Data Shaping for Improving Endurance and Reliability in Sub-20nm NAND

Eran Sharon, Stella Achtenberg, Idan Alrod, Avi Klein, Alon Eyal
Intelligent Memory Systems, SanDisk Corp.

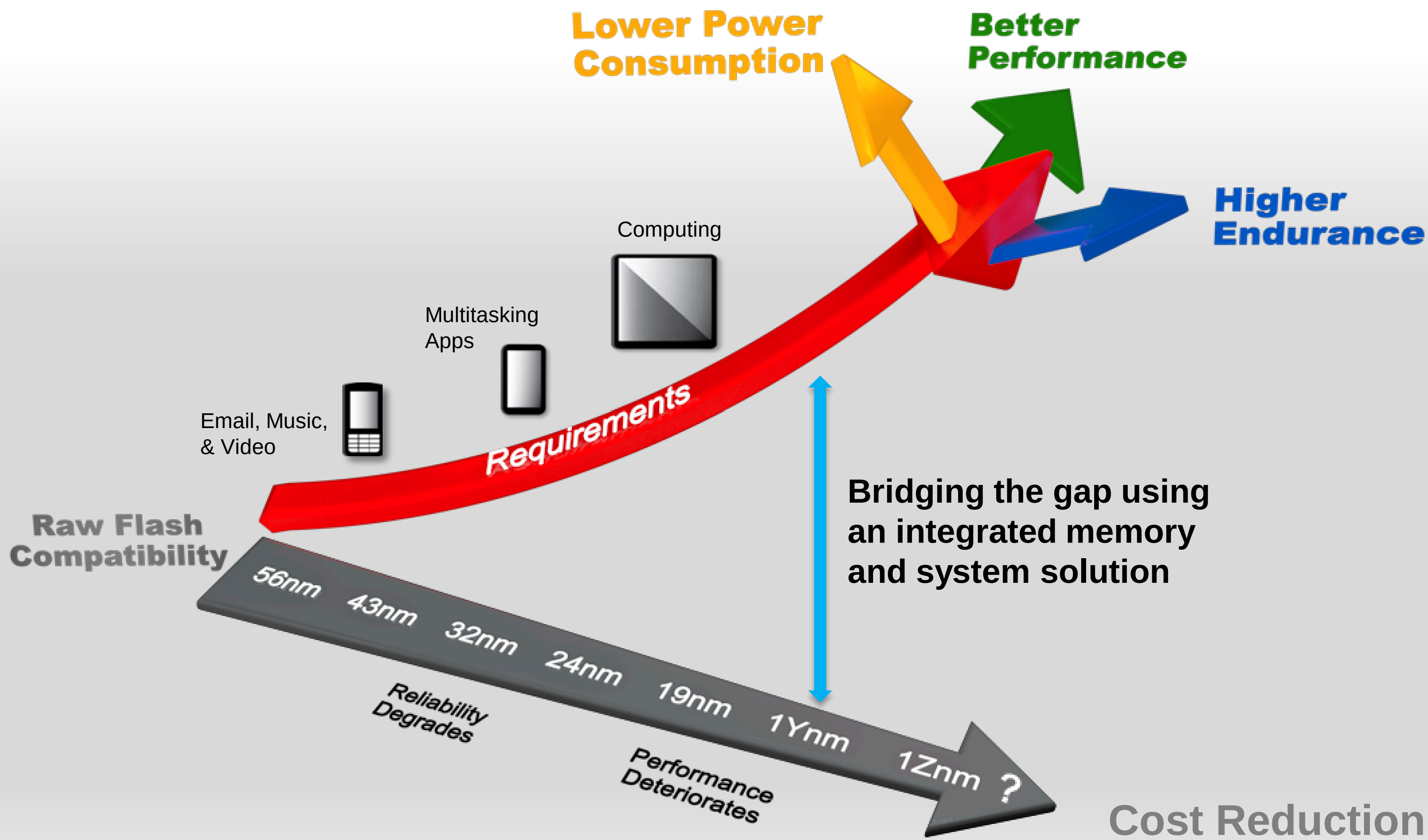
August 2014

Outline

- Gap Between Product Requirements and Technology Capability
- Multi Tier Integrated System & Memory Solution - Recap
- Low Data Entropy in Typical Hosts – Unrealized Potential
- Leveraging Low Host Data Entropy for Data Shaping
- Data Shaping Effect on Memory Endurance and Reliability
- Practical Approaches for Data Shaping
- Eco-System Consideration
- Summary

