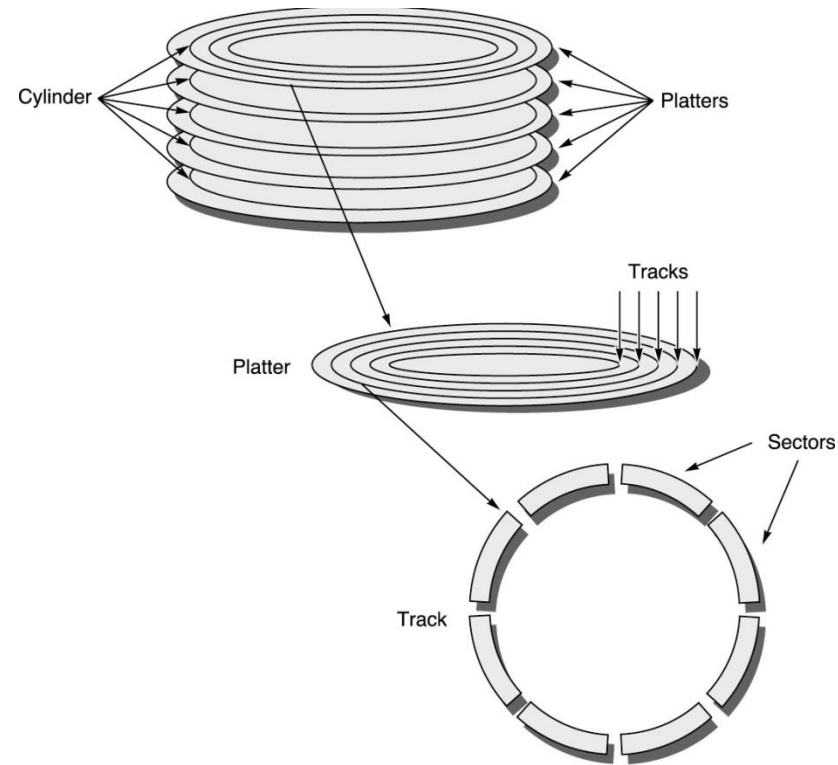

Storage

Magnetic Disks

- To control: Disk Controller
- Disk surfaces are divided into **tracks**
- Each track is divided into **sectors**
- Read/Write: a movable arm with a **read/write head** is located over each surface
- **Seek time**: time to move the arm over the proper track
(min seek time, max seek time, and avg seek time)
- **Rotation Latency** (Rotational Delay):
Time for the requested sector to rotate under the head



Average Rotation time = $0.5 / \text{Revolutions Per Minute}$

e.g. $0.5 / (3600 \text{ RPM}) = (0.5 * 60 / 3600) \text{ sec} = 8.3 \text{ ms}$

Redundant Arrays of (Inexpensive) Disks

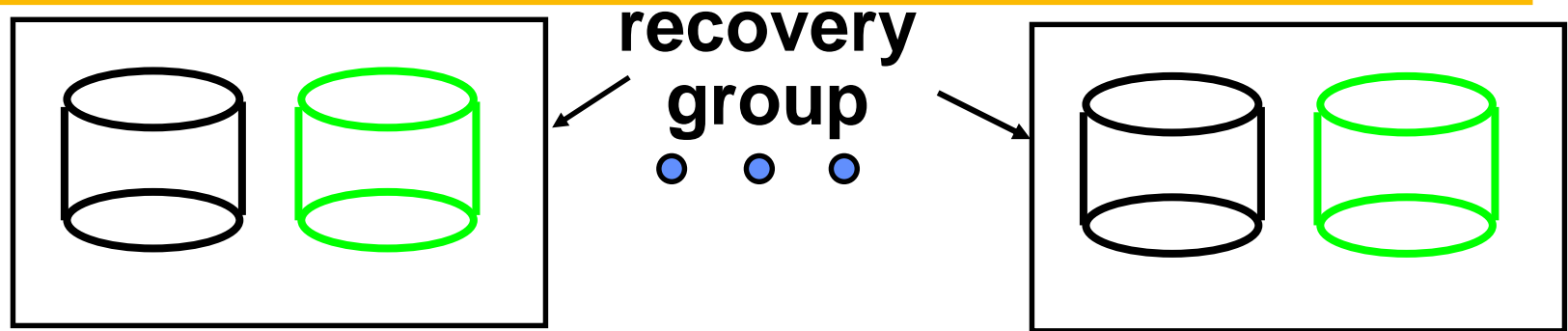
- Files are "striped" across multiple disks
(spreading data over multiple disks)
- Force accesses to several disks if the data files are large
- Redundancy yields high data availability
 - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
 - ⇒ Capacity penalty to store redundant info
 - ⇒ Bandwidth penalty to update redundant info

