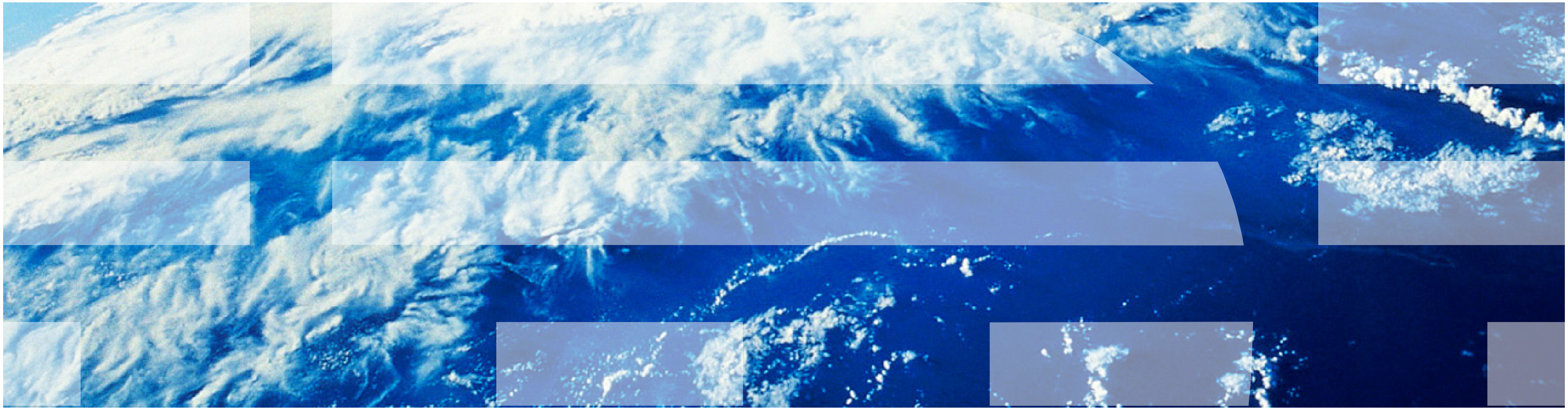
Sub-block Wear-leveling for NAND Flash

Roman Pletka, Xiao-Yu Hu, Ilias Iliadis, Roy Cideciyan, Theodore Antonakopoulos

Non-Volatile Memories Workshop, San Diego, March 6-8 2011



Work done in collaboration with University of Patras

Overview

- Motivation
  - – Experimental results from SLC and MLC NAND Flash memories
  - – Block vs. Page RBER
- Sub-block wear-leveling schemes
  - – Content-aware Wear-leveling
  - – Wear-leveling using page tracking
  - – Device-dependent wear-leveling
  - – Combinations of sub-block wear-leveling schemes
- Conclusion & Outlook

Introduction

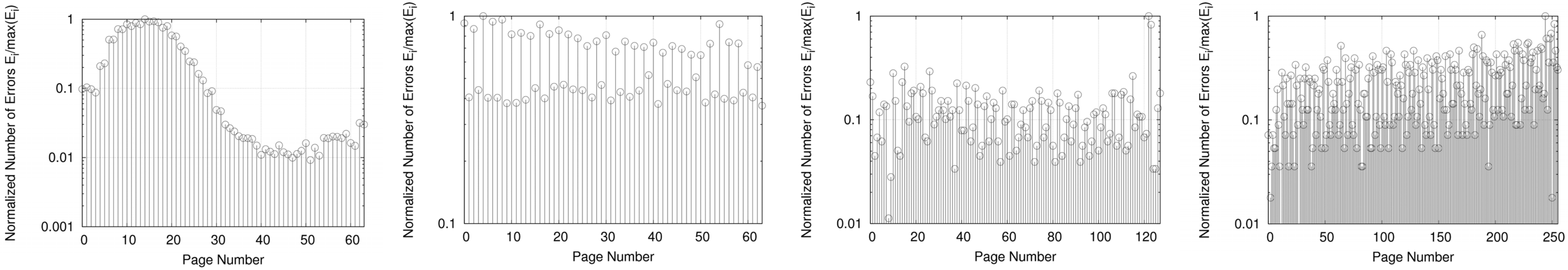
- Flash Translation Layer: Speeds up writes (no write-in-place) and better spreads wear over blocks.
- Garbage collection: Reclaims free space from invalid pages of a block by relocating still valid pages to a free block. Often combined with wear leveling.
- Existing wear-leveling schemes:
  - Dynamic wear-leveling:
    Allows to equalize wear among blocks with dynamic data (i.e., overwritten often), except for blocks with static data that never get touched.
  - Static wear-leveling:
    Periodically force relocation of static data to aged blocks from the free block pool.

