Solving 3D Flash's Error Recovery Problem

Conor Ryan

NVMdurance

Conor.Ryan@NVMdurance.com

Take Home Messages

- Errors cannot be avoided
  - They need to be handled gracefully
  - BCH doesn’t scale well; LDPC is difficult to implement
- SmartECC makes BCH usable with TLC
- LDPC can be simplified
  - Using hard reads only reduces size and power
- No testmode access required
- All sorts of useful information comes from our characterization

Minimizing Errors

- Flash is trimmed by manufacturer to maximize reliability
  - Trade-off between endurance and retention
- Different trade-offs can be chosen
  - Can compromise LLR tables for LDPC
- Flash **can** be trimmed to particular applications
  - Minimize errors at desired endurance/retention
- However, this requires test mode access
  - Obtaining this can be long, painful and unsuccessful!

The Harsh Reality of Failure

- Read fails **will** happen
- They are increasingly more likely with 3D TLC
- Will be even more likely with QLC
- How can we minimize the impact of failure?
- What is a pragmatic way to deal with failure so we don't impact SSD?

When a read fails

- BCH
  - Re-read with different read levels
  - Recover to RAID
- LDPC
  - (Possibly) read calibration and re-read
  - Read soft information
  - Decode soft information
  - Recover to RAID
- This all takes time

Reliability of SSD

- Trigger rate
  - How often a read fails
- Error Recovery Flow (ERF)
  - How fast and efficient the ERF is at recovering data
  - Plus, how quickly the ERF decides data is unrecoverable

Promoting SSD Reliability

- RBER/ECC
  - Keep RBER low, use powerful ECC
  - BCH doesn’t scale well
  - LDPC requires accurate LLR tables and effective ERF, complex to implement
- Active management
  - Make the best possible use of the blocks
- These are delaying the inevitable

LLR Tables

- Log-likelihood Ratios
  - Inform LDPC how to decode soft reads
  - How soft information decodes to hard read
- Manufacturers often only give “correlation” tables
  - Your mileage may (and probably will!) vary

LLR Tables

- Require a full characterization of chips
- LLR tables are specific to a certain time of life
  - Endurance **and** retention
- LLR tables are specific to use cases
- Need to be regenerated if extra soft data is needed

Flash Memory Summit

LLR Correlation Tables

- Correlation does not mean practical!
- Manufacturers often give “corner cases”
  - 0 cycles, no retention; 0 cycles full retention
  - Full cycles, no retention; full cycles full retention
- The lurking problems
  - Going off book in any way
  - Different cycles/retention; different temperatures
  - Less likely to work as geometry gets reduced

The Problem

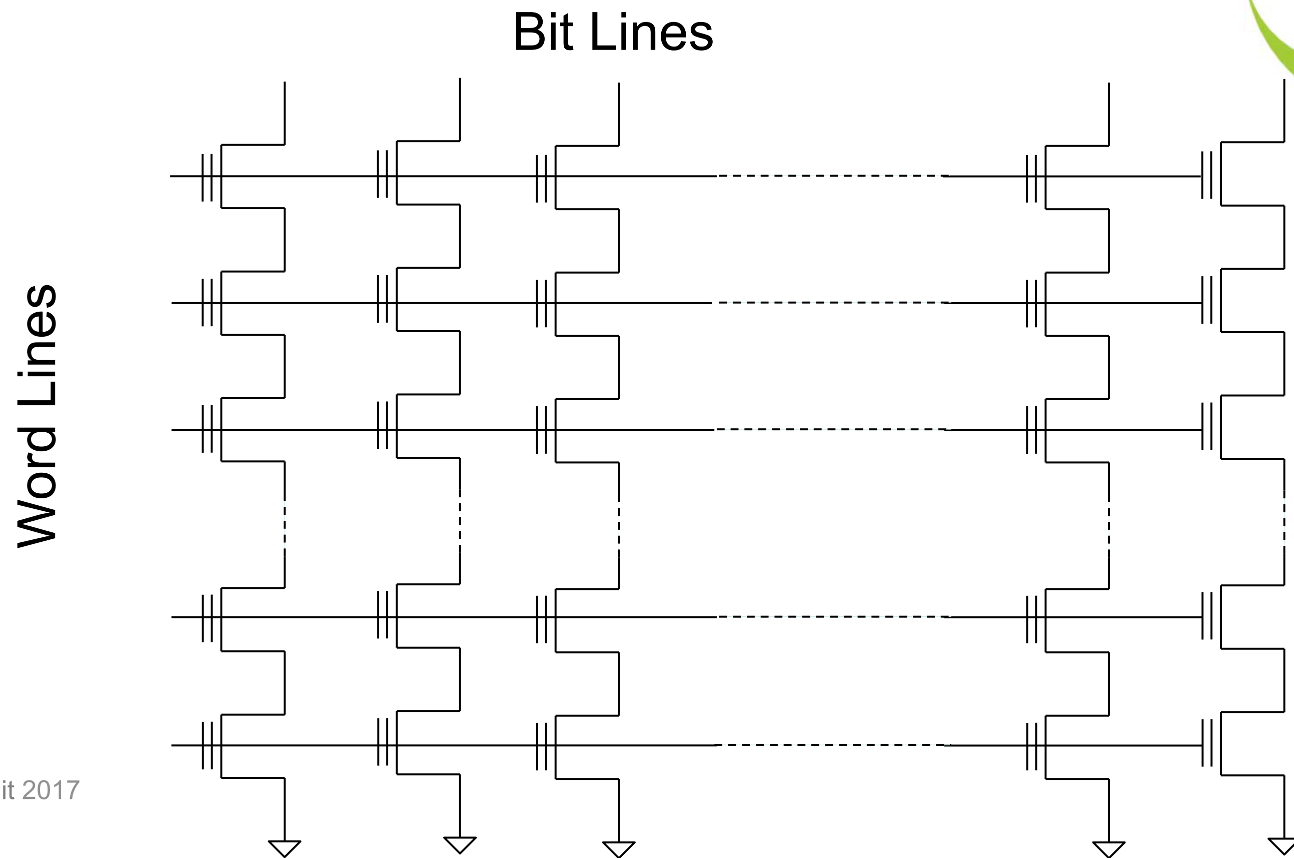
- Bit flips are **not** random
- Many are caused by cell-to-cell interference
- Closer neighbors gives higher interference
  - This is partly why planar flash hit a wall
- More neighbors gives higher interference
  - This is partly why TLC and QLC will be problematic even in 3D flash

