Homework, etc.

- ☐ Reading: Chapter 12 (on website)
- ☐ Homework:
  - Exercises from Ch. 12
  - Due: 10/12 (later than usual due to break)
- ☐ Prelim I:
  - Friday, 10/12
  - Covers: Everything through Friday (10/5) lecture
  - Only up to today's (RAID) in-depth, though

COS 140: Foundations of Computer Science

RAID: Redundant Array of Independent Disks

Fall 2018

The problem

- How to store data:
  - Reliably
  - So that we can maximize a lot of *requests* by different processes
  - So that we can maximize the amount of data transferred/second to each process
- These are conflicting, as we'll see!
- We'll concentrate on disk storage

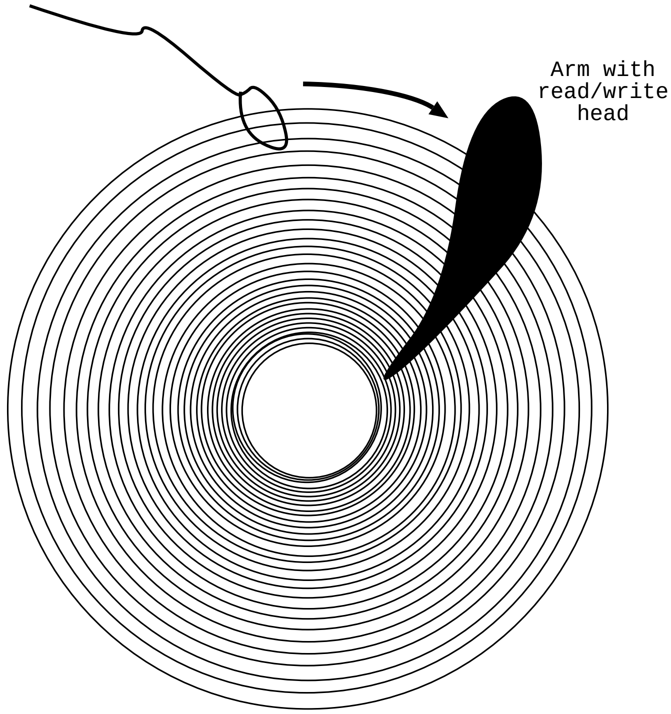
More About Magnetic Disks

- Type of external memory, like magnetic tape, flash, or optical disks (e.g., DVSs)
- Access method: *direct access*

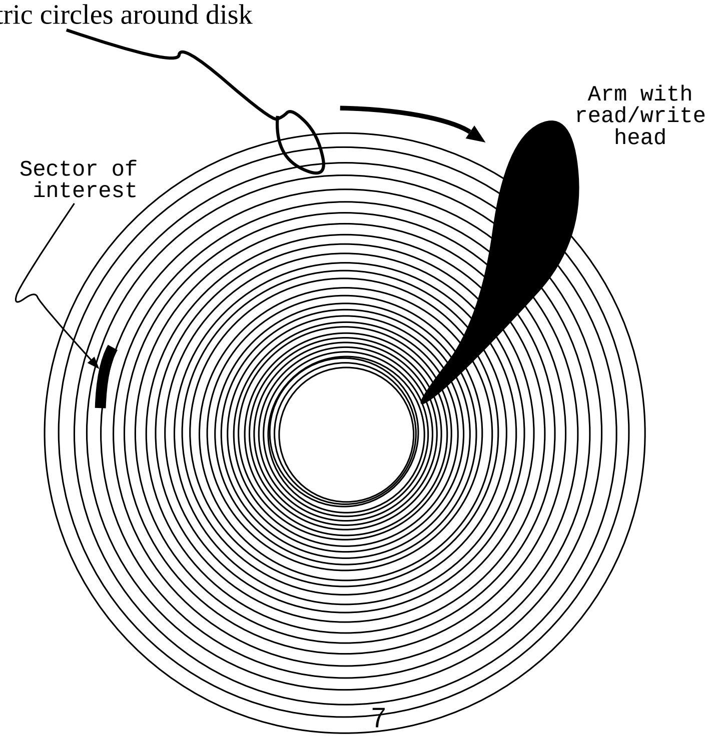
Arrangement of Data on Mag Disks

(animation)

