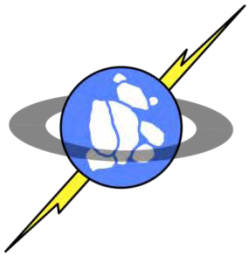
Characterizing Flash Memory:
Beyond the Datasheet

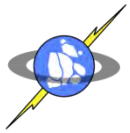
Laura M. Grupp*, Adrian M. Caulfield*, Joel Coburn*,
Eitan Yaakobi†, Steven Swanson*, Paul H. Siegel†

*Non-volatile Systems Laboratory, Department of Computer Science and Engineering
†Center for Magnetic Recording Research
Jacob's School of Engineering
University of California, San Diego



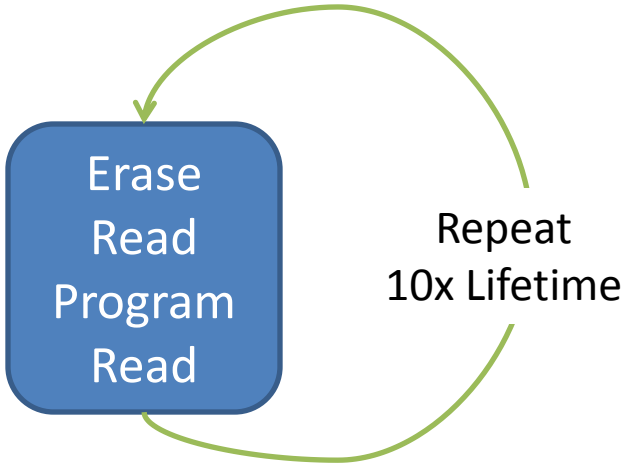
Possible Applications

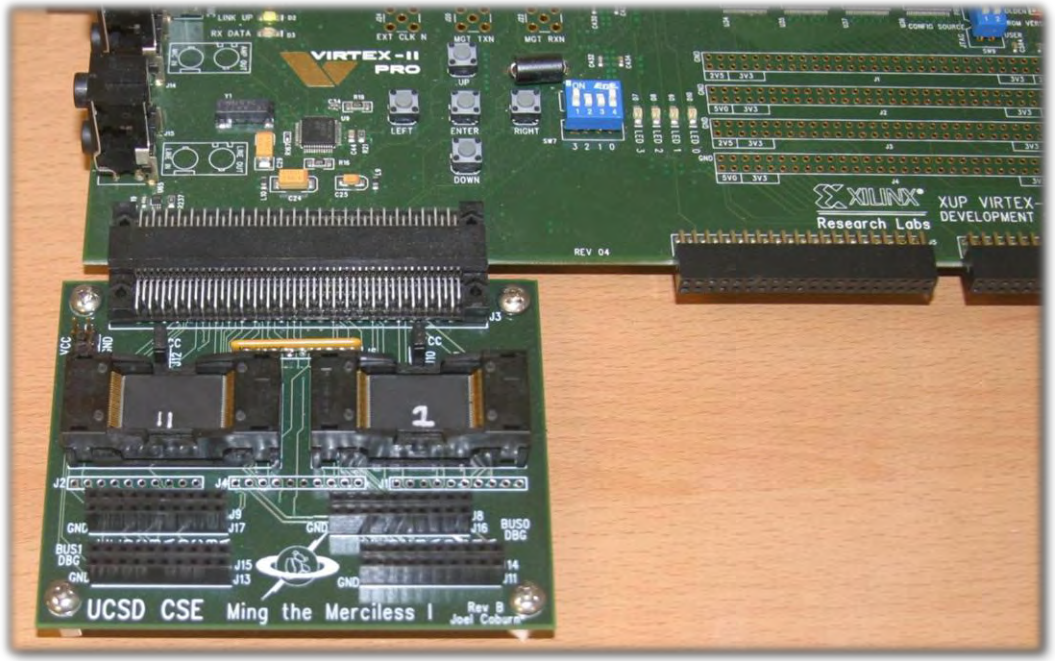
- Flash Translation Layers (FTLs)
- Operating System control
- Data encodings
- Heterogeneous SSD
- Data retention time



Test Setup

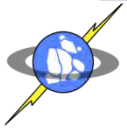
- Custom-Built Daughter Board
- Xilinx XUP Board
- Full-fledge Linux
- Kernel module





The Test Subjects

Chip Name	Manufacturer	Capacity (GB)	Page Size (B)	Pages/ Block	Block/ Plane	Planes/ Die	Dies
A-SLC2	A	2	2048	64	1024	2	1
A-SLC4	A	4	2048	64	4096	1	1
A-SLC8	A	8	2048	64	4096	2	1
B-SLC2	B	2	2048	64	2048	1	1
B-SLC4	B	4	2048	64	2048	2	1
E-SLC8	E	8	2048	64	4096	1	2
B-MLC8	B	8	2048	128	4096	1	1
B-MLC32	B	32	4096	128	2048	2	2
C-MLC64	C	64	8192	128	4096	1	2
D-MLC32	D	32	4096	128	4096	1	2
E-MLC8	E	8	4096	128	1024	1	2



