



Why Phase Change Memory (PCM) Makes Sense for Reference Designs

Background

Add new firmware, with less waiting

Erase cycles are a big waste of time

PCM: No erase needed

Understanding the benefits of PCM

Why Phase Change Memory (PCM) Makes Sense for Reference Designs

Hardware reference designs help engineers accelerate the development of next-generation solutions based on the latest technologies. Phase Change Memory (PCM), a new class of non-volatile memory, offers characteristics that deliver compelling advantages for storing updatable firmware in a reference design. Reference design and chipset vendors should consider adding PCM support to their products now to help their customers understand this leading-edge technology.

Background

As Figure 1 shows, Phase Change Memory (PCM) combines some of the best qualities of DRAM and non-volatile memory (NVM). Like DRAM, the low latency and bit-alterability of PCM make it a compelling choice for direct execution of firmware code in a reference design. Similar to DRAM, PCM imposes no requirement to erase prior code or data prior to writing updated code. And since PCM is non-volatile, there is no refresh power required to maintain code or data.

Attributes	PCM	DRAM	NAND	NOR	EEPROM
Bit Alterable	Green	Green	Red	Red	Green
Non-volatile	Green	Red	Green	Green	Green
Cost	Yellow	Green	Red	Yellow	Yellow
Read Speed	Yellow	Green	Yellow	Yellow	Red
Write Speed	Green	Yellow	Yellow	Yellow	Red

Figure 1. Phase Change Memory (PCM) is a new class of non-volatile memory that combines many of the best attributes of NOR flash, NAND flash and DRAM.

Add new firmware, with less waiting

Hardware reference designs help engineers accelerate the development of next-generation solutions based on the latest technologies. New computing and communications products often include enhanced firmware that enables differentiated feature sets and fast performance, with this code typically loaded into boot ROM.

An ever-present issue for engineers is that the normal course of product development often involves many firmware updates before the new product is ready for release. These firmware updates involve recompilation and “re-flashing” of code. The re-flash process requires software to manage two discrete steps: 1) erasing the original code from the non-volatile memory, and 2) reprogramming the memory device with the new software image. Most reference designs alert users that re-flashing is in progress with a blinking LED status light. When describing the firmware update procedure, the typical reference design manual will warn users not to turn off or interrupt AC power to the device during the software update process or risk corrupting the memory and damaging the device.

Engineers often find themselves engaged in the iterative process of rewriting bits of firmware code, recompiling and downloading it to onboard NOR flash storage. Spending several minutes of downtime waiting for each re-flash to complete is not only tedious, it wastes valuable engineering resources.

Erase cycles are a big waste of time

If we can assume that the interface used for the firmware update is a relatively fast link such as USB, we discover that much of the time spent watching that blinking LED is traceable to the time required to erase non-volatile memory during the re-flashing process.

As shown in Figure 2, updating a 32MB firmware image with code written to a typical 256Mb NVM device (NOR flash) involves a typical erase time of 3.33 minutes, compared to typical programming time of just 34 seconds.

Phase Change Memory (PCM) for reference designs

Roughly 85 percent of the time spent re-flashing firmware is spent erasing the NOR flash device. This lost-time bottleneck becomes all the more significant when the frequency of firmware updates during the course of a typical project is considered. In design lab testing, the maximum observed total erase time for a 32MB firmware image was more than 16 minutes.

	Erase	Programming	Total
Time (typical)	3.33 minutes	34 seconds	3.90 minutes
Time (maximum)	16.67 minutes	2.37 minutes	19.04 minutes

Figure 2. Time for each step of the firmware re-flash process, observed with a 256Mb NOR flash memory device (Numonyx® Axcell™ M29EW). Erase time is a bottleneck, accounting for about 85 percent of the total time required to re-flash firmware. (Source: Numonyx)

PCM: No erase needed

Phase Change Memory (PCM) is a class of non-volatile memory devices that employ reversible phase change in materials to store information. PCM relies on differences in the electrical resistivity exhibited by the different phases of the material.

PCM has some interesting characteristics which make it ideal for storing updatable firmware in a reference design:

- Like flash memory, PCM is a non-volatile memory technology.
- Unlike other floating-gate memory types, PCM is bit-alterable. This means that information stored in PCM can be switched from a 1 to 0 or from 0 to a 1 without a separate and time-consuming erase step.
- PCM features fast random access times. This enables code to execute directly from memory, without an intermediate copy to RAM. The read latency of PCM is similar to single-bit-per-cell NOR flash, while the read performance can match DRAM.
- PCM can achieve write speeds comparable to NAND flash, but with lower latency, since there is no separate erase step needed. By comparison, NOR flash features moderate write speeds but long erase times.

Understanding the benefits of PCM

The core value of PCM is that it will outlast electrostatic memory architectures, so developers will eventually need to become familiar with the technology. Adding PCM to a reference design provides an excellent way for design engineers to evaluate the advantages of the technology in actual applications, without having to make a commitment to volume manufacturing.

Compared to existing non-volatile memory, PCM has clear advantages in terms of latency and endurance (both for reads and writes) in addition to providing the ability to execute code directly from non-volatile memory.

PCM also provides the opportunity to implement low latency storage that can help developers simplify their software architecture by avoiding the necessity for caching algorithms to move code into RAM. In addition to eliminating extra architectural steps, PCM is byte-alterable, avoiding the need to copy full sectors when only bytes are necessary.


By enabling direct memory mapped, NVM interfaces, PCM not only speeds up programming, it can also provide opportunities to rethink I/O and storage architectures.

To illustrate, if random latency for PCM is normalized to one day, the latency of a NAND-based solid state drive (SSD) is 17 days, and storage-class PCM is 30 minutes. As a replacement for DRAM, PCM has no requirement for refresh power, which supports improved memory power management.

If you are a reference design or chipset vendor, placing a directly addressable non-volatile RAM interface in your product makes sense. Now is the time for your customers to get acquainted with the latency and endurance advantages of PCM as a leading-edge memory technology.

numonyx.com

Copyright © 2009 Numonyx B.V. All rights reserved. Numonyx, the Numonyx logo and Axcell are trademarks or registered trademarks of Numonyx B.V. or its subsidiaries in other countries. *Other names and brands may be claimed as the property of others.

1009/NLH/BMG/PDF 109138-001 Please recycle 



numonyx™