RAID 0

- **No redundancy**
- **Sometimes nicknamed JBOD (Just a Bunch of Disks)**
- **Used as a measuring stick for other RAID levels in terms of cost, performance and dependability.**

Redundant Arrays of Inexpensive Disks

RAID 1: Disk Mirroring/Shadowing



- Each disk is fully duplicated onto its “mirror”
Very high availability can be achieved
- **Bandwidth sacrifice on write:**
Logical write = two physical writes
 - Reads may be optimized
- **Most expensive solution: 100% capacity overhead**

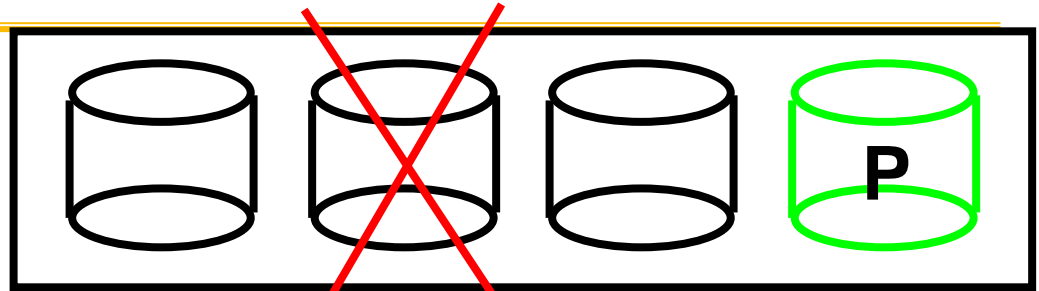
•

RAID 2

- Inspired by applying memory-style error correcting codes to disks
- RAID 2 not used since other more interesting schemes,

Redundant Array of Inexpensive Disks

RAID 3: Parity Disk



Striped physical
records

P contains sum of
other disks per stripe
mod 2 (“parity”)

If disk fails, subtract
P from sum of other
disks to find missing information

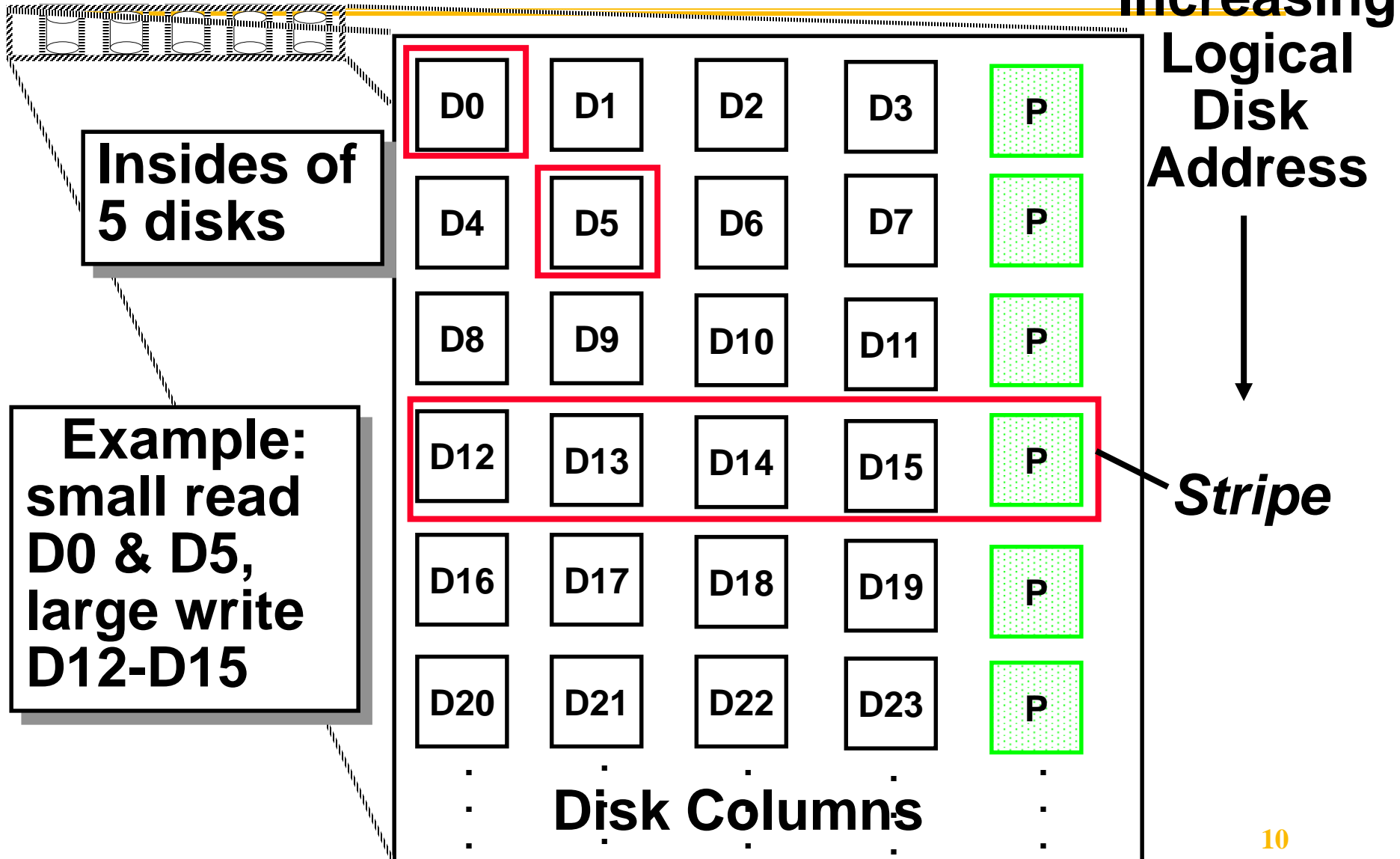
1	1	1	1
0	1	0	1
1	0	1	0
0	0	0	0
0	1	0	1
0	1	0	1
1	0	1	0
1	1	1	1