The Tests

Quantify known complexities, look for new ones

- Performance

- Energy Efficiency

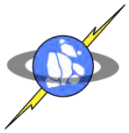
- Reliability

The Tests

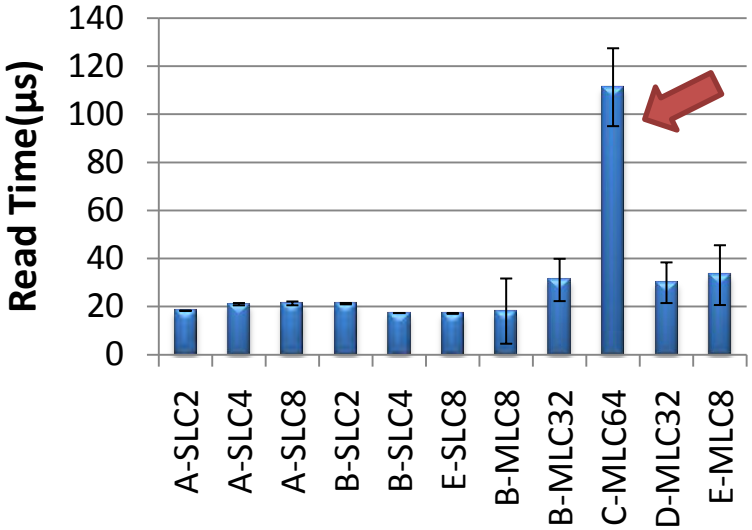
Quantify known complexities, look for new ones

