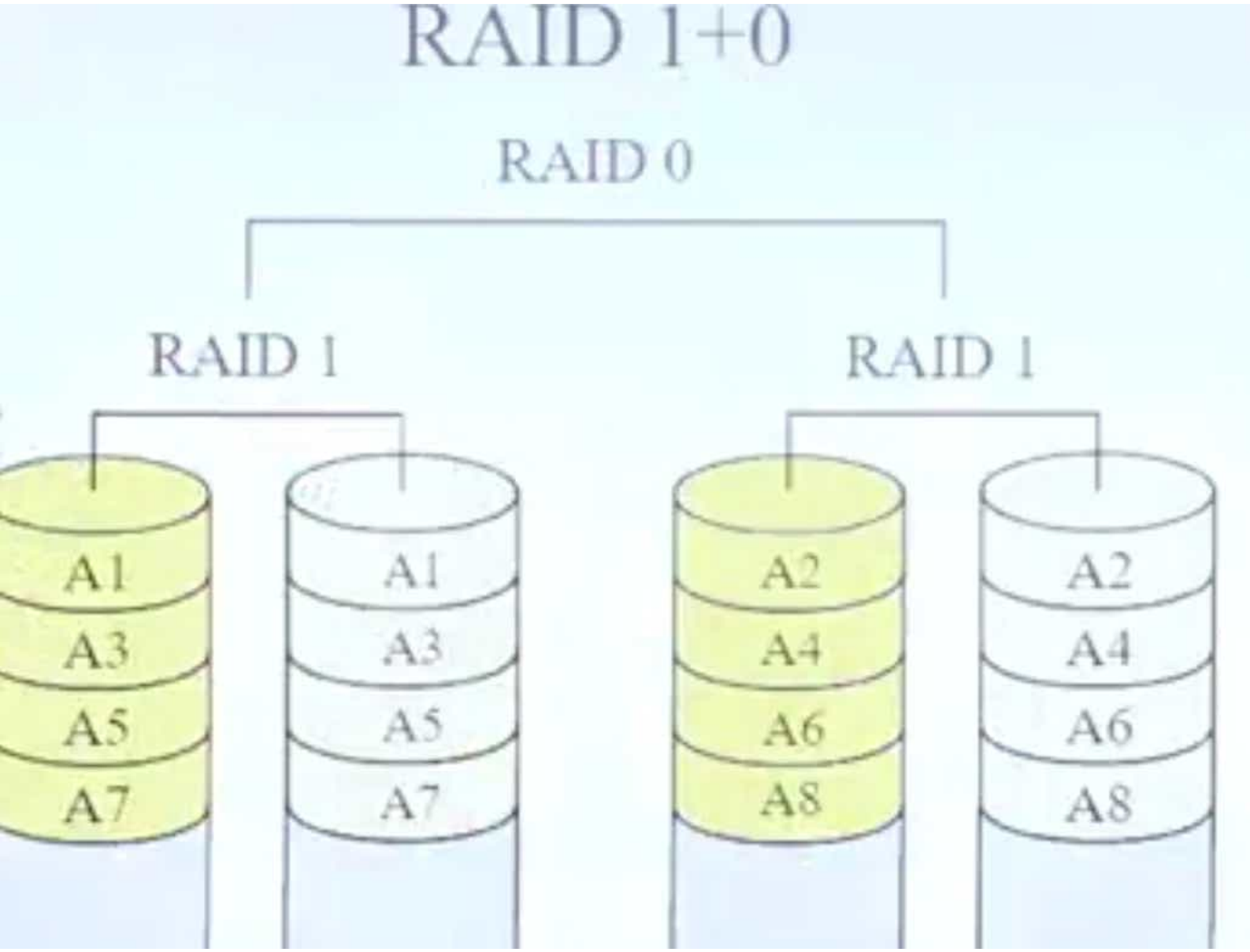
- **Redundant Array of Independent Disks**
- **Redundant Array of Inexpensive Disks**

RAID 0



RAID 1




RAID 1+0
RAID 0
RAID 1
RAID 1
A1
A3
A5
A7
A1
A3
A5
A7
A2
A4
A6
A8
A2
A4
A6
A8