RAID 3

- **Sum computed across recovery group to protect against hard disk failures, stored in P disk**
- **Logically, a single high capacity, high transfer rate disk: good for large transfers**
- **Wider arrays reduce capacity costs, but decreases availability**
- **33% capacity cost for parity if 3 data disks and 1 parity disk**

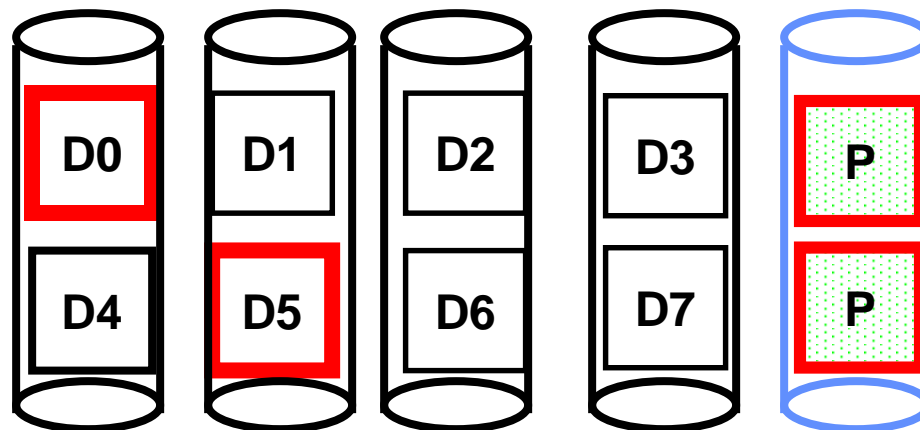
Redundant Arrays of Inexpensive Disks

RAID 4: High I/O Rate Parity



Small Writes

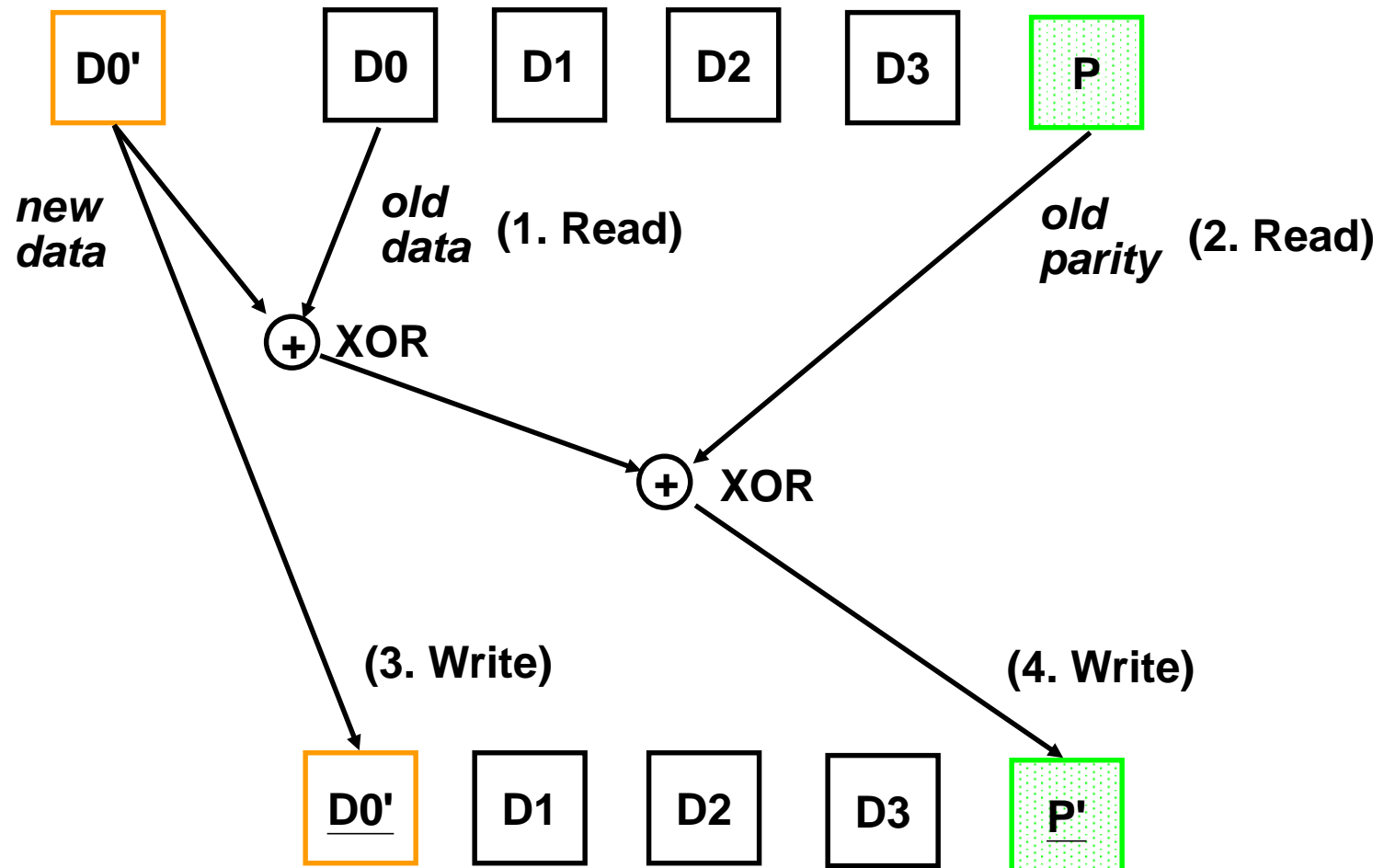
- RAID 4 works well for small reads
- **Small writes (write to one disk):**
 - Option 1: read other data disks, create new sum and write to Parity Disk
 - Option 2: since P has old sum, compare old data to new data, add the difference to P
 - **RAID4 takes Option 2**
- In RAID4: Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk



