

Empirical Evaluation of NAND Flash Memory Performance

Peter Desnoyers
Northeastern University
360 Huntington Ave.
Boston, MA 02115
pjd@ccs.neu.edu

ABSTRACT

Reports of NAND flash device testing in the literature have for the most part been limited to examination of circuit-level parameters on raw flash devices or prototypes, and system-level parameters on entire storage subsystems. However, there has been little examination of system-level parameters of raw devices, such as mean latency and endurance values.

We report the results of such tests on a variety of devices. Read, program, and erase latency were found to align closely with manufacturer’s specified “typical” values in almost all cases. Program/erase endurance, however, was found to exceed specified minimum values, often by as much as two orders of magnitude. In addition significant performance changes were found to occur with wear, providing mechanisms which may be used to track this wear as well as bearing significant implications for system performance over the lifespan of a device. Finally, random write patterns which incur performance penalties on current flash-based memory systems were found to incur no overhead on the devices themselves.

1. INTRODUCTION

Fixed magnetic disk has been the predominant media for secondary storage for over three decades. In the last five years, however, solid state storage in the form of NAND flash memory has come into increasing use, becoming the first competitor to magnetic disk storage to gain significant commercial acceptance.

With the increasing use of flash-based secondary storage, detailed understanding of behavior which affects operating system design and performance becomes important. However, while disk behavior has been extensively studied, there appear to be few sources for the information needed to predict performance and reliability of flash-based storage systems. Detailed studies of low-level electrical characteristics are available [5, 6, 9], as well as performance studies of complete storage assemblies (e.g. SSDs) containing flash devices and controllers [1, 8]. However, to the best of our knowl-

edge, there is no experimental study to date of actual flash devices giving measured values for read, write, and erase speed, power consumption, or write/erase longevity.

This paper reports measurements of these parameters for a range of raw flash devices. We focus on devices themselves, rather than flash-based systems such as USB drives or SSDs, in order to understand the capabilities and limitations of the underlying technology rather than that of any particular implementation.

Of the results found, the most unexpected were these:

- High write/erase endurance. Although NAND flash memory degrades with repeated write/erase cycles, measured lifetime varies greatly, and is often as much as two orders of magnitude higher than manufacturer specifications.
- Wear-dependent performance changes. On all devices tested, repeated write/erase cycling of a single block decreases write time and increases erase time—by as much as a factor of three or more—as that block wears out, changing overall system performance as well as providing a predictor of individual block failure.
- Random write speed. Although flash-based storage systems such as SSDs may have poor random write performance [1], the chips themselves perform as well on random writes as sequential ones.

In the remainder of this paper we first present an overview of flash memory technology from a system perspective in Section 2, followed by experimental results (Section 3) and conclusions (Section 4).

2. BACKGROUND

NAND flash is a form of electrically erasable programmable read-only memory based on a particularly space-efficient basic cell, optimized for mass storage applications. Unlike most memory technologies, NAND flash is organized in *pages* of typically 2K or 4K bytes which are read and written as a unit. Unlike block-oriented disk drives, however, pages must be erased in units of *erase blocks* comprising multiple pages—typically 32 to 128—before being re-written.

2.1 Technical Overview

To inform our discussion we present an overview of the circuit and electrical aspects of flash technology which are relevant to system software performance; a deeper discussion of these and other issues may be found in the survey by Sanvido *et al* [10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HotStorage '09 Big Sky, Montana
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

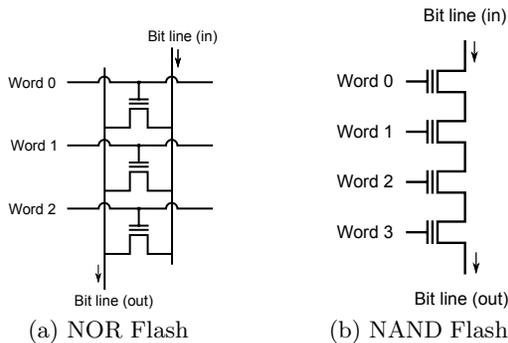


Figure 1: Flash circuit structure. NAND flash is distinguished by the series connection of cells along the bit line, while NOR flash (and other memory technologies) arrange cells in parallel between two bit lines.

