SECR2033
Computer Organization and Architecture

Lecture slides prepared by “Computer Organization and Architecture”, 9/e, by William Stallings, 2013.

Module 7

I/O and Storage Systems

Objectives:

- To understand how **I/O systems** work, including **I/O methods** and **architectures**.
- To familiar with **storage media**, and the **differences** in their respective **formats**.
- To understand how **RAID** improves disk performance and reliability, and which RAID systems are most useful today.
- Be familiar with **emerging data storage technologies** and the barriers that remain to be overcome.

Module 7

I/O and Storage Systems

7.1 Introduction

7.2 I/O Architectures

7.3 Data Storage

7.4 Summary

7.1 Introduction

- All computers have I/O devices connected to them, and to achieve good performance I/O should be kept to a minimum!
- In studying I/O, we seek to understand the different types of I/O devices as well as how they work.
- Data storage and retrieval is one of the primary functions of computer systems.
    - One could easily make the argument that computers are more useful to us as data storage and retrieval devices than they are as computational machines.

Module 7

I/O and Storage Systems

7.1 Introduction

7.2 I/O Architectures

7.3 Data Storage

7.4 Summary

- ❑ Overview
- ❑ I/O and Performance
- ❑ I/O Control Methods
- ❑ I/O Bus Operation

Overview

- Input/output (I/O) is define as a subsystem of components that moves coded data between external devices and a host system, consisting of a CPU and main memory.
- I/O subsystems include, but are not limited to:

❑ Blocks of *main memory* → devoted to I/O functions.

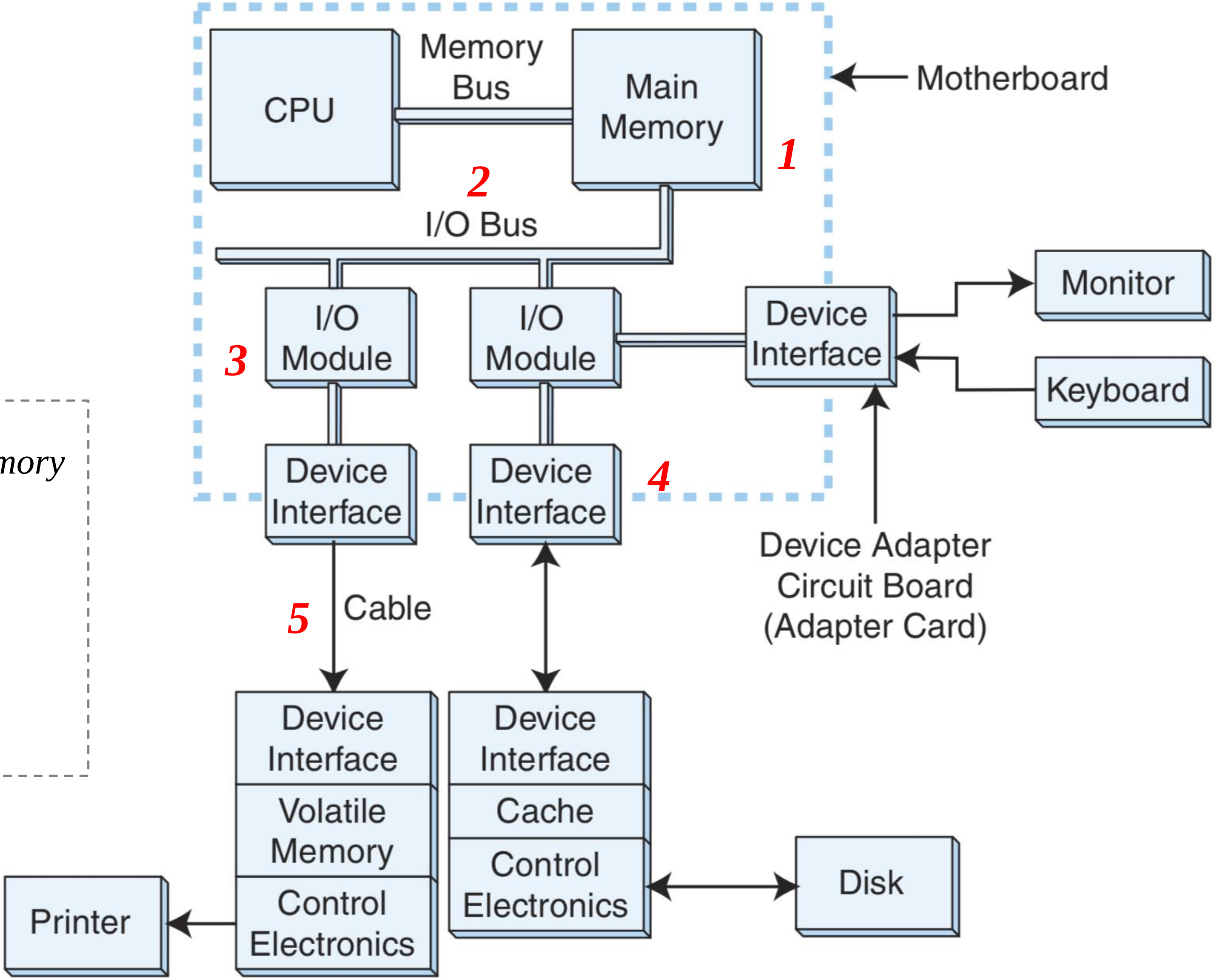
❑ *Buses* → move data into and out of the system.

❑ *Control modules* → in the host and in peripheral devices.

❑ *Interfaces* to external components → such as keyboards.

❑ *Cabling* or communications links → between the host system and its peripherals.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.275.



1 – Block of main memory

2 – Buses

3 – Control modules

4 – Interfaces

5 – Cabling

Figure: A model I/O configuration.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.276.

I/O Control Methods

- Computer systems employ any of four general I/O control methods.

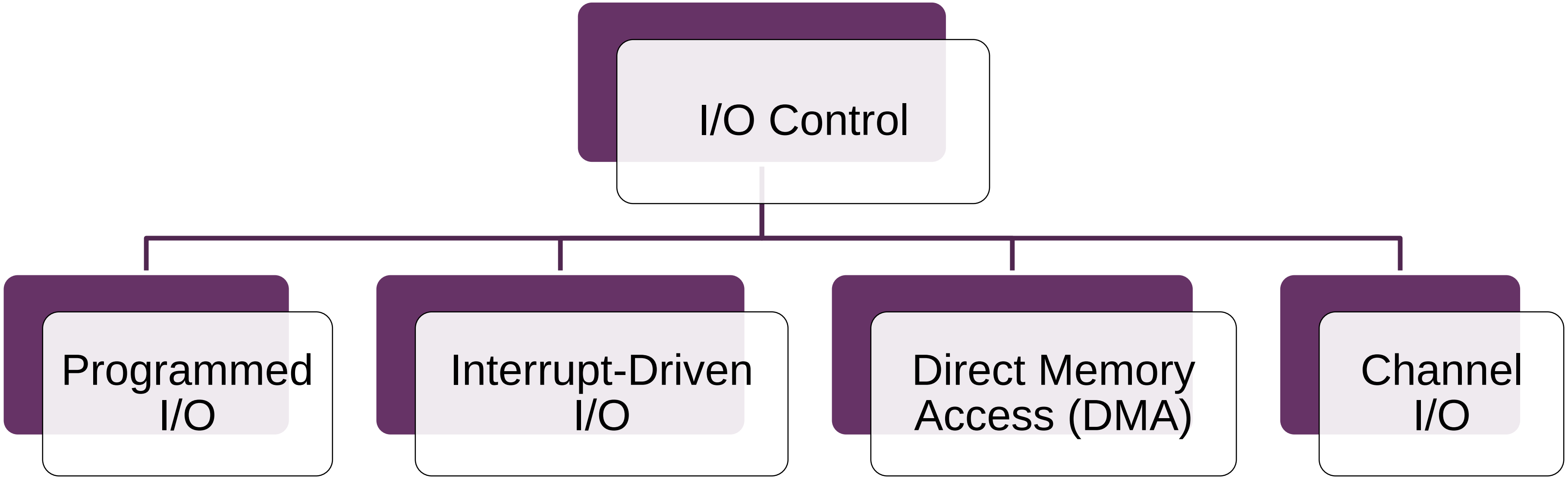


Figure: General I/O control methods

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.276.

(a) Programmed I/O

- Systems devote at least one register for the exclusive use of each I/O device.
- Programmed I/O sometimes referred *polled I/O* → The CPU continually monitors each register, waiting for data to arrive.

👍

- ❑ Simple.
- ❑ Less hardware.

👎

- ❑ Waste of processor's time (processor faster than I/O) – overhead in polling instead of actual data transfer.
- ❑ Worse for higher bandwidth I/O devices (*i.e*. disks)

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.276.

(b) Interrupt-Driven I/O

- Instead of the CPU continually asking its attached devices whether they have any input, the devices tell the CPU when they have data to send.
- Interrupts are usually signalled with a bit in the CPU flags register → *interrupt flag*.
- Once the *interrupt flag* is set, the operating system interrupts whatever program is currently executing, saving that program's state and variable information.
- After the CPU has completed servicing the I/O, it restores the information it saved from the program that was running when the interrupt occurred, and the program execution resumes.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.277.

INT (Interrupt)
INTA (Interrupt Acknowledge)

1 – Every peripheral device in the system has access to an interrupt request line.

2 – The controller signals the CPU when any of the interrupt lines are asserted.

3 – CPU is ready, assert `INTA` signal.

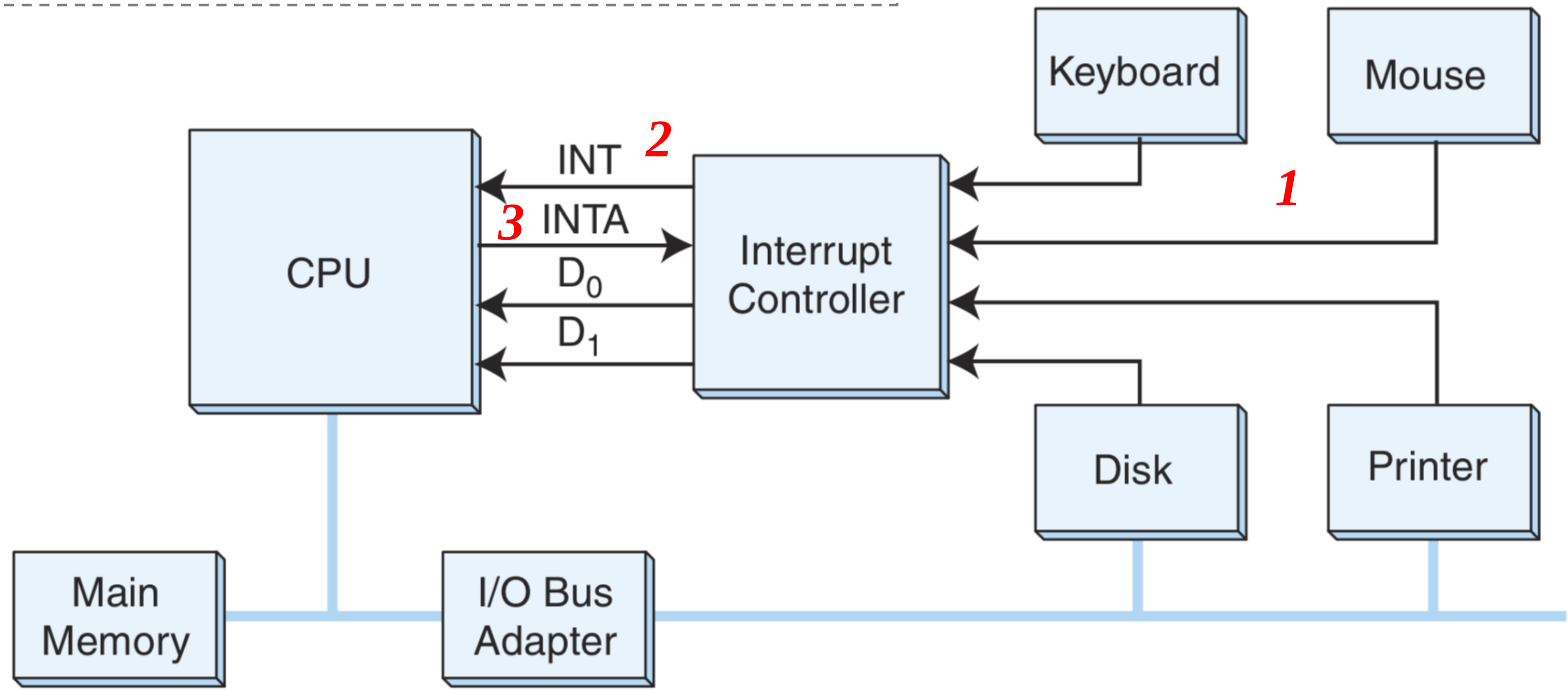


Figure: An I/O subsystem using interrupt.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.284-286.