Small Writes

Option2 of Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes



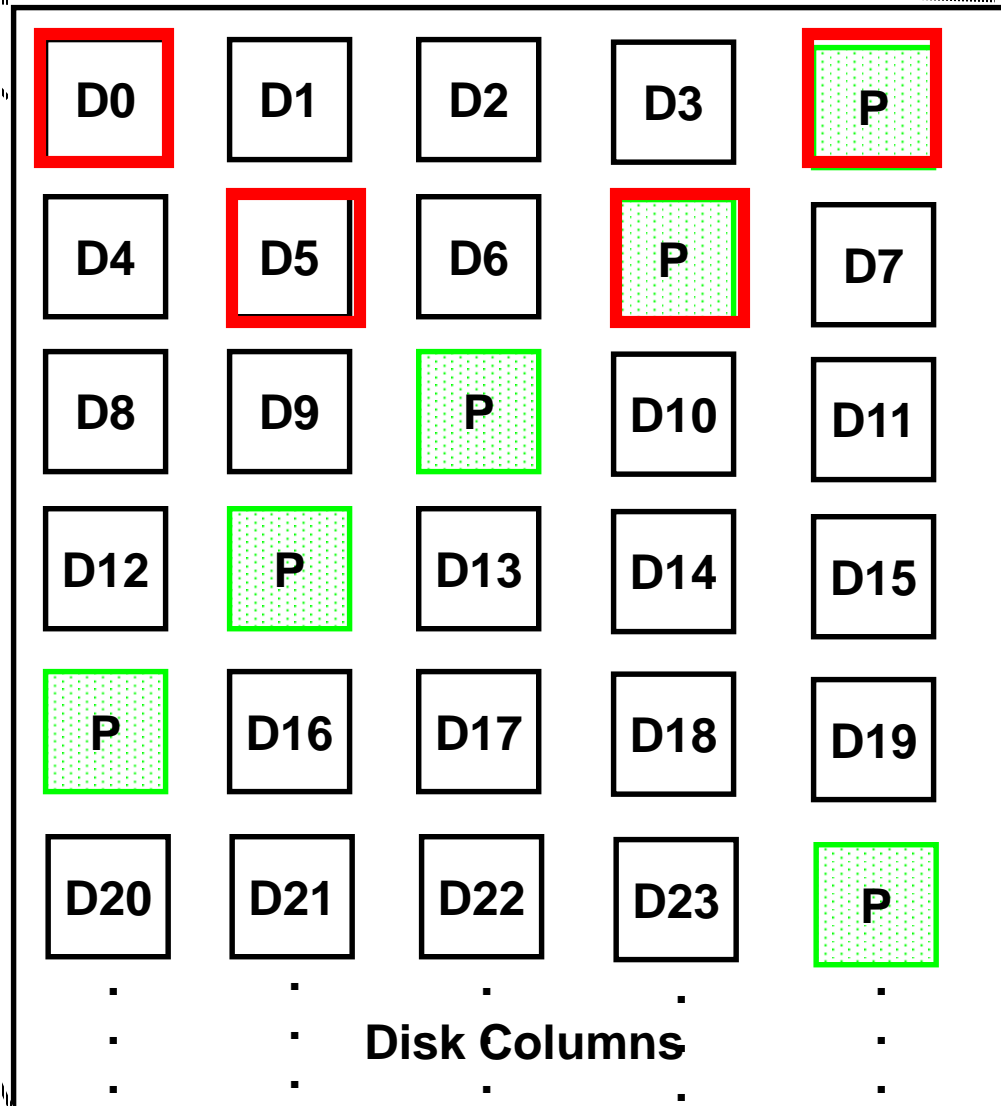
Redundant Arrays of Inexpensive Disks

RAID 5: High I/O Rate Interleaved Parity



Independent writes possible because of interleaved parity

**Example:
write to
D0, D5
uses disks
0, 1, 3, 4**



Increasing
Logical
Disk
Addresses



RAID 6: Recovering from 2 failures

- **Why > 1 failure recovery?**

- operator accidentally replaces the wrong disk during a failure
- since disk bandwidth is growing more slowly than disk capacity, the MTT Repair a disk in a RAID system is increasing
 - ⇒ increases the chances of a 2nd failure during repair since it takes longer
- reading much more data during reconstruction meant increasing the chance of an uncorrectable media failure, which would result in data loss

RAID 6: Recovering from 2 failures

- Network Appliance's *row-diagonal parity* or *RAID-DP*
- Like the standard RAID schemes, it uses redundant space based on parity calculation per stripe
- Since it is protecting against a double failure, it adds *two check blocks per stripe of data*.
 - If $p+1$ disks total, $p-1$ disks have data; assume $p=5$
- Row parity disk is just like in RAID 4
 - Even parity across the other 4 data blocks in its stripe
- Each block of the diagonal parity disk contains the even parity of the blocks in the same diagonal

Example $p = 5$

- Row diagonal parity starts by recovering one of the 4 blocks on the failed disk using diagonal parity
 - Since each diagonal misses one disk, and all diagonals miss a different disk
- Once the data for those blocks is recovered, then the standard RAID recovery scheme can be used to recover two more blocks in the standard RAID 4 stripes
- Process continues until two failed disks are restored



failures