Moving Target

- Voltage written to a cell does not remain constant
  - Quality of the cell
  - Number of p/e cycles of the cell
  - **Time** before read
  - **Number** of reads since write
- State and activity of neighboring cells

Flash Memory Summit

2D Flash



2D Flash

Bit Lines

Word Lines

"Victim" Cell

"Aggressor" cells

3D Flash

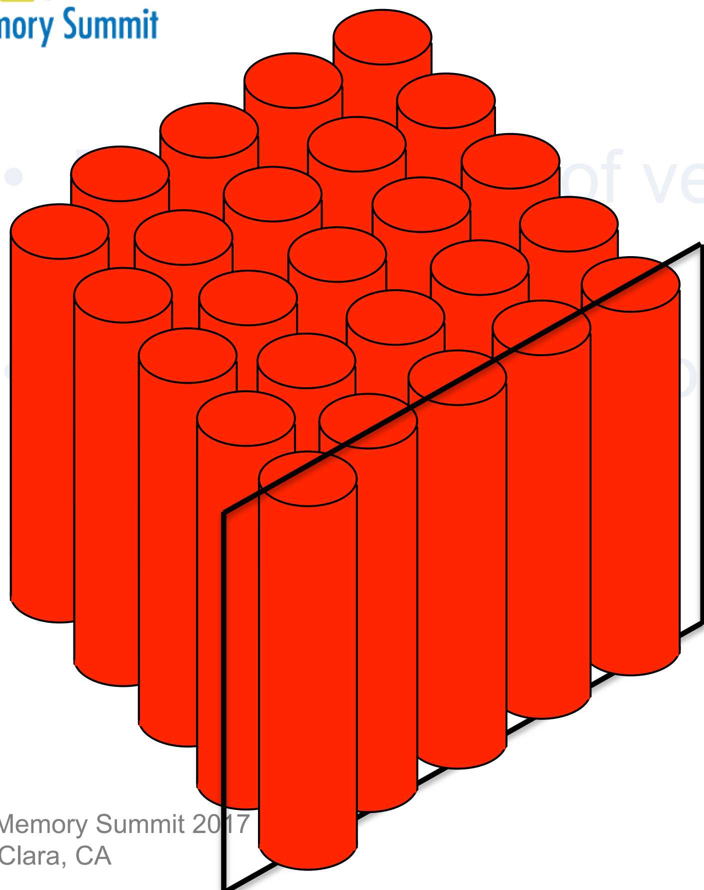
- Blocks consist of vertically-stacked layers of cells
- Cells have neighbors above and below as well as to the side

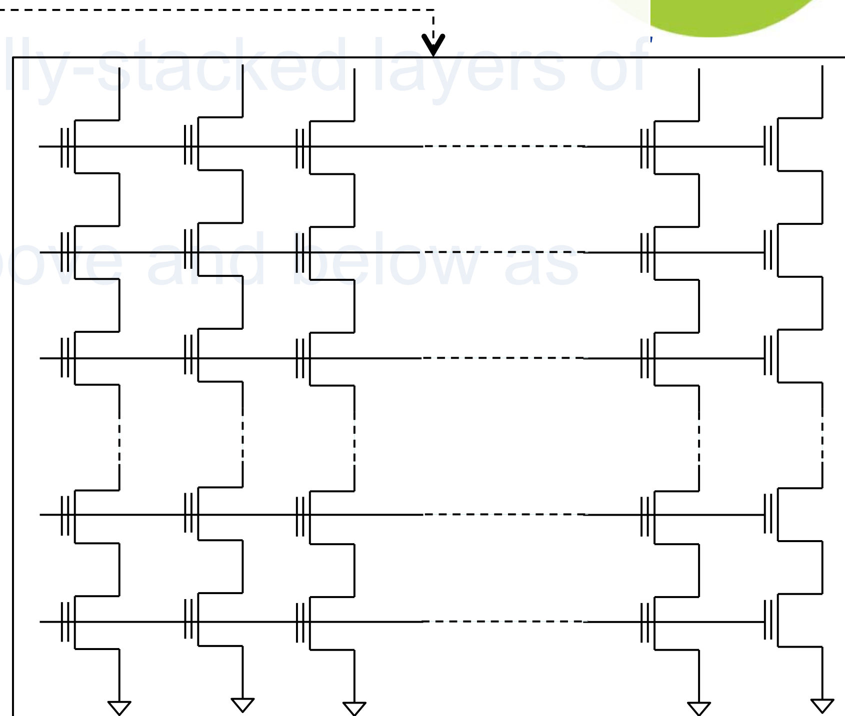
3D Flash - layers

- Blocks consist of vertically-stacked layers of cells
- Cells have neighbors above and below as well as to the side