  => But these schemes consider only wear of blocks, not individual pages.
- Other methods to increase the Flash lifetime:
  - ECC:
    - Typical recommendations by manufacturers: SLC: 1-4 bit/540B, MLC: 8-12 bit/540B
    - ECC schemes taking into account device statistics [1].
  - Multi-write coding: Reprogram without erasing using coding [2]
  - Write reduction using compression [3] or deduplication [4].

[1] – Eitan Yaakobi et al., Error Characterization and Coding Schemes for Flash Memories, Globecom 2010
[2] – Ashish Jagmohan et al., Write Amplification Reduction in NAND Flash through Multi-write Coding, MSST 2010
[3] – SandForce DuraWrite, http://www.sandforce.com
[4] – Feng Chen et al., CAFTL: A Content-aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives, FAST 2011

Experimental Results – Endurance

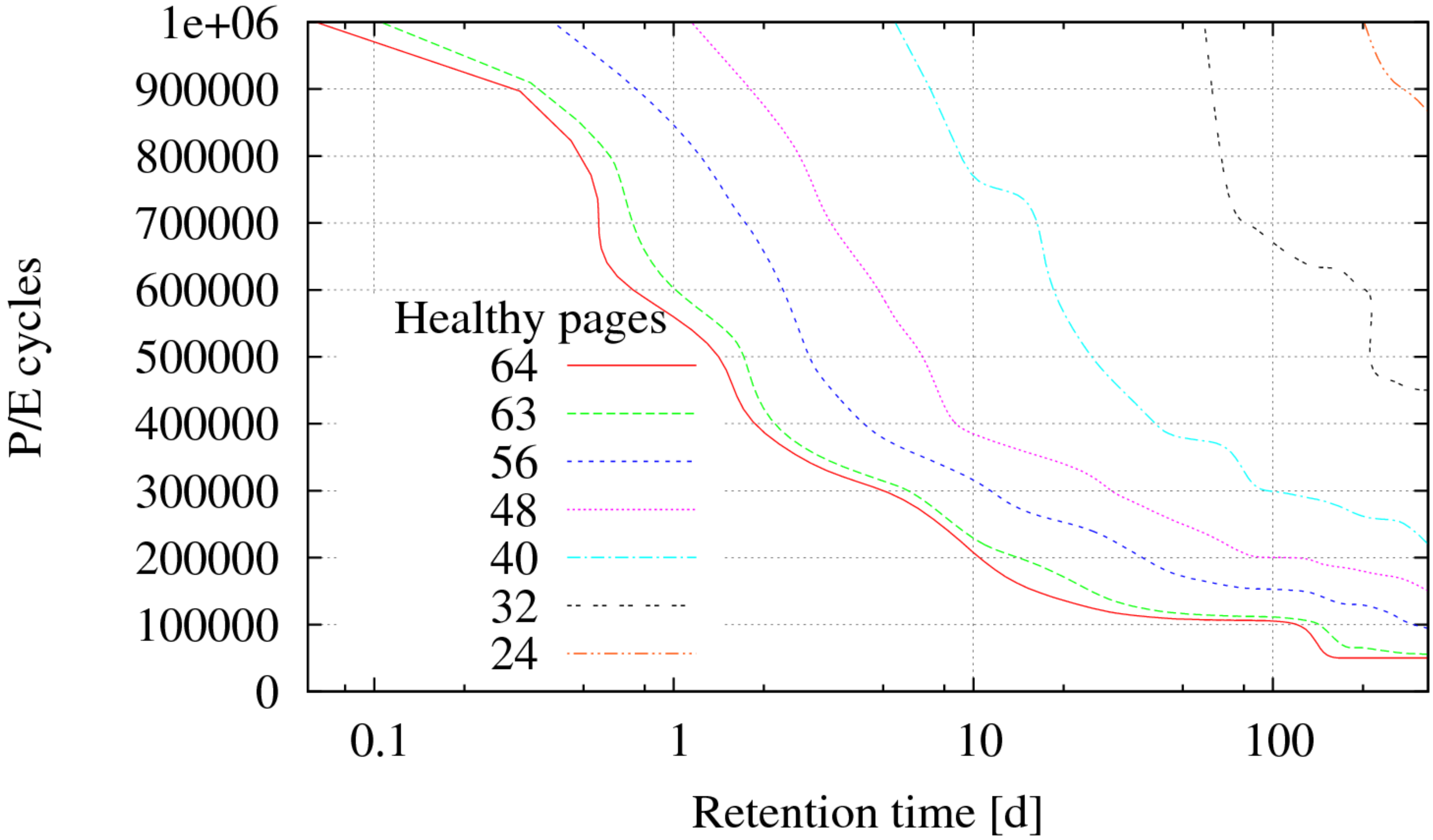
- Normalized number of errors for Flash pages within a single block for different NAND Flash devices, shown for the exercised P/E cycles.