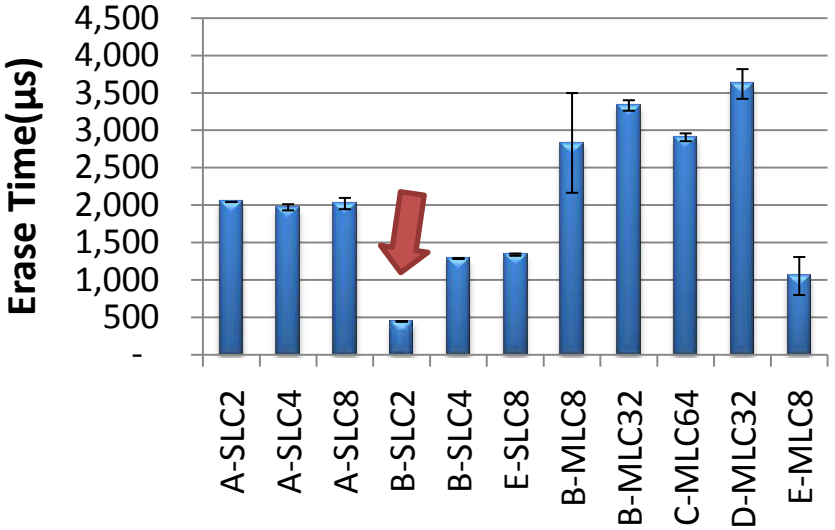
- **Performance**

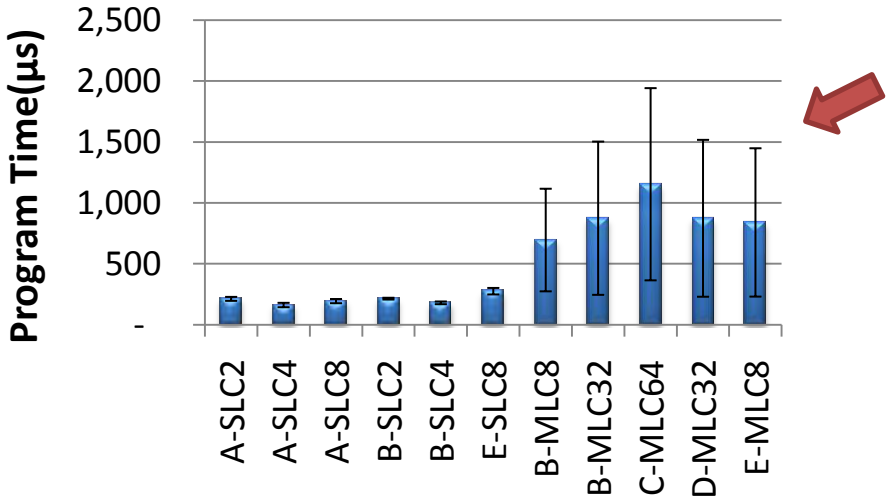
- Energy Efficiency
- Reliability



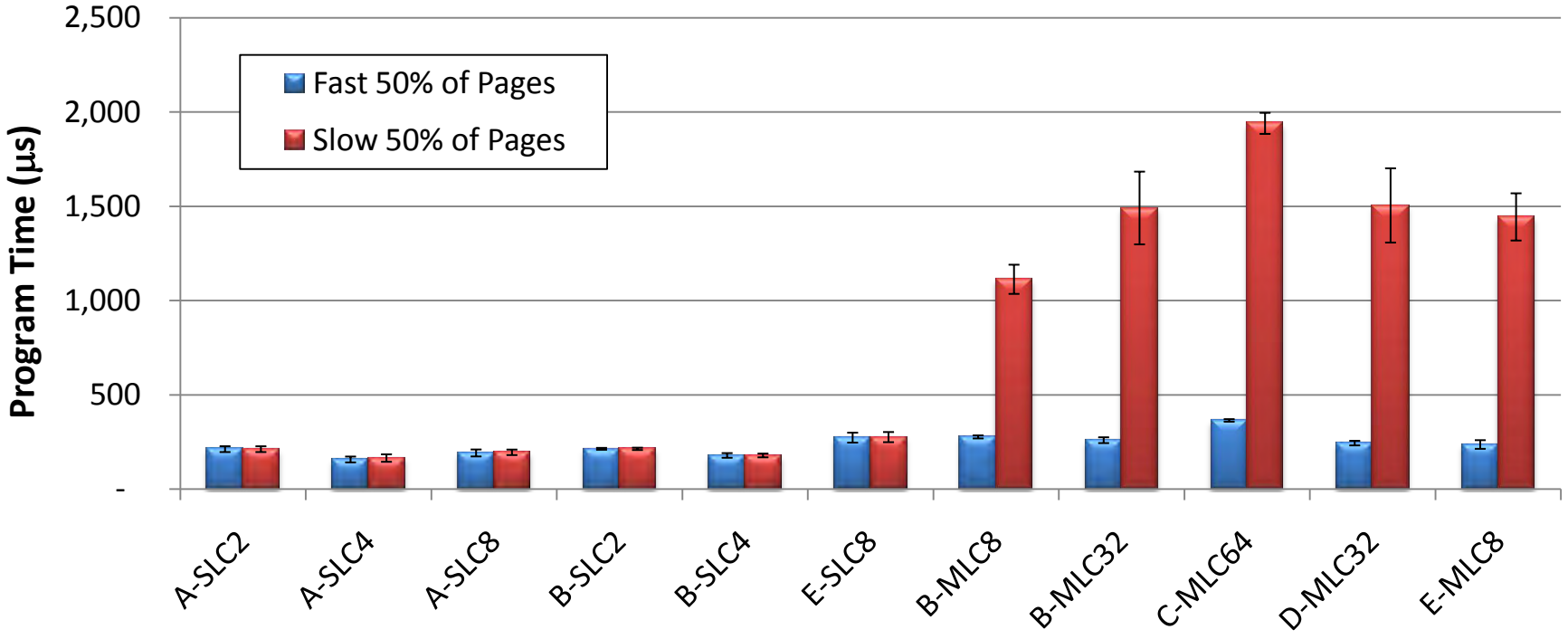
Operation Latency

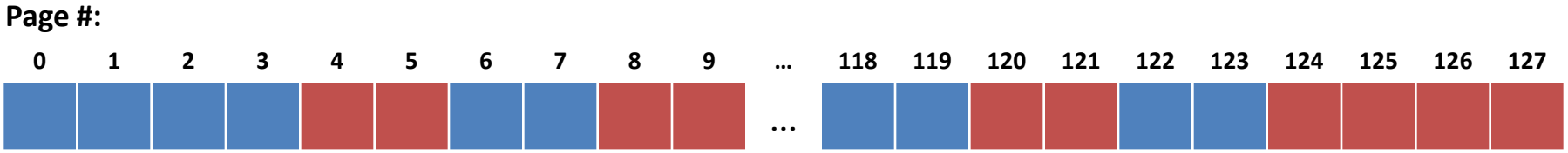






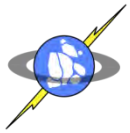
Program Speed Anomaly





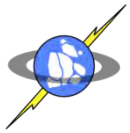
Variation-Aware FTLs

- Can now embellish existing FTLs
- Page based FTL from MSR [Birrell et. al., 2005]
  - "High priority" == write to fast page
  - "Low priority" == any page
- Effects of Skipping slow pages
  - Lower latency when it matters
  - Increased wear