Flash Memory Summit

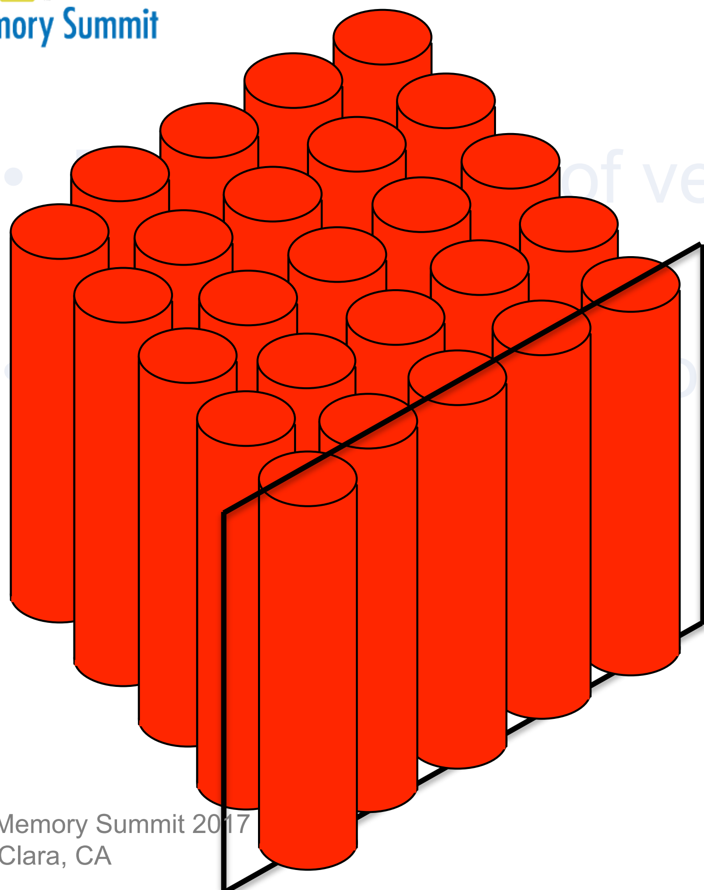
3D Flash - layers

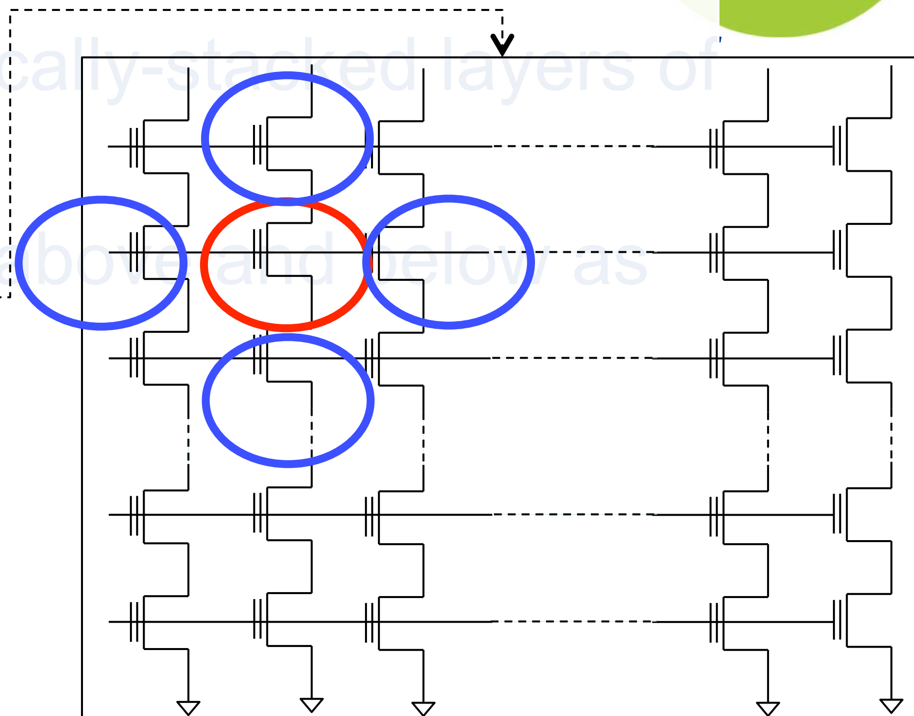




Flash Memory Summit

3D Flash - layers





The Solution - SmartECC

- We use Machine Learning to **automatically** discover how related cells influence a particular cell value at a given time of life
- SmartECC is a set of rules for flipping bits; sets are specific to a time of life
  - Important because manner of interference varies throughout life

SmartECC Flow

Uncorrectable errors in codeword

Is cell X's Vt close to another state?

Yes

Neighbors match known state?

Yes

No

No

Do nothing

Flip the bit

Typical Error Recover Flow

Hard read 100uS.

Read Calibration process 700uS

→ Under BCH limit?
- Yes → Read Successful
- No → Level 1 Cost 500uS 65-100% codeword recovery

Level 1 → Under BCH limit?
- Yes → Read Successful
- No → Level 2 Cost 300uS 97-100% codeword recovery

Level 2 → Under BCH limit?
- Yes → Read Successful
- No → Fail to RAID

Detailed Algorithm

- Perform hard read
- If BER less than ECC hard read capability then return codeword
- Else perform SmartECC Level 1 Recovery
  - 5 sectors toggled out and stored (similar timing to LDPC)
  - If BER less than hard read ECC capability return codeword
- Else perform SmartECC Level 2 Recovery
  - 3-5 further sectors toggled out and stored (60% time penalty over LDPC)
  - If BER less than hard read ECC capability return codeword

Detailed Algorithm

- Perform hard read
- If BER less than ECC hard read capability then return codeword
- Else perform SmartECC Level 1 Recovery
  - 5 sectors toggled out and stored (similar timing to LDPC)
  - If BER less than hard read ECC capability return codeword
- Else perform SmartECC Level 2 Recovery
  - 3-5 further sectors toggled out and stored (60% time penalty over LDPC)
  - If BER less than hard read ECC capability return codeword

Implementation - Discovery

- Perform characterization
- Extract data from devices every 1K cycles
  - Get data at zero and full retention
- Perform Machine Learning
  - Discover neighbor patterns at each checkpoint
- Generate LLR tables for LDPC

Implementation - Practicalities

- Characterization
  - 90 head temperature controlled test array
  - Arria 10-based SSD on Intel/NVMdurance reference design
  - Approximately three weeks
- Machine Learning
  - Cloud based parallel search; 2-3 days

Discovering Neighbor Patterns

- What is the **minimal** number needed?
  - Too many and run-time will be impacted
- Use Machine Learning
  - Discover the minimal set required

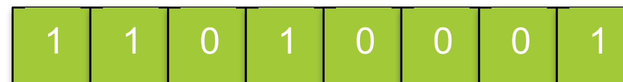
Size of the problem

- In TLC with Level 1 it may be possible to brute force the patterns, but...
  - Search space is huge
  - Increases exponentially with each extra bit
  - Increases with depth of characterization
  - Rules change over time
- Level 2 space is larger
- In QLC the space will be much larger