Device	X	Y	Y	Z
Flash Type	SLC	SLC	MLC	MLC
Page Size [B]	2112	4314	4314	4320
Block Size [pages/block]	64	64	128	256
Capacity [Gbits]	4	8	16	256
Endurance [P/E cycles]	100k	100k	10k	5k
Exercised P/E cycles	1M	1M	100k	100k

Experimental Results – Retention

- Number of healthy pages (i.e., with RBER less than 10^{-3}) as a function of retention time and P/E cycles (device X).

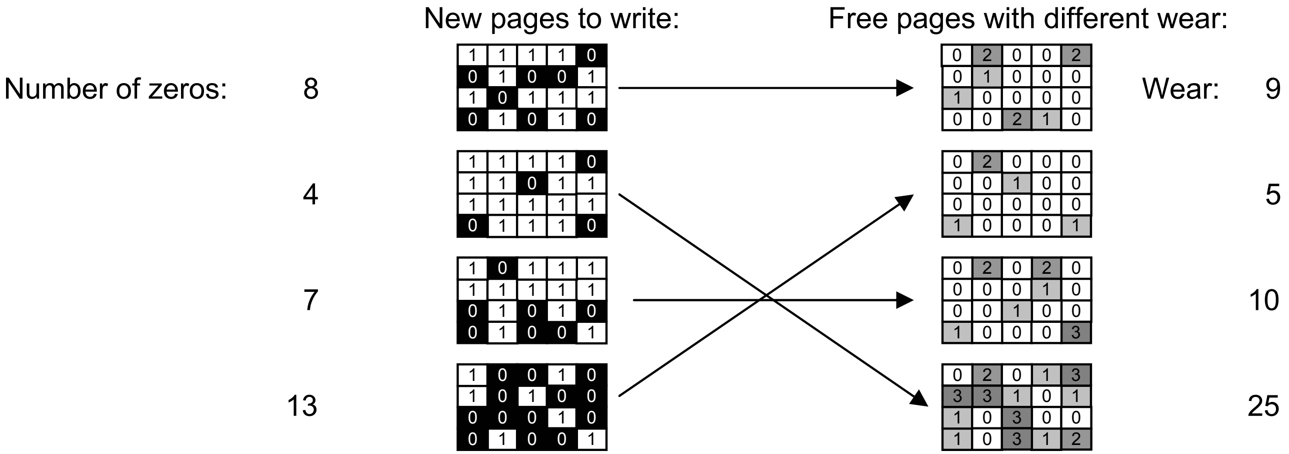


Conclusion from Experimental Results

- Significant RBER differences between pages within the same block have been found.
- The error pattern is similar for blocks in the same device, but absolute error rate may vary.
  - This stems from tolerances in the manufacturing process (e.g., slight variations in the thickness of the tunnel oxide).
- NAND Flash memories of the same type (manufacturer, technology) show similar error patterns.
  - This comes from the internal structure of NAND Flash memories.
- Measuring RBER:
  NAND Flash RBER should be measured at the page level, not over a full Flash block.
- Other error sources:
  - Program and read disturbs are best addressed using ECC.
  - Chip failures best addressed with higher level redundancy (e.g., RAID-like schemes).