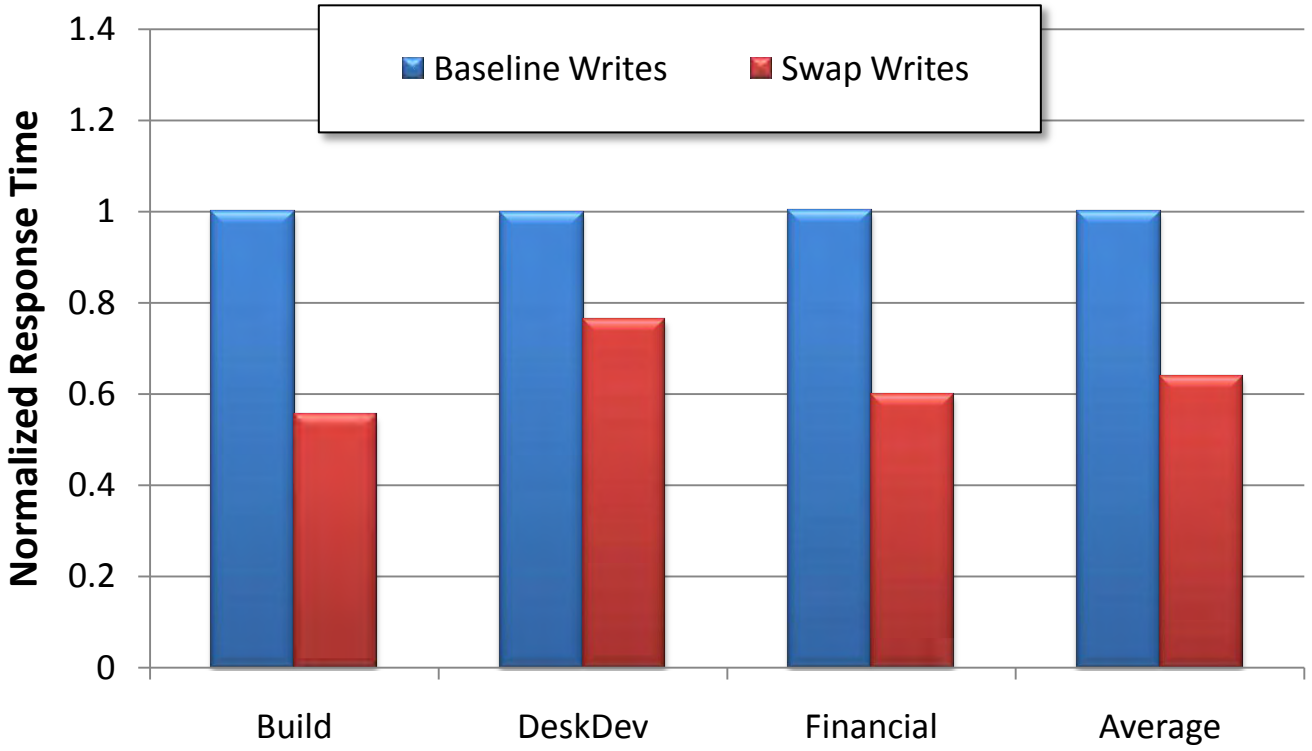
Improving Paging Latency

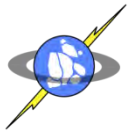
- Paging out Virtual Memory
  - 5-20% of total accesses
  - High Priority
- Goal: reduce swap latency
- Side effect: Increased Wear



Results

- Swap Writes: 1.5x faster
- Wear increased by only 3%





The Tests

Quantify known complexities, look for new ones

- Performance
- **Energy Efficiency**
- Reliability

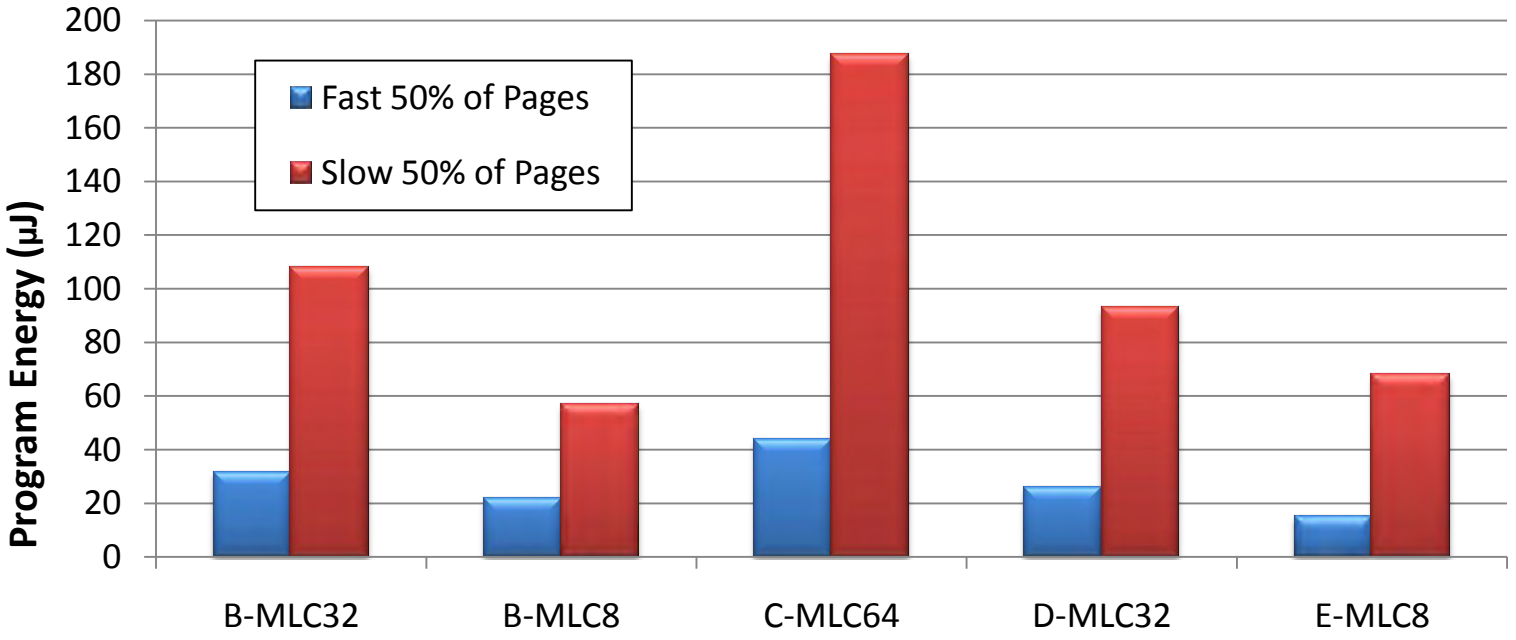
Power

Operation	SLC	MLC
Read	15.8 – 54.5 mW	10.6 – 123.5 mW
Program	32.8 – 88.5 mW	15.2 – 141.3 mW
Erase	18.3 – 50.3 mW	26.7 – 112.3 mW
Idle	1.9 – 17.7 mW	8.2 – 27.5 mW

