

NAND Flash Media Management Through RAIN

Scott Shadley, Senior Product Marketing Manager
Micron Technology, Inc.
Technical Marketing Brief

What Is RAIN, and Why Do SSDs Need It?

This brief compares redundant array of independent NAND (RAIN) protection in enterprise solid state drives (SSDs) to the redundant array of independent (and inexpensive) disks (RAID) protection scheme for hard disk drives (HDDs) and details how RAIN technology benefits systems using Micron's SSD products.

Existing Protection Techniques in Enterprise Storage

In the early days of using hard disk drives (HDDs), it became apparent that a single device had enough potential for failure to warrant the creation of a protection scheme to prevent unwanted data loss.

Several data protection mechanisms were developed, including the RAID protection scheme. Each mechanism implemented a slightly different form in terms of how it worked and its impact on array performance. For HDD arrays, the most common implementation became RAID 5 (rotating single-bit parity) because it protected a single drive from failing with minimal impact on performance.

Figure 1 shows an example of how RAID 5 works. In this example, parity data is rotated across four unique disks. The parity (in red) represents the data required for the system to recover any of the actual blocks of stored information. As with all parity protection, this parity occupies space on the disks, and therefore, affects the overall capacity of the array, which is important when considering how this technique and protection scheme can be applied to an SSD.

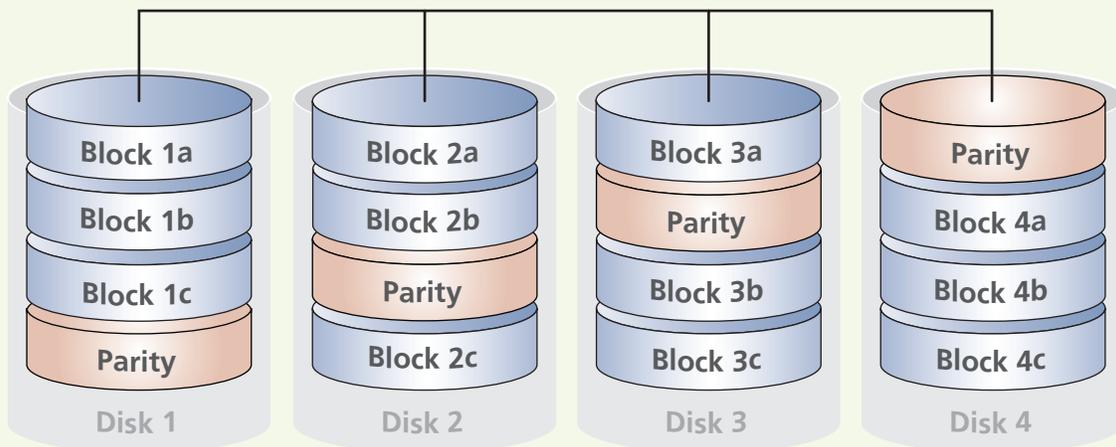


Figure 1: RAID 5 in HDDs

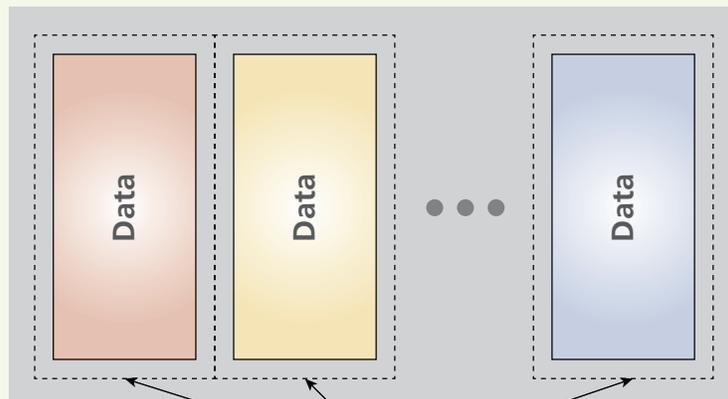


ECC Is No Longer Enough to Protect NAND in SSDs

Like HDDs, SSD devices also have the potential for failure and data loss. The original method for protecting data in SSD designs was simply to add the required levels of error correction code (ECC) to the given pages and then recover data using ECC. When the first SSDs were shipped, this initial level of protection was sufficient to minimize the chance of failure, much like with the early HDD products.

