Systémy odolné proti poruchám
Ochrana pamětí Flash

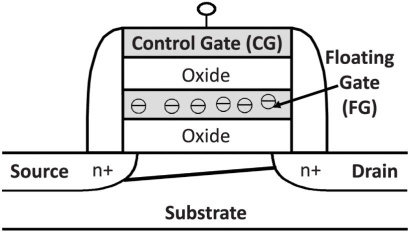
Martin Havlík

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

28 April, 2025

Flash memory cells – NAND flash

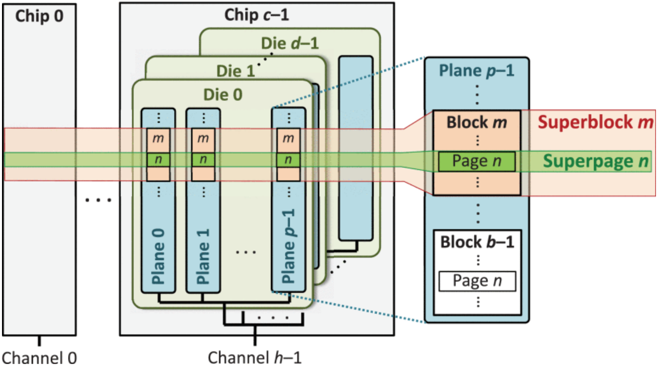
- modified MOSFET (most commonly), trapped electric charge changes V_T
- {Single, Multi (2)...} Level Cell – SLC, MLC, TLC...
- threshold voltage windows, prob. dens. hills between read ref. volts
- FG insulated → non-volatile
- Channel Hot Electron (CHE) injection & Fowler-Nordheim Tunneling (FNT) – quantum mechanics



String-like structures of cells sharing contacts (vs NOR flash) – higher density → cheaper but less reliable and more error-prone

SSD memory hierarchy organization

E.g. 4-16 chips, up to 16 dies per chip, 1-4 planes, 100s-1000s blocks, 2D array 100s rows of flash cells. Super- across chips & planes (same ID).

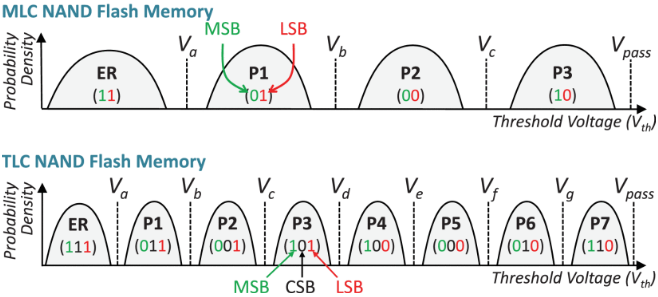


SSD = a group of chips connected via channels to a controller (ECC engine, scrambler, compression, DRAM manager & buffers etc.)

Program-erase cycles

1. Program-erase (P/E, PE)
   - program (write) only 1 → 0 with page granularity, smallest R/W unit
   - erase 0 → 1 only block granularity (made of pages)
   - disparity ⇒ garbage collection
   - degrades individual cells – trapped electrons → cycling noise
   - leads to raw bit errors needing to be corrected with ECC
   - up to an uncorrectable state → limited lifetime!
2. Example stats
   - older SLC NAND 150 000 P/E, 5x-nm MLC around 10 000
   - newer 1x-nm MLC NAND circa 3000, TLC only around 1000

Threshold voltage distribution between reference voltages



SSD Controller I.