Programming Energy

Pages have

- **Similar power**
- **Dissimilar latencies**

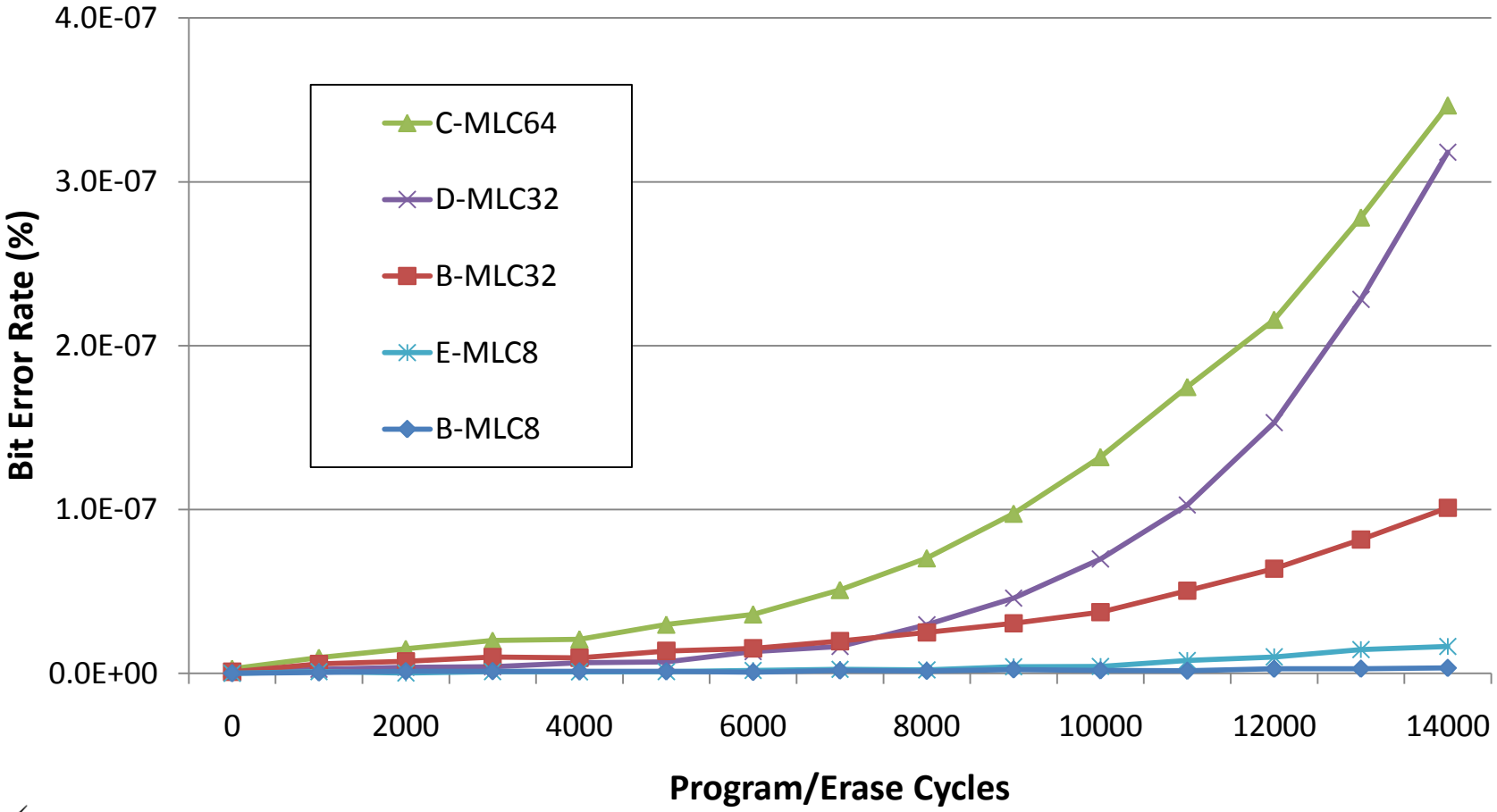


The Tests

Quantify known complexities, look for new ones

- Performance
- Energy Efficiency
- **Reliability**

Error Rates


4.0E-07
3.0E-07
2.0E-07
1.0E-07
0.0E+00
Bit Error Rate (%)
C-MLC64
D-MLC32
B-MLC32
E-MLC8
B-MLC8
0
2000
4000
6000
8000
10000
12000
14000
Program/Erase Cycles

