



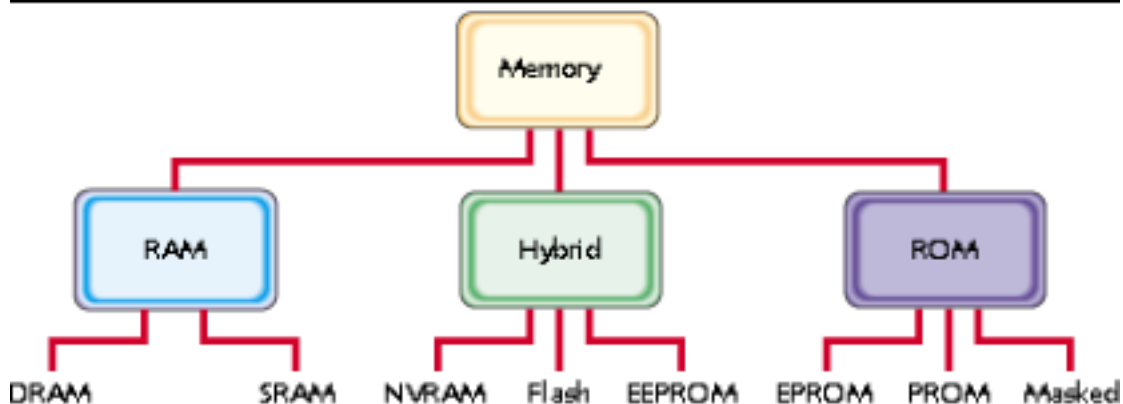
# Memory Types

by Michael Barr

Copyright & COPY; 2001 by CMP Media, Inc. All rights reserved.

Many types of memory devices are available for use in modern computer systems. As an embedded software engineer, you must be aware of the differences between them and understand how to use each type effectively. In our discussion, we will approach these devices from the software developer's perspective. Keep in mind that the development of these devices took several decades and that their underlying hardware differs significantly. The names of the memory types frequently reflect the historical nature of the development process and are often more confusing than insightful. Figure 1 classifies the memory devices we'll discuss as RAM, ROM, or a hybrid of the two.

**Figure 1: Common memory types in embedded systems**



## Types of RAM

The RAM family includes two important memory devices: static RAM (SRAM) and dynamic RAM (DRAM). The primary difference between them is the lifetime of the data they store. SRAM retains its

contents as long as electrical power is applied to the chip. If the power is turned off or lost temporarily, its contents will be lost forever. DRAM, on the other hand, has an extremely short data lifetime-typically about four milliseconds. This is true even when power is applied constantly.

In short, SRAM has all the properties of the memory you think of when you hear the word RAM. Compared to that, DRAM seems kind of useless. By itself, it is. However, a simple piece of hardware called a DRAM controller can be used to make DRAM behave more like SRAM. The job of the DRAM controller is to periodically refresh the data stored in the DRAM. By refreshing the data before it expires, the contents of memory can be kept alive for as long as they are needed. So DRAM is as useful as SRAM after all.

When deciding which type of RAM to use, a system designer must consider access time and cost. SRAM devices offer extremely fast access times (approximately four times faster than DRAM) but are much more expensive to produce. Generally, SRAM is used only where access speed is extremely important. A lower cost-per-byte makes DRAM attractive whenever large amounts of RAM are required. Many embedded systems include both types: a small block of SRAM (a few kilobytes) along a critical data path and a much larger block of DRAM for everything else.

## Types of ROM

Memories in the ROM family are distinguished by the methods used to write new data to them (usually called programming), and the number of times they can be rewritten. This classification reflects the evolution of ROM devices from hardwired to programmable to erasable-and-programmable. A common feature of all these devices is their ability to retain data and programs forever, even during a power failure.

The very first ROMs were hardwired devices that contained a preprogrammed set of data or instructions. The contents of the ROM had to be specified before chip production, so the actual data could be used to arrange the transistors inside the chip. Hardwired memories are still used, though they are now called “masked ROMs” to distinguish them from other types of ROM. The primary advantage of a masked ROM is its low production cost. Unfortunately, the cost is low only when large quantities of the same ROM are required.

One step up from the masked ROM is the PROM (programmable ROM), which is purchased in an unprogrammed state. If you were to look at the contents of an unprogrammed PROM, you would see that the data is made up entirely of 1's. The process of writing your data to the PROM involves a special piece of equipment called a device programmer. The device programmer writes data to the device one word at a time by applying an electrical charge to the input pins of the chip. Once a PROM has been programmed in this way, its contents can never be changed. If the code or data stored in the PROM must be changed, the current device must be discarded. As a result, PROMs are also known as one-time programmable (OTP) devices.

