

Memory Cycling Capability: Sorry, It's Complex.

Mark Webb, MKW Ventures Consulting, LLC.

I addressed this in a couple presentations at Flash Memory Summit but questions on cycling performance have grown based on 3D Xpoint and QLC. I will highlight why “how many times can you cycle it?” is 10x more complicated than most people realize.

Key items on cycling performance (Random thoughts.... there is a lot more detail)

- Cycling specifications are based on the criteria used to define success. Rarely do bits fail “dead”. They usually are slow to program or erase, read disturb easily, can’t hold charge for a long time, etc. A device spec could be 800us program time and 100us read time and data retention of 5 years. If I change that to 1200us program, 200uS, and 1 year retention, I could hypothetically increase the cycling capability from 3000 cycles to 10,000 cycles. And what if I changed data retention to 1 hour ???
- Cycling specs are based on allowed fail rate and die level/system level error correction to deal with it. This is published in detailed bit error rate data (BER). Basic NAND allows us to tolerate thousands of blocks failing to cycle, and millions of bits that sometimes read erratically. I can always increase redundant blocks and add more ECC capability. These “tricks” prevent the end customer from seeing errors. A QLC cell seems less reliable than TLC or MLC but it is possible to make it MORE reliable to the end user with error correction.
 - In addition to all of the error correction, we can adjust for end user fail rates as well. Is 2% AFR (annual fail rate) OK? 0.7%? 0.2%? Are you willing to pay 2x the price for the 0.2% fail rate?
- Based on the above statistics and BER, the size of the array matters. This is very important when we talk about emerging memory. Without error correction, my test chip (a few bits) for MRAM or ReRAM or 3DXP may cycle 10M times. When I put 64K bits in an array it might last 1M times. A 1Gbit array will last 100K times... and then we get into error correction and fail rates.
- Theory vs Actual. I have seen multiple papers saying that MRAM has infinite endurance or something like 1E12 cycles. But MRAM is a real product now and with that comes real cycling numbers. I don't believe any MRAM product is spec'd with 1E12 cycles. Think 6 order of magnitude less.... best case. There are ways to manage the cycling up and down. The capability starts at 1E12 and then starts to drop when you actually make real devices. This is the problem with universal memory claims because ... wait for it ... DRAM and SRAM can ACTUALLY last 1E12 cycles in real products. No emerging memory is even close.
- So, if we just take NAND and things we have seen over the years as examples of the above.
 - NAND theoretically can cycle 1E12... charge trapping is a non-ideality that I will ignore
 - A small array (1K) can cycle >1M times without any fails.
 - I can buy a 1Gbit chip today that can cycle >100K times (50nm SLC with ECC).
 - Planar TLC can cycle 3K times.... If I allow 1% failure, it last 10K cycles, the average (median) bit in that array lasts >30K cycles.
 - If I slow down the program and read timing dramatically, I can make a TLC 3K part last 10K+

- And hypothetically, If I build a QLC SSD with massive redundancy and overprovisioning (think RAID), I can have 10 drive writes per day for 10 years. Looks like 36,500 cycles to end user.. but it is not

What does all this mean? The memory companies (and some SSD companies) know all of this and have details on tradeoff and how to manage them... Sorry, it's complex. Ask the experts how it works in a given application. Asking "how many cycles does that last?" or saying "QLC isn't good enough" may not be useful

We can discuss specific questions to ask (BER, FIT, AFR, DPM, etc) and compare NAND, DRAM, MRAM, ReRAM, 3DXP in more detail. We published estimated 3DXP cycling performance numbers. Call for more information

next blog, I will answer the question "is Schrodinger's cat dead or alive?". As I tell my kids when we discuss at the dinner table "It's complex"

Mark Webb
www.mkwventures.com