TRACKS: Concentric circles around disk

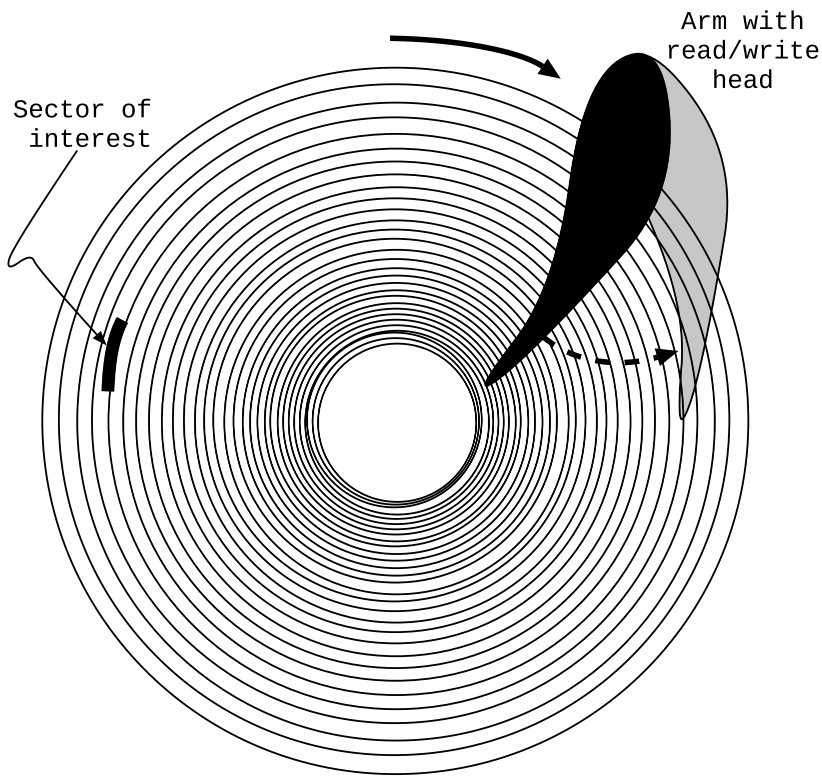


TRACKS: Concentric circles around disk



Access Time for Disks

- *Seek time:* time till head on correct track
- *Rotational latency:* time till the correct sector under head
- Access time = Seek time + Rotational latency



Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency:

$$\frac{60s}{5400\text{rev}} \times 0.5\text{rev} \approx 6 \text{ ms average rotational latency}$$

  - Transfer time:

$$4KB \times \frac{1GB}{2^{20}KB} \times \frac{8Gb}{1GB} \times \frac{1s}{3Gb} \approx 0.01\text{ms}$$

  - Total time $\approx 12 + 6 = 18$ ms

Types of Disks

- Type: depends on how close head gets to surface
- Closer the head ⇒ narrower head can be ⇒ narrower tracks ⇒ more data
- Closer the head ⇒ increased chance of errors due (e.g.) to impurities, dust, etc.
- Standard disks: head floats on a cushion of air – does not come in contact with the disk
- Floppy: head touches the disk when reading and writing
- Winchester: in a sealed unit so head can get closer to the disk because there are no contaminants

Example: Seagate 3.5 in. hard disk

(Eric Gaba Wikimedia Commons user: Sting)

Performance Issues for External Memory

- Reliability
- Speed
  - *Transfer capacity* - how much data can be read from or written to the disk in a given amount of time
  - *I/O request rate* - how many reads or writes can be accomplished in a given amount of time
- Cost

How to Measure Speed: Transfer Capacity

- Amount of data that can be read from or written to the disk per second
- Important → large amount of data/request
- Depends on: buses, disk device, other factors

How to Measure Speed: I/O Request Rate

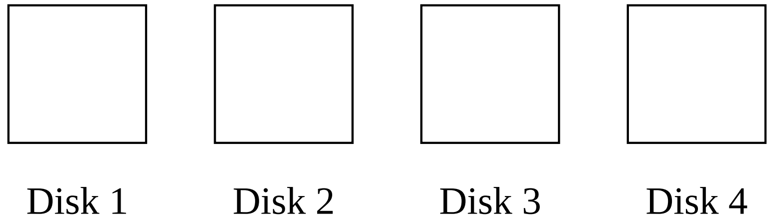
- Number of requests/second that are serviced by disk (reads or writes)
- Important → many requests generated per second

RAID

What RAID Hopes to Accomplish

- Improve performance through parallelism.
  - increase speed
  - increase reliability
- But: extra disks (for parallelism) ⇒ higher cost.