But as NAND geometries continue to shrink, and controller complexity increases, a protection scheme beyond

simple ECC is needed—like RAID was needed for HDDs. Similar solutions to protect the user data in SSDs have emerged as the market matures, but unlike with HDD arrays, where the industry collectively settled on a very few specific versions of array-level RAID implementations (most commonly, single-bit parity RAID 5 and double-bit parity RAID 6—both rotating), SSDs utilize approaches at the device level. Thus, no two versions of SSD data protection technology are guaranteed to be exactly the same. With many competitive architectures being created by multiple SSD manufacturers, it is important to understand the details of the approach being used. These details can be thought of as tools in the SSD designer’s toolkit and can be tuned and adjusted to best match the SSD design to the intended use, workload, endurance, and a host of other factors.



ECC protection on every page

Figure 2: Original method of SSD protection – Required levels of ECC added to every page

To help differentiate between array-level (traditional HDD) parity protection and SSD-specific protection, the term RAIN is used in reference to SSD protection.

SSD Architecture Review

To show how RAIN can be beneficial in SSD design and to show the relevance of parity-based RAIN, it is important to first look at how data is organized in the SSD. Unlike HDD designs that have rotational storage organizations (platter, track, sector), SSDs use Flash devices as the storage media inside, enabling new design flexibility.

The SSD layout in Figure 3 shows some similarities between how SSD NAND devices are arranged compared to the multiple-drive arrangement of HDD arrays. This high level of similarity suggests possible integration of RAID into a single SSD device rather than using many individual drives. This represents the first level at which RAIN comes into play within a RAIN-enabled SSD.

Most SSD controllers use parallelism in order to increase SSD performance and locate stored data across the many smaller Flash devices. For example, to create a 350GB SSD, smaller 8GB NAND devices, along with parallel data placement and retrieval, are used to create this individual drive capacity.