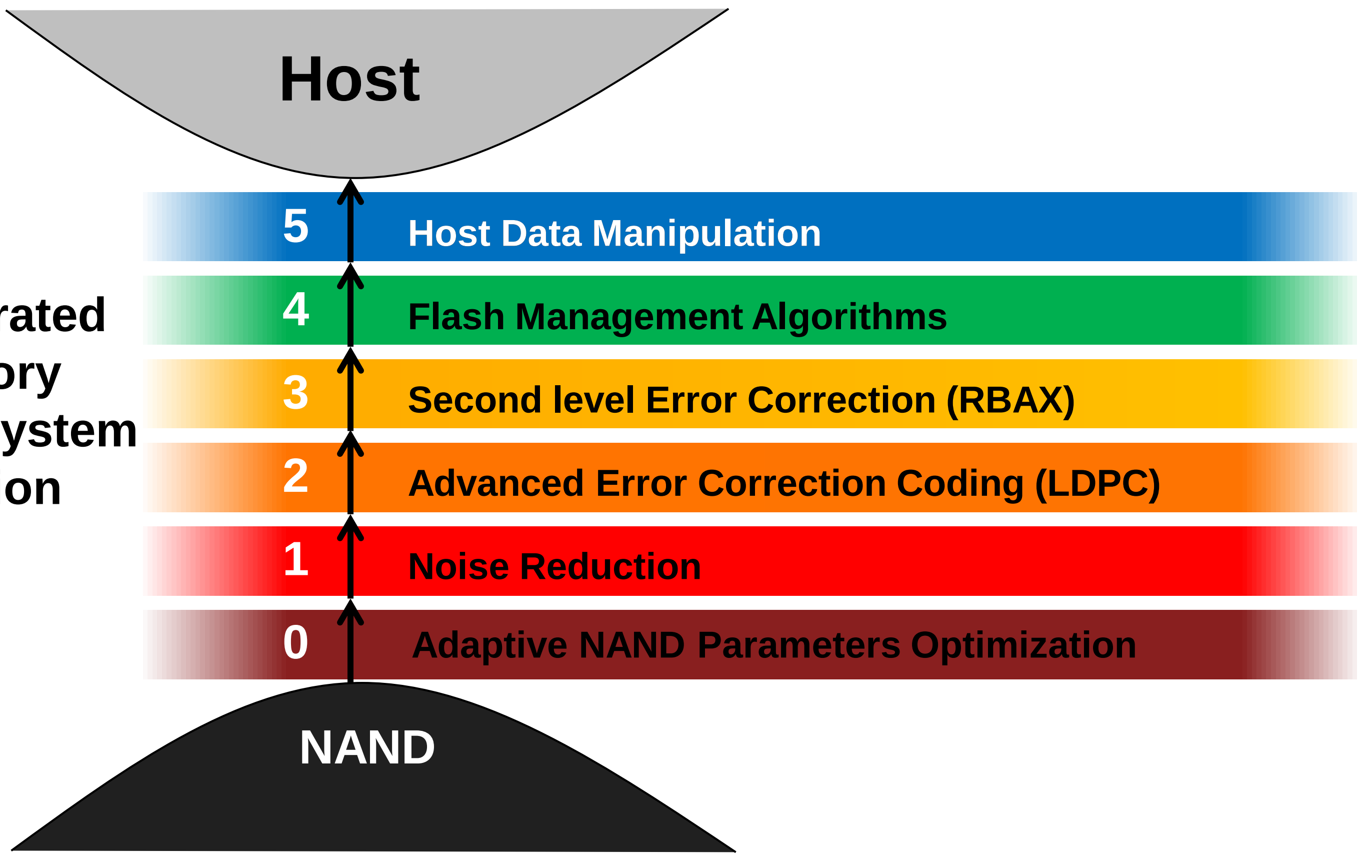
Disclaimer: This tutorial provides an overview of various techniques and concepts, some or all of which may not necessarily reflect what SanDisk is actually using in their products.

Gap Between Raw Memory Capability and Applications Requirements



SanDisk

Integrated Memory & System Solution



Integrated memory and system solution

Low Data Entropy in Typical Hosts – Unrealized Potential

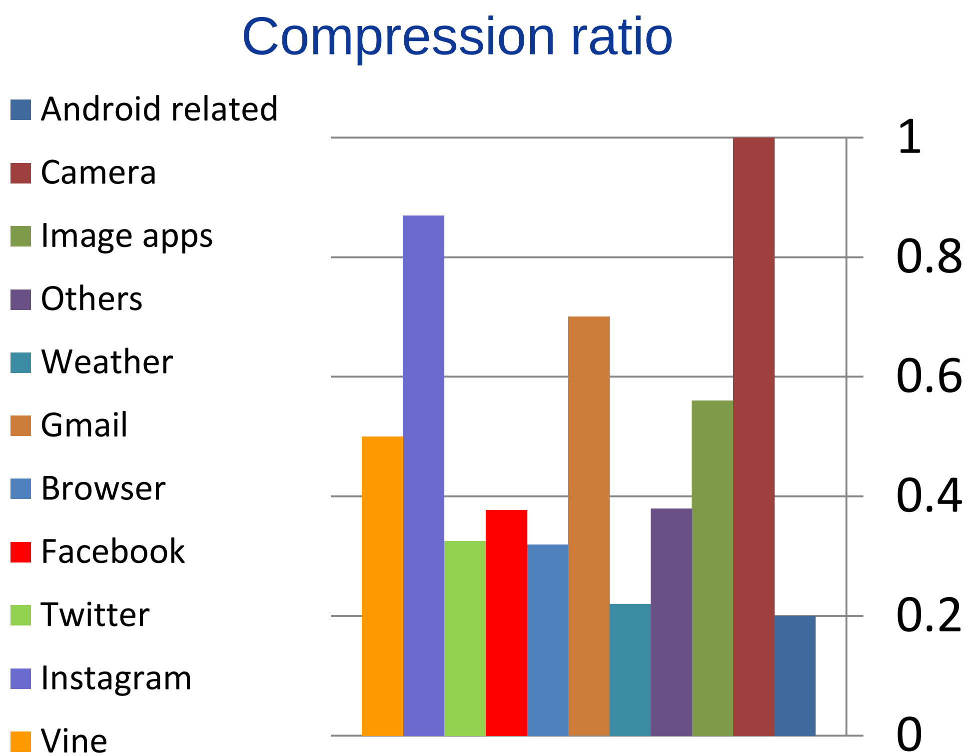
- Examination of typical hosts data traffic shows that significant fraction of the data is of low entropy, having many repetitive data patterns

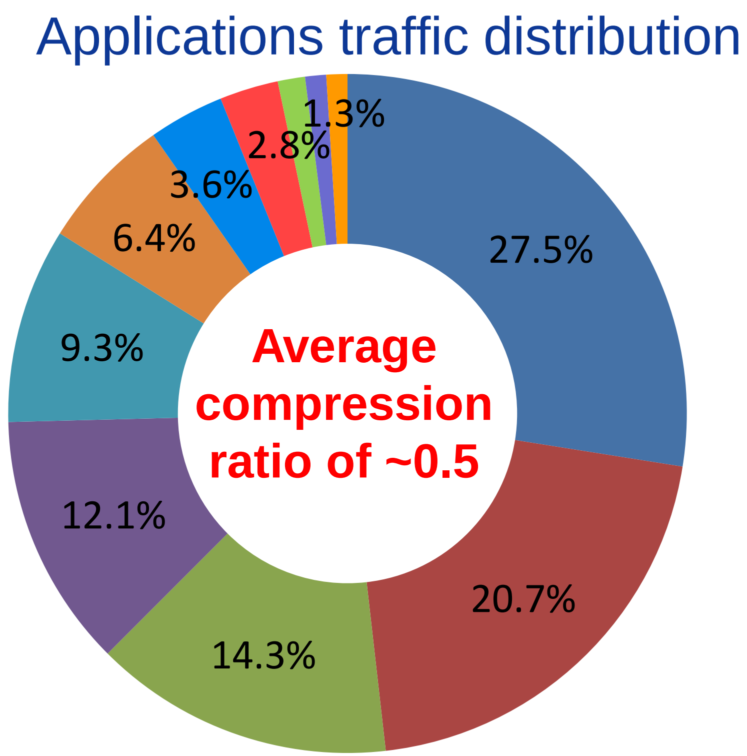
Entropy – measure of data randomness

- **Unrealized potential**: the inherent “redundancy” in the host data can be leveraged for improving endurance, reliability, performance and power, by manipulating host data:
  - **Deduplication**
  - **Compression**
  - **Data Shaping – a.k.a Endurance Coding**