Architecture of RAID

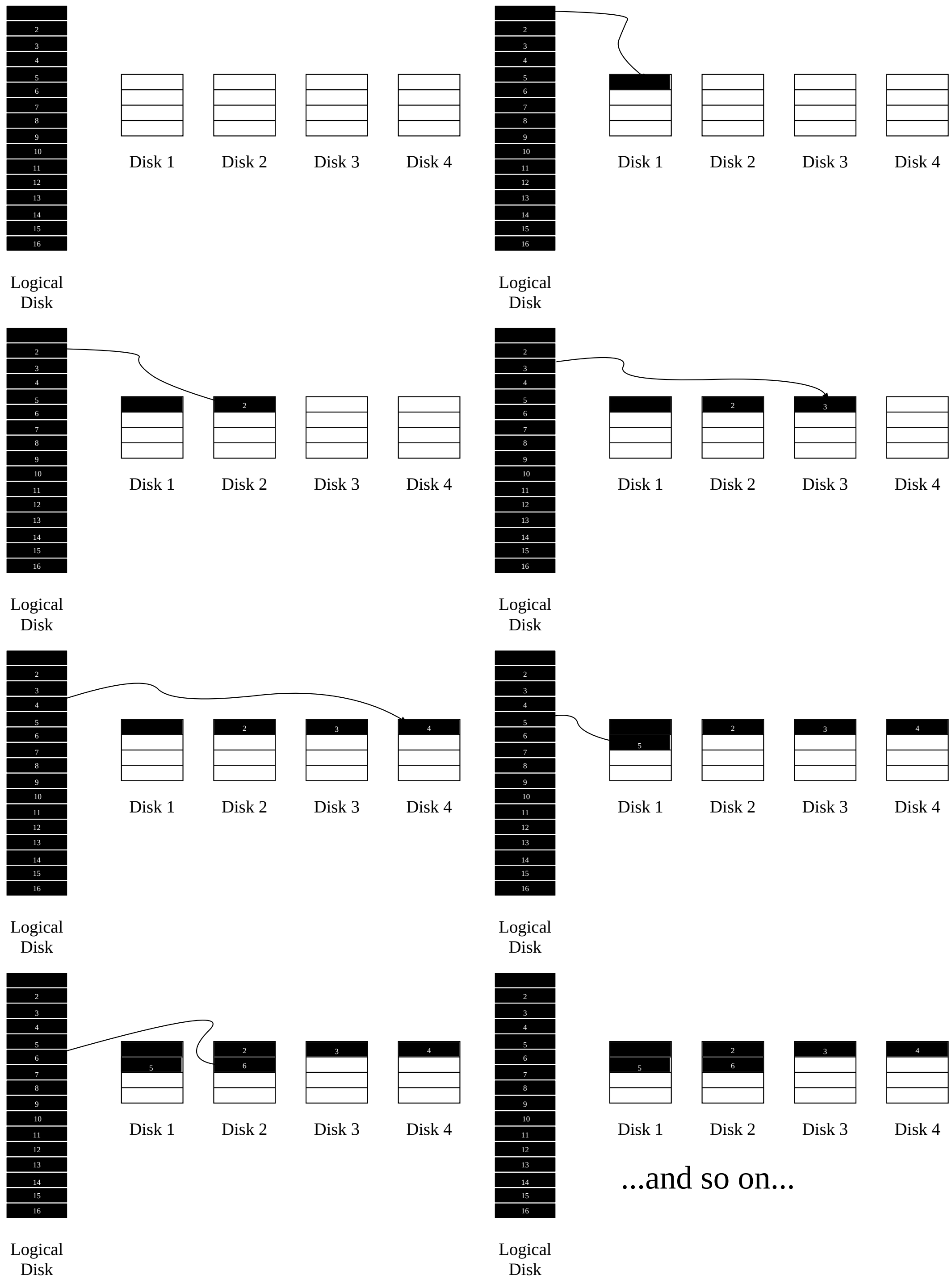


- Several disks in the *array*
- Different *RAID levels* specify how disks are used
  - Each level: addresses different issue(s)
  - Order of levels not significant