(c) Direct Memory Access (DMA)

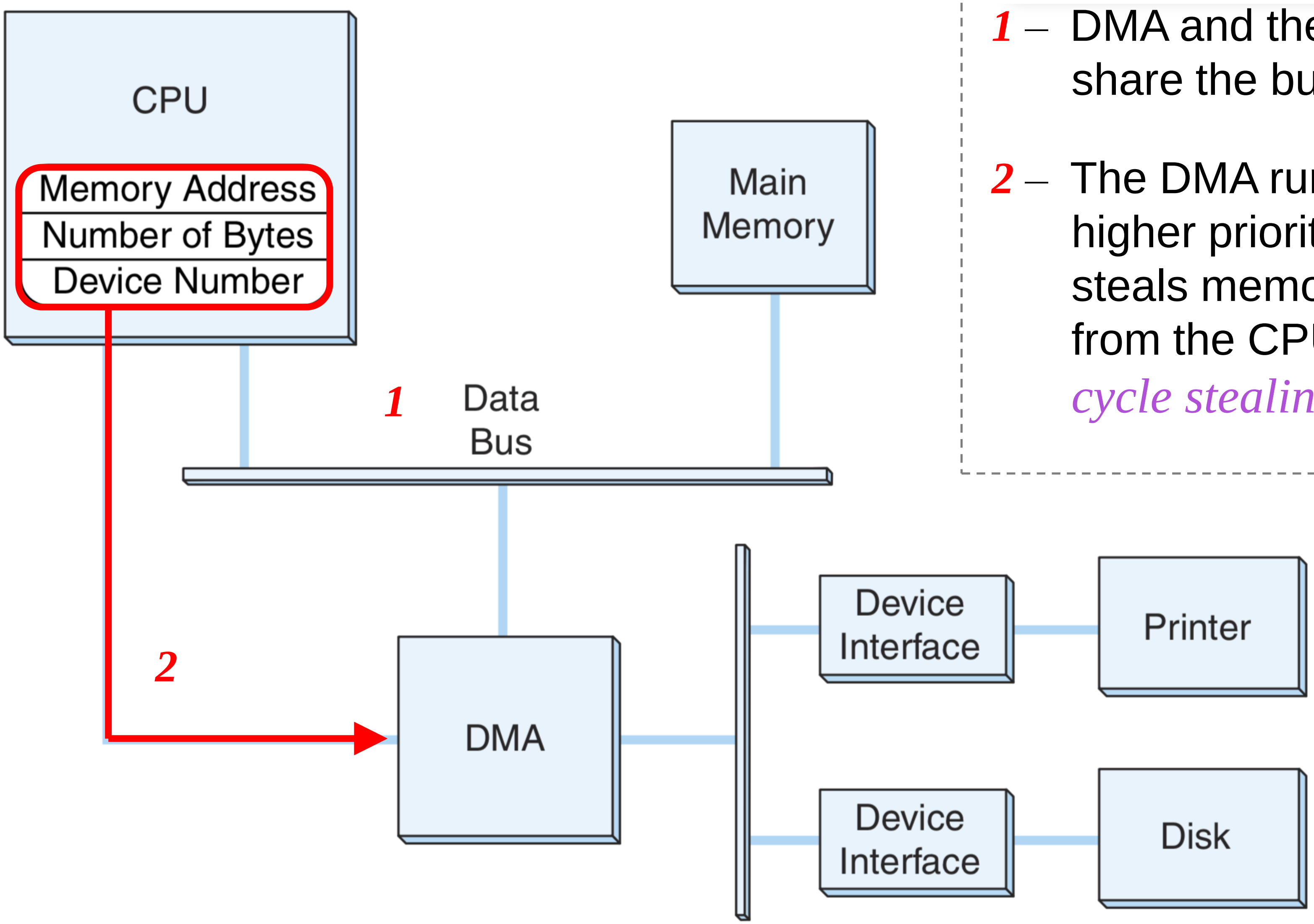
- With both programmed I/O and interrupt-driven I/O, the CPU moves data to / from the I/O device.
- When a system uses DMA, the CPU offloads execution of tedious I/O instructions.
- To effect the transfer, the CPU provides the DMA controller with:

✓ the location of the bytes to be transferred,

✓ the number of bytes to be transferred, and

✓ the destination device or memory address.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.277-278.



1 – DMA and the CPU share the bus.

2 – The DMA runs at a higher priority and steals memory cycles from the CPU (aka *cycle stealing*).

Figure: An example of DMA configuration.

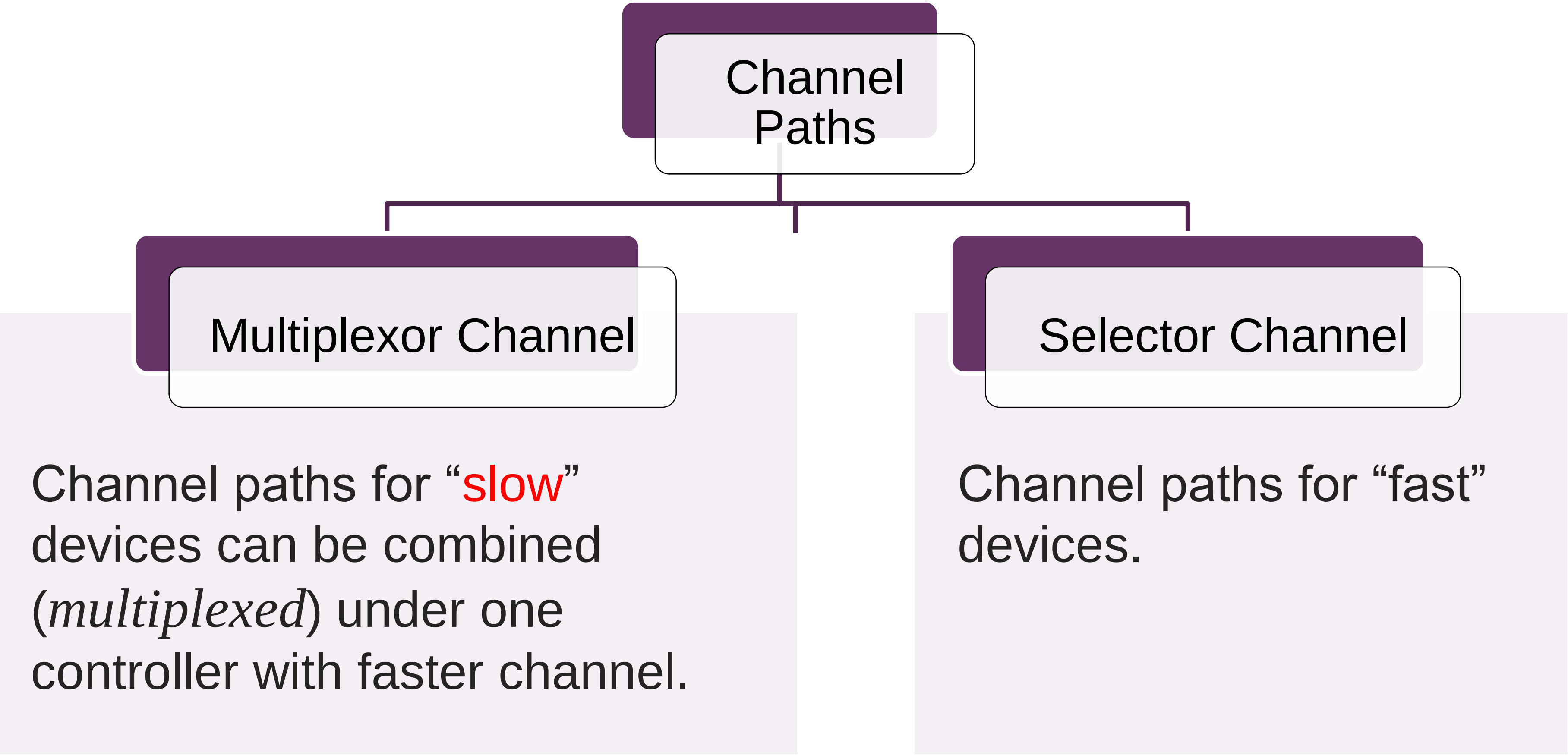
Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.278.

(d) Channel I/O

- DMA I/O requires only a little less CPU participation than does interrupt-driven I/O.
- Such overhead is fine for small, single-user systems; but it not scale well to large, multi-user systems such as mainframe computers.
- Therefore, an intelligent type of DMA interface known as an I/O channel is used

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.279.

- I/O channel → consists of one or more *I/O processors* (IOPs) that control various I/O pathways (*channel paths*).



Channel Paths

Multiplexor Channel

Channel paths for “slow” devices can be combined (*multiplexed*) under one controller with faster channel.

(e.g. terminals, printers)

Selector Channel

Channel paths for “fast” devices.

(e.g. disk drives)

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.279.

1 – IOPs = small CPUs.

2 – I/O buses – help to isolate the host from the I/O operation.

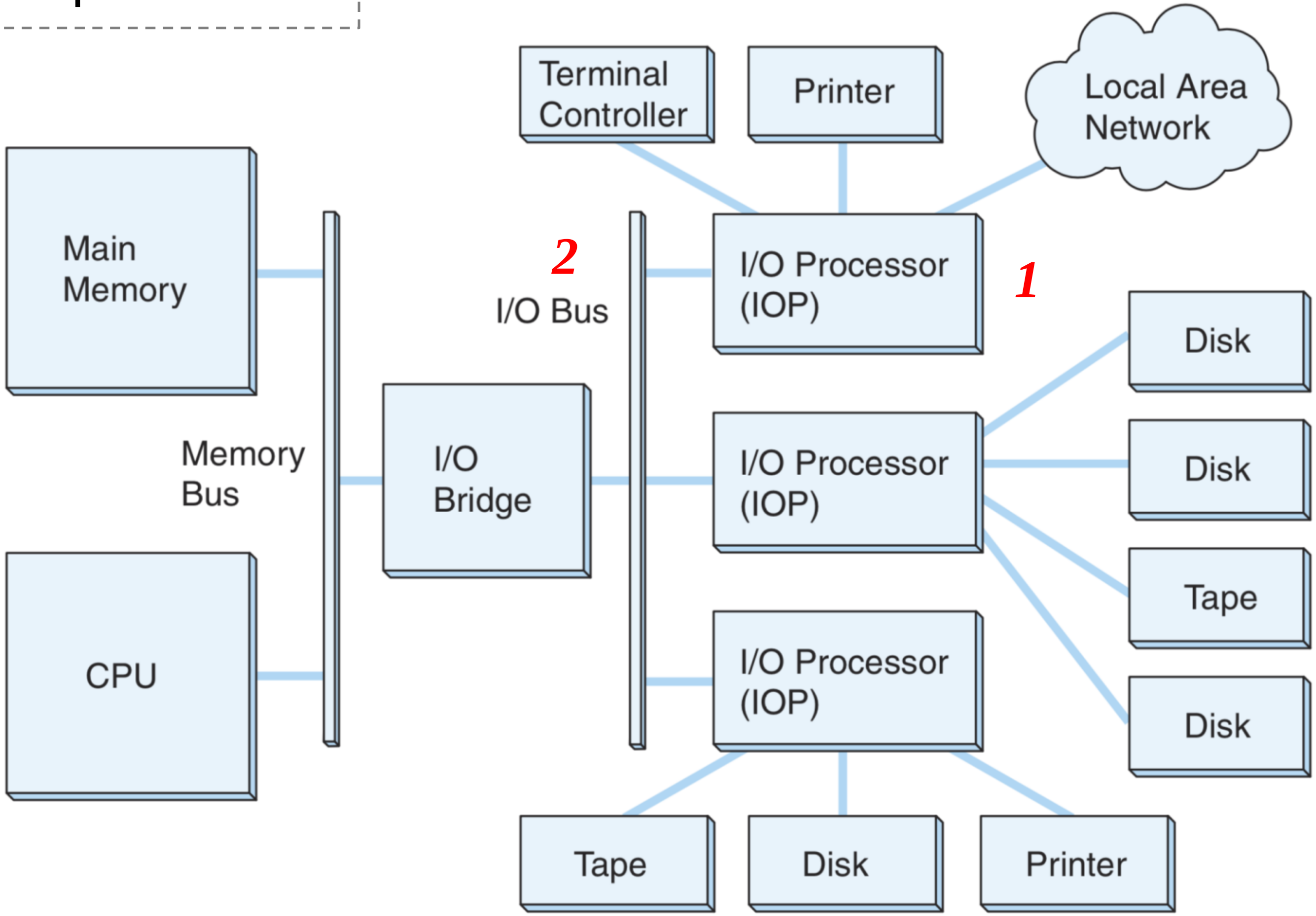


Figure: A channel I/O configuration.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.280.

1 – IOPs = small CPUs.

2 – I/O buses – help to isolate the host from the I/O operation.

<u>Example</u>:

Copying files from **disk** to **tape**, IOP uses only system <u>memory bus</u> to fetch its instruction from <u>main memory</u>.

The remainder transfer is affected using only <u>I/O bus</u>.

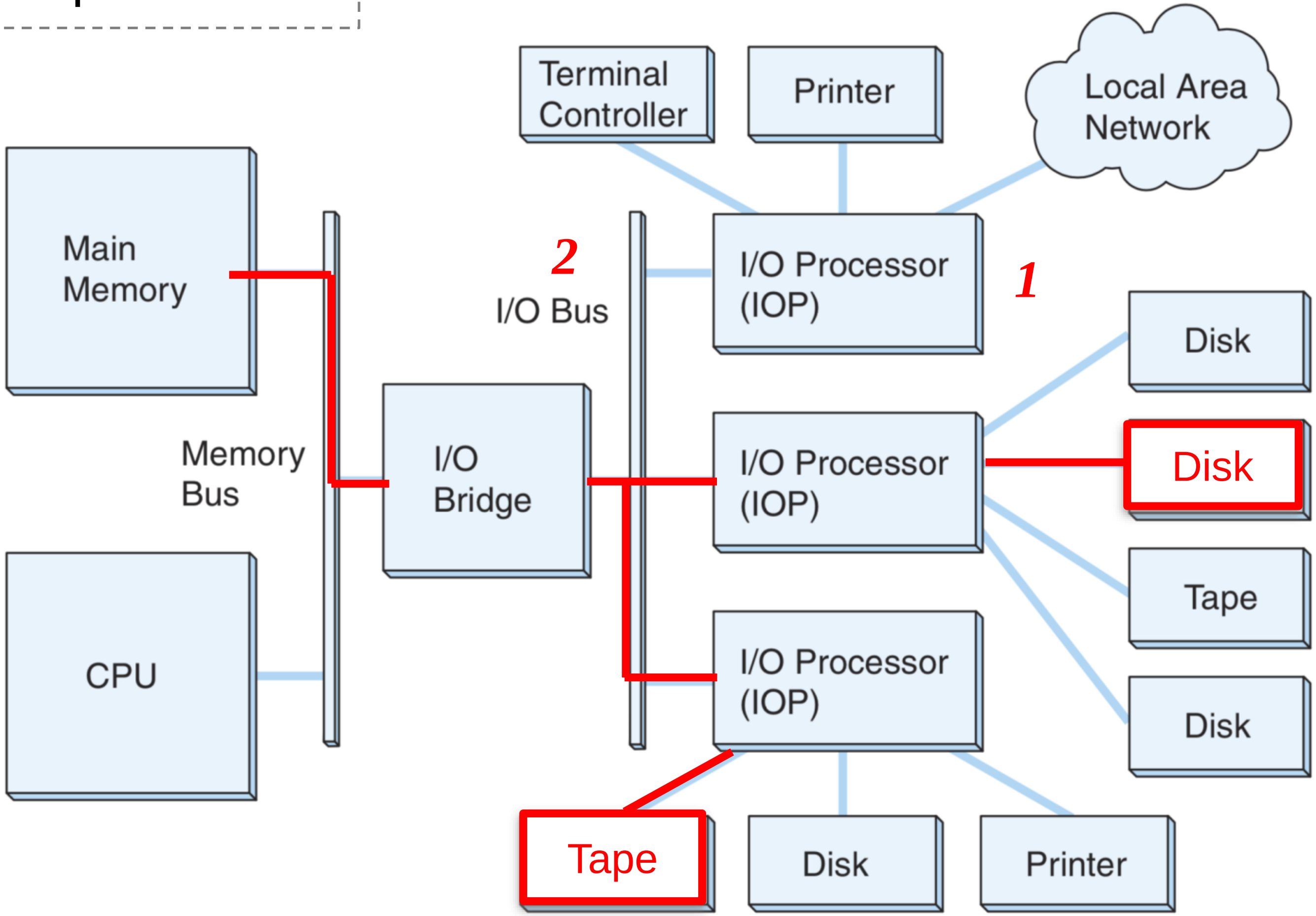


Figure: A channel I/O configuration.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.280.