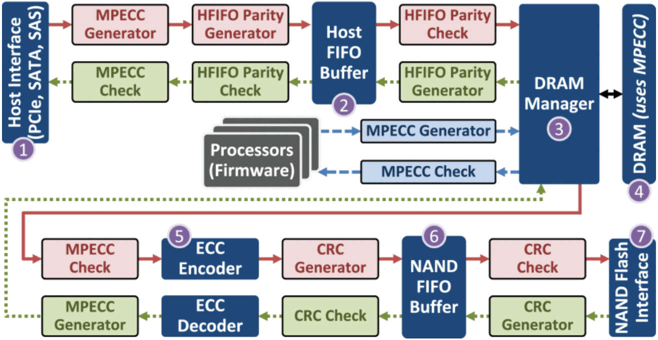
1. Flash Translation Layer (FTL)
   - LA → PA transparent mapping, GC (fewest valid pages)
   - wear leveling – reduce heterogeneity of block wearout
2. Flash Reliability Management
   - refresh (in-place, remapping-based), hot data
   - read-retry to mitigate voltage shifts, latency!
3. Compression
   - e.g. LZ77/LZ78, optional (can be already compressed/encrypted)
4. Data Scrambling and Encryption
   - errors data dependent → randomly distributed 1s & 0s
   - LSFR seeded with LA $\oplus$ data $=$ scrambled data
   - self-encrypting drive (SED) typically using AES
5. Bad Block Management
   - process variation, uneven wearout, few blocks high raw bit error rate (RBER) → avoid usage
   - original & growth bad blocks (OBBs & GBBs), reserved blocks
   - expected less than 2% OBBs

Page smallest R/W unit, smallest erase by blocks!

SSD Controller II.

6 Data Path Protection

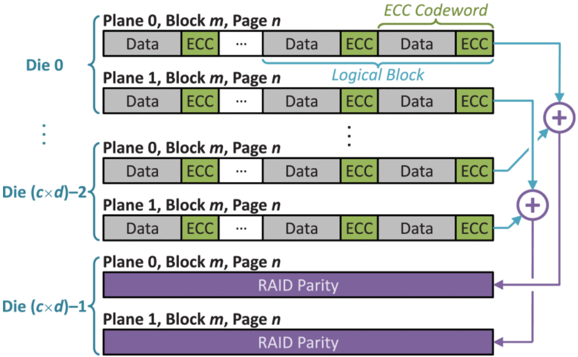
- parity checks for SRAM (HFIFO) & DRAM (MPECC)
- weaker ECC than (BCH & LDPC) for NAND (↓ expected error rate)
- host → flash, flash → host, firmware metadata



memory protection ECC, host FIFO, DRAM buffers before writing, check & discard MPECC, encode into ECC, generate CRC, write to NAND FIFO, CRC check & NAND write. DRAM buffering + metadata (mapping table etc.) – uses MPECC too

SSD Controller III.

7. Superpage-Level Parity
   - protect from ECC failures within chip/plane
   - $\oplus$ all pages in plane 0, write to plane 0 parity die
   - OS accesses with logical block (LB) granularity, typ. 4kB
   - hidden GBB or ECC failure
   - LB read failure, $\oplus$ LBs from all other dies in superpage (those must read correctly!)



Write amplification (WA) & over-provisioning (OP)

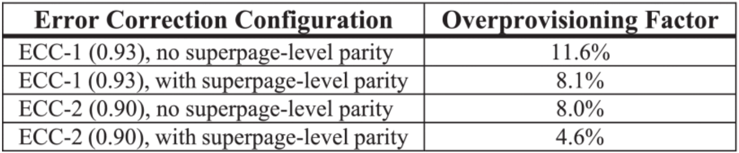
Granularity mismatch between program (page) and erase (block)

- writing to a block needs it erased! (can only write 1 → 0)
- garbage collecting moves valid pages to free up block for erasing
- results in additional flash writes = write amplification

Over-provisioning (inaccessible extra capacity) decreases WA – trade-off

- ECC strength requires more space → less for OP – another trade-off!

More ECC more power consumption!