Program Disturb - Procedure

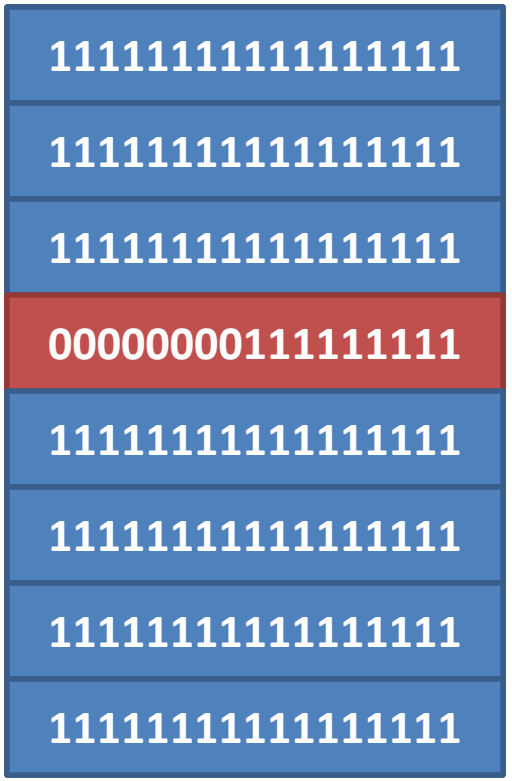
Procedure:

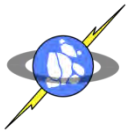
Erase Block

Repeat:

Program Each Page

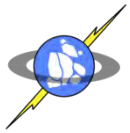
Use 0's on half of one page





Program Disturb - Results

- SLC: no errors for at least one reprogram
- MLC: errors for reprograms of certain pages



Write-Once Memory (WOM) Codes

Reprogram

Write-Once Memory

1. 01110110
2. 10000111

Logical Bits	First Generation	Second Generation
00	111	000
01	110	001
10	101	010
11	011	100

Program:	01	11	01	10
Reprogram:	10	00	01	11

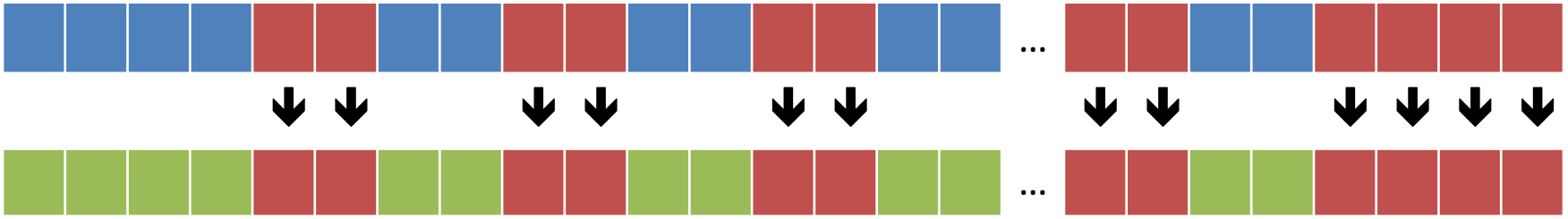
Physical	**110**	**011**	**110**	**101**	**1st Gen.**
Physical	**010**	**000**	**110**	**100**	**2nd Gen.**

WOM Codes and Flash

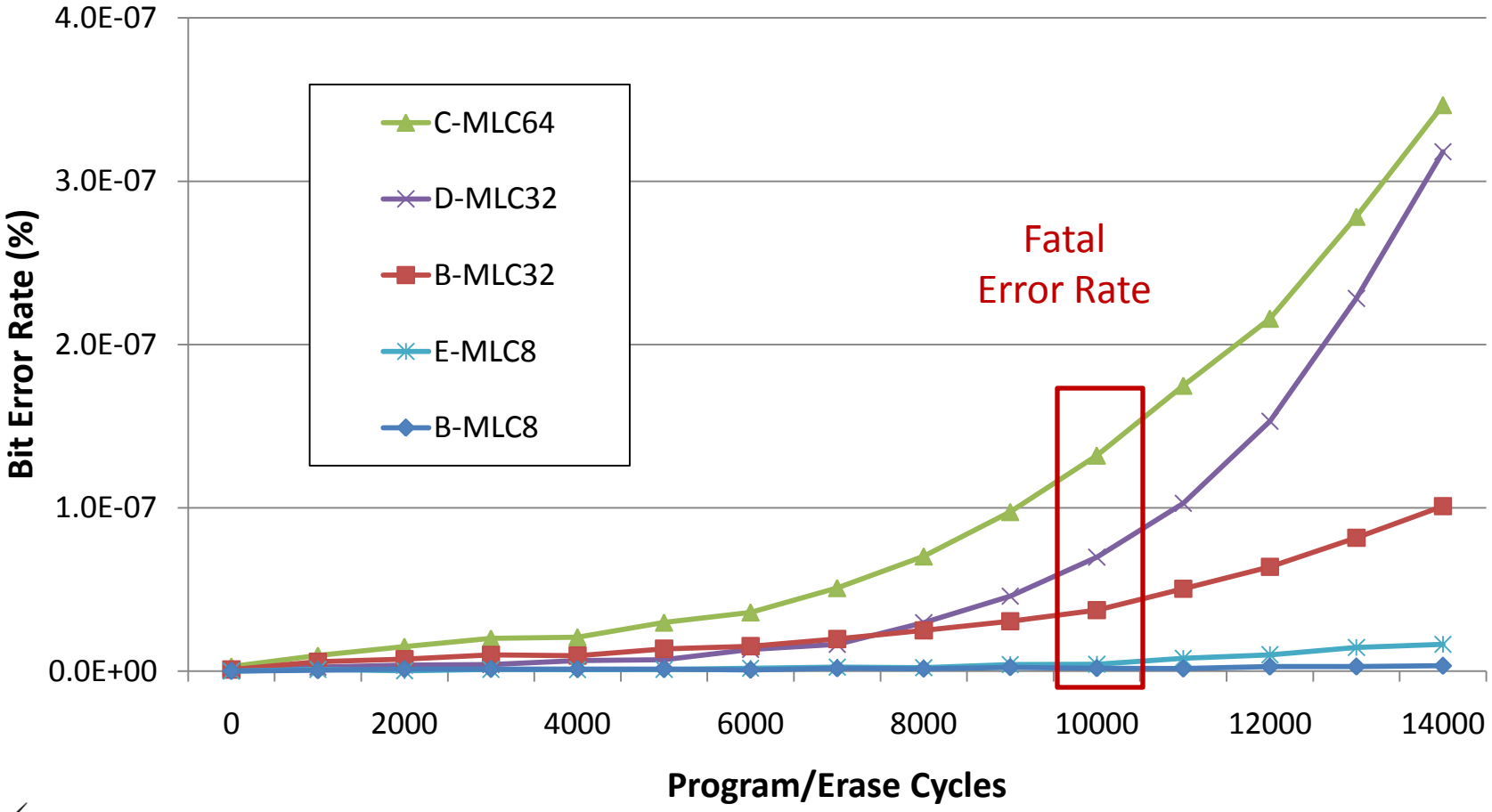
Procedure: Erase, Program, Reprogram, Repeat