I/O Bus Operation

- The memory bus and the I/O bus can be **separate entities**.
- One good reason for having memory on its own bus:
  → memory transfers can be *synchronous*, using some multiple of the CPU's clock cycles to retrieve data from main memory.

Never an issue of the memory being offline that afflict peripheral equipment, such as a printer running out of paper.

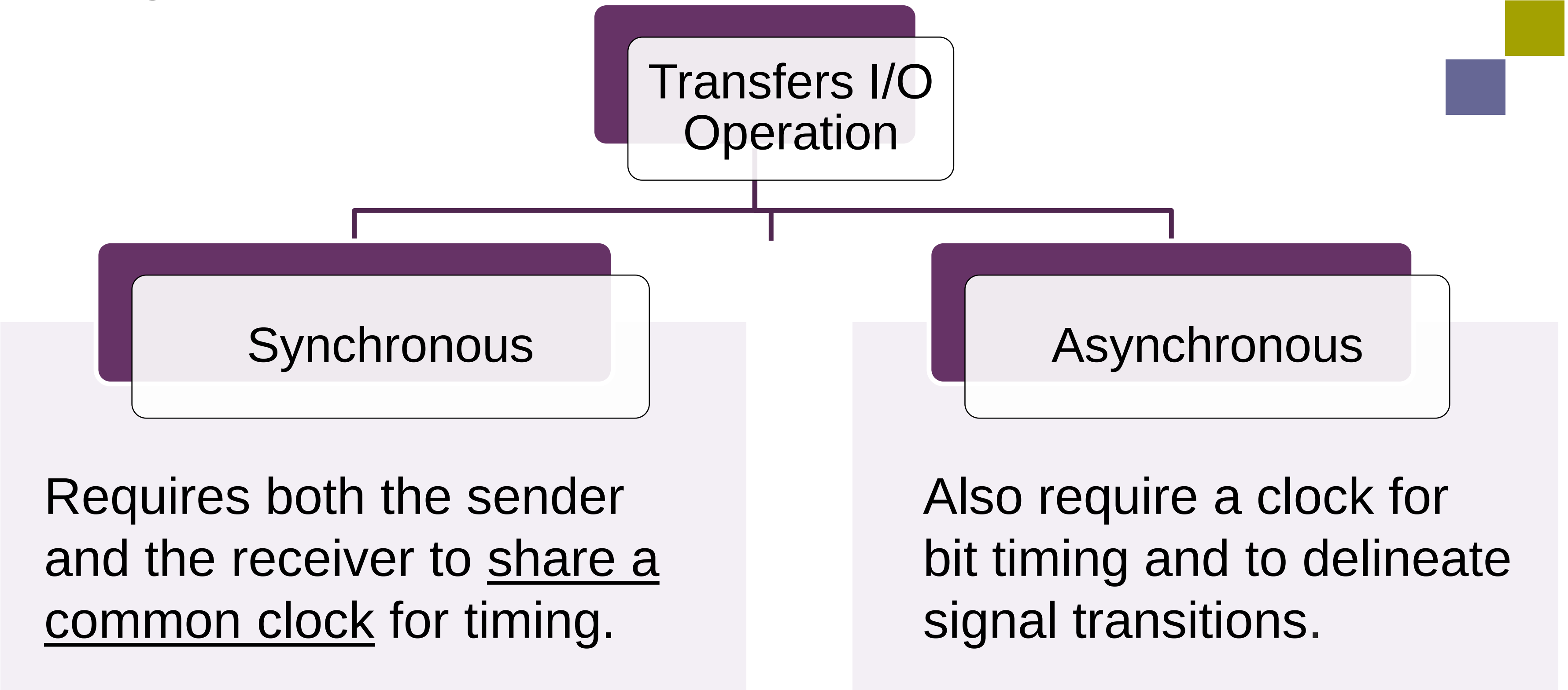
Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.280.

I/O buses, on the other hand, cannot operate synchronously because I/O devices cannot always be ready to process an I/O transfer.

- I/O control circuits placed on the I/O bus and within the I/O devices negotiate with each other to determine the bus to be used.
- Because these handshakes take place every time the bus is accessed, I/O buses are called *asynchronous*.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.281.

- In general:



Transfers I/O Operation

Synchronous

Requires both the sender and the receiver to share a common clock for timing.

(Memory bus)

Asynchronous

Also require a clock for bit timing and to delineate signal transitions.

(I/O buses)

To distinguish *synchronous* from *asynchronous* transfers will become clear after we look at the next example.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.281.

- The connection between the DMA circuit and the device interface circuits is more accurately represented by the following figure, which shows the individual component buses.

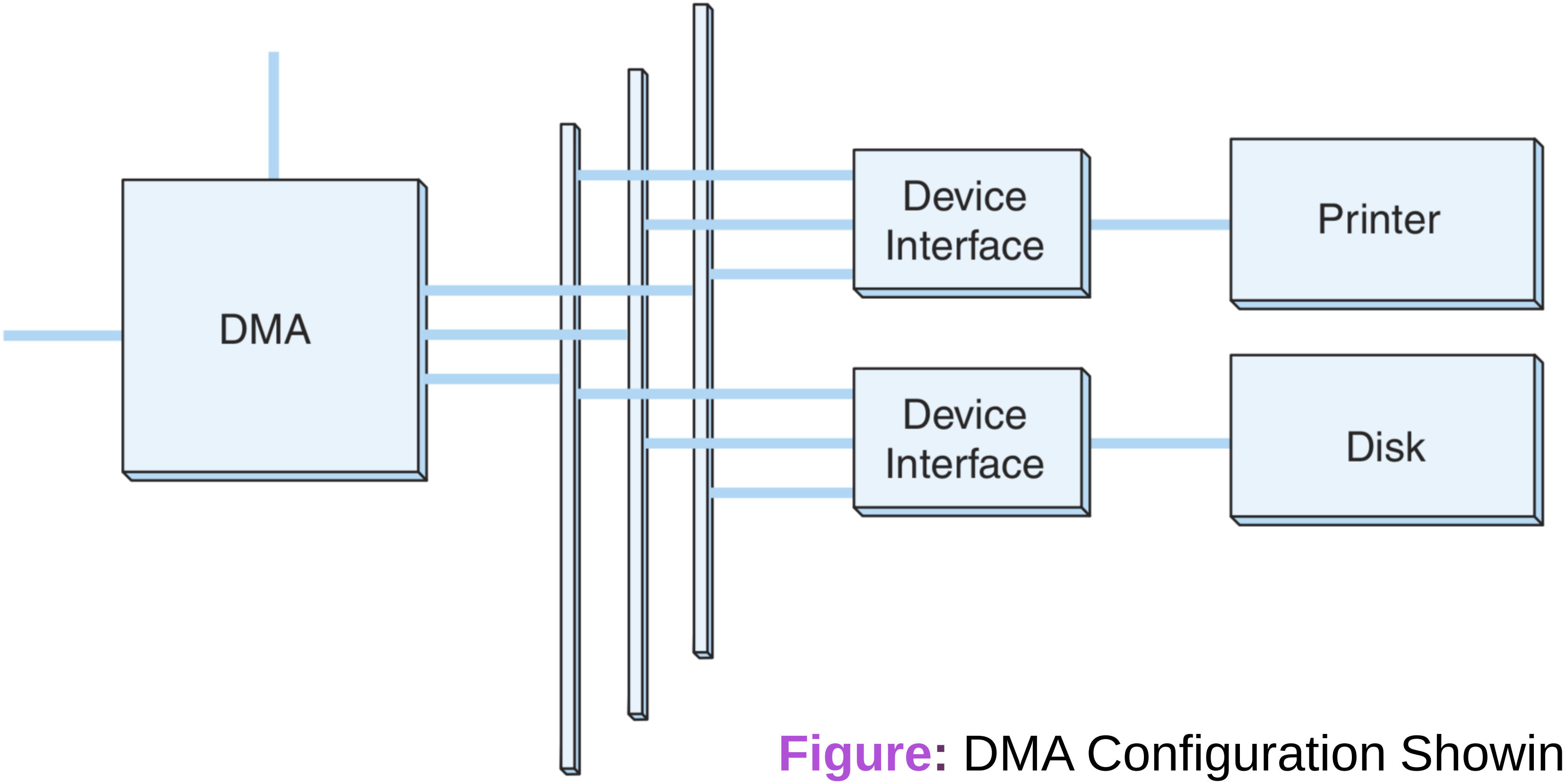
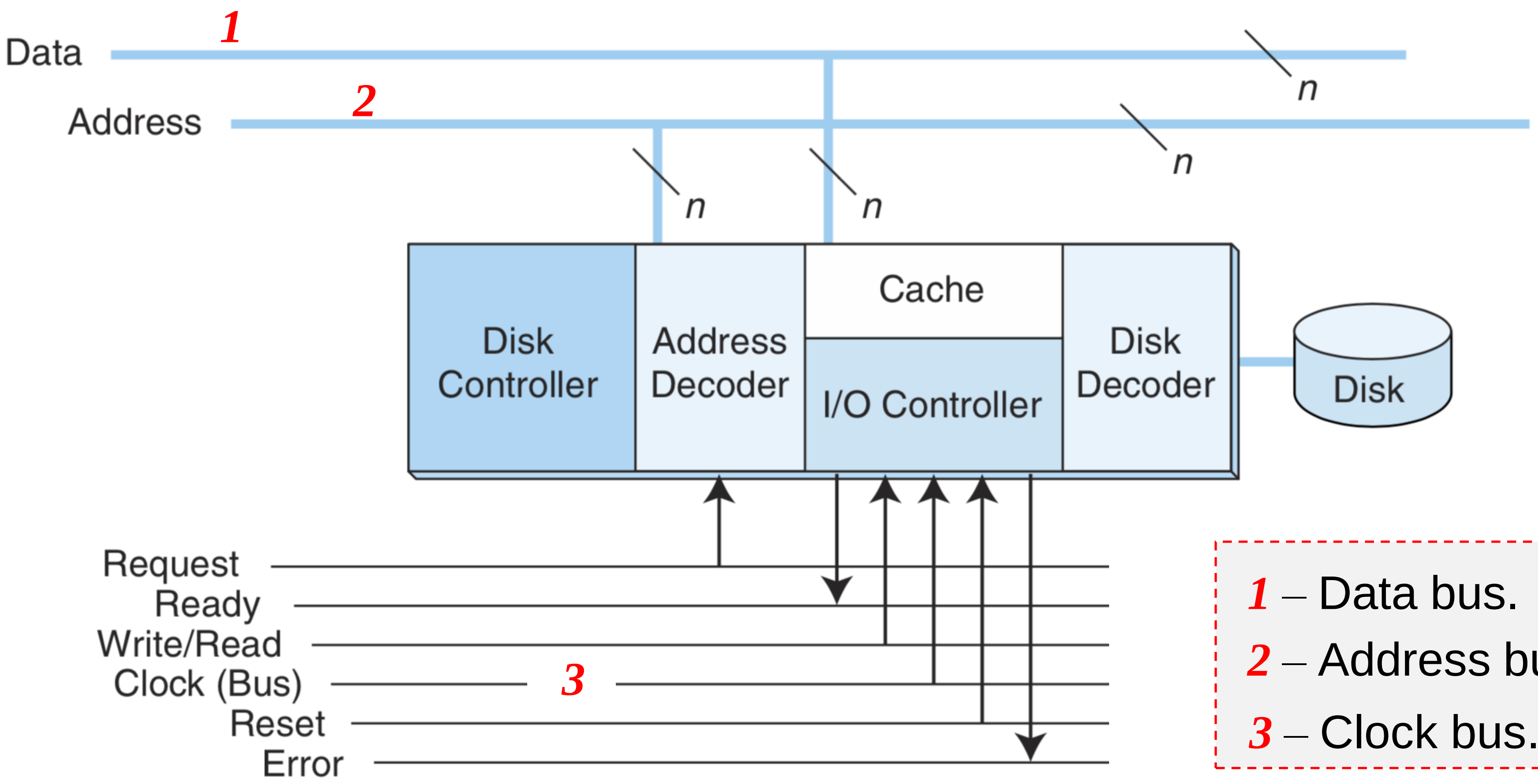


Figure: DMA Configuration Showing Separate Address, Data, and Control Lines.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.281.

Example : To write data to the disk drive.



1 – Data bus.

2 – Address bus.