Analyzing mobile traffic of a sample user

- Record the traffic between the host and the controller during sample usage.
- Platform: Android 4.2.2. based Smart Phone
- **Average compression ratio of ~50%**

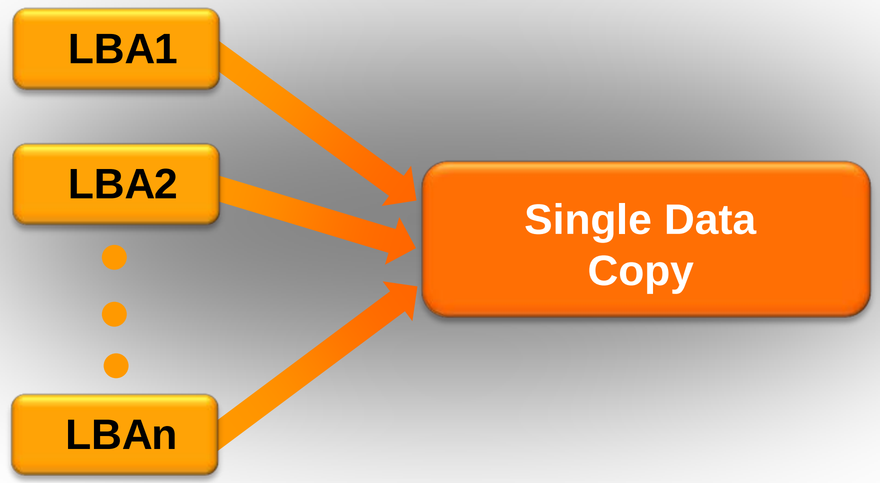




SanDisk

Deduplication

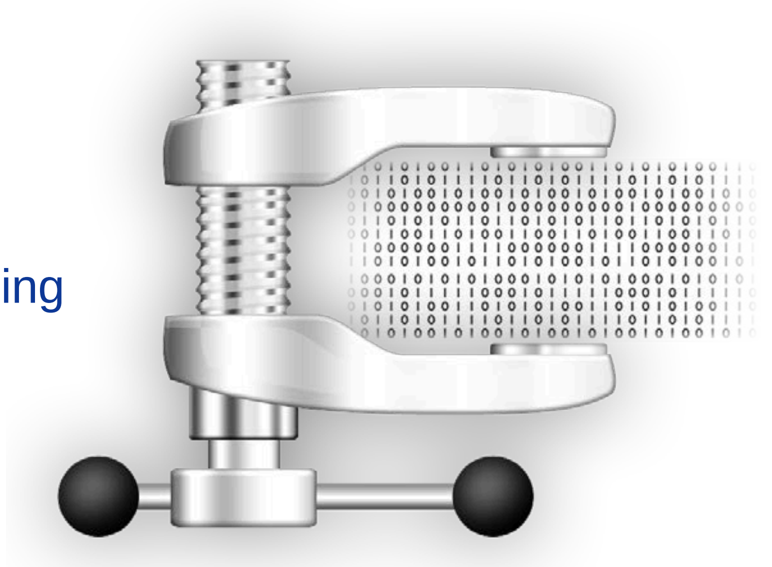
- Specialized data compression technique for eliminating duplicate copies of repeating data
  - Manage multiple pointers to a single stored copy
- Operates on the file system level
  - Less suitable for eMMC level implementation (operates on 4KB sectors, unaware of files)
- Highly suitable for enterprise backup applications



Compression

- Typical traffic in Mobile applications is highly compressible

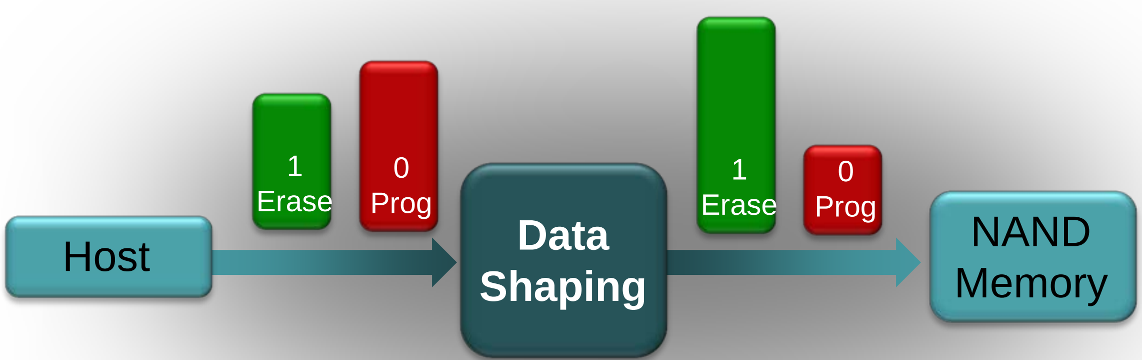
- Compression can provides significant endurance & performance gains
  - Less P/E cycles per GygaByte (GB) written
  - Increases the effective memory over provisioning
    - Improved garbage collection efficiency
    - Reduced write amplification
    - Performance stability

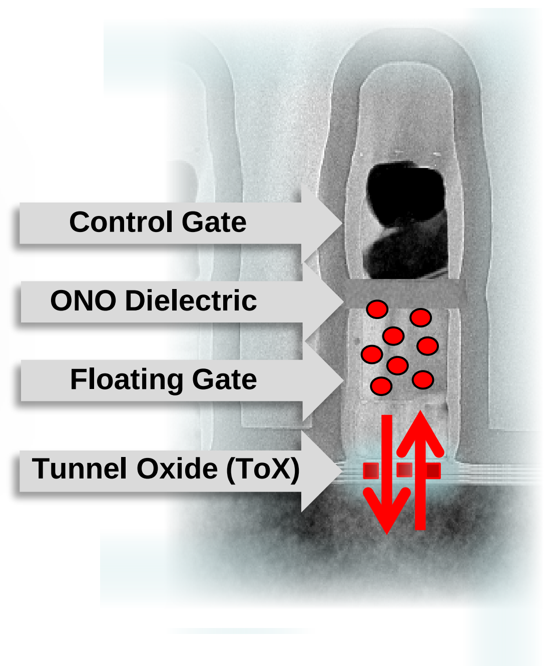


- System level considerations – impact on controller complexity power & cost
  - Requires significant changes in the Flash Management – mapping the logical address space into a variable physical space
  - High throughput, low power and cost compression engine design is challenging