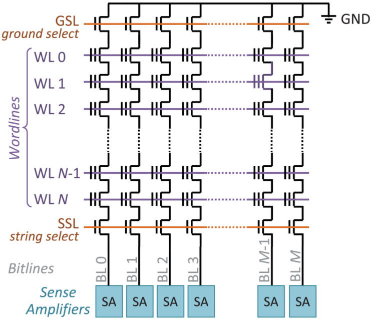


Error Correction Configuration	Overprovisioning Factor
ECC-1 (0.93), no superpage-level parity	11.6%
ECC-1 (0.93), with superpage-level parity	8.1%
ECC-2 (0.90), no superpage-level parity	8.0%
ECC-2 (0.90), with superpage-level parity	4.6%

Assume 20% extra phys space over advertised (2TB → 2.4TB)
(coding rate) – % non-ECC data → lower = higher redundancy
factor – amount of SSD space left for overprovisioning

Error characterization and mitigation

1. Cell-to-cell program interference and shadow program sequencing
   - programming adjacent wordlines interference → 2 (+) step zig-zag LSB (CSB) MSB program ⇒ minimize interference on fully programmed wordline
2. Neighbor cell Assisted error Correction (NAC)
   - V_T shift correlated to values in adjacent wordlines → know thy neighbor and adjust reference for reading
3. Refreshing
   - retention & read disturbances, long term accumulating raw bit errors → remap, in-place x fixed intervals, adaptive (wearout, temperature)
   - worse with increasing PE cycles
4. Read-retry
   - on ECC failure to correct, retry reading with slightly different ref. voltage



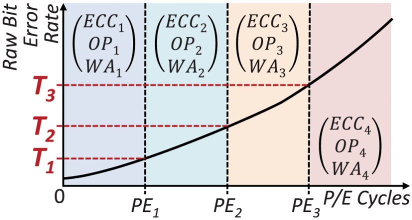
Adaptive error mitigation

1. Multi-rate ECC
    - initial weaker ECC, more for OP, less WA
    - PE interval based, measure Raw Bit Error Rate (RBER), threshold, switch to stronger ECC (decode with 1, encode with 2)

   | User data | OP space <-|-> ECC |

2. Dynamic Cell Levels (block granularity)
    - voltage distribution hills widen over time, overlap
    - downgrade e.g. TLC to MLC – ↑ margins
    - read-hot data in special downgraded blocks to minimize read disturb

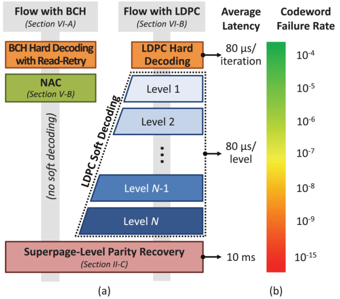
3. Slower program erase operations
    - when write request throughput is low
    - slower more precise programming → reduce oxide degradation
    - higher latency – can perform during GC (SSD already idle)

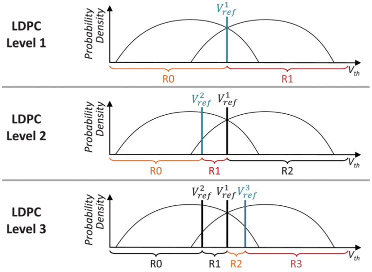


BCH & LDPC ECC I.

1. RBER overview
   - BCH & LDPC, able to correct around 10^{-3} RBER
   - data to host with post-correction error rate target 10^{-15} (JEDEC standard)
2. BCH decode
   - BCH decode → read-retry, max attempts → NAC, max attempts → superpage parity (most time expensive!) → uncorrectable
3. LDPC decode – more levels more latency!
   - no read-retry, do soft decoding instead
   - soft information: prob. the cell contains 0/1 – from multiple reads w/ diff. ref. voltages
   - log likelihood ratio (LLR)
   - each level adds new info about cell (vs hard which replaces information) → increases strength of error correction

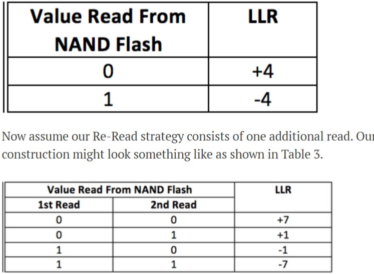
$$LLR = log\frac{P(x=0|V_{th})}{P(x=1|V_{th})}, LLR^{Rj}_{level}$$





BCH & LDPC ECC II.