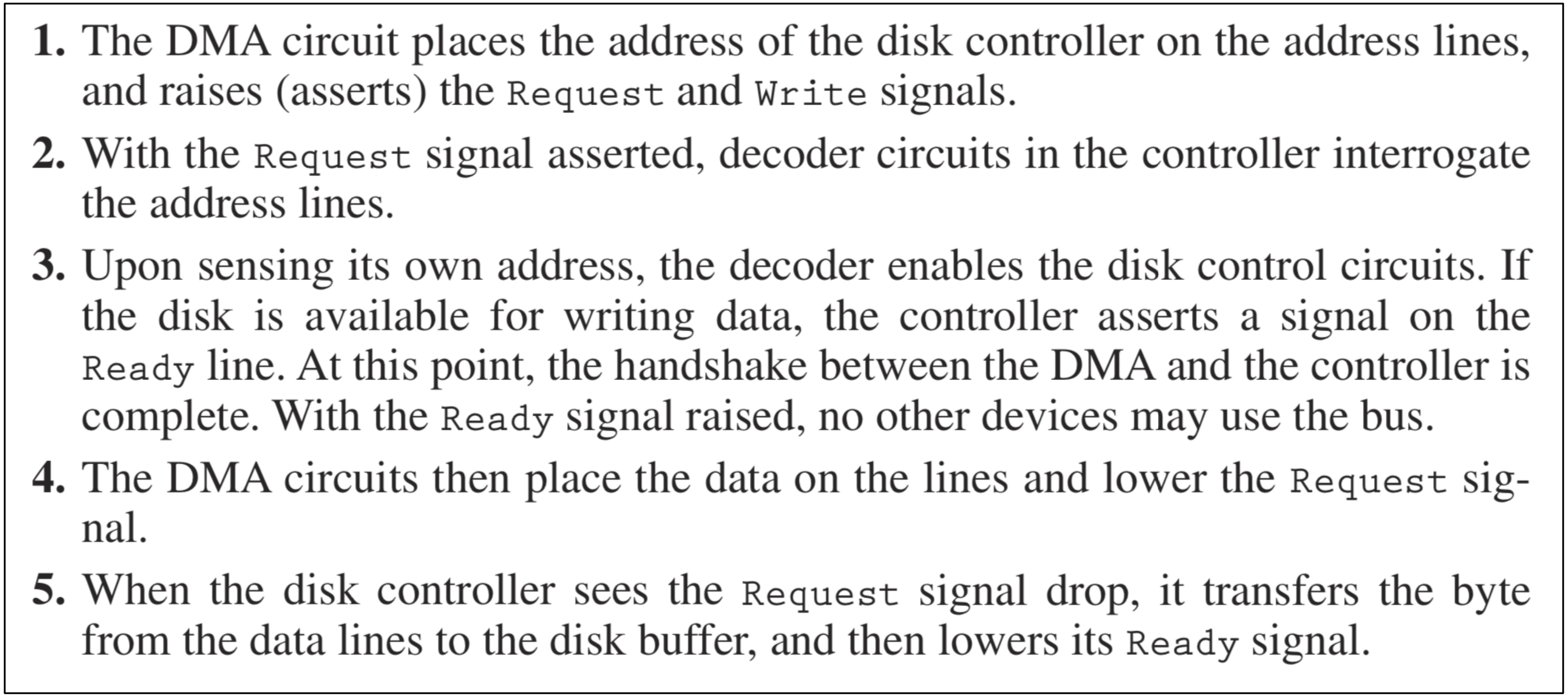
3 – Clock bus.

Figure: A Disk Controller Interface with Connections to the I/O Bus.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.282.

- The number of data lines determines the *width* of the bus.
- A data bus having eight data lines carries one byte at a time.

- To write data to the disk drive, the example of bus executes the operations as the following sequence:



1. The DMA circuit places the address of the disk controller on the address lines, and raises (asserts) the `Request` and `Write` signals.

2. With the `Request` signal asserted, decoder circuits in the controller interrogate the address lines.

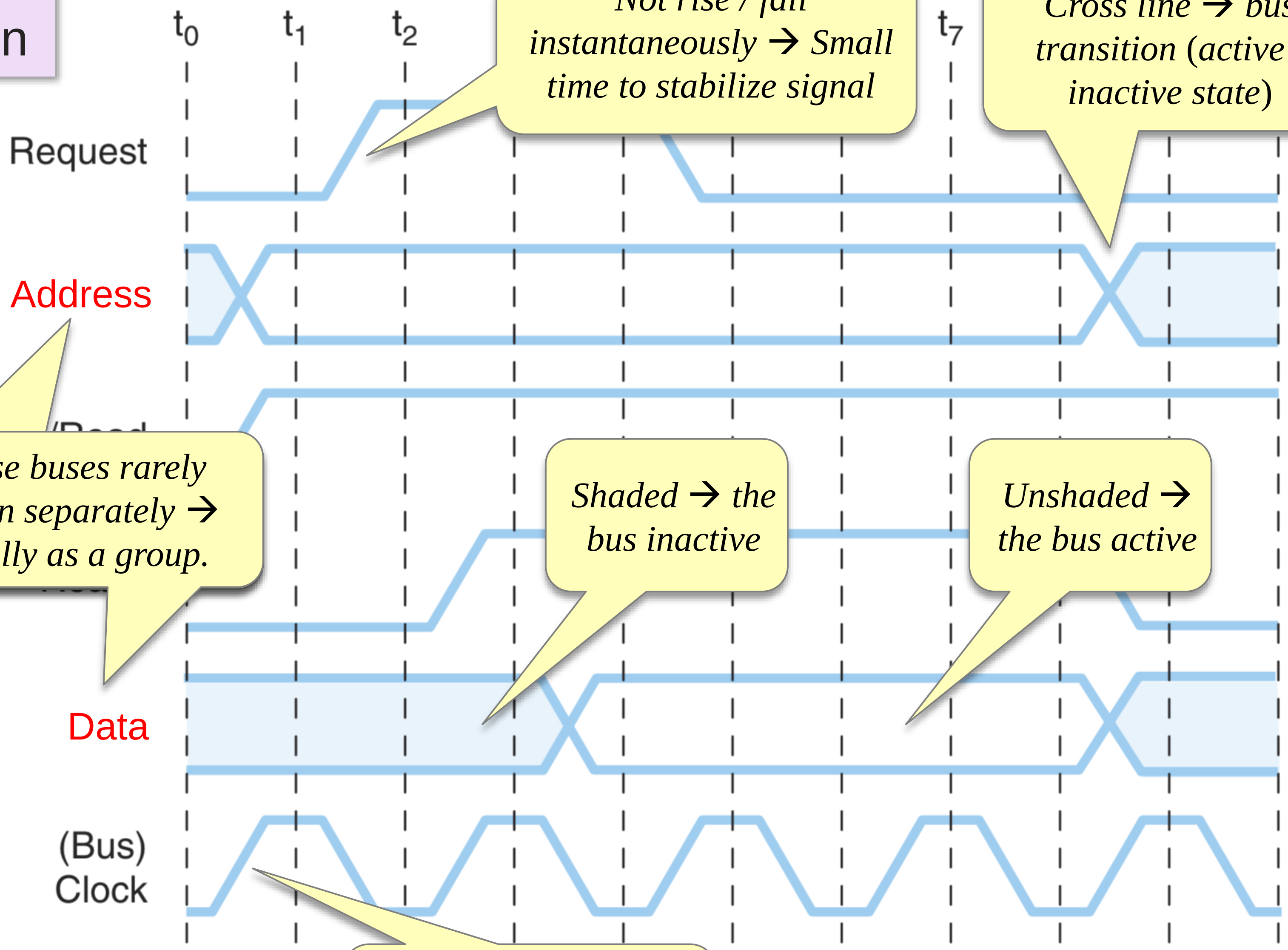
3. Upon sensing its own address, the decoder enables the disk control circuits. If the disk is available for writing data, the controller asserts a signal on the `Ready` line. At this point, the handshake between the DMA and the controller is complete. With the `Ready` signal raised, no other devices may use the bus.

4. The DMA circuits then place the data on the lines and lower the `Request` signal.

5. When the disk controller sees the `Request` signal drop, it transfers the byte from the data lines to the disk buffer, and then lowers its `Ready` signal.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.282.

Disk write operation



Not rise / fall instantaneously → Small time to stabilize signal

Cross line → bus transition (active / inactive state)

These buses rarely shown separately → usually as a group.

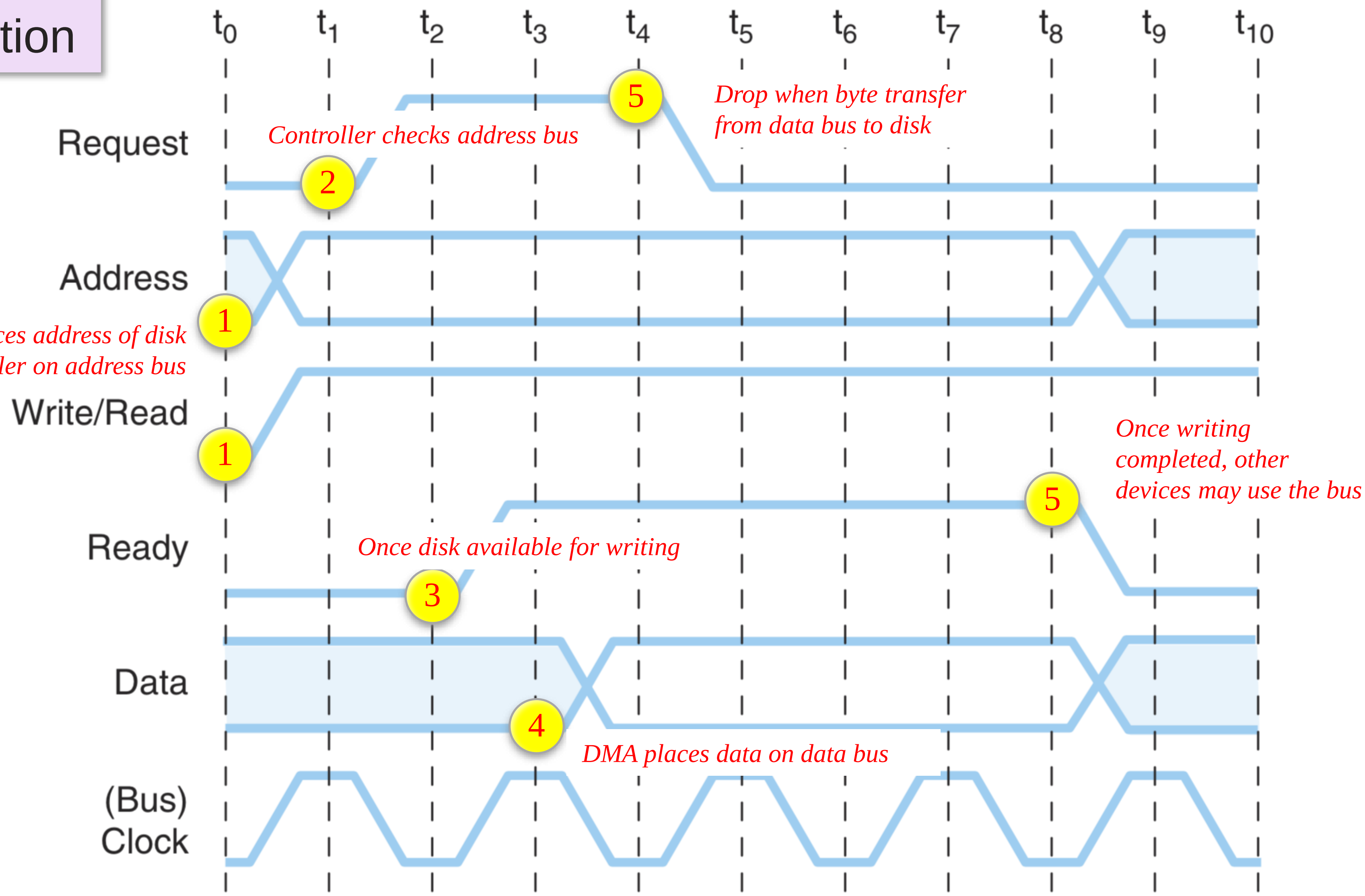
Shaded → the bus inactive

Unshaded → the bus active

Signals on bus only change during clock cycle transition.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.282.

Disk write operation



Controller checks address bus

Drop when byte transfer from data bus to disk

DMA places address of disk controller on address bus

Once writing completed, other devices may use the bus

Once disk available for writing

DMA places data on data bus

Handshake between DMA and controller completed

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.282.

Table: The bus operation in detail based on previous timing diagram.

Time	Bus signal	Meaning
t_0	*Write*	Bus is needed for writing.
t_0	*Address*	Indicates where bytes will be written.
t_1	*Request*	Request write to address on address lines.
t_2	*Ready*	Acknowledges write request, bytes placed on data lines.
$t_{3\text{-}7}$	*Data Lines*	Write data (requires several cycles).
t_8	*Lower Ready*	Release bus.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.282.

Module 7

I/O and Storage Systems

7.1 Introduction

7.2 I/O Architectures

7.3 Data Storage

7.4 Summary

- □ Overview
- □ Magnetic Disk Technology
- □ Optical Disks
- □ Magnetic Tape
- □ RAID
- □ Future Data Storage

Overview

- This section examines a range of external memory devices and systems.
- It begins with the most important device, the magnetic disk.
- An increasingly important component of many computer systems is the solid state disk.
- The external optical memory is also examined, then magnetic tape is described.
- Finally, examines the use of disk arrays to achieve greater performance, looking specifically at the family of systems known as RAID (*Redundant Array of Independent Disks*).

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.195.

(1) Magnetic Disk Technology

- *Magnetic disks as the most important device are the foundation of external memory on virtually all computer systems.
- Magnetic disks offer large amounts of durable storage that can be accessed quickly.
- Disk drives are called *random* (sometimes *direct*) access devices because each unit of storage, the *sector*, has a unique address that can be accessed independently of the sectors around it.
- Magnetic disk organization is shown on the following slide.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.287.
*William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.195.

- *Sectors* are divisions of concentric circles called *tracks*.

Disk tracks are consecutively numbered starting with track 0 at the outermost edge of the disk.