Data Shaping ("Endurance Coding")

- **The Challenge**: Increasing Endurance
- **The Means**: Data Shaping – transform input data sequence into a "shaped" data sequence which induces less wearing when programmed to the NAND.
- **SLC Example**: transform the input data into shaped data having **less 0's**





Minimize number of programmed cells per P/E cycle

- Minimize average number of electrons tunneling in & out of the ToX per P/E cycle
- Slow down ToX quality degradation

SanDisk

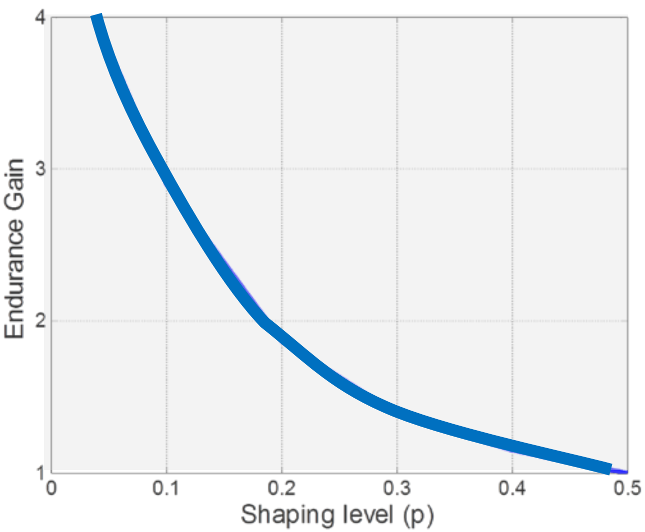
Achievable Endurance Enhancement via Data Shaping

- Cell wearing is proportional to the probability p of the cell to be programmed
- **Simplified model**:
  - W_E – Wearing of an erased cell during a P/E cycle
  - W_P – Wearing of a programmed cell during a P/E cycle
  - $W_P >> W_E$ (Much more electrons passing through ToX for programmed cells)
  - Total wearing as a function of the shaping level p: $W(p) = (1-p) \cdot W_E + p \cdot W_P$
  - Endurance gain due to using Shaped data ($p < 0.5$) vs. Scrambled data ($p = 0.5$):

$$Gain(p) = \frac{W(0.5)}{W(p)} = \frac{\alpha + 0.5}{\alpha + p},$$

$$\text{where } \alpha = \frac{W_E}{W_P - W_E}$$

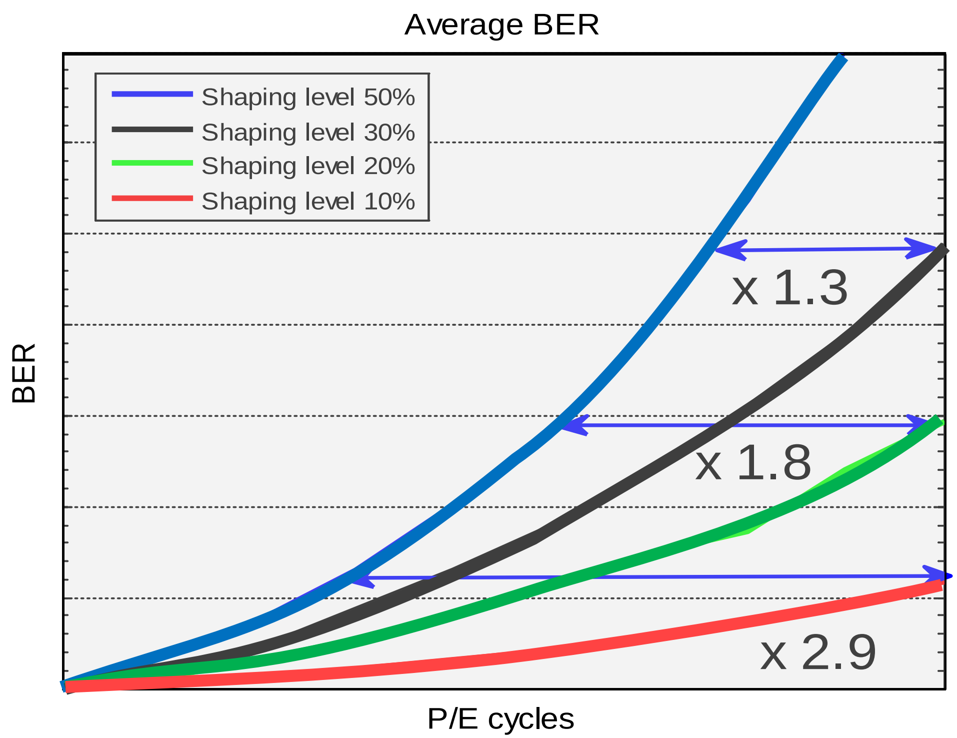
(α is specific per memory technology)



SanDisk

Achievable endurance Enhancement via Data Shaping - Empirical measurements

- **Objective**: measure the cumulative wearing reduction effect of shaping
- **Experiment**:
  - Cycle the memory with shaped data (different shaping levels, up to different cycles)
  - At the last cycle, program with scrambled data ($p = 0.5$) and measure BER
  - Compare BER deterioration with cycling as a function of the average shaping level