Content-aware Sub-block Wear-leveling

- Equally distribute the number of 0’s written among fixed-sized data segments according to the current wear of each Flash segment.
  - Because the tunnel oxide of a memory cell is stressed when electrons are pushed into (writing a 0) or out of (erasing a 0) the floating gate, which leads to irreparable defects.
- Segment granularity: Sub-page, page, block, …
  - Generally, the finer the granularity, the better the wear is spread.
  - For practical reasons page granularity is the most promising.
- How does it work?
  - The number of 0’s in a new data segment to be written are counted. Use this information for tracking wear and/or placement decision.
  - For MLC devices, the intermediate levels can be weighted accordingly.
- Implementation alternatives:
  - Tracking the wear of each data segment. -> Significant amount of additional meta-data.
  - Write segments with similar wear to the same block -> Requires a set of blocks from which data segments are allocated (more complex write page allocator).



New pages to write:

Number of zeros: 8

1	1	1	1	0
0	1	0	0	1
1	0	1	1	1
0	1	0	1	0

4

1	1	1	1	0
1	1	0	1	1
1	1	1	1	1
0	1	1	1	0

7

1	0	1	1	1
1	1	1	1	1
0	1	0	1	0
0	1	0	0	1

13

1	0	0	1	0
1	0	1	0	0
0	0	0	1	0
0	1	0	0	1

Free pages with different wear:

Wear: 9

0	2	0	0	2
0	1	0	0	0
1	0	0	0	0
0	0	2	1	0

5

0	2	0	0	0
0	0	1	0	0
0	0	0	0	0
1	0	0	0	1

10

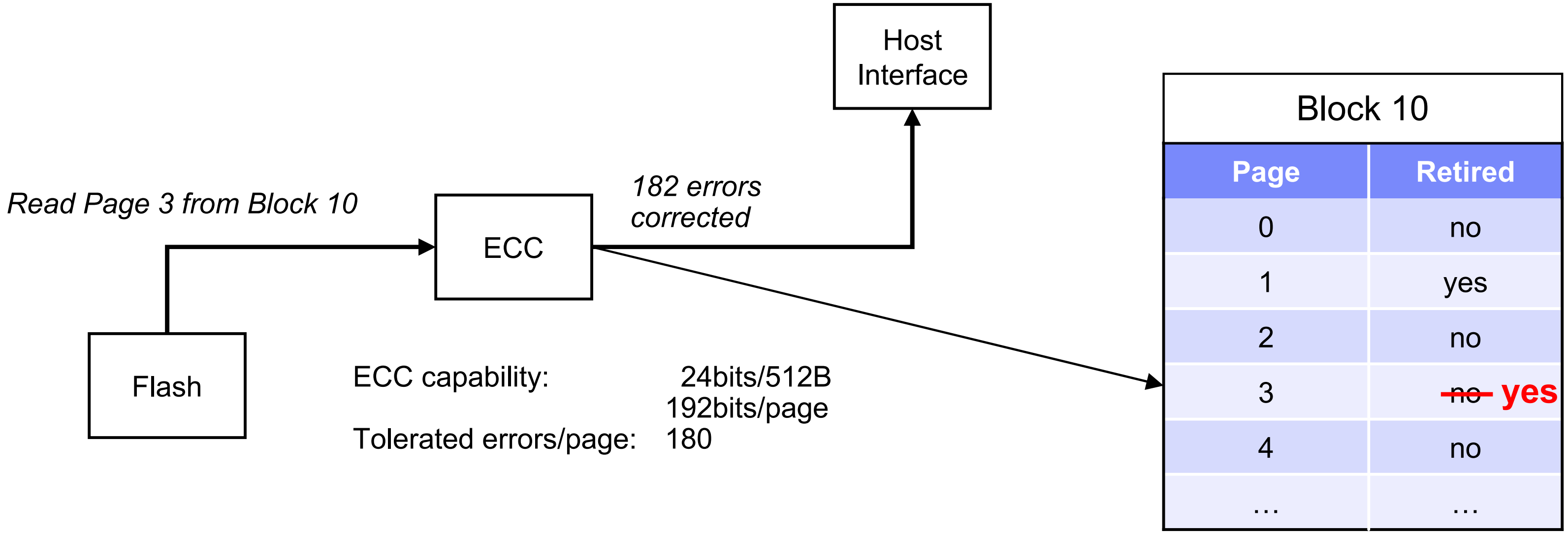
0	2	0	2	0
0	0	0	1	0
0	0	1	0	0
1	0	0	0	3

25

0	2	0	1	3
3	3	1	0	1
1	0	3	0	0
1	0	3	1	2

Sub-block Wear-leveling using Page Tracking

- Use the number of errors that have been corrected by ECC when reading a page combined with a page error threshold (i.e., maximum number of tolerated errors per page).
- Retire pages individually when the page error threshold is hit. Remaining pages in the Flash block are still used.
- Tracking valid pages requires one additional bit per page of meta-data and allows to utilize a page to its utmost limit. However, the page must be periodically read to determine the current number of corrected errors, due to retention errors.
  - Frequency of reads increases with P/E cycles.



