Storage Coding for Wear Leveling in Flash Memories

Anxiao JIANG *ajiang@cs.tamu.edu* Texas A&M

Robert MATEESCU *mateescu@caltech.edu* Caltech

Eitan YAAKOBI *eyaakobi@ucsd.edu* UC San Diego

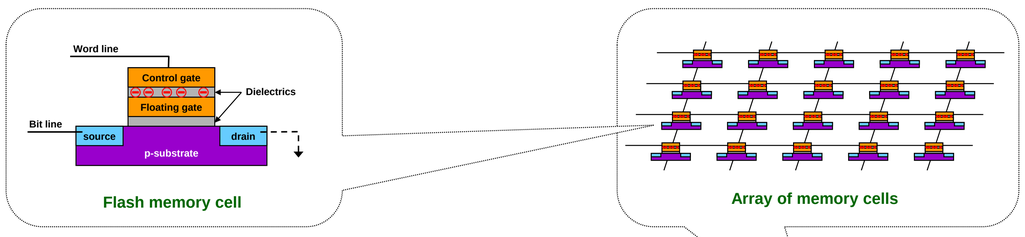
Jehoshua BRUCK *bruck@caltech.edu* Caltech

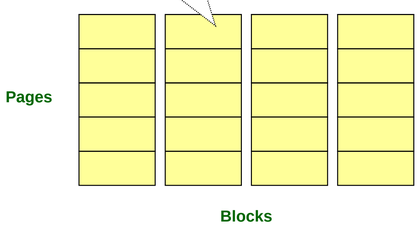
Paul H. SIEGEL *psiegel@ucsd.edu* UC San Diego

Alexander VARDY *avardy@ucsd.edu* UC San Diego

Jack K. WOLF *jwolf@ucsd.edu* UC San Diego

Flash Memory





NVM Non-Volatile Memory

EEPROM Electrically Erasable Programmable Read-Only Memory

Erasing Data

ERASE by BLOCKS

ERASURE is DESTRUCTIVE for the physical substrate (10^4 ~ 10^5 erasures)

Challenge

Given the destination block for each page, reorganize data

while MINIMIZING BLOCK ERASURES

5	6	8	6	1	1	4	8
3	7	6	3	2	4	3	5
2	7	2	1	4	5	8	7

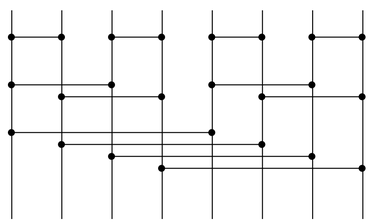
→

1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8
1	2	3	4	5	6	7	8

Hall Theorem Decomposition

5	6	8	6	1	1	4	8
3	7	6	3	2	4	3	5
2	7	2	1	4	5	8	7

	6	8				4	
3				2			
			1		5		7



Sorting network

We can pick one number from each column such that they are all different and form a permutation

With **two** extra blocks, **O(n log n)** erasures can be achieved with an AKS [Ajtai,Komlós,Szemerédi 1983] sorting network, and no coding techniques.

Simple Coding Techniques - XOR

Hall theorem decomposition of matrix into "lines" - permutations of {1,…,n}

Process blocks in a fixed order, forward and backward:

Forward: each page maintains old + new (XOR) information

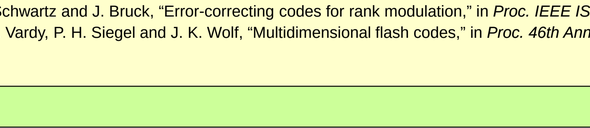
Backward: decode new information for each page

With **one** extra block, **2n** erasures can be achieved with efficient XOR coding techniques.

Permutation decomposition into cycles

	6	8				4	
3				2			
			1		5		7

Each "line" is a permutation made of cycles



b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	
	3	6	8	1	2	5	4	7	
3+1		6	8	1	2	5	4	7	$b_1 + b_4$
3+1	6+2		8	1	2	5	4	7	$b_2 + b_5$
3+1	6+2	8+3		1	2	5	4	7	$b_3 + b_0 + b_4$
3+1	6+2	8+3	1+4		2	5	4	7	$b_4 + b_7$
3+1	6+2	8+3	1+4	2+5		5	4	7	$b_5 + b_6$
3+1	6+2	8+3	1+4	2+5	5		4	7	copy b_6
3+1	6+2	8+3	1+4	2+5	5	4+7		7	$b_7 + b_8$
3+1	6+2	8+3	1+4	2+5	5	4+7	7		copy 8
3+1	6+2	8+3	1+4	2+5	5	4+7	7		$b_7 + b_6 + b_3 + b_0 + b_2$
3+1	6+2	8+3	1+4	2+5	5	4+7		8	$b_6 + b_3 + b_0 + b_2 + b_8$
3+1	6+2	8+3	1+4	2+5	5		7	8	$b_5 + b_4 + b_1$
3+1	6+2	8+3	1+4	2+5		6	7	8	$b_4 + b_1 + b_6$
3+1	6+2	8+3	1+4		5	6	7	8	$b_3 + b_0 + b_2 + b_8$
3+1	6+2	8+3		4	5	6	7	8	$b_2 + b_8$
3+1	6+2		3	4	5	6	7	8	$b_1 + b_6$
3+1		2	3	4	5	6	7	8	$b_0 + b_3$
	1	2	3	4	5	6	7	8	

References

[1] M. Ajtai, J. Komlós and E. Szemerédi, "An O(n log n) sorting network," in *Proc. ACM Symposium on Theory of Computing*, pp. 1–9, 1983.
[2] K.E. Batcher, "Sorting networks and their applications," in *Proceedings of the AFIPS Spring Joint Computer Conference*, pp. 307–314, 1968.
[3] B. Bollobas, *Modern Graph Theory*, Chapter 3, Springer, 2002.
[4] Y. Cassuto, M. Schwartz, V. Bohossian and J. Bruck, "Codes for asymmetric limited-magnitude errors with application to multi-level flash memories," in *Proc. IEEE ISIT*, 2007.
[5] H. Finucane, Z. Liu and M. Mitzenmacher, "Designing floating codes for expected performance," in *Proc. 46th Annual Allerton Conference*, 2008.
[6] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," in *ACM Computing Surveys*, vol. 37, no. 2, pp. 138-163, June 2005.
[7] A. Jiang, V. Bohossian and J. Bruck, "Floating codes for joint information storage in write asymmetric memories," in *Proc. ISIT*, 2007, pp. 1166-1170.
[8] A. Jiang, R. Mateescu, M. Schwartz and J. Bruck, "Rank modulation for flash memories," in *Proc. IEEE ISIT*, 2008, pp. 1731-1735.
[9] A. Jiang, R. Mateescu, E. Yaakobi, J. Bruck, P. H. Siegel, A. Vardy and J. K. Wolf, "Storage coding for wear leveling in flash memories," Caltech Tech. Rep., online: *http://www.paradise.caltech.edu/etr.html*.
[10] A. Jiang, M. Schwartz and J. Bruck, "Error-correcting codes for rank modulation," in *Proc. IEEE ISIT*, 2008, pp. 1736-1740.
[11] E. Yaakobi, A. Vardy, P. H. Siegel and J. K. Wolf, "Multidimensional flash codes," in *Proc. 46th Annual Allerton Conference*, 2008.