1. Computing LLR (estimation)
   - model a cell as a communication channel → expected signal & additive noise due to errors
   - $V_{th} \sim_{model}$ Gaussian distribution
   - PE cycling noise $\sim_{model}$ additive white Gaussian noise (AWGN)
   - LLR estimation either during runtime or precomputed in tables by SSD manufacturer
   - online & offline empiric training based on PE cycle count, retention time, read disturb count etc.
2. Number of soft decoding levels
   - latency!
   - e.g. 5-level up to 480 µs
   - trade-off, helps to reduce triggering expensive superpage-level parity recovery
   - diminishing returns



Value Read From NAND Flash	LLR
0	+4
1	-4

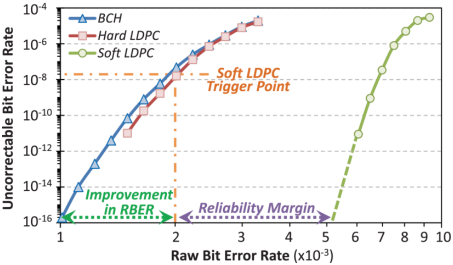
Now assume our Re-Read strategy consists of one additional read. Our construction might look something like as shown in Table 3.

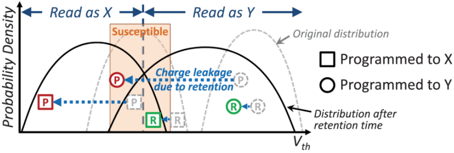
Value Read From NAND Flash		LLR
1st Read	2nd Read	
0	0	+7
0	1	+1
1	0	-1
1	1	-7

Example source: https://www.eetimes.com/soft-decoding-in-ldpc-based-ssd-controllers/

BCH & LDPC III.

1. Error correction strength comparison
   - coding rates: 0.935 BCH, 0.936 LDPC
   - hard decoding very similar, no benefit of LDPC
   - assume UBER target 10^{-16}, then BCH can correct up to RBER 1.0×10^{-3}, soft LDPC up to 5.0×10^{-3} (latency!)
   - as SSD wears out, RBER increases, upon some threshold, e.g. 2.0×10^{-3} we can switch to soft LDPC decoding to further maintain low UBER
   - lifetime vs read latency trade-off
2. Retention Failure Recovery (RFR)
   - pre-ECC for end-of-lifetime SSD reads
   - read fails (uncorrectable), read-retry, record magnitude of V_{th} shift, determine susceptible vs resistant
   - susceptible likely Y, resistant likely X
   - can reduce RBER of failed pages by up to 50%





Main literature source

[1] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, “Error characterization, mitigation, and recovery in flash-memory-based solid-state drives,” Proceedings of the IEEE, vol. 105, no. 9, pp. 1666–1704, 2017. DOI: 10.1109/JPROC.2017.2713127.

Other related topics. . .

1. Differential ECC for 3D NAND[1]
   - 2nd LDPC with lower code rate (stronger ECC) for read-hot data
   - trade-off, stronger ECC less read retry, energy cost, parity space cost
   - identification, re-coding, extra info & book-keeping overhead
2. FTRM: A Cache-Based Fault Tolerant Recovery Mechanism[2]
   - Cache Mapping Table (CMT) for FTL, reconstruction on recovery
   - Out Of Bounds (OOB) page area validity flag, access counter
3. ECC Caching[3]
   - adaptively increase ECC protection level, minimize induced WA
   - overlong ECC across pages boundary → 2 r/w ops per 1 r/w request
   - extra check bits (ECB) in $ CAM[PPN] (content-addressable mem.)

[1] https://doi.org/10.1145/3566097.3567853

[2] https://doi.org/10.3390/electronics9101581

[3] https://doi.org/10.1007/s10836-023-06075-6