ECC capability: 24bits/512B
192bits/page

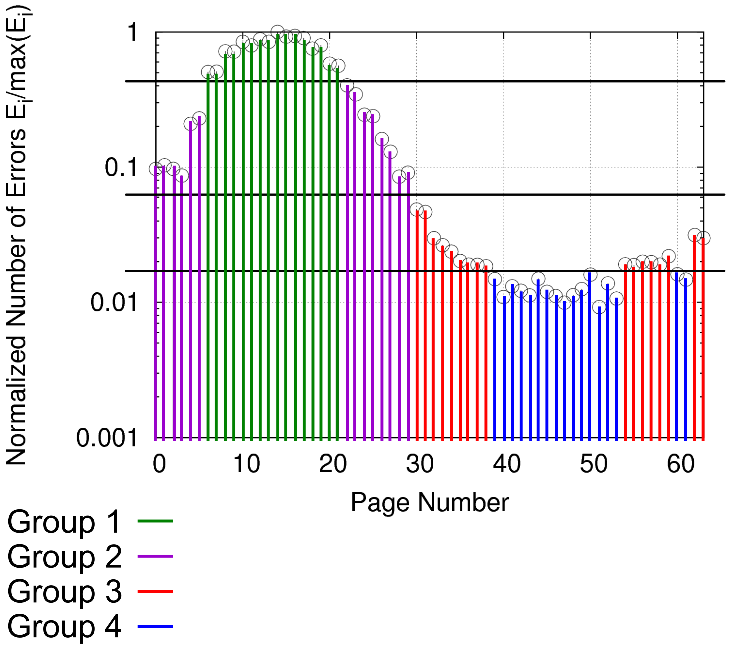
Tolerated errors/page: 180

Block 10

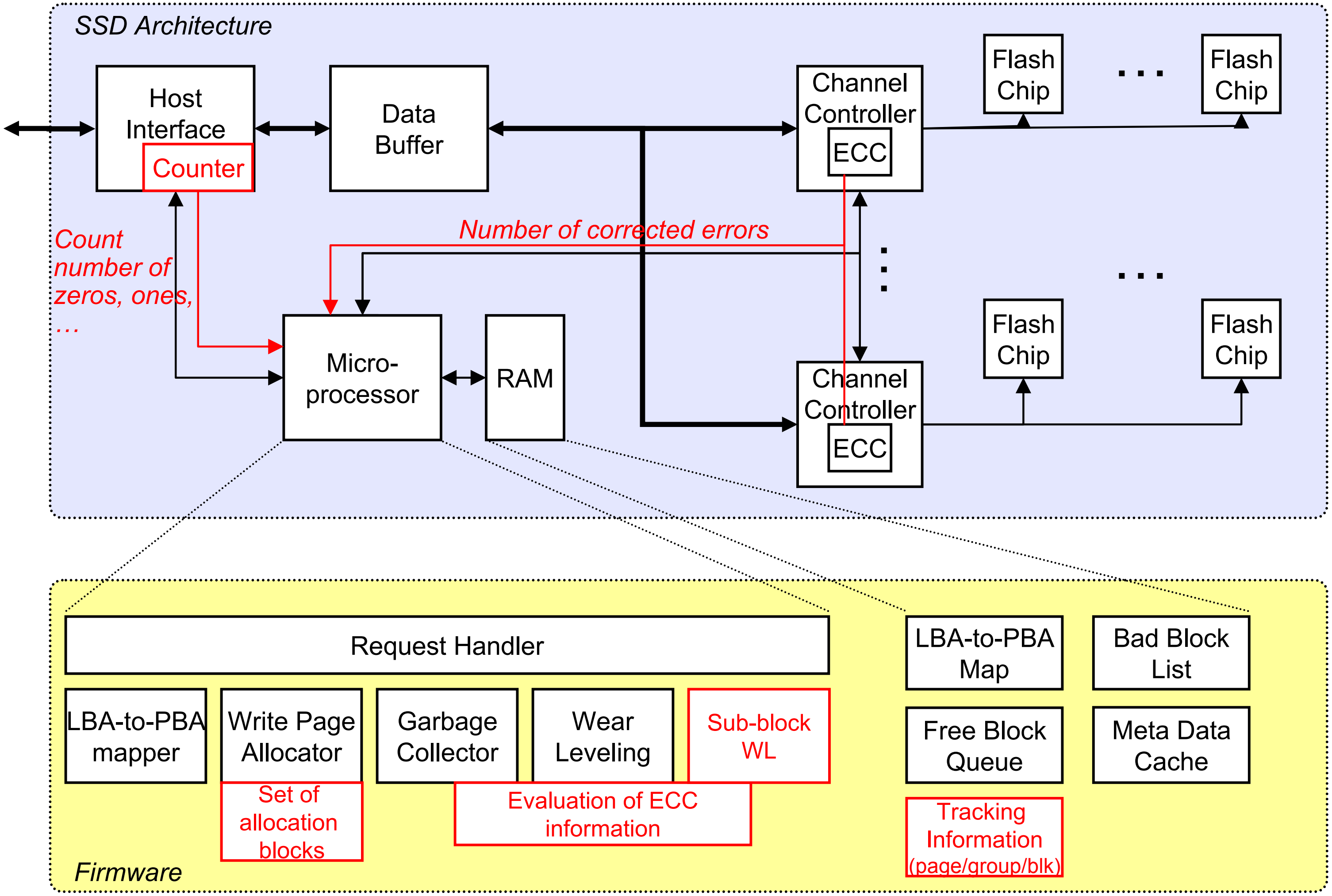
Page	Retired
0	no
1	yes
2	no
3	~~no~~ yes
4	no
…	…

Device-dependent Sub-block Wear-leveling

- Characteristics for a given device type are consistent and can be utilized.
- Concept: Partitioning of a block into two or more *groups* with pages in the same group having similar characteristics: First group holds pages that are most likely to fail early; the last groups holds the most robust ones. Groups from the same block are selectively retired.
- The mapping of pages to groups is obtained from device characterization.
- The defective block marker used in ONFi (8-bits) can be used to indicate which groups are still healthy in a block.
- Alternative: Utilize pages from different groups with varying relative frequency (e.g., periodically skip allocation from groups). Per block P/E counter together with the device characteristics can be used to determine when a group is skipped.
  -> Advantages:
  - Allows to retire full blocks instead of individual groups (less meta-data).
  - Device capacity (including over-provisioning) doesn't vary over the whole lifetime.



How to integrate Sub-block Wear-leveling into SSDs



Combinations of Sub-block Wear-leveling schemes

- Content-aware WL & page tracking:
  - Utilize number of corrected errors from ECC of most recent read operation to determine wear-out of a page.
- Content-aware WL & device-dependent WL:
  - Place new pages with more zeros to Flash pages from a group with better characteristics.
- Page tracking & device-dependent WL:
  - Retire an entire group when the first page in the group hits the maximum tolerated error rate during a read.
- Combination of all 3 WL schemes:
  - Place new pages with more zeros to Flash pages with better characteristics.
  - Determine characteristics from the number of corrected ECC errors. Page reads are done when block is garbage collected (small amount of meta-data).
  - Retire an entire group when the first page in the group hits the maximum tolerated error rate during a read.

Conclusion

- Sub-block wear-leveling can significantly extend the lifetime of SSDs.
- RBER at block and page level are not the same thing!
- Three sub-block wear-leveling schemes have been presented.
  => The different sub-block wear-leveling schemes can be combined.
- Suitable combinations of sub-block wear-leveling schemes exist for different use cases (Flash cache, SSD).
  - Stable vs. decreasing r/w performance over time.
  - Device capacity: Fixed vs. shrinking.
- Outlook: Investigate PCM devices.

Questions…