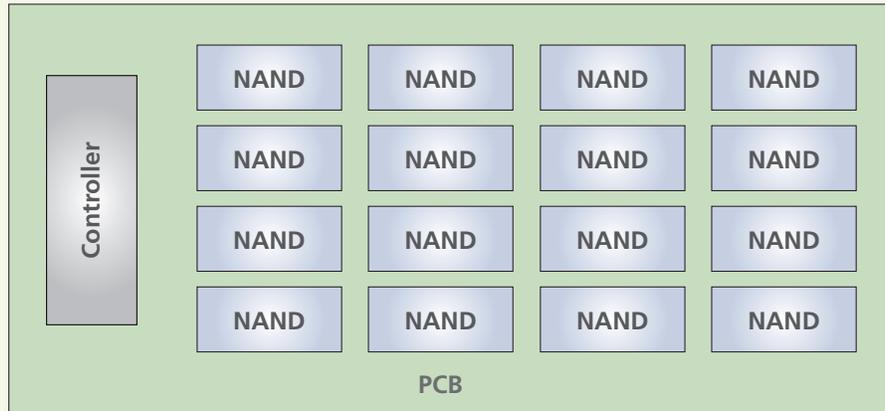


Figure 3a: Physical NAND device arrangement on an SSD

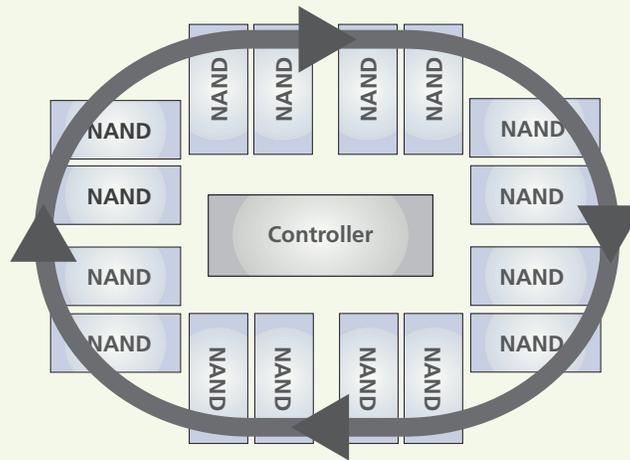


Figure 3b: Logical data placement/arrangement on an SSD

This parallelism used in SSDs works like an HDD RAID 0 array where data is spread across multiple HDDs in the array. This is also referred to as “striped data, without parity.”

The data architecture inside a NAND component can be organized in many ways, and each controller’s architecture and firmware can have a unique implementation. Moving data the most efficient way possible is the key to a successful design. The better the methodology, the faster and more compelling a user will find the end product.

How Does RAIN Work in Micron SSDs?

RAIN technology adds user data protection that extends beyond ECC, minimally impacts drive performance, and optimizes NAND management. With a high degree of parallelism already in place within the SSD (striping), adding a parity protection architecture is the next logical step.

However, there are many possible ways to implement this parity protection scheme—each with potential tradeoffs that must be considered when designing the SSD.



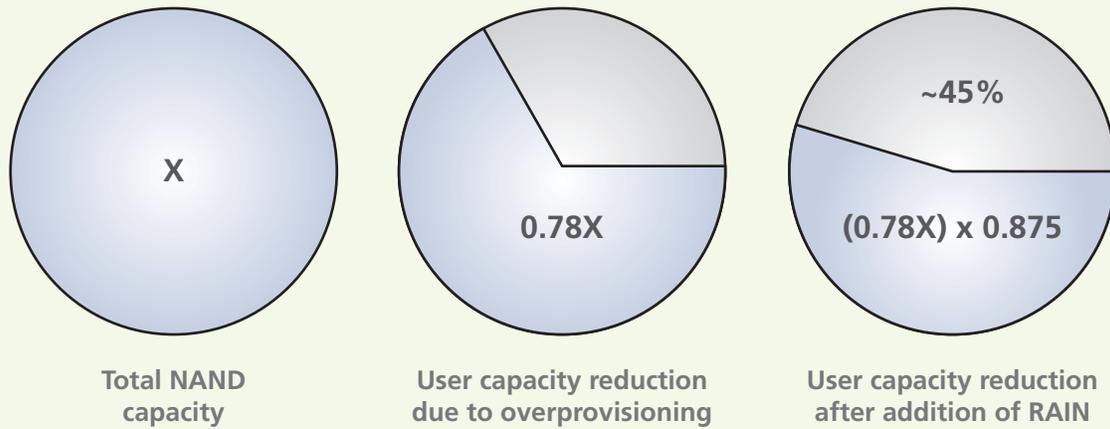


Figure 4: How implementing RAIN can affect the end-user capacity of the drive

Figure 4 shows a hypothetical example of how implementing a protection scheme like RAIN can impact the amount of NAND capacity available to the end user. Like in the HDD environment, where this is already standard practice, having SSD capacity consumed by a protection scheme is an acceptable tradeoff for the data protection benefits.

The example above assumes the following:

- **SSD RAW NAND capacity:** $X = 512\text{GB}$
- **Overprovisioning level:** $Z = 0.78 ((100-22)/100)$
- **RAIN design with 1 parity element for each 7 storage elements:** $Y = 7/8$ (a 1:7 ratio) = 0.875

Therefore:

- **User-available capacity of 512GB drive:**
 $512\text{GB} \times 0.875 \times 0.78 = 350\text{GB}$

Rain Options for SSD Designers

Unlike HDD arrays where the most common descriptions are the parity-generated and parity-placement mechanisms (for example RAID 5 or RAID 10), the common descriptions associated with RAIN differ slightly.

Most commonly, the description states the stripe level or stripe length, which expresses how many user data elements are associated with a single parity element. This is often referred to as "N to 1" where N is the number of user data elements.