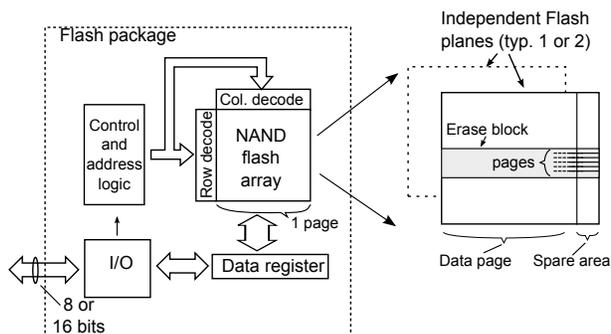


Figure 2: Typical flash device architecture. Read and write are both performed in two steps, consisting of the transfer of data over the external bus to or from the data register, and the internal transfer between the data register and the flash array.

The basic cell in a NAND flash is a MOSFET transistor with a floating (i.e. oxide-isolated) gate. Charge is tunneled onto this gate during write operations, and removed (via the same tunnelling mechanism) during erasure. This stored charge causes changes in V_T , the threshold or turn-on voltage of the cell transistor, which may then be sensed by the read circuitry. NAND flash is distinguished from other flash technologies (e.g. NOR flash, E^2 PROM) by the tunnelling mechanism (Fowler-Nordheim or FN tunnelling) used for both programming and erasure, and the series cell organization shown in Figure 2.1(b).

Many of the more problematic characteristics of NAND flash are due to this organization, which eliminates much of the decoding overhead found in other memory technologies. In particular, in NAND flash the only way to access an individual cell for either reading or writing is *through* the other cells in its bit line. This adds significant noise to the read process, and also requires care during writing to ensure that adjacent cells in the string are not disturbed. During erasure, in contrast, all cells on the same bit string are erased.

In order to ensure precise programming and erasure in the face of process, temperature, and other variations, an internal state machine repeatedly programs (or erases) a page and reads it back until the operation has succeeded. Earlier generations of NAND flash (and high-performance modern

devices) use what is termed Single-Level Cell (SLC) technology, storing a single bit on each cell. High-capacity Multi-Level Cell (MLC) devices use more than two levels for each cell, storing 2 to as many as 4 [12] bits each.

With few exceptions today’s flash devices correspond to the block diagram in Figure 2.1. Cells are arranged in pages, typically containing 2K or 4K bytes plus a spare area of 64 to 256 bytes for system overhead. Between 16 and 128 pages make up an *erase block*, or *block* for short, which are then grouped into a flash *plane*. Devices may contain independent flash planes, typically storing odd and even blocks, allowing simultaneous operations for higher performance. Finally, a static RAM buffer holds data before writing or after reading, and data is transferred to and from this buffer via an 8- or 16-bit wide bus.

This architecture evolved to meet storage demands for digital photography and MP3 players, with modest performance requirements and strict cost constraints. This is reflected in current flash devices, with low-cost interfaces limited to a peak bandwidth of 40 MB/sec. More recently, the market for high performance SSDs has generated demand for higher transfer rates, resulting in efforts such as ONFI 2.1 [3] to standardize 100MB/s to 200MB/s DDR interfaces.

In this study we are interested in the performance of basic operations—i.e. writing from the internal buffer to the flash plane, reading from the flash plane to the buffer, or erasing a block. These represent the fundamental performance limits of any particular NAND flash design, while I/O interfaces with sufficient performance are readily available. (e.g. as used in DRAM)

2.2 Related Work

Prior experimental studies of flash memory performance and endurance may be classified as circuit-oriented and system-oriented. Circuit-level studies have examined the effect of program/erase stress on internal electrical characteristics, often using custom-fabricated devices to remove the internal control logic and allow e.g. measurements of the effects of single program or erase steps. A representative study is by Lee et al. at Samsung [6], examining both program/erase cycling and hot storage effects across a range of process technologies. Similar studies include those by Park et al. [9] and Yang et al. [13], both also at Samsung.