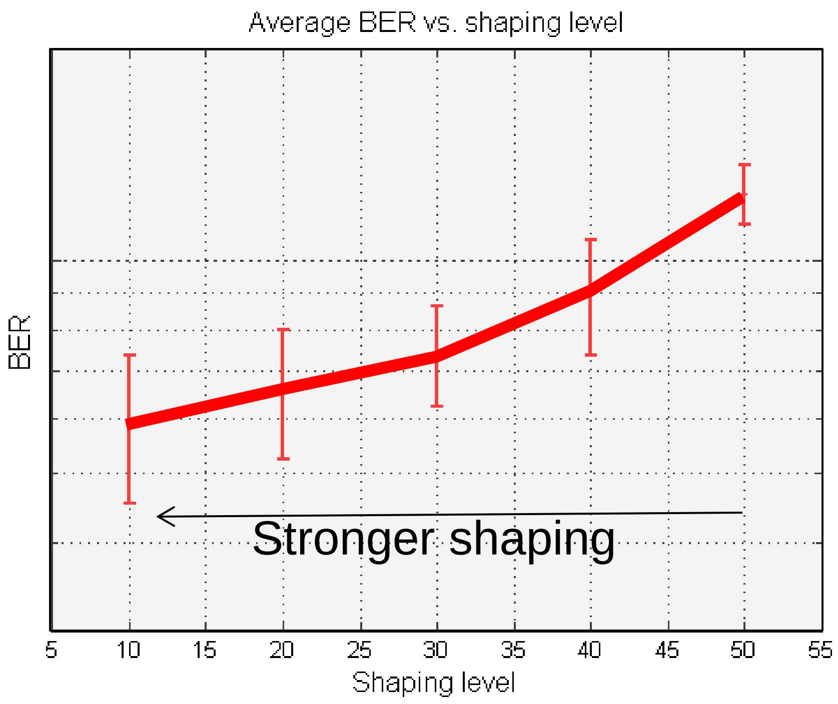


Measured endurance gain increases as the fraction of programmed cells (p) reduces

SanDisk

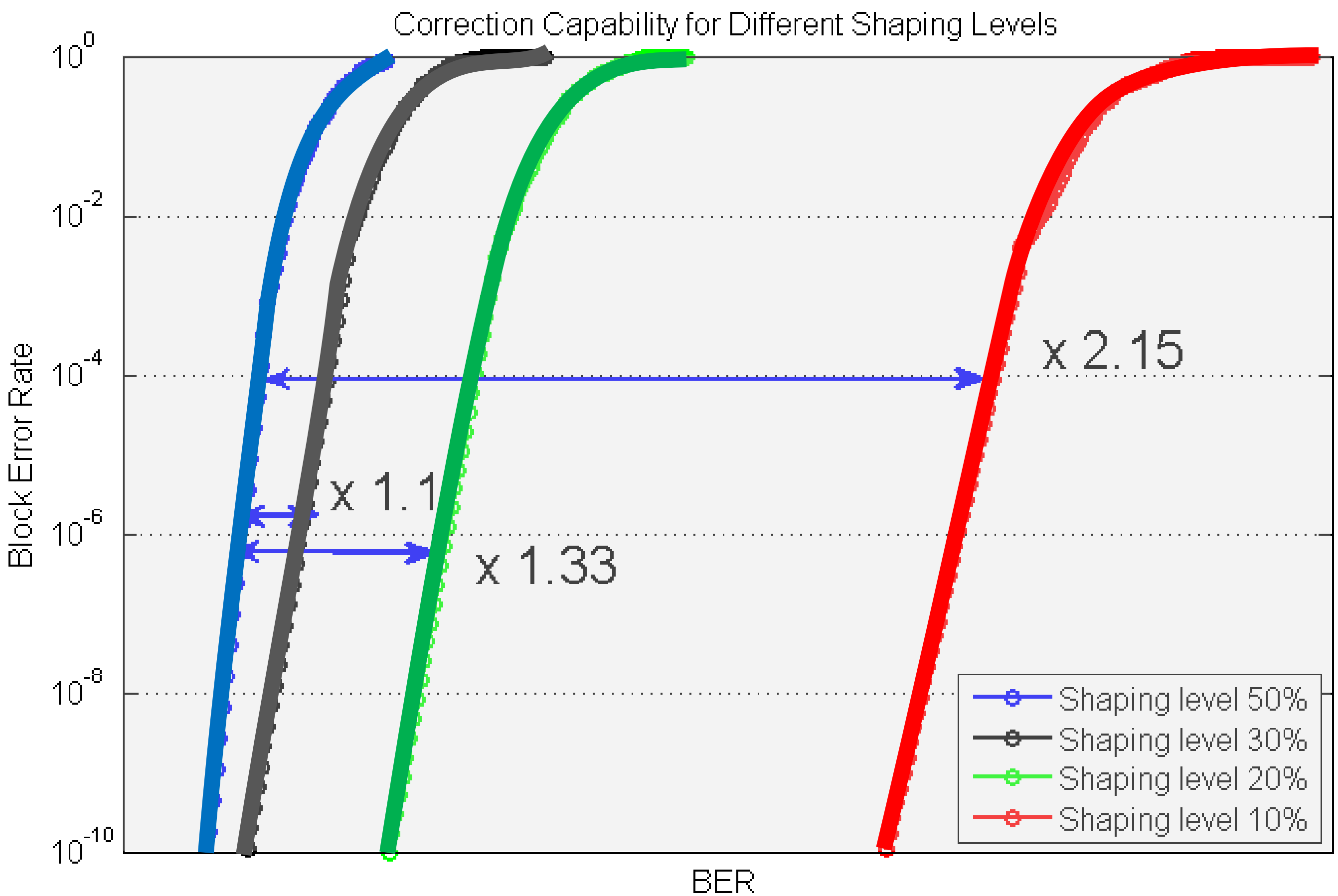
“Noise” Reduction due to Data Shaping - Empirical measurements

- **Objective**: measure the local BER reduction effect of shaping
- **Experiment**:
  - Cycle the memory with scrambled data
  - Program with shaped data at the last cycle and measure BER
  - Compare BER level at the last cycle as a function of the shaping level *p*.