For example, if an SSD associates seven user data elements to one parity element, that SSD would be described as having a 7:1 RAIN stripe. This specific RAIN stripe implementation is shown below in Figure 5a. The P in this diagram represents the parity element being generated and stored along with the user data it is protecting.



Figure 5a: 7:1 RAIN stripes



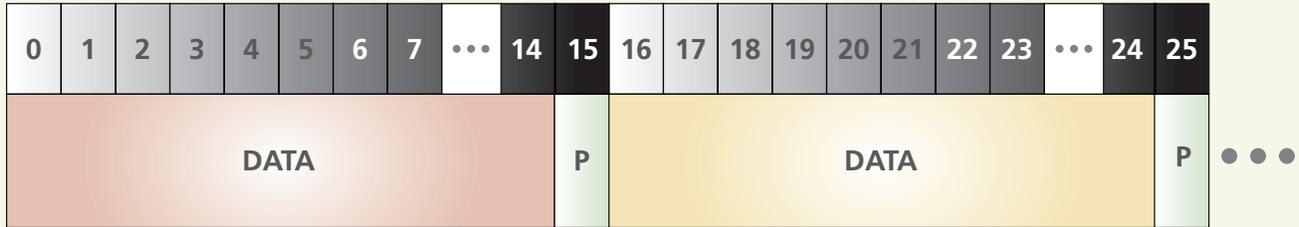


Figure 5b: 15:1 RAIN stripes



Figure 5c: 127:1 RAIN stripes

Other RAIN stripes exist, as shown above in Figure 5b (15:1 stripe) and Figure 5c (128:1 stripe), and work similarly.

It is important to understand that the stripe length is a parameter that is fixed when the SSD is designed and is optimized for a myriad of factors. Many different parity lengths can be considered and implemented, and each have a unique impact on the overall design, capacity, and, potentially, performance of the drive. In the case of Micron’s SSD firmware development, the differences in the performance of these data stripes are carefully

characterized to determine the best performance of the SSD and the best protection method required by the SSD design fundamentals.

Table 1 shows how data stripe lengths were considered during the design of Micron’s P320h PCIe SSD. Only the 1 + 1P version of the parity architecture has an impact on performance. This method of protection consumes 50% of the NAND in parity; therefore, it does not provide a viable choice for SSD design because it is simply not economical.

Attribute	256GB Raw Capacity SSD			
	No Parity	7 + 1P	3 + 1P	1 + 1P
Sequential read @ 64KB transfer size	1.5 GB/s	1.5 GB/s	1.5 GB/s	1.2 GB/s
Sequential write @ 64KB transfer size	1.5 GB/s	1.5 GB/s	1.5 GB/s	1.2 GB/s
Random read @ 4KB transfer size	375,000	375,000	375,000	300,000
Random write @ 4KB transfer size (full drive)	171,000	171,000	143,000	94,000

Table 1: P320h PCIe SSD solution data stripe consideration



By definition, RAIN is a redundant array of independent NAND, and taking this definition literally is the most efficient way to discern what is occurring in the SSD. While most simple SSD designs are focused only on performance (parallelism) to the point of failure, Micron has developed a solution that ensures independence as well as parallelism, as shown in the following example.

Example of Failure Recovery With RAIN

This example walks through a failure and recovery of the P320h drive in Table 1 using a 7:1 data stripe; it shows how the new data structure will be intact and can be carried forward through the life of the SSD. (Additional details and unique features of RAIN and the SSD firmware management are not discussed here due to the in-depth technical aspects and Micron-specific IP. These details are available through more direct discussions with Micron.)

In this example, each user data element number is shown as an individual data point. In actual drive applications, the level of data recovery can also be on either a smaller, individual NAND page or a much larger failure recovery.

While data is being moved around within the SSD through either host-initiated data writes or drive-level background operations, RAIN successfully recovers data, as shown in Figure 6.