System-level studies have instead examined characteristics of entire flash-based storage systems, such as USB drives and SSDs. The most recent of these presents uFLIP [1], a benchmark for such storage systems, with measurements of a wide range of devices; this work quantifies the degraded performance observed for random writes in many such devices. Additional work in this area includes [2] and [8]. There has been a small amount of empirical testing of raw flash devices in the wireless sensor network community [7], but this work has focused primarily on energy usage and has not addressed performance or endurance.

3. EXPERIMENTAL RESULTS

3.1 Methodology

In order to test a wide range of devices, flash chips were acquired both through traditional distributors and by purchasing and disassembling mass-market devices. A programmable flash controller was constructed using software control of general-purpose I/O pins on a micro-controller to

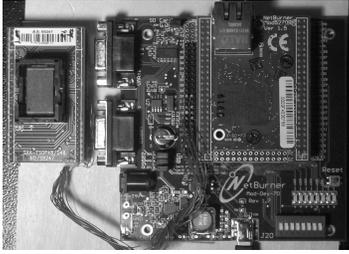


Figure 3: Flash device test apparatus. Test system is based on a NetBurner 5270 controller and TSOP48 programming socket.

Device	Mfr	Size	Cell	Nominal endurance
NAND128W3A2BN	ST	128Mbit	SLC	10^5
HY27US0812JA	Hynix	512Mbit	SLC	10^5
MT29F2G08AAD	Micron	2Gbit	SLC	10^5
MT29F4G08AAC	Micron	4Gbit	SLC	10^5
NAND08GW3B2C	ST	8Gbit	SLC	10^5
MT29F8G08MAAWC	Micron	8Gbit	MLC	10^4
29F16G08CANC1	Intel	16Gbit	SLC	10^5
MT29F32G08QAA	Micron	32Gbit	MLC	10^4

Table 1: Devices tested

implement the flash interface protocol for 8-bit devices; this test setup may be seen in Figure 2.2. Flash devices tested ranged from early 128Mbit (16MB) SLC devices to recent 16Gbit and 32Gbit MLC chips. A complete list of devices tested may be seen in Table 2.2. Unless otherwise specified, all tests were performed at 25° C.

3.2 Endurance

Limited write endurance is a key characteristic of flash memory, and all floating gate devices in general, which is not present in competing memory and storage technologies. As blocks are repeatedly erased and programmed, the oxide layer isolating the gate degrades, as described in more detail in [5]. This in turn causes a change in the response of the cell to a fixed programming or erase step, as shown in Figure 4. In practice this degradation is compensated for by adaptive programming and erase algorithms internal to the device, which use multiple program/read or erase/read steps to achieve the desired state. If a cell has degraded too much, however, the program or erase operation will terminate in an error, after which the external system must consider the block *bad* and remove it from use.

Program/erase endurance was tested by repeatedly programming a single page with all zeroes, and then erasing the containing block. Although rated device endurance ranges from 10^4 to 10^5 program/erase cycles, in Figure 5 we see that measured endurance was higher, often by nearly two orders of magnitude, with a small number of outliers.

Operations were timed by measuring the period during which the device indicated that it was busy after accepting a command, thus eliminating any dependency on the speed at which the test system was able to read or write data over the bus. Timing traces were collected during endurance tests, and a representative trace is shown in Figure 6. Cell degradation of V_T as seen in Figure 4 may be seen affecting the iterative programming and erase algorithms here, as program

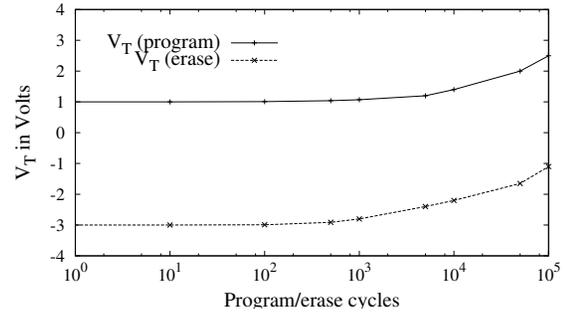


Figure 4: Typical V_T degradation with program/erase cycling. Data is abstracted from [6], [9], and [13].