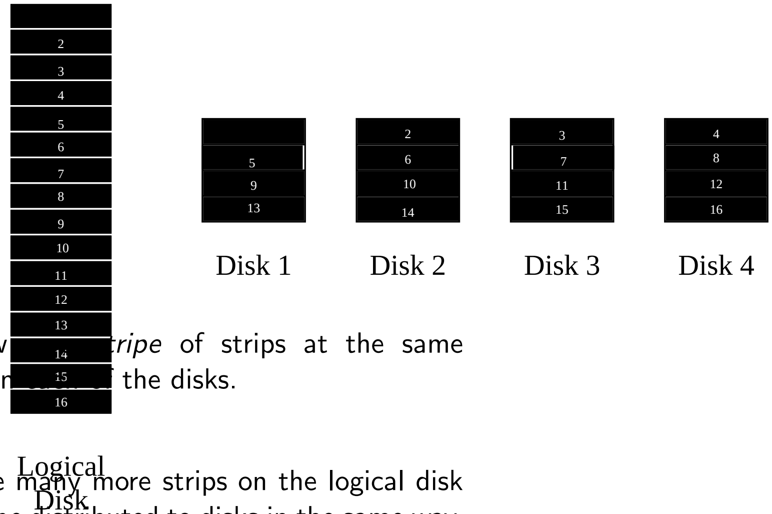
Distributing Data on the Disks

- *Logical disk:*
  - Abstraction of real disks
  - Think of single virtual disk on which data is stored
- Divide data unto equal-length chunks called *strips*
- Put strips on real disks in (e.g.) *round-robin* fashion
- *Stripe*: all the strips at correspondign locations on the disks

Distributing Data on the Disks



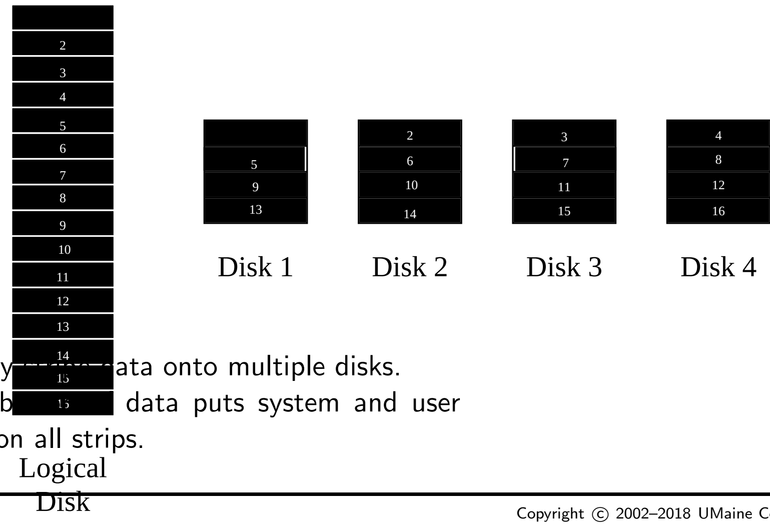
Distributing Data on the Disks



Each row [illegible]*tripe* of strips at the same location i[illegible]f the disks.

There are many more strips on the logical disk that will be distributed to disks in the same way. This small example is just for illustration.

RAID Level 0



- Simply [illegible]ata onto multiple disks.
- Distrib[illegible] data puts system and user data on all strips.

RAID 0 benefits

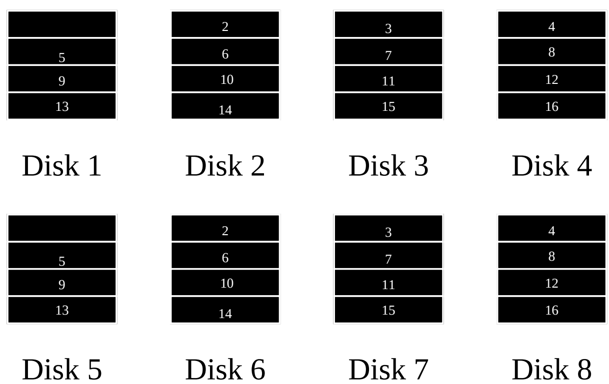
- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times
- Effect on request rate?
  - Suppose $r \leq s$
  - $\Rightarrow$ disk active per strip $\Rightarrow$ multiple requests handled at once
  - $\Rightarrow$ increased request rate
  - Ideally $r = s$: request rate increased up to n times
- So which? Depends on system characteristics, goals

RAID level 0 cost

- If n disks were going to be used anyway ⇒ no additional cost
- If not, then n small disks likely more expensive than 1 large disk

RAID 1

RAID Level 1



- Two sets of disks, mirror images of each other.