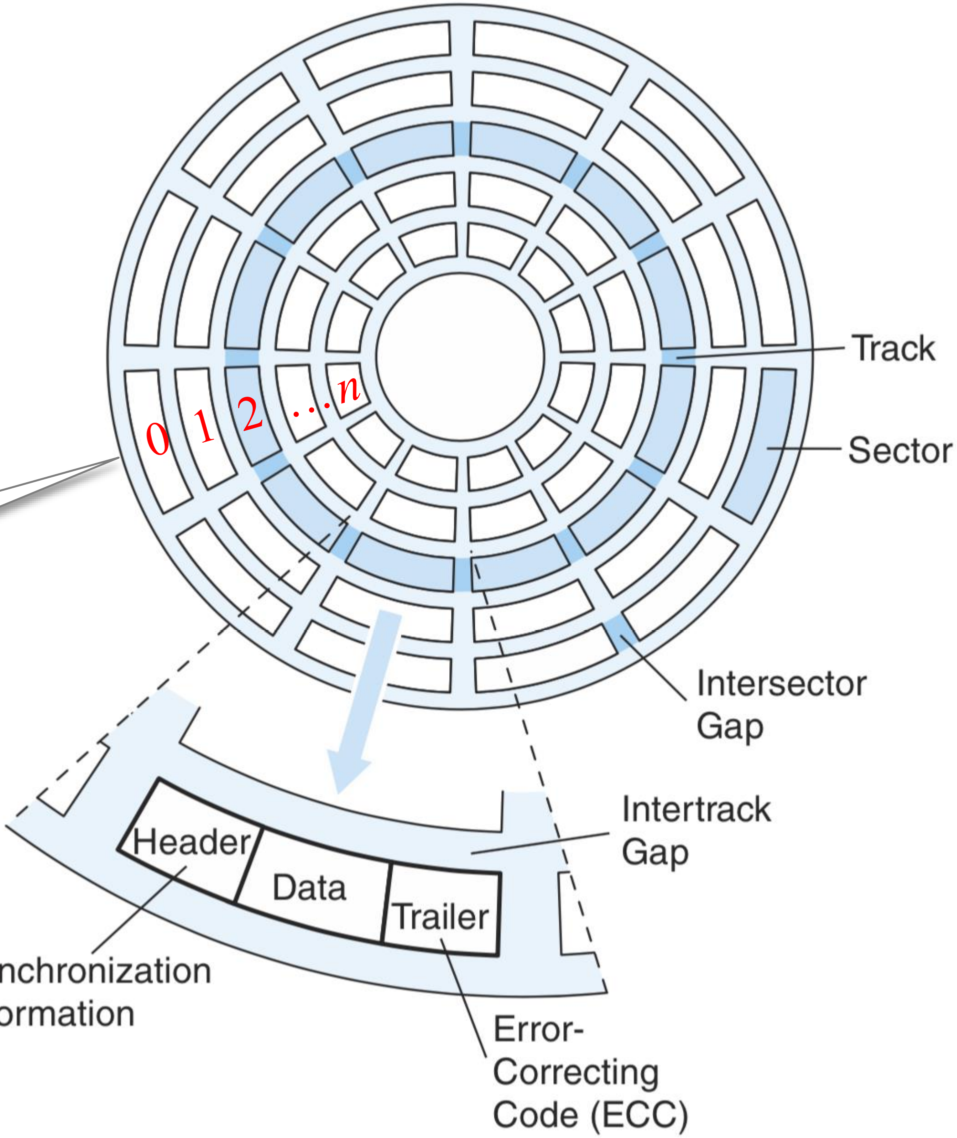


Figure: Disk Sectors Showing Intersector Gaps and Logical Sector Format.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.287.

Rigid Disk Drives

- Rigid (“hard” or fixed) disks contain control circuitry.
- One or more metal or glass disks called *platters* is/are stacked on a *spindle*.
- *Read/write heads* are mounted on a comb that swings radially to read the disk.

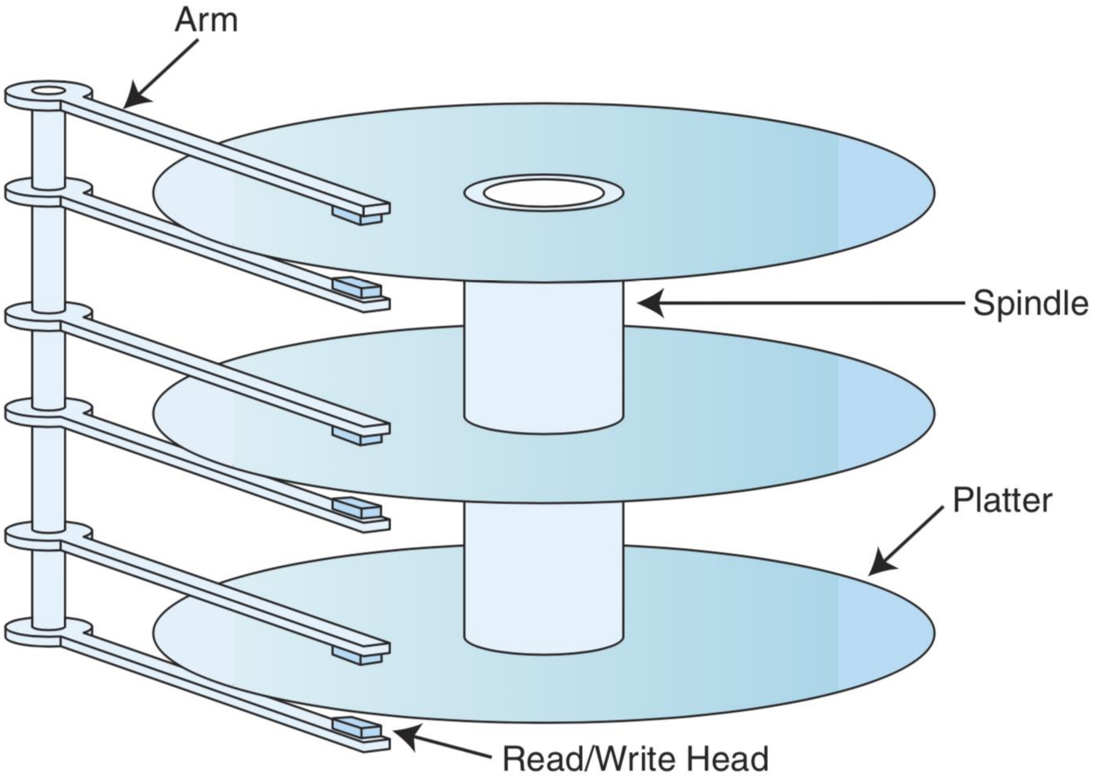
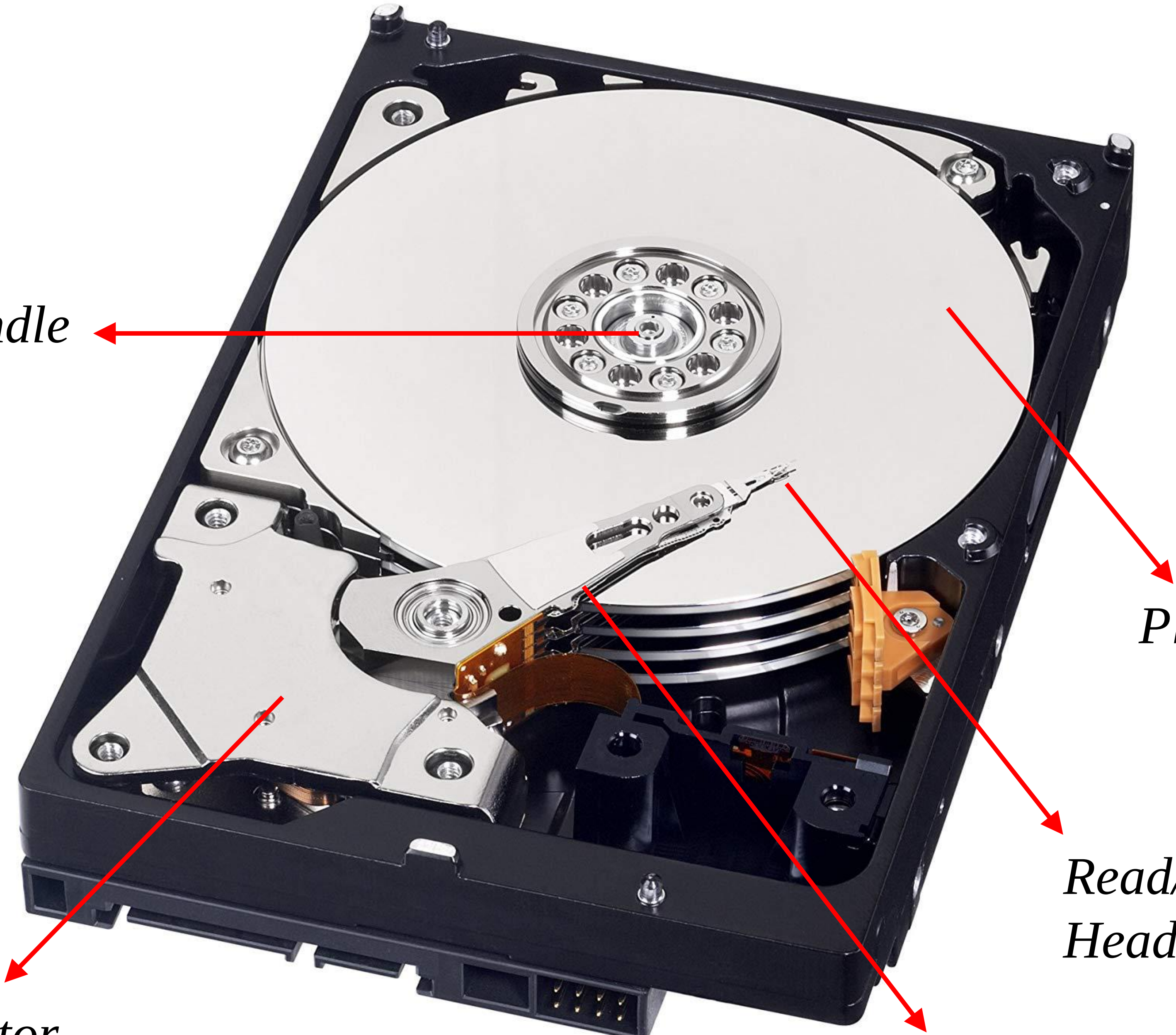


Figure: Rigid Disk Actuator (with Read/Write Heads) and Disk Platters.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.288.



Spindle

Platter

Read/Write Head

Actuator

Arm

https://images-na.ssl-images-amazon.com/images/I/81SYu2aMXRL._SL1500_.jpg (5 Jan 2019)

Flexible (Floppy) Disks

- Flexible disks are organized in much the same way as hard disks, with addressable *tracks* and *sectors*.
- Physical and logical limitations restrict floppies to much lower densities than hard disks.
- A major logical limitation of the DOS/Windows floppy diskette is the organization of its *File Allocation Table* (FAT).

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.292.
https://www.silicon.co.uk/wp-content/uploads/2014/05/shutterstock_166068341.jpg (5 Jan 2019)

(2) Optical Disks

- Optical storage systems offer (practically) unlimited data storage at a cost that is competitive with tape.
- Optical disks come in a number of formats, the most popular format being the ubiquitous CD-ROM (*Compact Disk-Read Only Memory*).

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.293.
http://www.ommdvd.com/images/services/Duplication-Optical-Media-sm.jpg (5 Jan 2019)

- Variation of optical disks: CD-R (*CD-Recordable*), CD-RW (*CD-Rewritable*), and WORM (*Write Once Read Many*) disks are optical storage devices often used for :
  - long-term data archiving and
  - high-volume data output.

- For long-term archival storage of data, some computer systems send output directly to optical storage rather than paper or microfiche → COLD (*Computer Output Laser Disk*)

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.294.

CD-ROM

(*Compact Disk-Read Only Memory*)

The total capacity data is 650MB

- CD-ROMs are polycarbonate (plastic) disks which a reflective aluminium film is applied.
- Compact disks are written from the <u>center to the outside edge</u> using a single spiraling track of bumps in the polycarbonate substrate.
- CD-ROMs were designed for storing <u>music</u> and other sequential <u>audio signals</u>.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.294, 297.

DVD

(*Digital Versatile Disk*)

**The total capacity data are 17GB (double sided, dual-layer).*

- DVDs (formerly called *digital video disks*), can be thought of as quad-density CDs.
- DVDs rotate at about three times faster than CDs.
- Unlike CDs, DVDs can be single-sided or double-sided, called *single layer* or *double layer*.
- One can expect that DVDs will eventually replace CDs for long-term data storage and distribution.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.297-298.
*William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.221.

High-Definition (HD) Optical Disks

- Designed to store high-definition videos and to provide significantly greater storage capacity compared to DVDs

- Two competing disk formats and technologies :

Blue-Ray DVD	HD-DVD
Can store 25GB on a single layer on a single side.	Can store 15GB on a single layer on a single side.

The big difference → HD-DVD is *backward compatible* with red laser DVDs, while Blu-Ray is not.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.221.

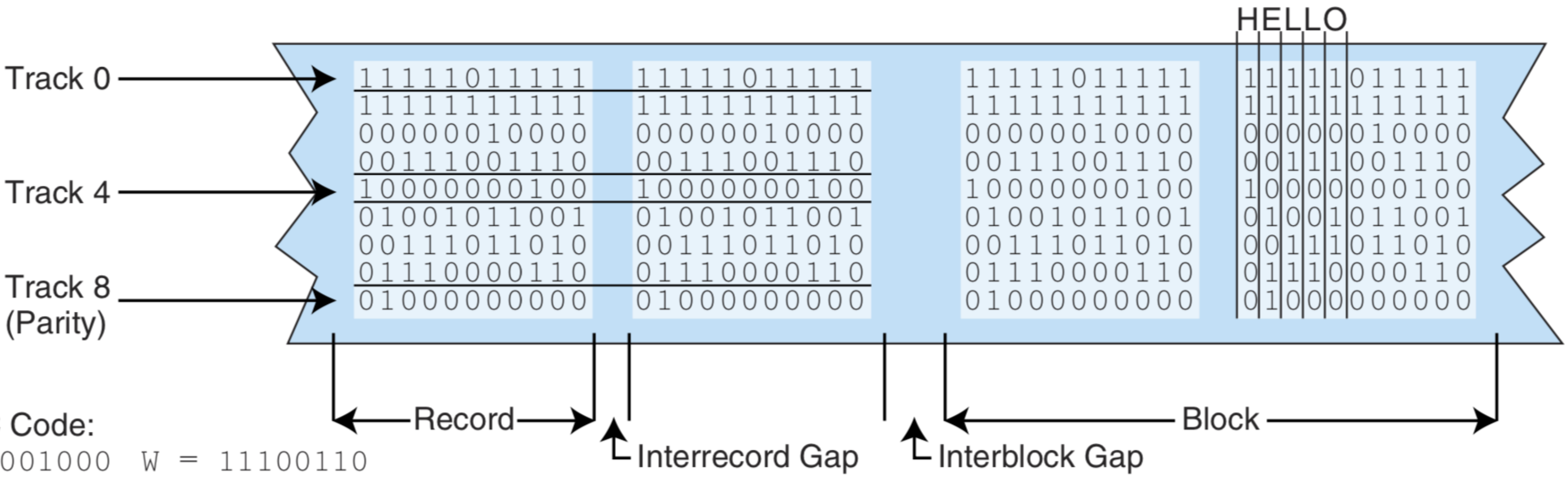
(3) Magnetic Tape

- Magnetic tape is the oldest and most cost-effective of all mass-storage devices.
- First-generation magnetic tapes were made of the same material used by analog tape recorders.