Evolutionary Algorithms

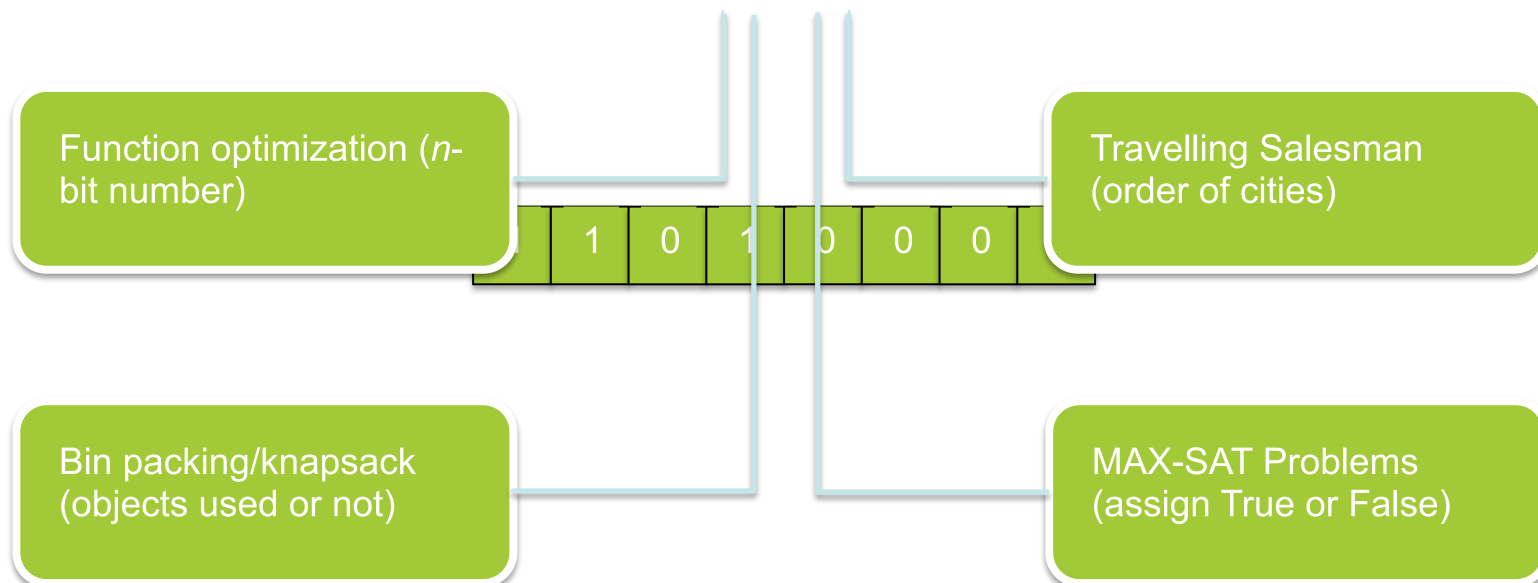
- Machine Learning algorithm based on simulation of natural evolution
- Maintains a **population** of solutions (“individuals”)
- Drive by *fitness function*
  - Measures how good solutions are at solving problem
- Recombine best solutions to create increasingly better ones