1. As data being written to the drive goes through the process of being programmed, data is arranged so that parity can be created and stored in the eighth page of this 7:1 example stripe. Over time, as ECC corrects data, RAIN does not need to be applied for any form of data recovery. RAIN is focused on defects that occur due to unexpected NAND issues; it is also focused on defects that occur as the device wears and ECC limits are exceeded.
2. As data is read from the drive, it is processed through the RAIN engine. In the event that any portion of the stripe is flagged as an unrecoverable error, the primary RAIN algorithm kicks in and recovers the data via the parity process of using an XOR algorithm on the remaining data. In the event that this data failure is a read process only, the newly recovered data is then stored in a new location on the drive, and other data is moved around in the drive as part of

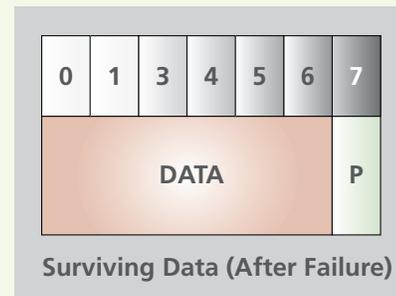
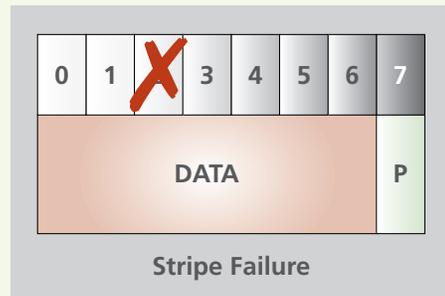
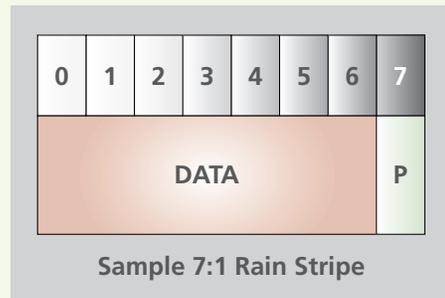


Figure 6: RAIN data recovery during data moves

- the normal background operations. Instead of simply moving existing data and replacing it with new write data, the newly recovered data is instantaneously moved into normal drive management routines.
3. The existing location where the failure occurs is not instantaneously moved around within the drive. That type of recovery is suboptimal, and, although effective, it creates unnecessary wear and write amplification. Micron’s RAIN has the ability to leave the existing data intact until it is absolutely necessary to reprogram the entire block. This has no impact on



performance or drive data management, which is a key feature of the independent nature of the solution.

4. Data is then gathered by the controller to indicate that this recovery has occurred. It is managed by the internal processes of the drive.

Conclusion

Data integrity is a key attribute that enterprise customers require, and potential customers prefer the assurance that nothing will be lost in their data stores.

Current solutions are focused heavily on data reliability. With HDD solutions, RAID is not optional, but *required*, to ensure this reliability. These solutions focus on ways to store more data in smaller, redundant locations. The continued use of tape-based storage design is another example of the need for secure data storage.

SSDs have moved into enterprise data environments, and they no longer fit the definition of RAID where the

“I” stands for inexpensive. Therefore, implementers need solutions that do not require them to create data sets that need redundant safety drives. Finding ways to solve this unusual situation can best be addressed by looking *inside* the SSD, rather than at ways to add layers on top of the use case.

Micron’s RAIN provides this internal solution to customers. SSDs are more robust than HDD solutions currently in use, but, as with all technologies, there is always a small chance to experience issues. RAIN is that next layer within the SSD that provides customers the peace of mind to implement solutions with fewer SSDs, yet still have existing HDD RAID protection levels.

Technology will continue to advance both inside the SSD and in the end-storage solutions. Micron’s goal is to ensure that features like RAIN are always one step ahead of customers’ growing needs for cost-effective data storage solutions with high reliability and endurance.

micron.com

Products are warranted only to meet Micron’s production data sheet specifications. Products and specifications are subject to change without notice.

©2011 Micron Technology, Inc. Micron, and the Micron Logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners. All rights reserved. 11/13 P:11446