Early tapes had capacities under 11MB, and required nearly a half hour to read or write the entire reel.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.299.

- Data was written across the tape <u>one byte at a time</u>, creating <u>one track for each bit</u>.
- An additional track was added for *parity*, making the tape nine vertical tracks wide:



```
11111011111   11111011111   11111011111   11111011111
11111111111   11111111111   11111111111   11111111111
00000010000   00000010000   00000010000   00000010000
00111001110   00111001110   00111001110   00111001110
10000000100   10000000100   10000000100   10000000100
01001011001   01001011001   01001011001   01001011001
00111011010   00111011010   00111011010   00111011010
01110000110   01110000110   01110000110   01110000110
01000000000   01000000000   01000000000   01000000000
```

EBCDIC Code:

```
H = 11001000  W = 11100110
E = 11000101  O = 11010110
L = 11010011  R = 11011001
L = 11010011  L = 11010011
O = 11010110  D = 11000100
    [space] = 01000000
```

Figure: A nine-track tape format.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.299.

- Tapes support various track densities and employ two dominant recording methods:

Serpentine

- ❑ Recording with 50 or more tracks per tape.
- ❑ *e.g.: Digital linear tape* (*DLT*) and *Quarter Inch Cartridge* (*QIC*) systems.

Helical Scan

- ❑ Pass tape over a tilted rotating drum (*capstan*), which has two read heads and two write heads.
- ❑ *e.g.: Digital Audio Tape* (*DAT*)

Both are distinguished by how the read-write head passes over the recording medium.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.300.

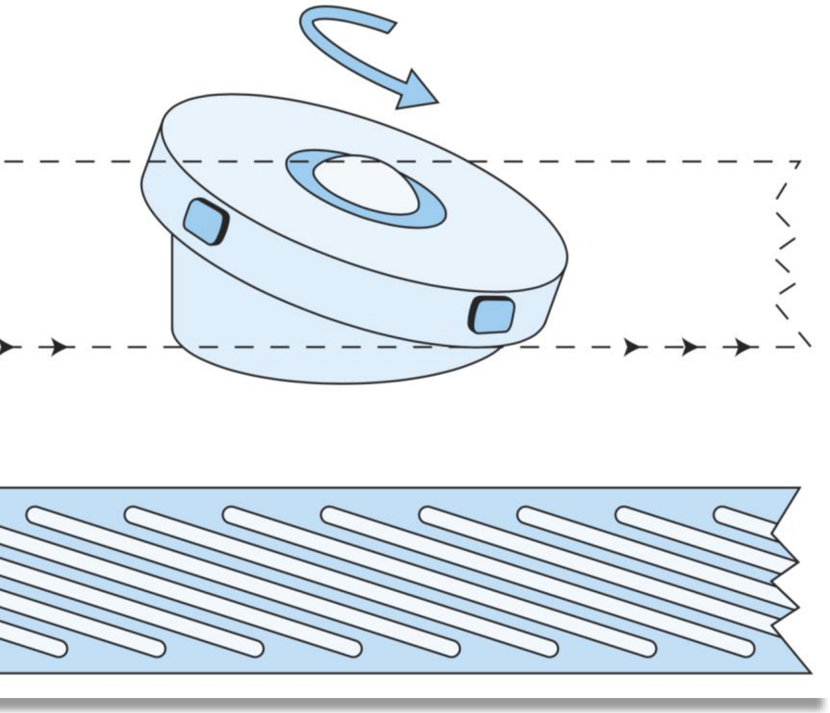


Figure: A *Helical Scan* Recording
(a) The Read-Write Heads on Capstan
(b) Pattern of Data Written on the Tape.

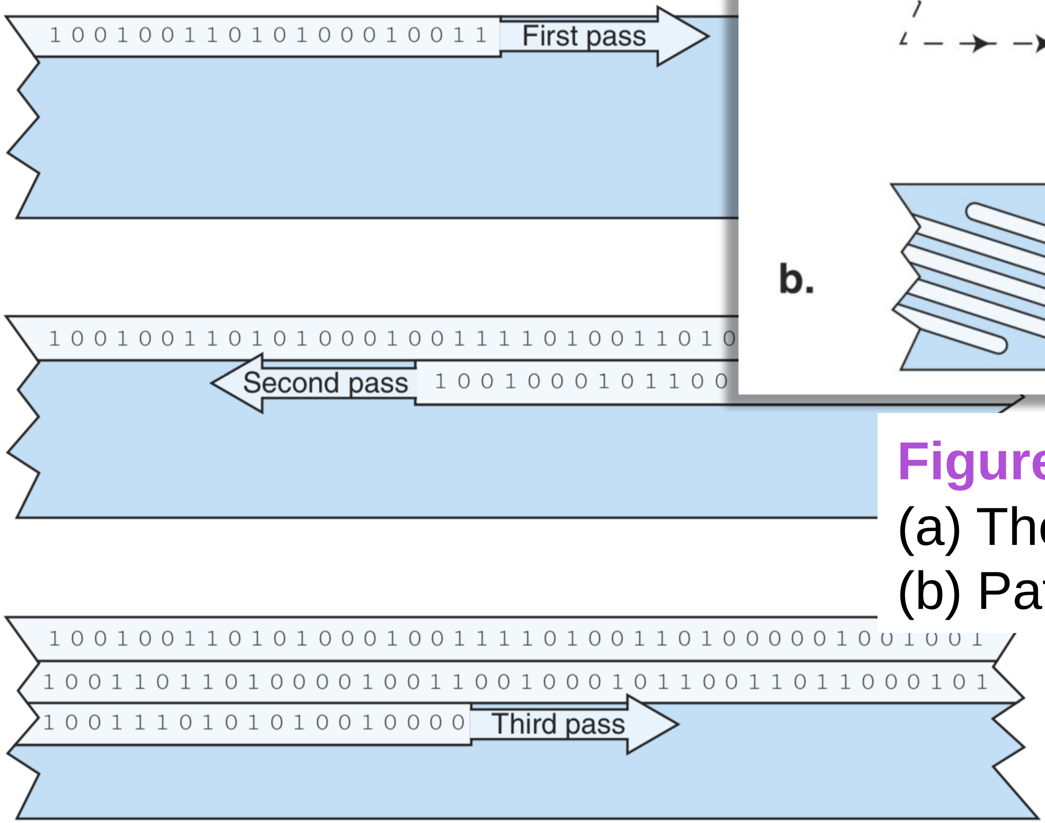


Figure: Three Recording Passes
on a *Serpentine* Tape.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.300, 301.

(4) RAID

(Redundant Array of Independent Disks)

- RAID → invented to address problems of disk reliability, cost, and performance.
- The disk storage can gains in performance by developing an array of disks that operates independently using multiple parallel components.

How it work?

- ❑ data is stored across many disks.
- ❑ extra disks added to the array to provide error correction (redundancy).

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.301.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.204.

- There is a wide variety of ways in which data can be organized and which redundancy can be added to improve reliability in multiple disks.
- There are 7 types (called *levels 0 – 6*) of RAID, each having different performance and reliability of 3 common characteristics:

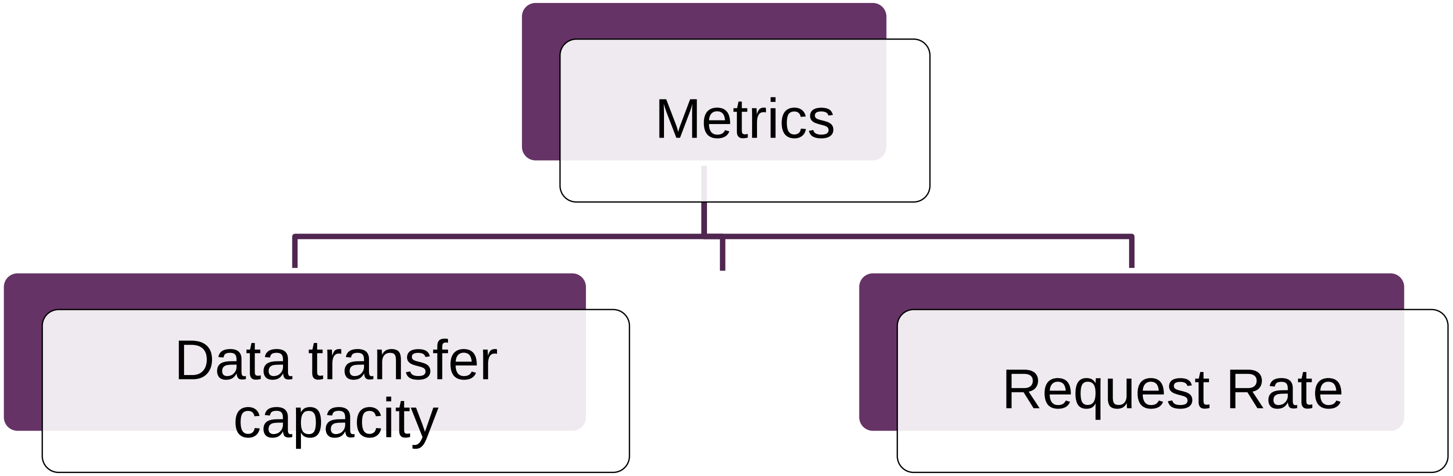
RAID → a set of physical disk drives viewed by the OS as a single logical drive.

Data → distributed across the physical drives of an array.

Redundant disk capacity → used to store parity information to guarantee data recoverability when disk failure.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.204.

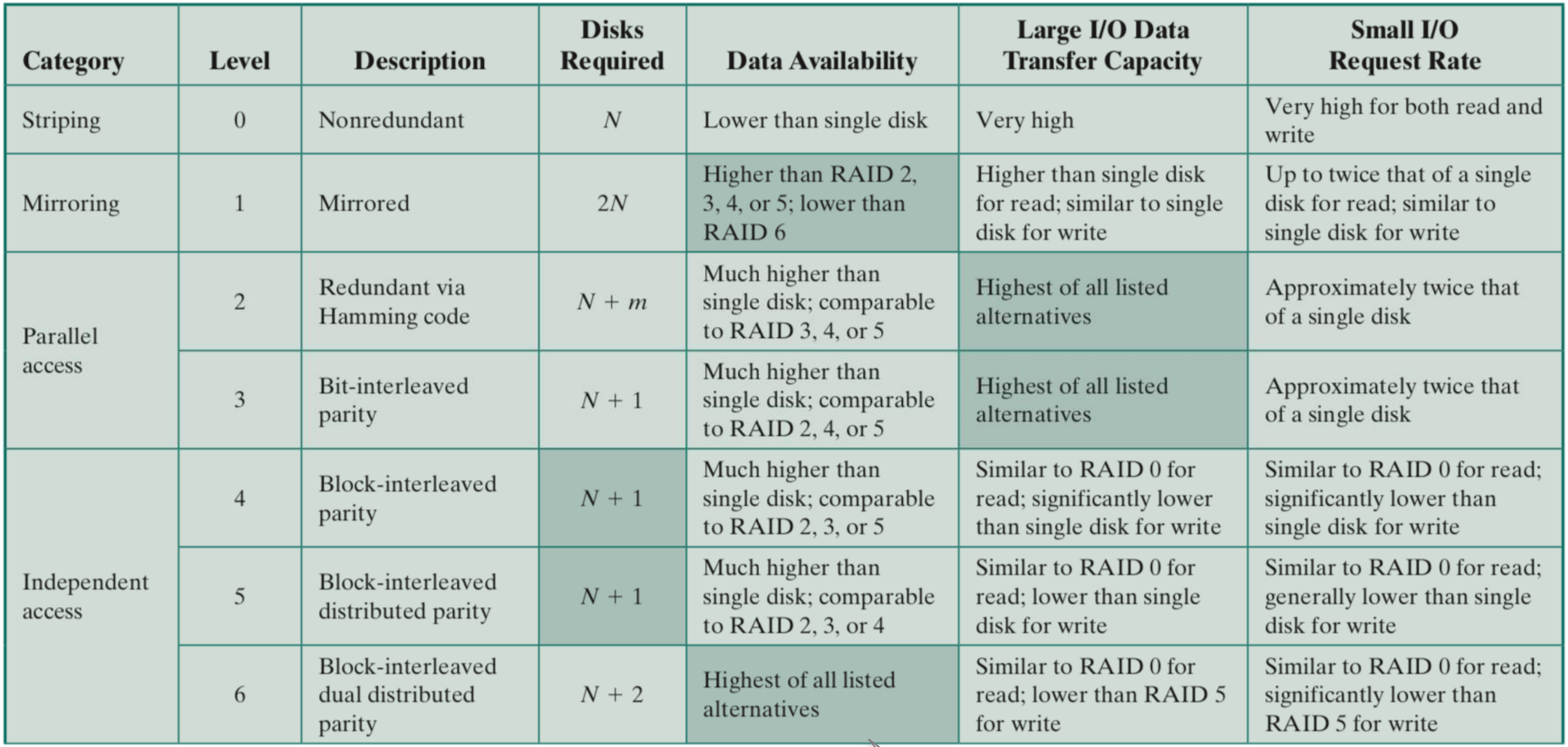
- The RAID levels inherently perform differently relative to TWO metrics of I/O performance:



- The next table provides a rough guide to the seven levels.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.26-27.

Table: RAID levels.



Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

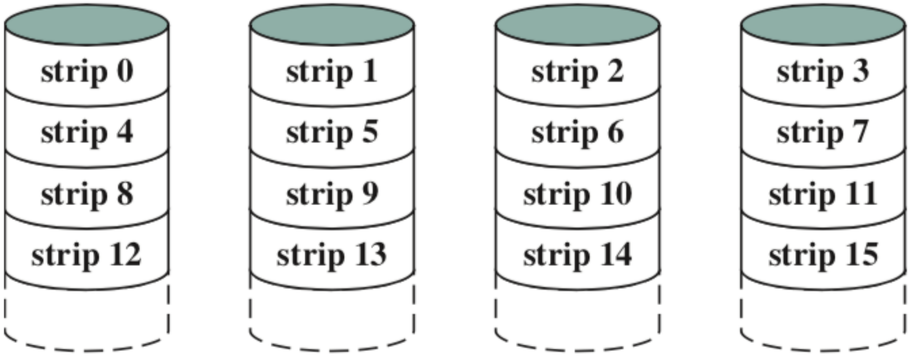
Note: N = number of data disks; m proportional to log N

Darker shading indicate the strong point for each level

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.206.

RAID Level 0

(Striping)



(a) RAID 0 (Nonredundant)

- Not a true of the RAID family because it has no redundancy to improve performance.
- Some applications primary concerns the performance and capacity, but the low cost is more important than improved reliability (*e.g*. supercomputer).
- In RAID 0, the user and system data are distributed across all the disks in the array.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205.

- All users and system data are viewed as being stored on a logical disk.
- The logical disk is divided into *strips* (*e.g*. physical blocks, sectors, or some other unit).
- The strips are mapped round-robin to consecutive physical disks in the RAID array (Refer next figure for data mapping).

👍 Simple data distribution across the disk array.

👎 Low reliability when failure of any disk.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205.

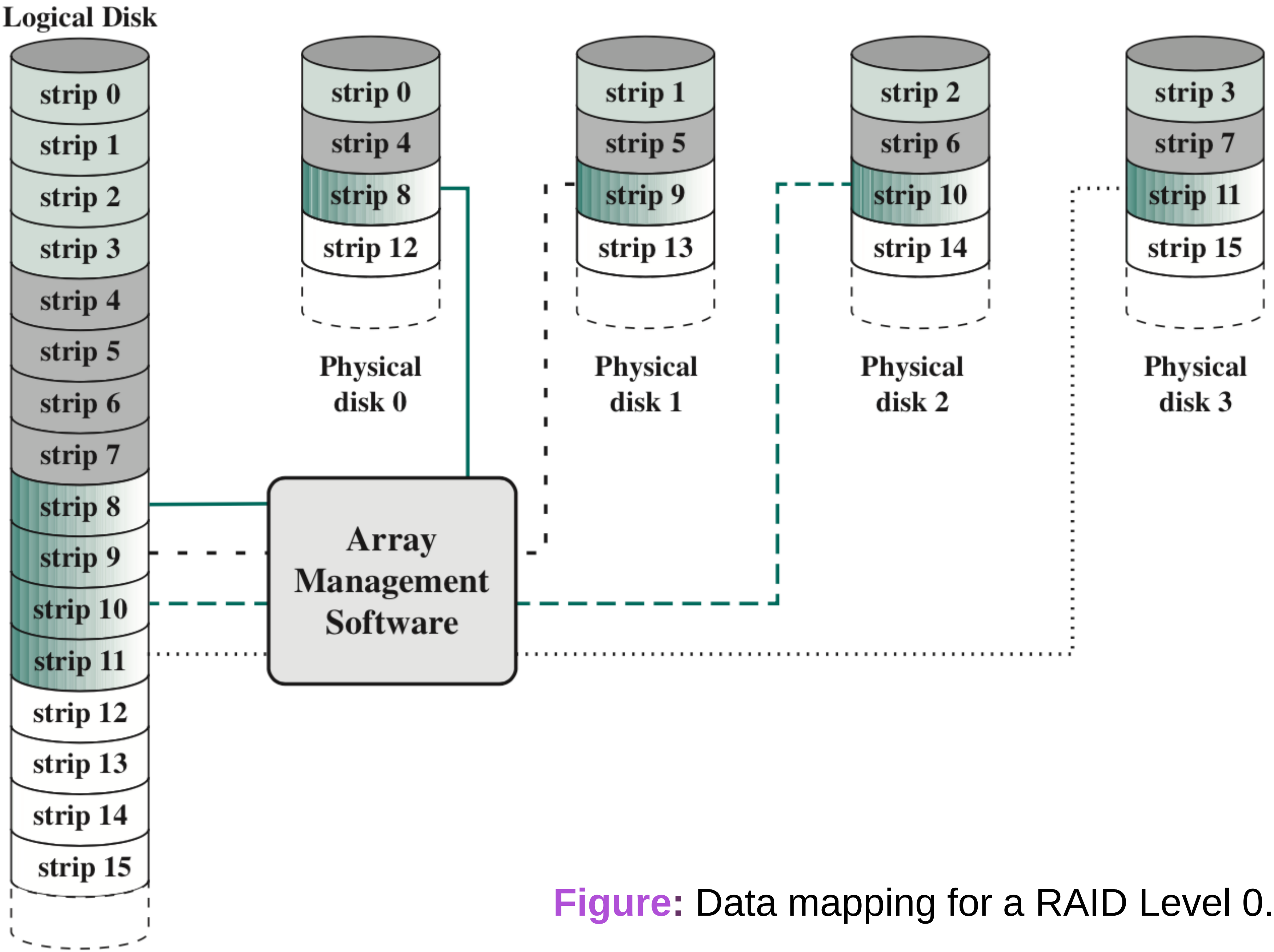


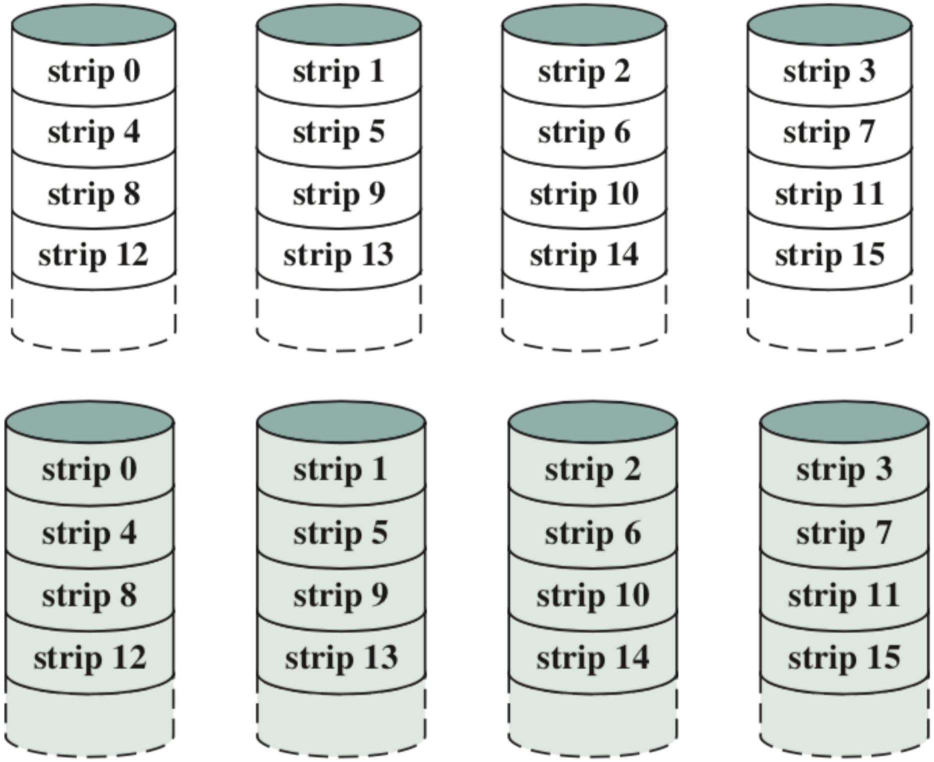
Figure: Data mapping for a RAID Level 0.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.209.

RAID Level 1

(Mirrored)

- Some form of parity calculation is used to introduce redundancy (duplication of all data).
- Each logical strip is mapped to two separate physical disks.



(b) RAID 1 (Mirrored)

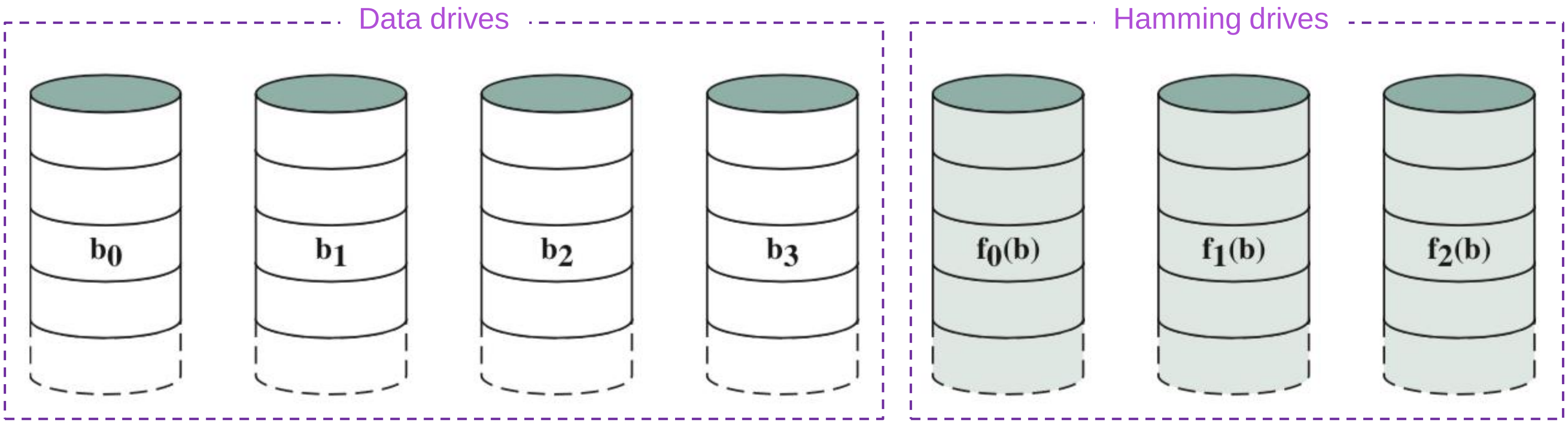
👍 100% redundancy and good performance.

👎 Cost is expensive.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.207, p.209.

RAID Level 2

(*Parallel access*)



(c) RAID 2 (Redundancy through Hamming code)

- This RAID 2 make uses of a parallel access technique.
- All members disks participate in the execution of every I/O request.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.208, p.210.

- Typically, Hamming code is used for error correction.

  - An error-correcting code is calculated across corresponding bits on each data disk (*Data drives*);
  - The bits of the code are stored in the corresponding bit positions of multiple parity disks (*Hamming drives*).

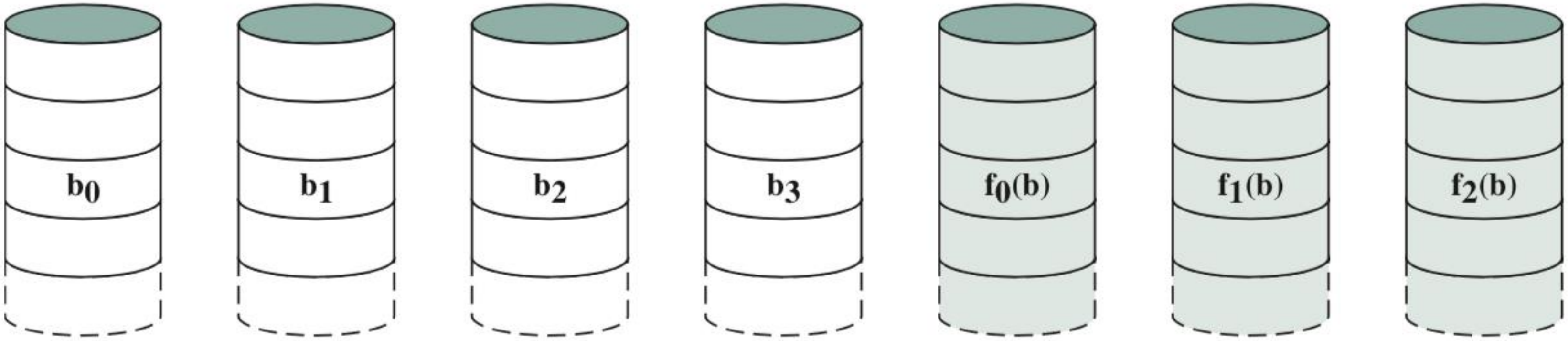
👍 Provides error correction.

👎 Performance is poor and the cost is relatively high.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.208, p.210.

RAID Level 3

(Parallel access)



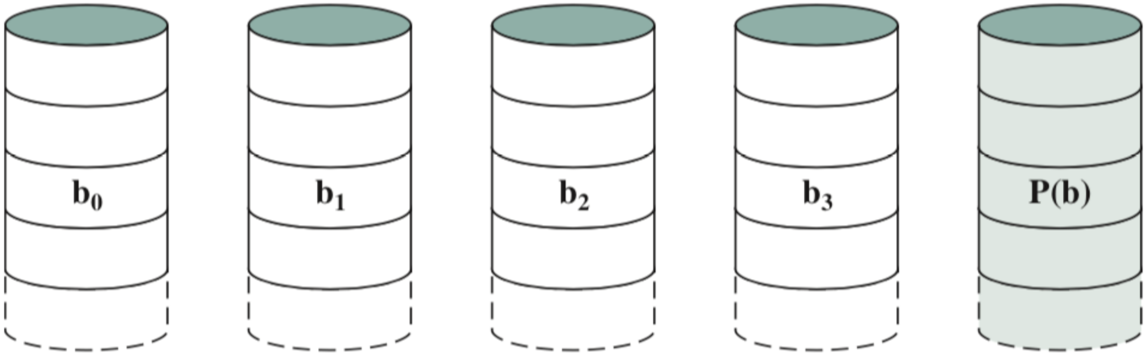
(c) RAID 2 (Redundancy through Hamming code)

- Organized similarly to RAID 2.
- RAID 3 requires only a single redundant parity disk.
- Parity is the XOR of the data bits.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.210.

RAID Level 3

(Parallel access)



(d) RAID 3 (Bit-interleaved parity)

- Organized similarly to RAID 2.
- RAID 3 requires only a single redundant parity disk.
- Parity is the XOR of the data bits.