Representation



Flash Memory Summit

Representation



Representation

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

Flash Memory Summit

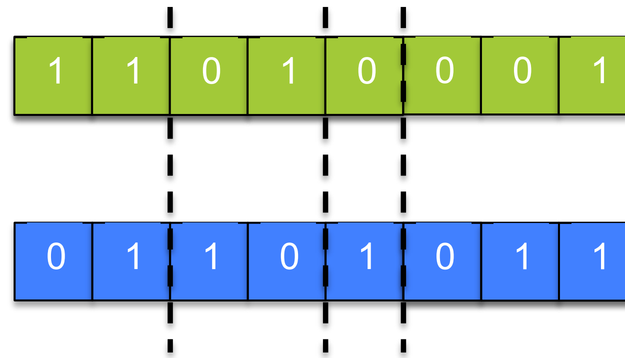
Representation

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

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

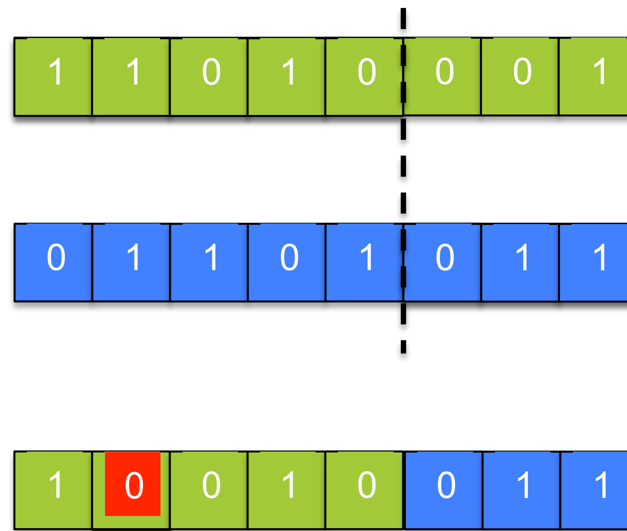
Flash Memory Summit

Recombination



Flash Memory Summit

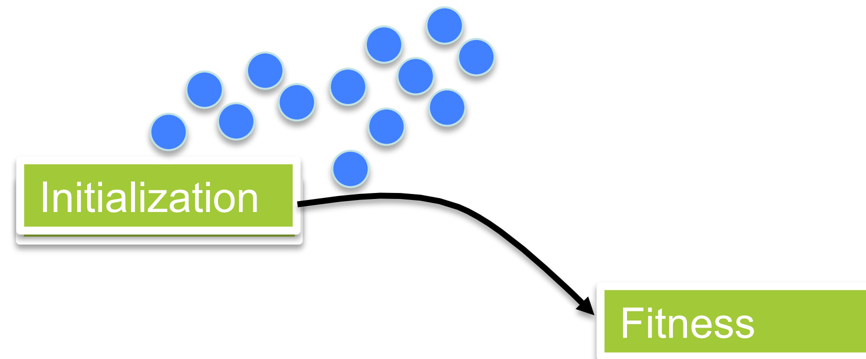
Mutation



Algorithm

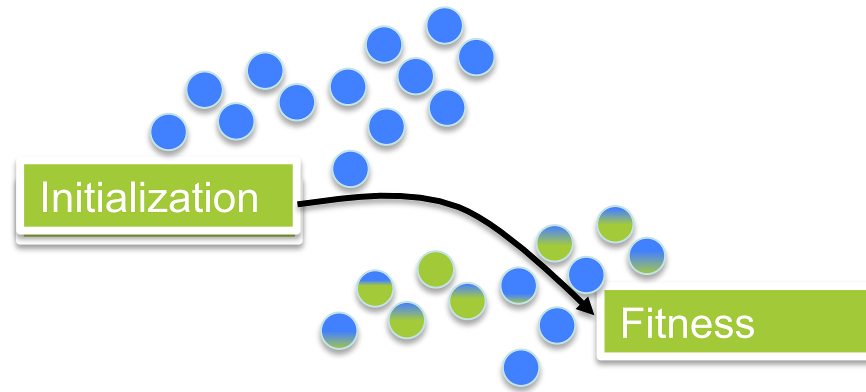
Flash Memory Summit

Algorithm



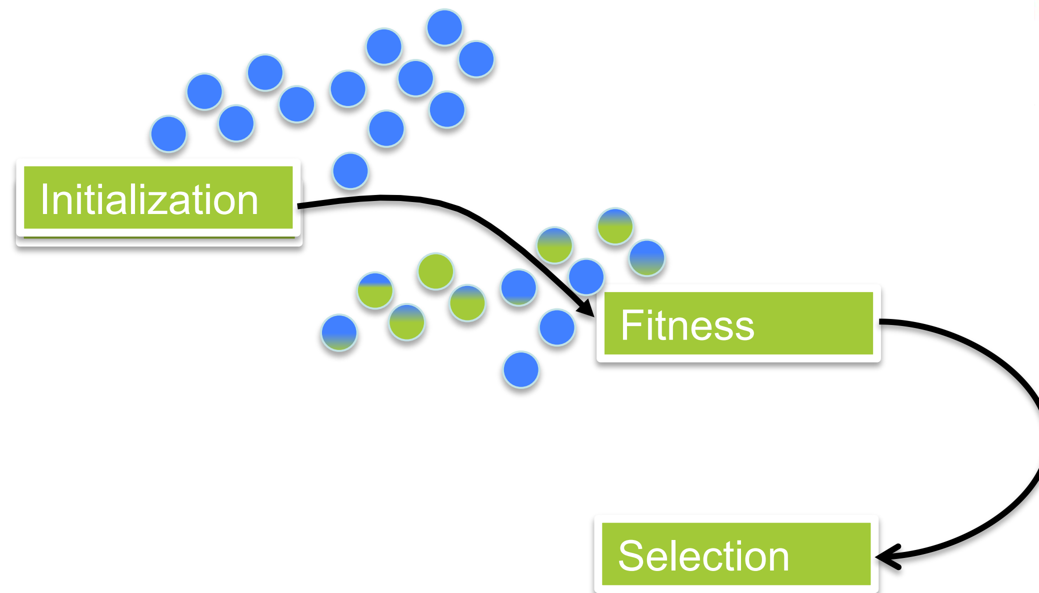
Flash Memory Summit

Algorithm



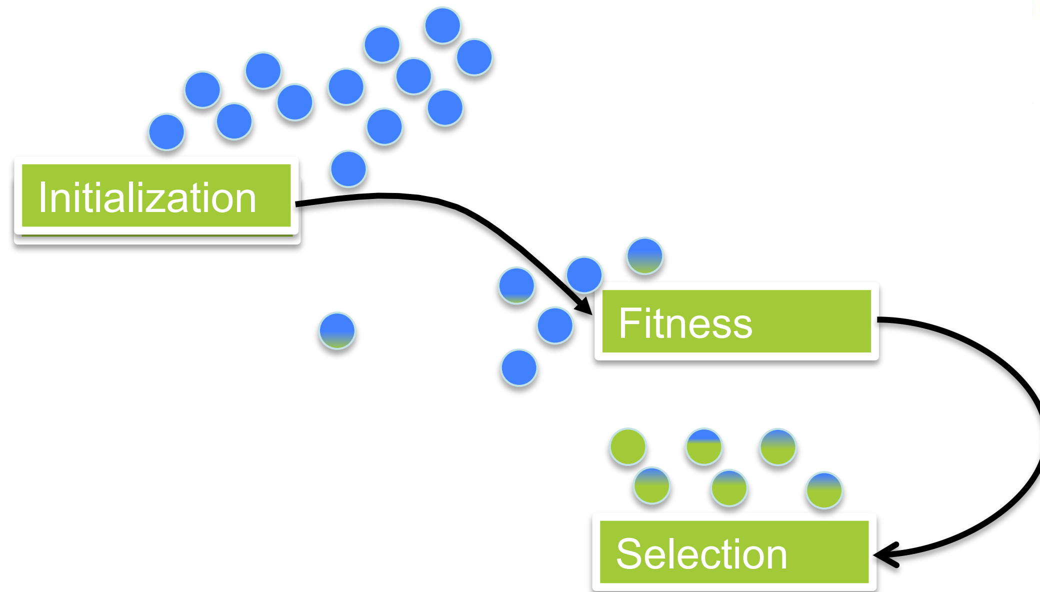
Flash Memory Summit

Algorithm



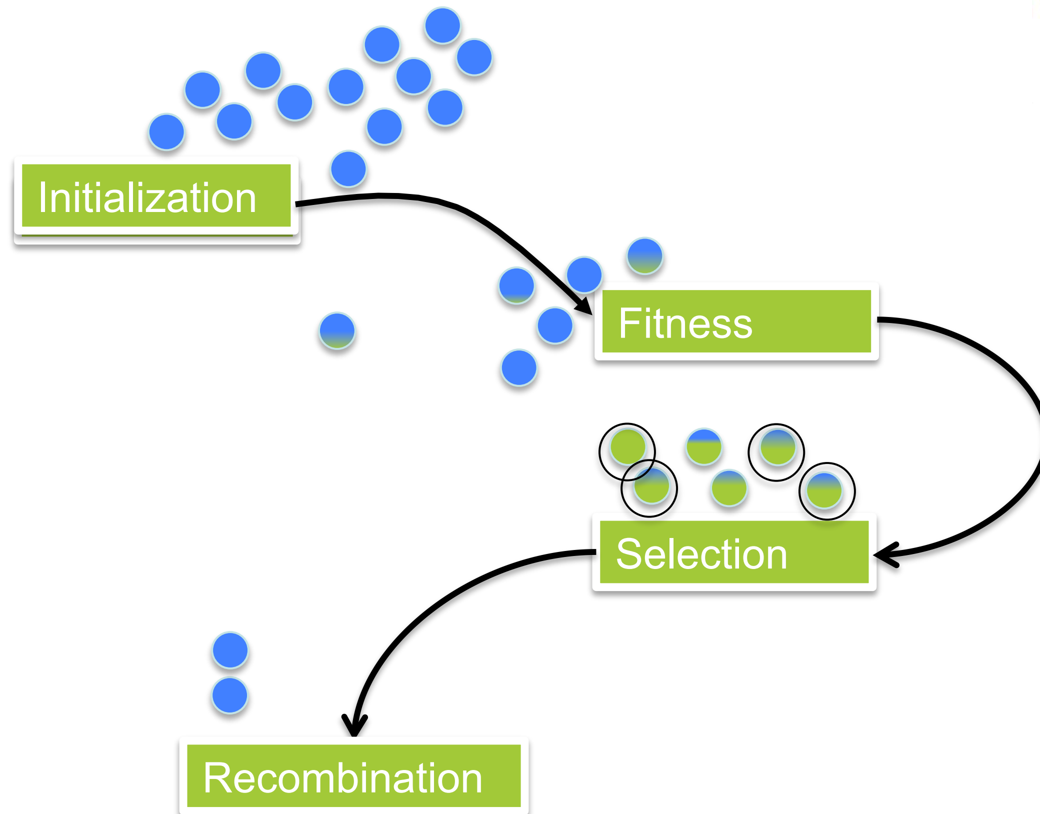
Flash Memory Summit

Algorithm



Flash Memory Summit

Algorithm



Flash Memory Summit

Algorithm

Initialization

Fitness

Selection

Recombination

Mutation

Flash Memory Summit

Algorithm

Complete

Initialization

Fitness

New Generation

Selection

Mutation

Recombination

Testing SmartECC

- Device
  - 3D TLC
  - 5K p/e with 4 months retention using LDPC
  - No spec with BCH!
- Conditions
  - Cycle to 5K in 1K increments
  - Retention recorded at 0 and 4 months
  - Devices obtained from three non-consecutive lots
- ECC
  - BCH ranging from 40 to 90

