Non-volatile memory (NVM) serves widely different purposes whether it is in an embedded system or a desktop computer. Both rely on memory that retains data without power, the overall need for implementation is the outcome for the performance, power efficiency, & storage capacity.

When dealing with an embedded system, usually found in smart appliances, automotive controls, and most medical equipment, prioritize reliability and efficiency over raw storage space. The flash memory used allows for faster code execution & data storage. NAND flash, more typically seen in devices where higher memory density is required at a lower cost. In addition to those, there are some other styles of ram that provide faster writes, longer lifespan, & overall better performance than traditional ram, albeit at a higher cost.

Due to strict power & size constraints, embedded systems often have limited NVM memory available to them. Some portable microcontrollers have built in flash that eliminates the need for an external storage. Other even have a fail-safe that is backed up to the battery in times of power loss. This give us different use cases, performance, low power usage, & long-term reliability based on what environment the NVM is in.

NVM used in a desktop, is more so general-purpose computing. These require much larger storage capacity (96 GB in my desktop). They are purposed to be used in conjunction with either an HDD or SSD for storage. HDD fits the billing for higher density at a lower speed & cost in comparison to the newer SSD that has advanced controllers for speed & durability, with greatly increased cost.

Embedded systems and desktop computers take fundamentally different approaches to meet their distinct requirements. Embedded devices prioritize efficiency and specialization, using microcontrollers with just enough processing power for their dedicated tasks. This approach keeps costs low and minimizes power consumption, but limits flexibility since these systems typically run lightweight real-time operating systems or no OS at all. In contrast, desktop systems employ powerful general-purpose processors capable of handling multiple demanding applications simultaneously. While this provides greater versatility and user control, it comes at the cost of higher power requirements and more complex system architectures.

Further trade is showcased by the choice of system architecture. Embedded systems can be streamlined to be simple & slower, faster & more complex, or a delicate balance between the two. Many leverage RISC (Reduced Instruction Set) computer processors for their energy efficiency and predictable timing, while some employ FPGAs (Field Programmable Gate Arrays) for customizable high-speed processing or SoCs for compact integration. On the other hand, desktop computers can afford to use more resource intensive architectures that make use of raw computing power. Think of it as brute forcing, mixed in with multi-tasking over slim, power efficient alternatives. Each of these approaches has a targeted audience that excel in various scenarios.

Ltd., A. (n.d.). *What is FPGA?* https://www.arm.com/glossary/fpga#:~:text=Field%20Programmable%20Gate%20Arrays%20(FPGAs,requirements%20after%20the%20manufacturing%20process.

What is RISC? (n.d.). https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/whatis/index.html#:~:text=RISC%2C%20or%20Reduced%20Instruction%20Set,History

Gillis, A. S., & Lutkevich, B. (2024, August 13). *What is an embedded system?: Definition from TechTarget*. Search IoT. https://www.techtarget.com/iotagenda/definition/embedded-system#:~:text=An%20embedded%20system%20is%20a,or%20have%20a%20fixed%20functionality.