WOM-safe: use WOM encoding

WOM-unsafe: use the same un-encoded data both times

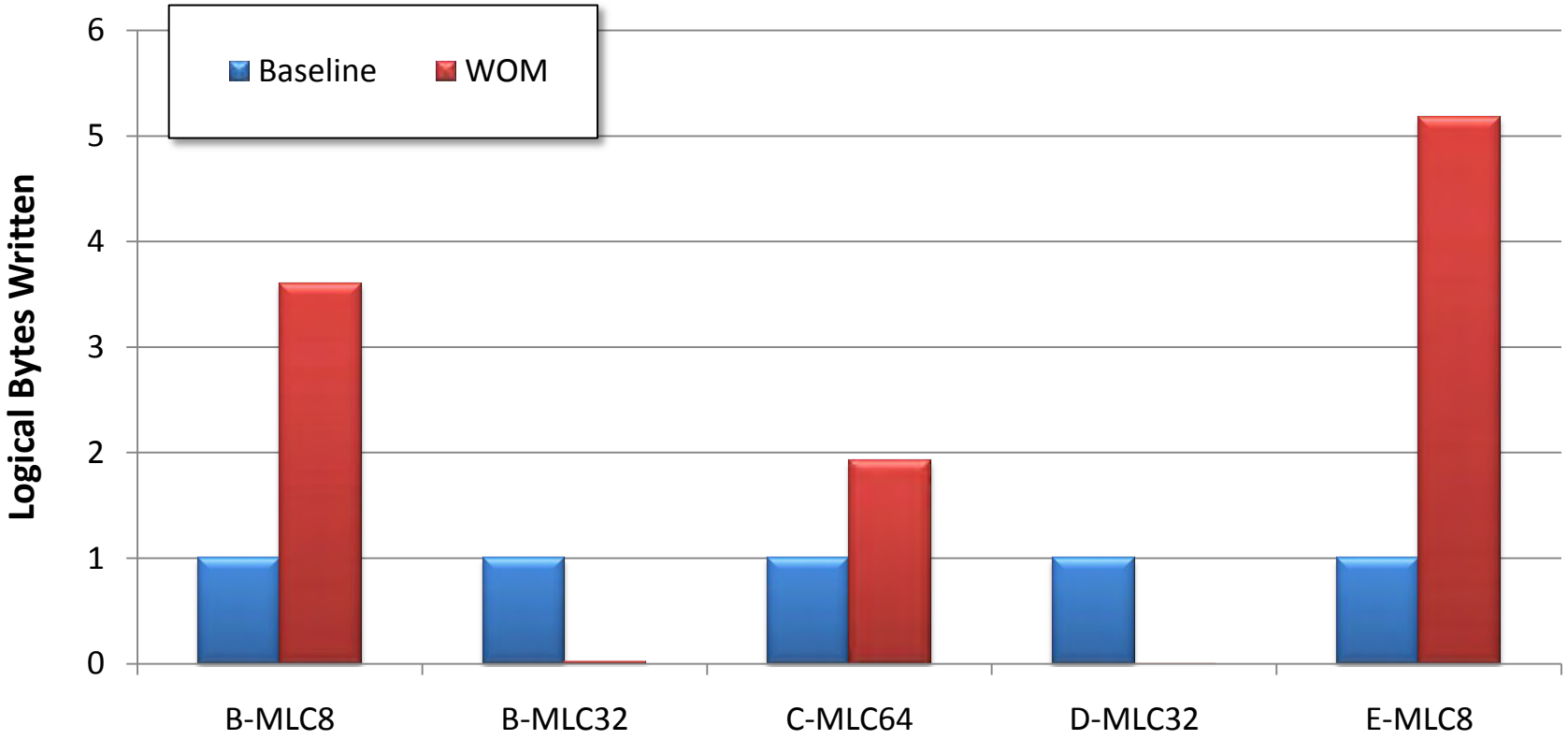


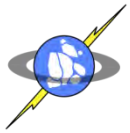
Error Rates


4.0E-07
3.0E-07
2.0E-07
1.0E-07
0.0E+00
Bit Error Rate (%)
C-MLC64
D-MLC32
B-MLC32
E-MLC8
B-MLC8
Fatal
Error Rate
0
2000
4000
6000
8000
10000
12000
14000
Program/Erase Cycles

WOM Codes – Lifetime Extension