Instead of an error-correcting code, a simple *parity bit* is computed for the set of individual bits in the same position on all data disks.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.210.

👍

- **Redundancy**: In the event of a data drive failure, the parity drive is accessed and data is reconstructed from the remaining devices.
- **Performance**: Can achieve very high data transfer rates due to the data striped on very small strips.

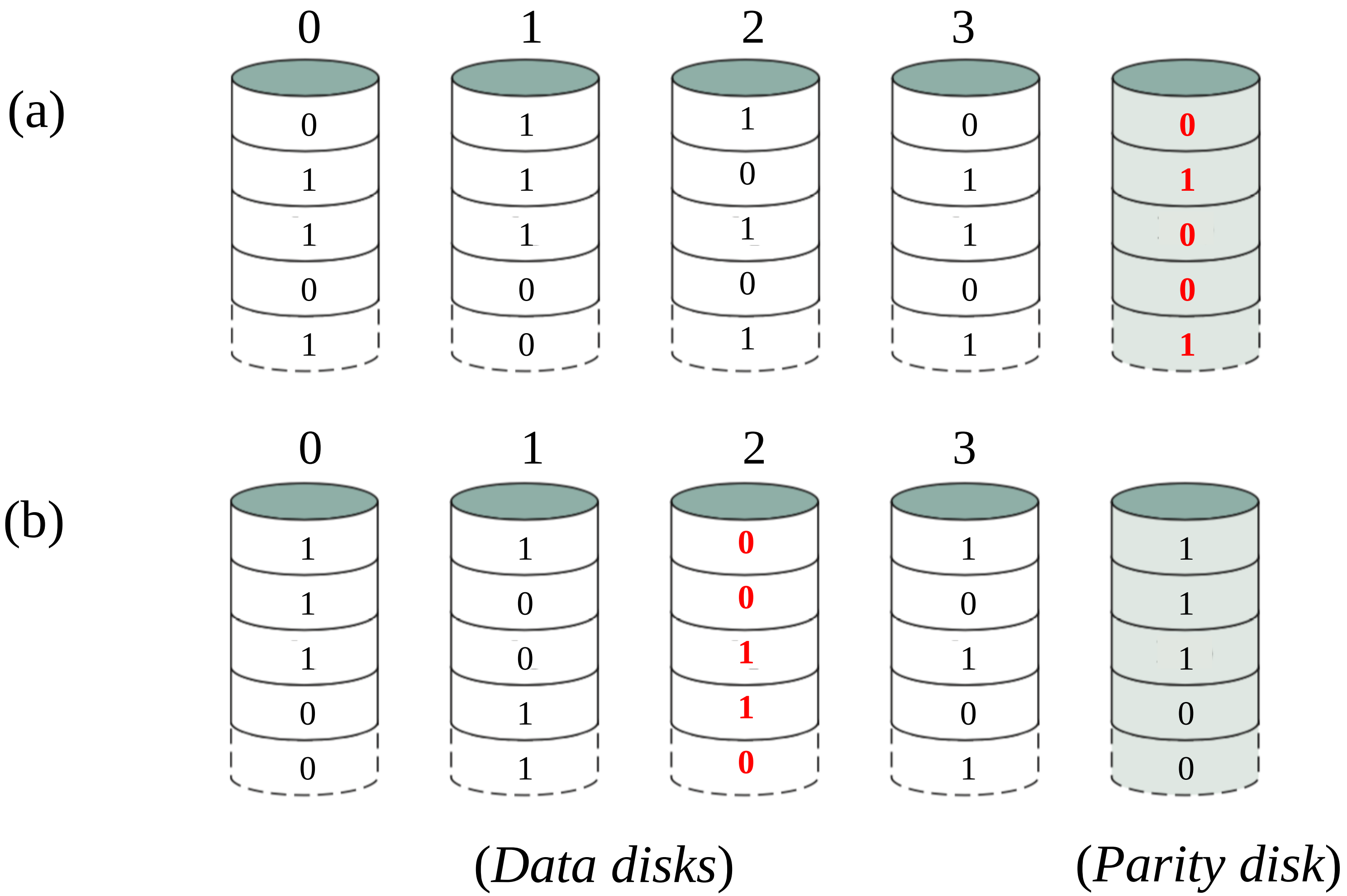
👎

- Consistency of the parity must be maintained for later regeneration.
- Not suitable for commercial applications, but good for personal systems.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.210.

Example :

To write (a) the parity bits and (b) recover the data drive 2.



(a)

0	1	2	3	
0	1	1	0	**0**
1	1	0	1	**1**
1	1	1	1	**0**
0	0	0	0	**0**
1	0	1	1	**1**

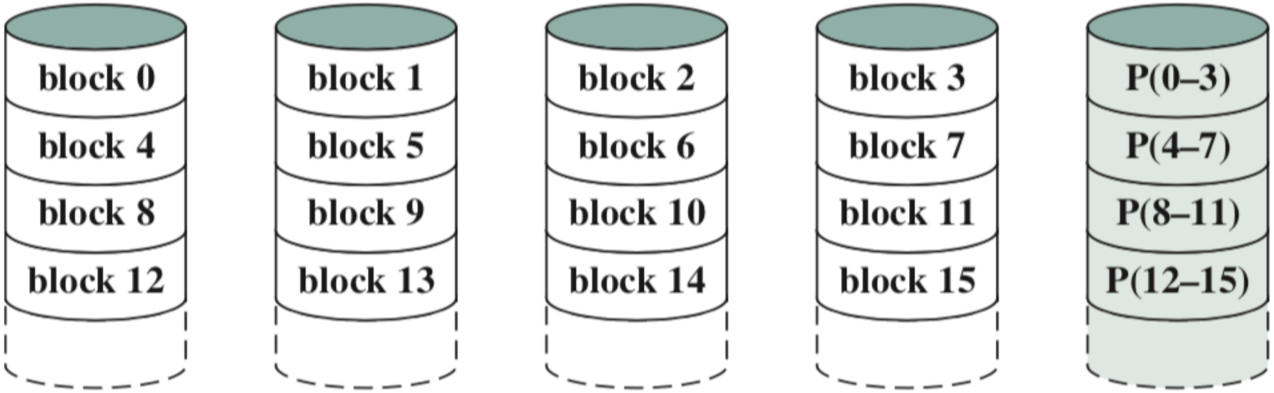
(b)

0	1	2	3	
1	1	**0**	1	1
1	0	**0**	0	1
1	0	**1**	1	1
0	1	**1**	0	0
0	1	**0**	1	0

(Data disks) *(Parity disk)*

RAID Level 4

(*Independent access*)



(e) RAID 4 (Block-level parity)

- This RAID 4 make uses of an independent access technique.
- Each member disk operates independently (while RAID 2 & 3, all disks must sync together) so that separate I/O requests can be satisfied in parallel.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.211.

- Strips are relatively large in RAID 4.

  - A bit-by-bit parity strip is calculated across corresponding strips on each data disk (*Data drives*);
  - The parity bits are stored in the corresponding strip on the parity disk (*Parity drives*).

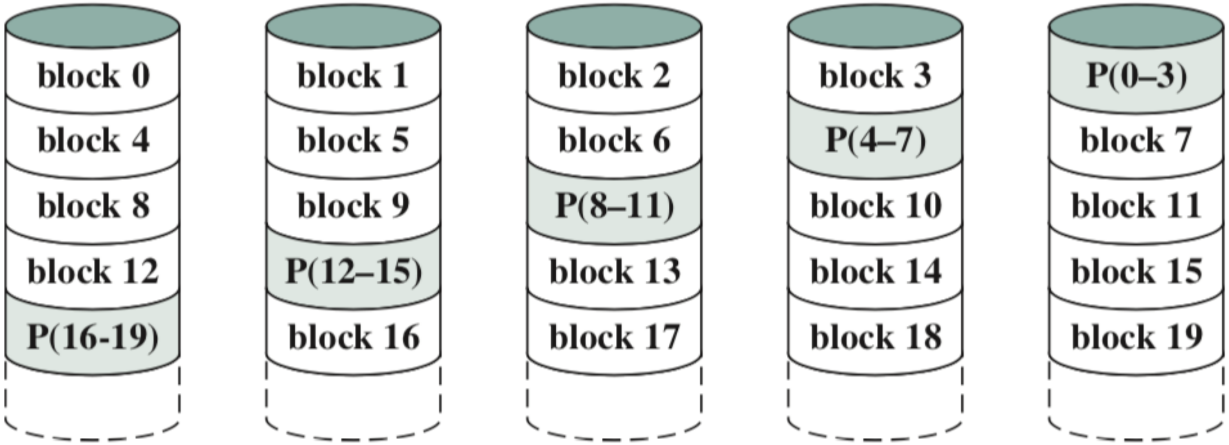
👍 Suitable for applications that require high I/O request rates.

👎 Less suited for applications that require high data transfer rates because each write operation involve the parity disk (bottleneck).

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.211.

RAID Level 5

(*Independent access*)



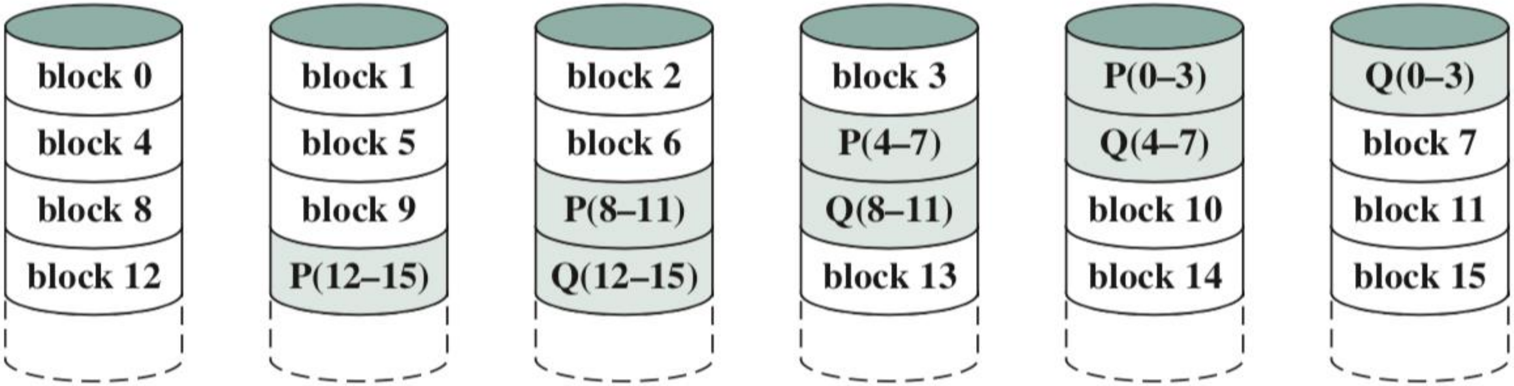
(f) RAID 5 (Block-level distributed parity)

- Similar to RAID 4 but this distributes the parity strips across all disks with a typical allocation (round-robin scheme).
- 1 disk fault tolerance
- The distribution of parity strips avoids the potential bottleneck found in RAID 4.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.212.

RAID Level 6

(*Independent access*)



(g) RAID 6 (Dual redundancy)

- Two different parity calculations are carried out and stored in separate blocks on different disks.
- Up to 2 disks fault tolerance
- This make it possible to regenerate data even if two disks containing user data fail.

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205.

👍 Provides extremely high data availability.

👎 Suffer more than 30% drop in overall write performance compared with a RAID 5.

* *RAID* 5 *and RAID* 6 *read performance is comparable.*

William Stallings (2016). *Computer Organization and Architecture: Designing for Performance* (10th Edition). United States: Pearson Education Limited, p.205, p.212.

Video with examples

- RAID 0, 1, 2, 3, 4, 5, 6 and 10 explained with animation
- https://www.youtube.com/watch?v=wTcxRObq738

Mb/in^2 (Megabits per square inch)
Gb/in^2 (Gigabits per square inch)

Future Data Storage

- Advances in technology have defied all efforts to define the ultimate upper limit for magnetic disk storage.
  - In the 1970s, the upper limit was thought to be around 2Mb/in^2.
  - Today's disks commonly support 20Gb/in^2.

- Improvements have occurred in several different technologies including:
  - Materials science.
  - Magneto-optical recording heads.
  - Error correcting codes.

Gb/in^2 (Gigabits per square inch)

- As data densities increase, bit cells consist of proportionately fewer magnetic grains.
- There is a point at which there are too few grains to hold a value, and a 1 might spontaneously change to a 0, or vice versa.
- This point is called the *superparamagnetic limit*.
  - In 2006, the superparamagnetic limit is thought to lie between 150Gb/in^2 and 200Gb/in^2 .
- Even if this limit is wrong by a few orders of magnitude, the greatest gains in magnetic storage have probably already been realized.

Future advancement in data storage most likely will occur through the use of totally new technologies.

- Research into finding suitable replacements for magnetic disks is taking place on several fronts.

- Some of the more interesting technologies include:
  - Biological materials.
  - Holographic systems, and
  - Micro-electro-mechanical devices.

7.4 Summary

- I/O systems are critical to the overall performance of a computer system.
  - consist of memory blocks, cabling, control circuitry, interfaces, and media.

- I/O control methods include programmed I/O, interrupt-based I/O, DMA, and channel I/O.
- Buses require control lines, a clock, and data lines. Timing diagrams specify operational details.

- Magnetic disk is the principal form of durable storage.
- Optical disks provide long-term storage for large amounts of data, although access is slow.
- Magnetic tape is also an archival medium. Recording methods are track-based, serpentine, and helical scan.

- RAID gives disk systems improved performance and reliability.
- Any one of several new technologies including biological, holographic, or mechanical may someday replace magnetic disks.
- The hardest part of data storage may be end up be in locating the data after it’s stored

Review Questions

7

7.1 Explain how programmed I/O is different from interrupt-driven I/O.

7.2 What is polling?

7.3 How is channel I/O different from interrupt-driven I/O?

7.4 How is channel I/O similar to DMA?

7.5 What distinguishes an asynchronous bus from a synchronous bus?

7.6 How do DVDs store so much more data than regular CDs?

7.7 Explain how serpentine recording differs from helical scan recording.

7.8. Name the four types of I/O architectures.

Linda Null and Julia Lobur (2003). *The Essentials of Computer Organization and Architecture.* United States: Jones and Bartlett Publishers. p.330.