RAID level 1 benefits

- Reliability $\uparrow\uparrow$
  - Data completely redundantly stored
  - Disk fails: read from/write to copy
- Transfer rate, request rate
  - Same arguments as for RAID 0 re: strip size vs. request size
  - Additionally:
    - If disk busy, can read from duplicate $\Rightarrow \uparrow$ speeds
    - Could handle up to $2\times$ requests of RAID 0 if $s = r$
    - Could make $S = \frac{1}{2}r \Rightarrow$ all disks involved, up to $2 times$ request rate of RAID 0
  - Writes must be done to both disks, but they can be done in parallel.

RAID 1 cost

- If need n disks worth of data, need $2n$ disks
- I.e., doubles cost

So what is RAID 1 good for?

- ☐ Critical data for which a failure cannot be tolerated and where the cost is not a problem.
- ☐ Additional ↑ transfer or request rates

RAID 2

RAID Level 2

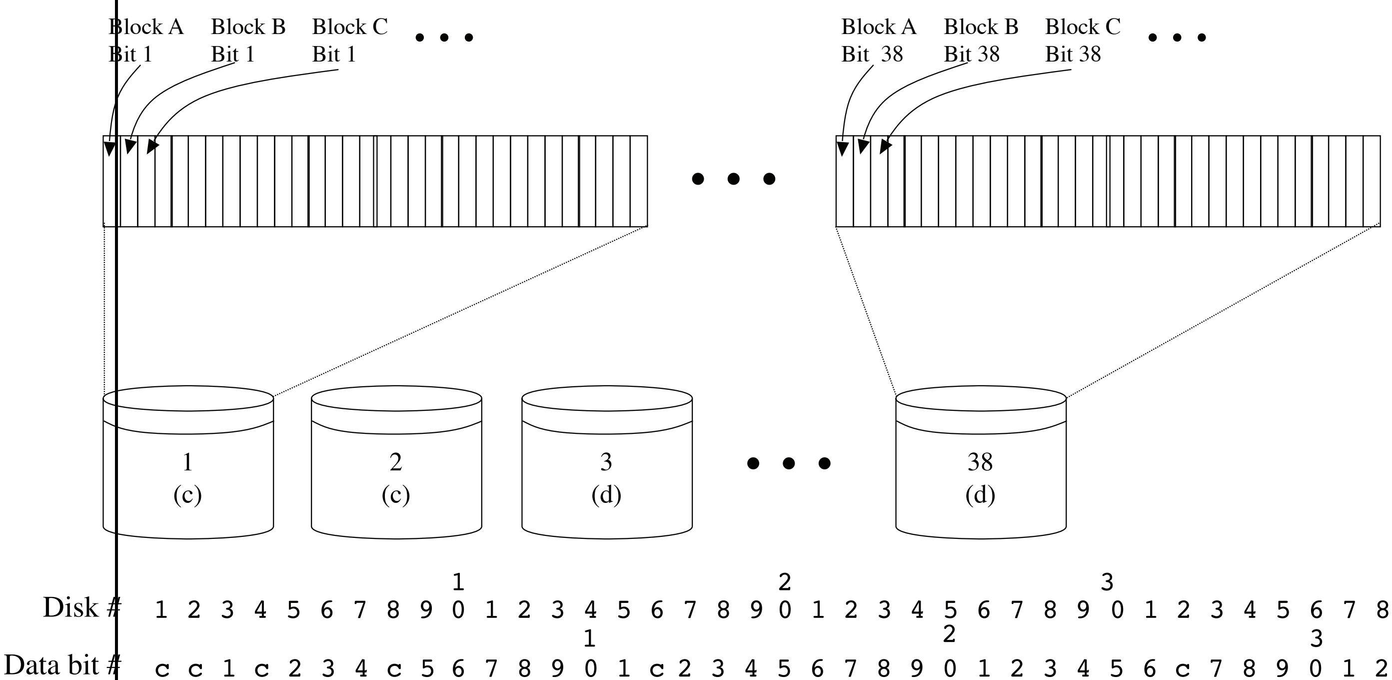
- ☐ Idea: use some additional space to store an *error-correcting code*
- ☐ When an error occurs (on read or write), use that to fix it
- ☐ Uses a *Hamming code* (we'll study this later)
- ☐ For corresponding bit locations on each data disk, create Hamming code
  - Hamming code requires about $\log_2 n$ additional bits for n data bits
  - $\Rightarrow$ extra disks needed

RAID 2 benefit

- Reliability
- Correction of 1-bit errors on read or write
- Reconstruct data if one disk fails.