Effect of Shaping on Error Correction Capability

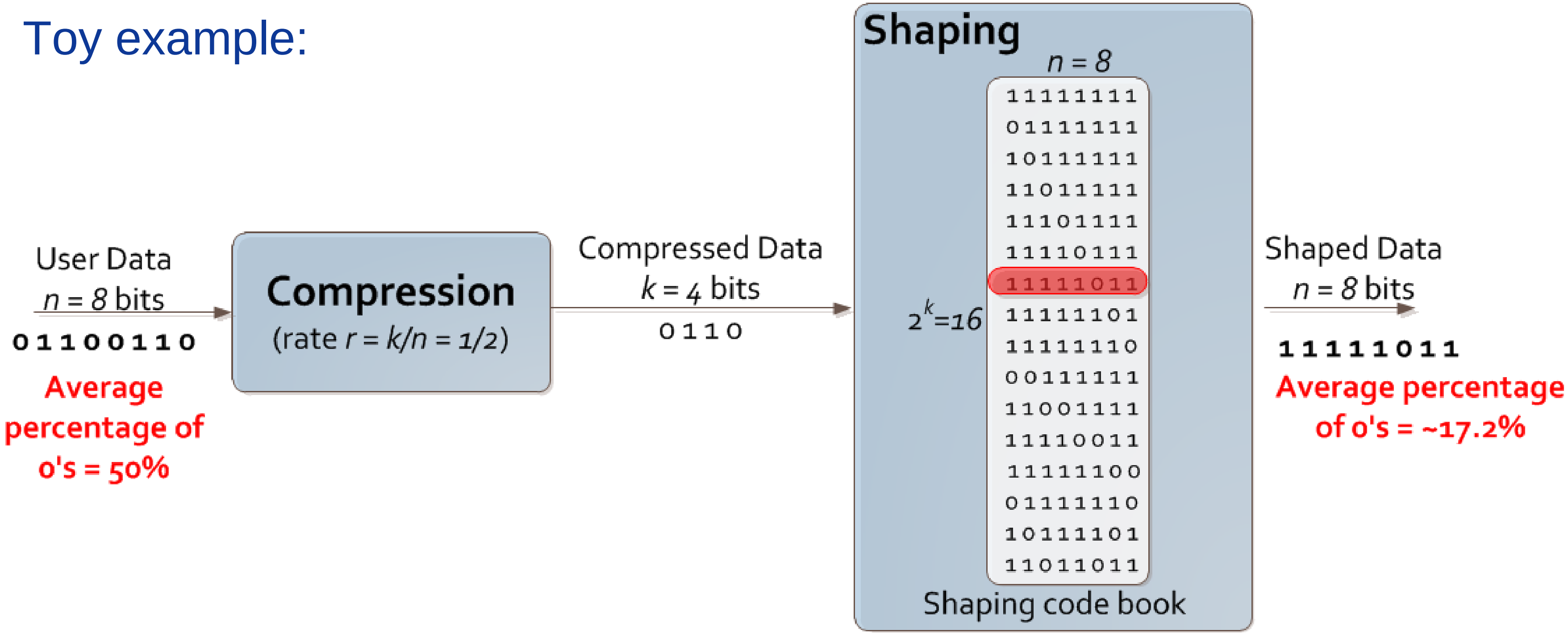
- Decoder correction capability can be significantly improved for shaped data
- Adjust decoder soft input metrics based on the estimated shaping level



Leveraging low host data entropy for shaping

- Compression – Expansion approach
  - Compress: *n* user bits into *k* compressed bits
  - Expand: *k* compressed bits into *n* shaped bits

- Toy example:



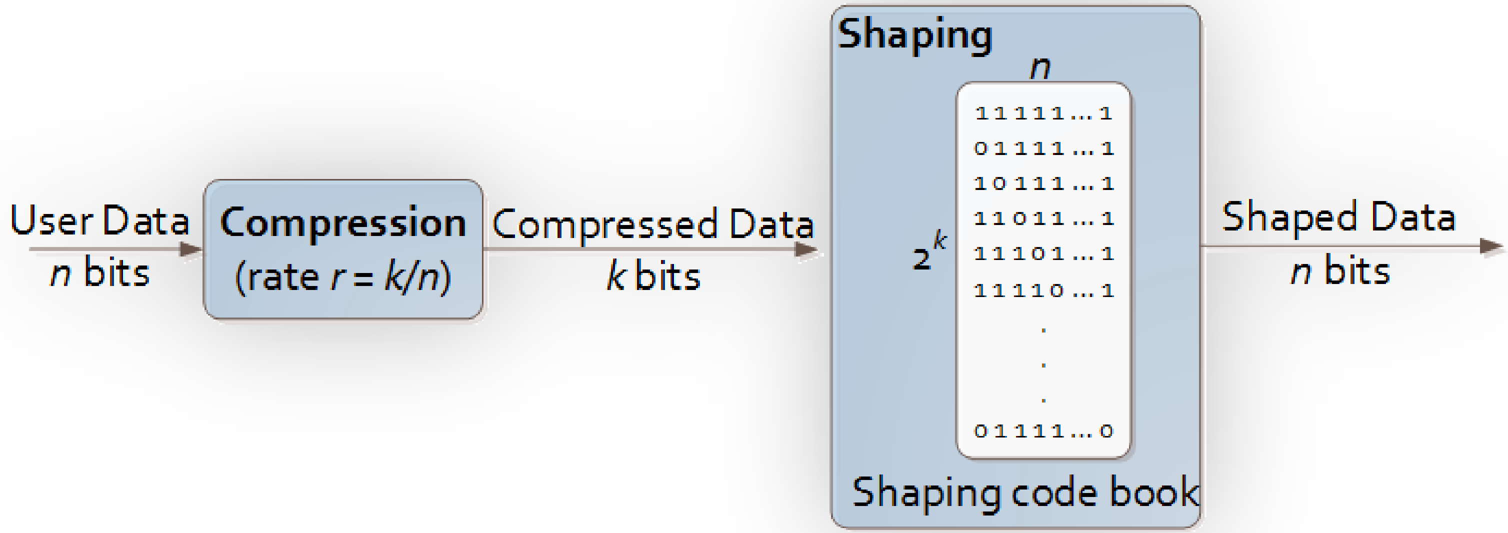
Achievable Shaping Level as a function of Data Compressibility

- What is the achievable shaping level of the Compression- Expansion approach?
- Assume data compression rate $r=k/n$, an optimal shaping code book will include all the 2^k length n binary vectors having a minimal number of 0's j, up to at most m.

$$\Rightarrow 2^k = \sum_{j=0}^{m} \binom{n}{j} \underset{n \to \infty}{\cong} 2^{n \cdot Hb\left(\frac{m}{n}\right)} \quad \Rightarrow H_b\left(\frac{m}{n}\right) = \frac{k}{n} = r$$

where $H_b(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function

$\Rightarrow$ The achievable shaping level of an optimal scheme is: $\boldsymbol{p = \frac{m}{n} = H_b^{-1}(r)}$