An EPROM (erasable-and-programmable ROM) is programmed in exactly the same manner as a PROM. However, EPROMs can be erased and reprogrammed repeatedly. To erase an EPROM, you simply expose the device to a strong source of ultraviolet light. (A window in the top of the device allows the light to reach the silicon.) By doing this, you essentially reset the entire chip to its initial-unprogrammed-state. Though more expensive than PROMs, their ability to be reprogrammed makes EPROMs an essential part of the software development and testing process.

## Hybrid Types

As memory technology has matured in recent years, the line between RAM and ROM has blurred. Now, several types of memory combine features of both. These devices do not belong to either group and can be collectively referred to as hybrid memory devices. Hybrid memories can be read and written as desired, like RAM, but maintain their contents without electrical power, just like ROM. Two of the hybrid devices, EEPROM and flash, are descendants of ROM devices. These are typically used to store code. The third hybrid, NVRAM, is a modified version of SRAM. NVRAM usually holds persistent data.

EEPROMs are electrically-erasable-and-programmable. Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light. Any byte within an EEPROM may be erased and rewritten. Once written, the new data will remain in the device forever-or at least until it is electrically erased. The primary tradeoff for this improved functionality is higher cost, though write cycles are also significantly longer than writes to a RAM. So you wouldn't want to use an EEPROM for your main system memory.

Flash memory combines the best features of the memory devices described thus far. Flash memory devices are high density, low cost, nonvolatile, fast (to read, but not to write), and electrically reprogrammable. These advantages are overwhelming and, as a direct result, the use of flash memory has increased dramatically in embedded systems. From a software viewpoint, flash and EEPROM technologies are very similar. The major difference is that flash devices can only be erased one sector at a time, not byte-by-byte. Typical sector sizes are in the range 256 bytes to 16KB. Despite this disadvantage, flash is much more popular than EEPROM and is rapidly displacing many of the ROM devices as well.

The third member of the hybrid memory class is NVRAM (non-volatile RAM). Nonvolatility is also a characteristic of the ROM and hybrid memories discussed previously. However, an NVRAM is physically very different from those devices. An NVRAM is usually just an SRAM with a battery backup. When the power is turned on, the NVRAM operates just like any other SRAM. When the power is turned off, the NVRAM draws just enough power from the battery to retain its data. NVRAM is fairly common in embedded systems. However, it is expensive-even more expensive than SRAM, because of the battery-so its applications are typically limited to the storage of a few hundred bytes of system-critical information that can't be stored in any better way.

Table 1 summarizes the features of each type of memory discussed here, but keep in mind that different memory types serve different purposes. Each memory type has its strengths and weaknesses. Side-by-side comparisons are not always effective.

| <i>Type</i>       | <i>Volatile?</i> | <i>Writeable?</i>              | <i>Erase Size</i> | <i>Max Erase Cycles</i>     | <i>Cost (per Byte)</i>     | <i>Speed</i>                      |
|-------------------|------------------|--------------------------------|-------------------|-----------------------------|----------------------------|-----------------------------------|
| <b>SRAM</b>       | Yes              | Yes                            | Byte              | Unlimited                   | Expensive                  | Fast                              |
| <b>DRAM</b>       | Yes              | Yes                            | Byte              | Unlimited                   | Moderate                   | Moderate                          |
| <b>Masked ROM</b> | No               | No                             | n/a               | n/a                         | Inexpensive                | Fast                              |
| <b>PROM</b>       | No               | Once, with a device programmer | n/a               | n/a                         | Moderate                   | Fast                              |
| <b>EPROM</b>      | No               | Yes, with a device programmer  | Entire Chip       | Limited (consult datasheet) | Moderate                   | Fast                              |
| <b>EEPROM</b>     | No               | Yes                            | Byte              | Limited (consult datasheet) | Expensive                  | Fast to read, slow to erase/write |
| <b>Flash</b>      | No               | Yes                            | Sector            | Limited (consult datasheet) | Moderate                   | Fast to read, slow to erase/write |
| <b>NVRAM</b>      | No               | Yes                            | Byte              | Unlimited                   | Expensive (SRAM + battery) | Fast                              |

*Table 1. Memory Type Characteristics*

This article was published in the May 2001 issue of [Embedded Systems Programming](#). If you wish to cite the article in your own work, you may find the following MLA-style information helpful:

Barr, Michael. "Memory Types" [Embedded Systems Programming](#), May 2001, pp. 103-104.