RAID 2 costs

- Synchronized disks, write penalty (to compute ECC)
- May need a large number of disks – or a large number of reads/writes per disk



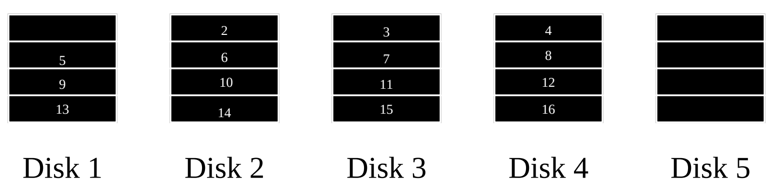
RAID 2, 32-bit data blocks

What is RAID Level 2 Good For?

- ☐ Not commercially implemented
- ☐ Would be good if many single-disk errors...
- ☐ ...but unlikely and...
- ☐ ...disks themselves use ECC!
- ☐ Bit error rates ≈ 1 per 10^{14} bits read
- ☐ If are really worried about data, use Level 1.

RAID 3

RAID Level 3



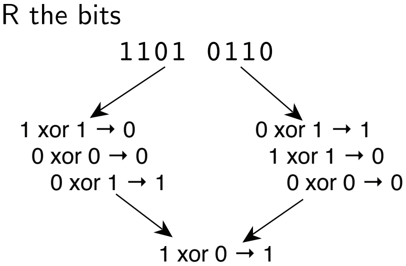
- ☐ Single extra disk stores a *parity bit*.
- ☐ Strip size: byte or word, access disks in parallel (synchronized)

What is a Parity Bit?

- Consider # 1s in data:
  - 0100 1011
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even ⇒ *even parity*
  - Set to make total # 1s odd ⇒ *odd parity*
- E.g., even parity:
  - 0100 1011 ⟶ 0100 1011 0
  - 0100 1111 ⟶ 0100 1111 1
- Store both data and parity bit

How is the Parity Bit Used?

- When writing: compute, store parity bit
- When reading:
  - Know if parity *supposed* to be even or odd
  - Compute parity: if not correct ⇒ error
- How to compute?
  - Remember homework asking about odd # of 1s in a number?
  - For even parity, can just XOR the bits



RAID 3 benefits

- Reliability $\uparrow$
  - Detect errors; can try re-reading
  - If disk drive fails $\longrightarrow$ *reduced mode*
    - *Every* read will $\rightarrow$ parity error
    - But now know which bit is wrong!
- Transfer rate $\uparrow$ due to small strip size
- Request rate: no change

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?
  - Modern disk drives are very good – maybe 1 error per 10^{14} bits read
  - P(error) in a 4 KB read: about 3×10^{-10}
  - P(2 errors) in a 4KB read $= (3 \times 10^{-10})^2 \approx 10^{-19}$
  - I.e., 0.0000000000000000001, or expect 1 2-bit error in 10,000,000,000,000,000,000 4KB reads

What is RAID Level 3 Good For?

- ☐ When some error detection is needed
- ☐ High transfer rate and low number of outstanding requests.

RAID 4 & 5

RAID Level 4, 5, 6, and beyond

- ☐ Same as RAID Level 3, but uses *independent access array*: disks in array operate independently
  - Uses larger strips ⇒ strip-level parity
  - Increases IO request rates at the expense of transfer capacity.
  - Write penalty: have to read old data strip, old parity, then write them.
  - Parity disk can become bottleneck.
- ☐ RAID Level 5 distributes the parity bits across the disks instead of having them all on one disk, to solve a potential problem with parity disk bottlenecks for Level 4.
- ☐ RAID Level 6: two parity blocks per stripe
- ☐ Others: some proprietary, some combinations of others (e.g., RAID 0+1)