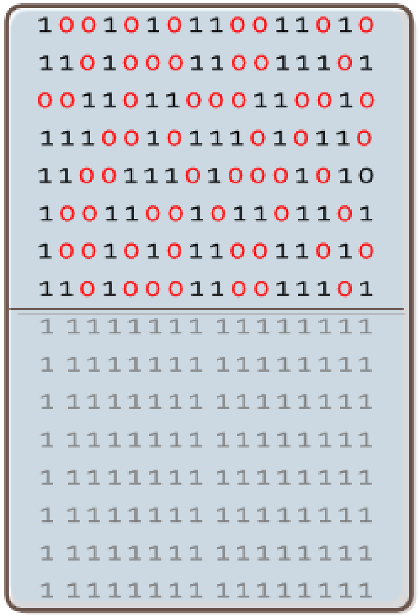
SanDisk

Endurance Enhancement Potential - Shaping Vs. Compression

- **Example:** Compression rate $r = \frac{1}{2}$, achievable shaping level = $p = H_b^{-1}(r) = 0.11$

Reference system
Host data is scrambled
~50% of the cells are programmed per GB

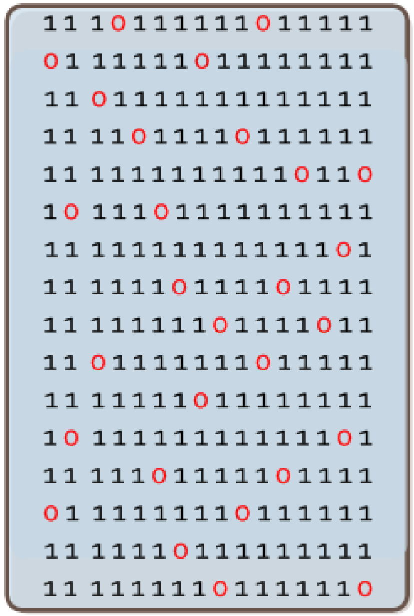
System employing compression
Host data is compressed
~25% of the cells are programmed per GB



Potentially X2 the endurance

(excluding indirect effects like write amplification reduction)

System employing shaping
Host data is shaped
~11% of the cells are programmed per GB



Potentially X2.8 the endurance

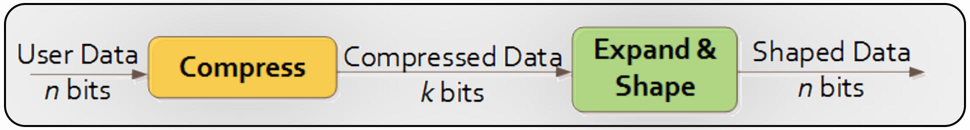
(depending on the specific memory technology and the shaping scheme optimality)

Flash Management Implications - Shaping Vs. Compression

- **Impact on the Flash Management (Backend Firmware):**
  - **Compression** – Significant impact
    Converts *n* bits to *k* bits ($k < n$)
    Changes the logical to physical address management –
    Map logical sectors to variable physical sub-sectors
  - **Shaping** – Transparent to the Flash Management
    Converts *n* bits to *n* bits
    Logical to physical address mapping unchanged –
    Can be considered simply as a different type of scrambler…

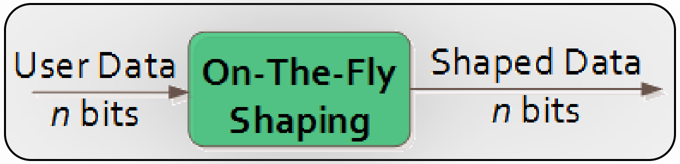
Practical approaches for data shaping

- **Compression – Expansion approach:**



  - Two stage approach:
    - Compress using a lossless compression algorithm – e.g. LZ compression
    - Expand using a shaping code - e.g. Adaptive Reverse Huffman/Run-Length
  - **Pros:** near optimal – can closely approach the theoretical shaping limit
  - **Cons:** High complexity, High power consumption, Large latency (need to support variable compression-expansion rates)

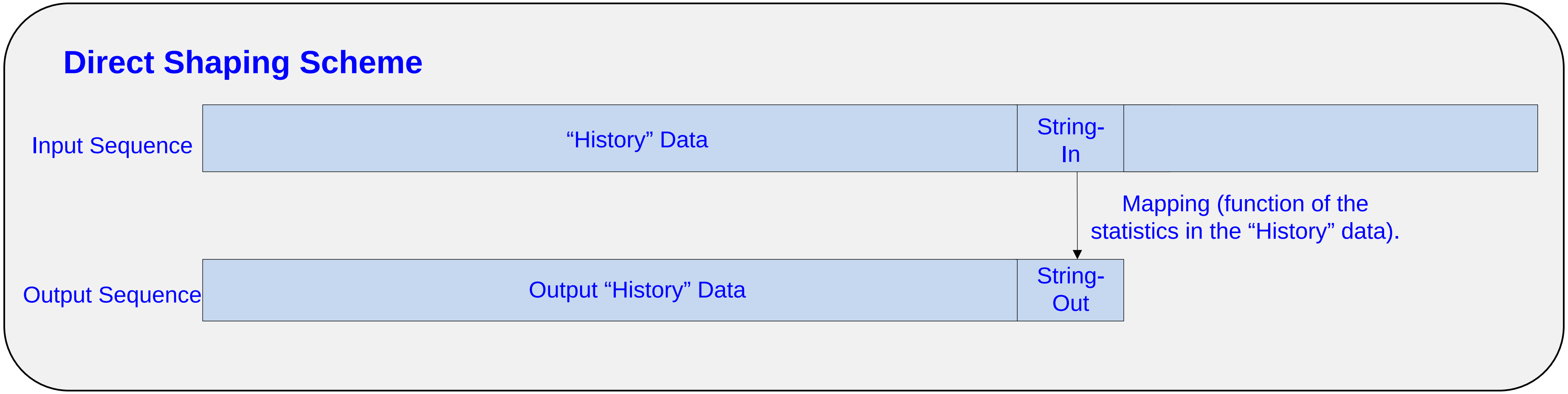
- **Direct shaping approach:**



  - Single stage approach: Direct shaping transformation from n (compressible) input bits to n(shaped) output bits
  - **Pros:**
    - Negligible complexity
    - Negligible power consumption
    - Can be done On-The-Fly at very high throughputs
  - **Cons:** Sub-optimal – achieves lower shaping level than theoretical limit

Direct Shaping

- Transform *n* **compressible** bits into *n* **shaped** bits
- Convert each input string into a shaped output string using an adaptive mapping
- The mapping used for the current input string is a function of the statistics of previous strings, matching the most frequent "historic" strings to the most shaped strings



- **Reversibility**: all mapping decisions are based on the "history" and hence can traced back by the De-Shaping algorithm → No need to store any side information
- **Amenable to an extremely slim design (few Kgates), negligible power consumption, OTF operation at high throughput**

Direct Data Shaping – How does it Work?