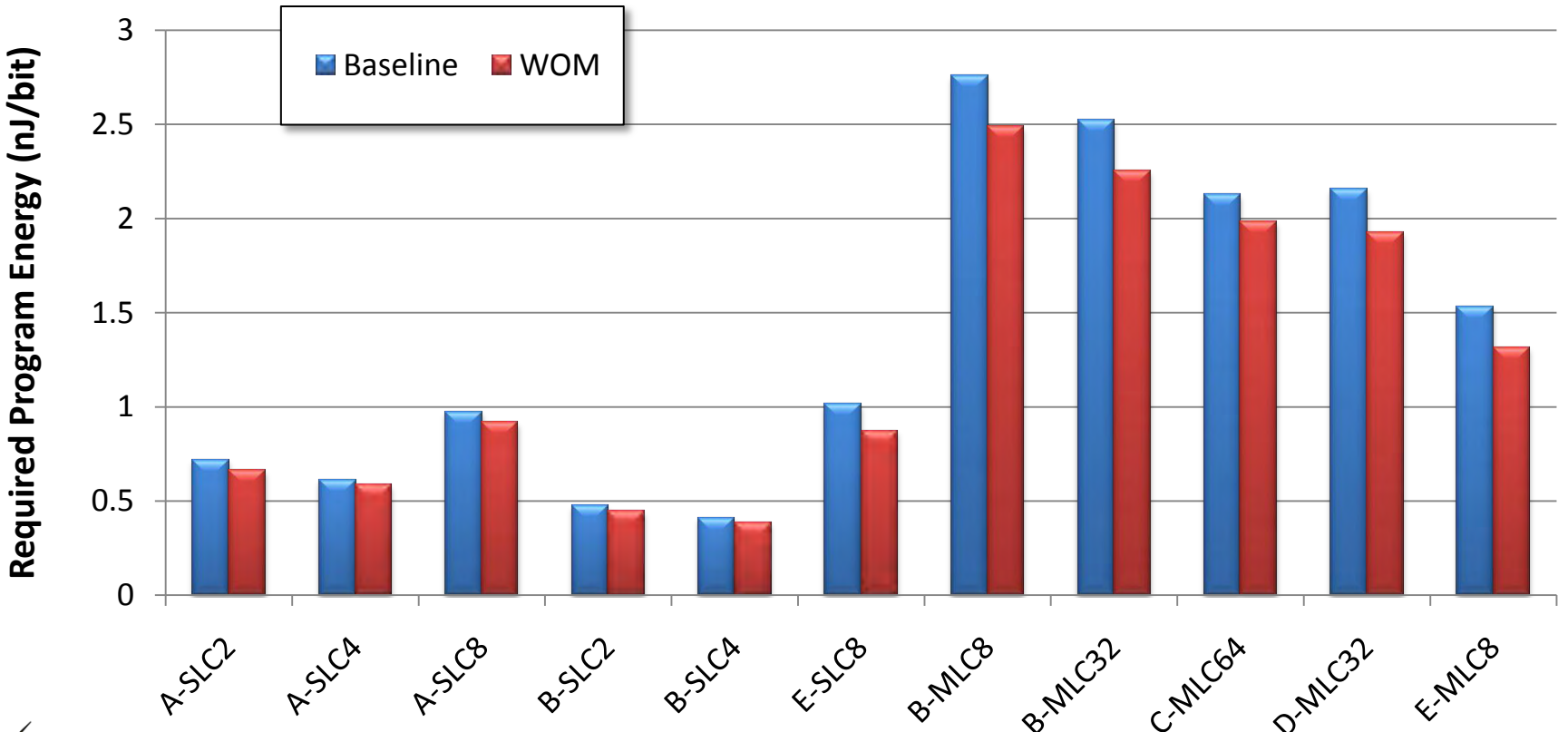
Fatal Error Rate: error rate at quoted lifetime

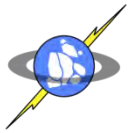




WOM Codes – Energy Reduction

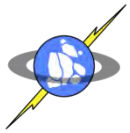
Fewer erases per written bit





Conclusion

- Aim: how to make the best use of flash
- Wealth of data beyond the data sheet
- Tune the interface or application



Thank You