Results – 5K p/e, 4 months

Note

- 864,000 codewords tested per level

ECC	BCH
40	13421
45	5652
50	2235
55	836
60	285
65	109
70	39
75	16
80	3
85	2

Results – 5K p/e, 4 months

Note

- 864,000 codewords tested per level

ECC	BCH	Level 1
40	13421	362
45	5652	192
50	2235	121
55	836	65
60	285	43
65	109	21
70	39	8
75	16	2
80	3	1
85	2	0

Results – 5K p/e, 4 months

Note

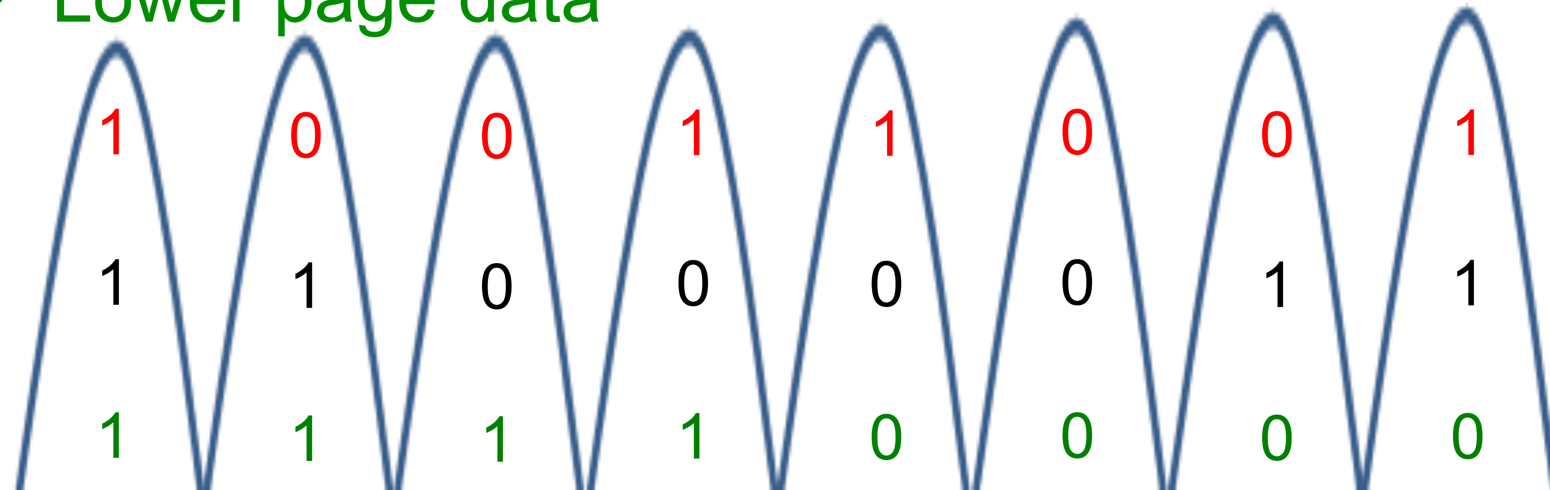
- 864,000 codewords tested per level

ECC	BCH	Level 1	Level 2
40	13421	362	9
45	5652	192	0
50	2235	121	0
55	836	65	0
60	285	43	0
65	109	21	0
70	39	8	0
75	16	2	0
80	3	1	0
85	2	0	0

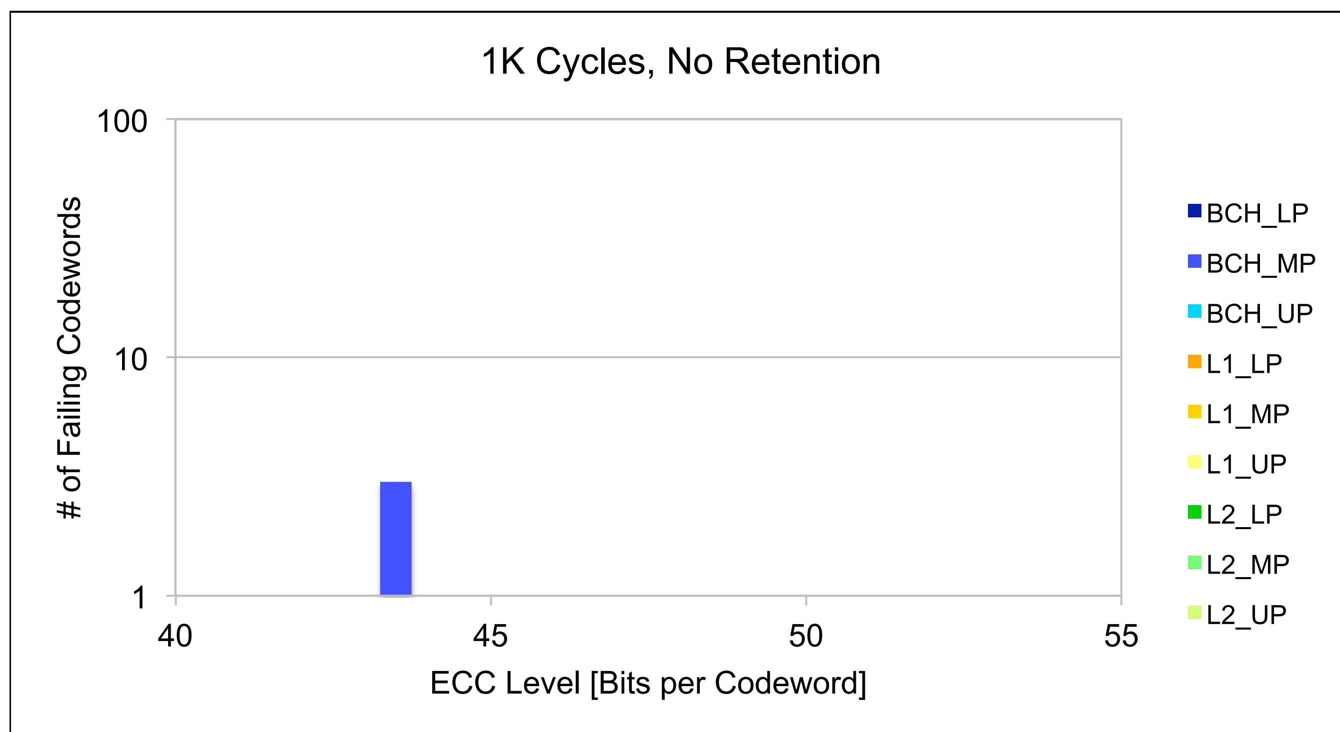
Not all pages are created equal!