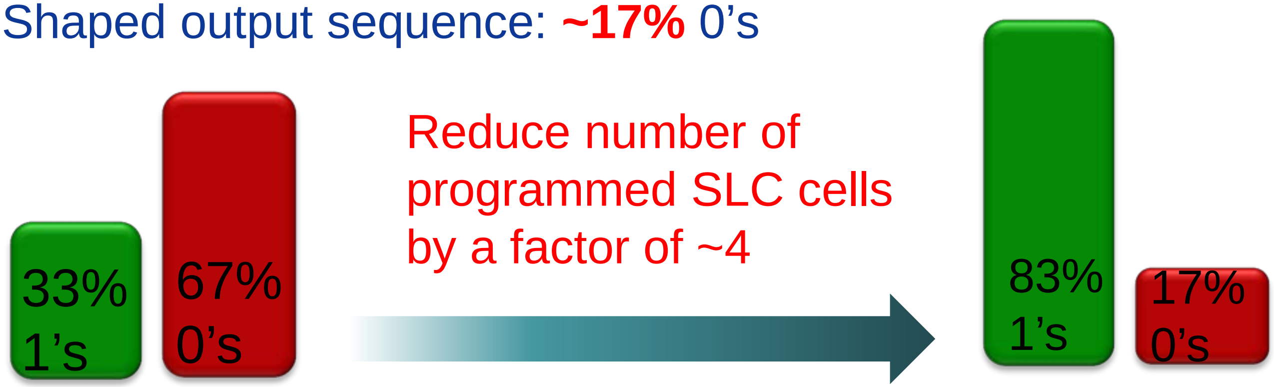
- **Toy example:**
- Convert a 64 bit compressible input sequence into a 64 bit shaped output sequence
- At step *j* map the most frequent 4 bit strings up to step *j*-1 to 4 bit strings with less 0's

input	output	count
1 0 1 0	1 1 1 1	9
0 0 0 0	1 1 1 0	4
0 0 1 0	1 1 0 1	2
1 0 0 0	1 0 1 1	1
0 0 0 1	0 1 1 1	0
0 1 0 0	1 1 0 0	0
0 0 1 1	1 0 1 0	0
0 1 0 1	0 1 1 0	0
1 0 0 1	1 0 0 1	0
0 1 1 0	0 1 0 1	0
1 1 0 0	0 0 1 1	0
0 1 1 1	1 0 0 0	0
1 0 1 1	0 1 0 0	0
1 1 0 1	0 0 1 0	0
1 1 1 0	0 0 0 1	0
1 1 1 1	0 0 0 0	0

Compressible input sequence: **~67%** 0's

1010	0010	1010	0000	1010	1010	1000	0000	1010	1010	1010	0000	0010	1010	1010	0000

Shaped output sequence: **~17%** 0's



SanDisk

Shaping Advantages - Summary

1. **Reduced cell wearing** (main motivation)
2. **Less disturb effects**
3. **Higher ECC capability**

Orthogonal advantages

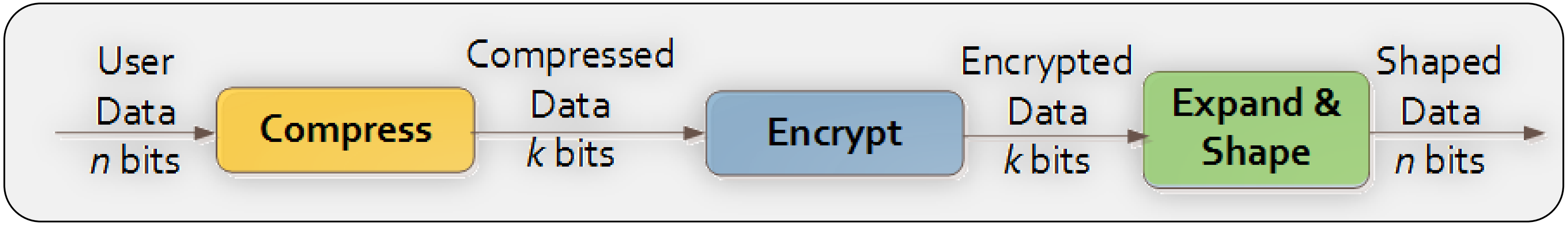
Shaping gain is threefold

- First advantage is **cumulative** – cell wearing is a function of the entire history of shaped and non-shaped data that was programmed to it
- The second two advantages are **local** – observed only when currently programmed data is shaped - improve the average performance, power and reliability

4. **Negligible complexity & power, High throughput On-The-Fly operation**
5. **FW transparent – can be considered as a different type of scrambling**

Leveraging Low Host Data Entropy - Ecosystem Considerations

- Data encryption results in high data entropy (randomizes the data)
- Data encryption at the host side should be avoided in order to take advantage of the low host data entropy via compression or shaping
- Encryption and Data Shaping can co-exist if they are performed at the memory controller level in the following order:
  - Compress
  - Encrypt
  - Expand via shaping



SanDisk

Summary

- **Analyzing mobile traffic reveals low host data entropy**
  - ~0.5 average compression rate measured for sample usage on an Android based Smart Phone
- **Unrealized potential: the inherent "redundancy" in the host data can be leveraged for improving endurance, reliability, performance and power**
  - Apply methods of Deduplication, Compression and Shaping
- **Shaping provides a FW transparent low complexity & power approach for taking advantage of the low host data entropy**
  - Reduced cell wearing
  - Reduced error rates
  - Increased error correction capability

  } Improved endurance, performance and reliability
- **Ecosystem cooperation is required in order to take advantage of the low host data entropy, under security and encryption requirements**

Thank you!

Questions?

Contact:

eran.sharon@sandisk.com or

stop by SanDisk booth # 204

© 2013 SanDisk Corporation. All rights reserved. SanDisk is a trademark of SanDisk Corporation, registered in the United States and other countries. iNAND and Adaptive Flash Management are trademarks of SanDisk Corporation. Other brand names mentioned herein are for identification purposes only and may be the trademarks of their respective