- Upper page data
- Middle page data
- Lower page data

- Lower page data is typically most reliable



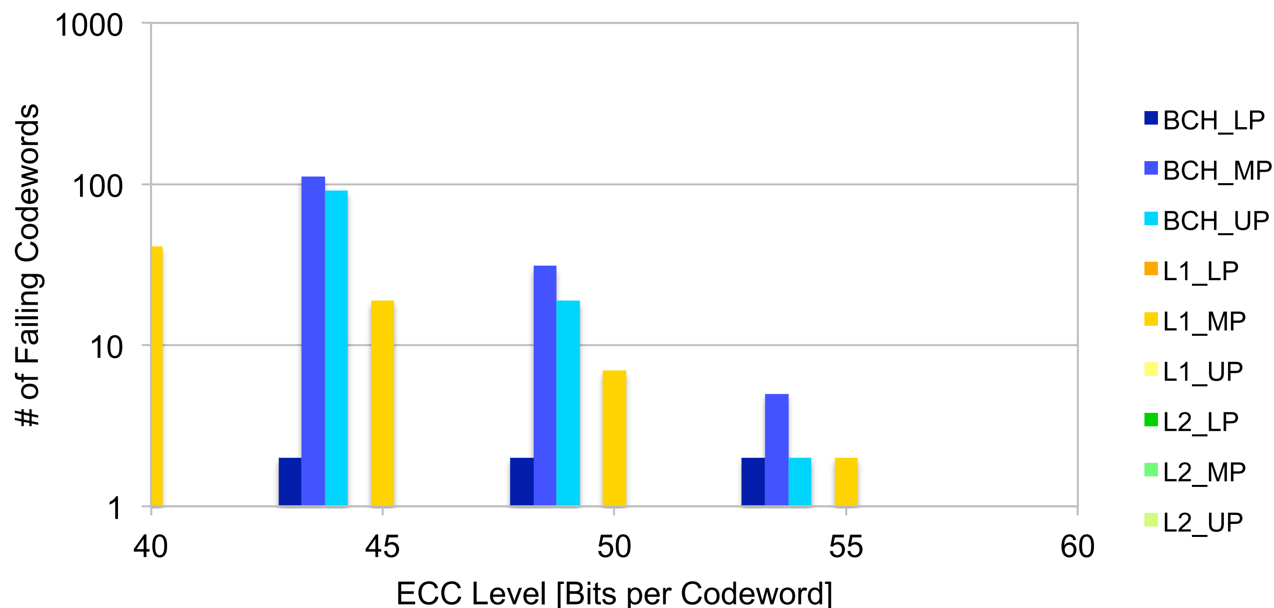
Results by page type



Notes

- Only MP contains errors
- True for all zero retention plots

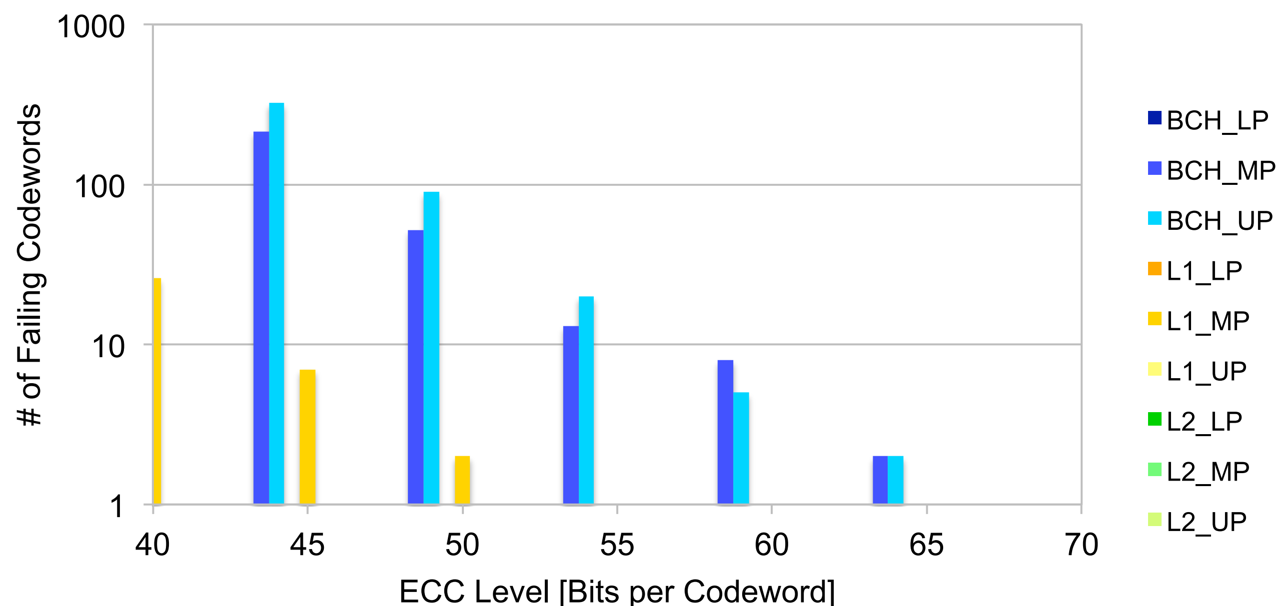
Results



Notes

- MP and UP consistently worst
- BCH recovers all data with 60 bit ECC
- L1 recovers all data with 60 bit ECC
- MP worst for L1
- L2 recovers all data with 40 bit ECC

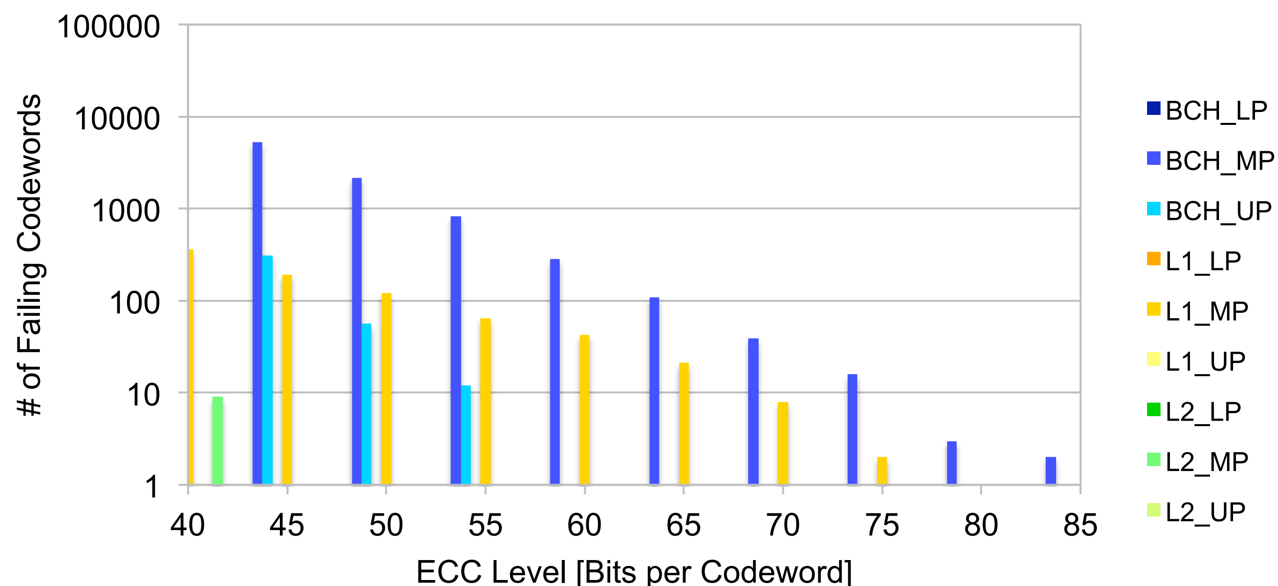
Results



Notes

- MP and UP consistently worst
- BCH recovers all data with 70 bit ECC
- L1 recovers all data with 60 bit ECC
- MP still worst for L1
- L2 recovers all data with 40 bit ECC

Results



Notes

- MP and UP consistently worst
- BCH recovers all data with 90 bit ECC
- L1 recovers all data with 85 bit ECC
- L2 recovers all data with 45 bit ECC
- MP worst for L1/L2

Trigger Rate

- SmartECC does not change the trigger rate
  - Where trigger rate means **intervention**
- Lowering the hard read ECC raises the trigger rate
- Why bother with SmartECC?
  - Because its implementation is simple

SmartECC FPGA Module

Level 1 data

Hard read bits
Level 1, sector read 1
Level 1, sector read 2
Level 1, sector read 3
Level 1, sector read 4

Comparator bit flip

Corrected data buffer

ECC Engine

Rules based pattern lists

SmartECC FPGA Module

Level 1 data

Level 2 data

Hard read bits
Level 1, sector read 1
Level 1, sector read 2
Level 1, sector read 3
Level 1, sector read 4
Level 2, Other data 1
Level 2, Other data 2
Level 2, Other data 3

Comparator bit flip

Corrected data buffer

ECC Engine

Rules based pattern lists

FPGA Resources - Sample

- We implemented a 20 bit BCH ECC engine in a modest FPGA (Intel Altera Cyclone 5)
- Occupied 7K (of 16K) ALUs and 1.3m (of 2.7m) memory bits
- *50 bit ECC used the whole FPGA*
- SmartECC requires
  - 532 ALUs and 0.6 Mbit of memory
  - Approx 3% of space and 25% of memory

(Lack of) Costs

- Latency is not the issue
  - LDPC soft decode takes the same time as SmartECC Level 1; SmartECC Level 2 is 300uS extra
- FPGA resources is not an issue
  - SmartECC uses a simple comparator arrangement although data needs to be stored between reads
- Trigger rate may be an issue
  - Trade off between lowering the hard bit correction capability and the rate at which time must be taken to recover the data

Trade Offs

- Hard ECC capability vs Trigger rate
  - Trigger rate is unaffected by SmartECC
  - **But** exploiting potential reduction in the power of the ECC engine affects the trigger rate
  - Higher hard ECC gives lower trigger rate

Trade Offs Summary

- High ECC gives low trigger rate
  - But occupies space
- SmartECC L2 is slower than LDPC
  - But occupies far less space
- Low ECC with Smart ECC L2
  - Cheapest in terms of space
  - Slowest in terms of time
- SmartECC L1 virtually the same as LDPC
  - But occupies far less space

Trade Offs

- Advice
  - Use highest hard ECC capability available
  - Use in appropriate applications
    - Force to use BCH, FPGA controller, QLC, archiving, etc.
  - Manage wear to prioritize reliable blocks
    - Extensive data on blocks comes free with characterization process
  - Manage wear-out at block level
    - Use NVMdurance Navigator or similar

Generating Solutions

- Perform full characterization with ML-enabled temperature controlled test heads
- Provides
  - LLR tables at 1K p/e intervals
  - Rule based bit patterns for SmartECC module
  - Error models
  - Reliability information at page and block level to inform map and wear-leveling algorithms

Summary

- Read Failures are inevitable
  - LDPC too costly and/or difficult for some applications
  - Generating accurate LLR tables is not trivial
- SmartECC use Machine Learning to clean up data
  - Simple sets of rules to flip bits
  - Can use **any** ECC method
  - LLR tables created as a by-product
- Tunable trade off between level of ECC and trigger rate

Practicalities

- SmartECC implemented on working SSD reference platform
  - PCIe NVMe interface; 3D Micron TLC flash
  - Custom channel controller, FTL and 40 bit SmartECC engine
- NVMdurance can automatically discover SmartBCH rules
  - Customized for specific use cases

Contact Us

- NVMdurance provide
  - SmartECC – Rules and ERF to extend BCH to be usable in current generation 3D flash
  - SmartLDPC – Rules and ERF to enable hard LDPC to be used at lower cost (space, power, etc.)
    - Includes LLR tables and full ERF for LDPC
- Conor Ryan
  - Conor.Ryan@NVMdurance.com
  - Tel (580) 672 9004 or +353 86 816 3653