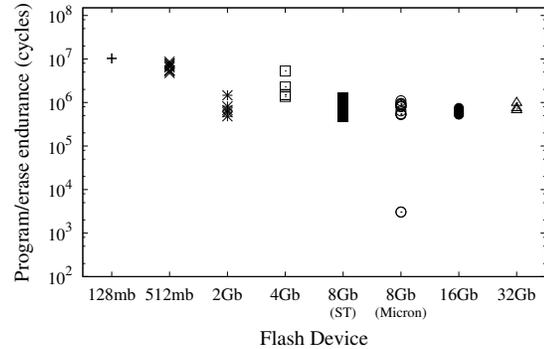


Figure 5: Write/Erase endurance by device. Measured lifetimes of individual blocks are plotted. Nominal endurance of devices tested is 10^5 cycles for all devices except the 8Gb Micron and 32Gb device, which are rated for 10^4 cycles.

times decrease and erase times increase over the lifetime of a block. This effect was seen in all devices tested except for those based on the oldest technology: for the 128Mbit part erase times remained constant and program times decreased, while both remained constant for the 512Mbit device.

3.3 Performance

Read performance was tested under a number of scenarios, including random and sequential reads. Again, latency was measured from the end of the read command until the device indicated that data was ready to be transferred, thus avoiding effects of varying transfer speed. No significant difference was found between random and sequential speeds, nor was read performance seen to vary with program/erase cycling, and so a single average is reported for each device. Results may be seen in Figure 7, where measured speeds are compared to speeds specified by the manufacturer when available.

Specified read latency (across all environmental and circuit conditions) is typically $25\mu s$ for current-generation SLC devices and $50\mu s$ for MLC ones, although early small-page SLC devices are rated at $12\mu s$. Measured speeds under test conditions are seen to be somewhat better than specification, but not by large margins except in the case of the smallest device. We speculate that this anomaly may be due to the device being produced in a newer process technology than it was originally designed for.

As described above, write and erase performance vary over

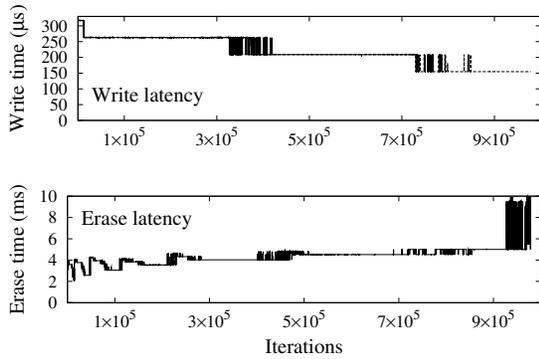


Figure 6: Wear-related changes in latency. Data points are subsampled rather than averaged to illustrate the quantized latency values due to iterative internal algorithms.

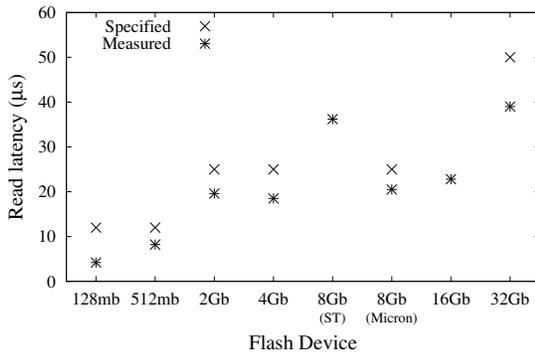


Figure 7: Read latency by device. Measured values were unaffected by access pattern or block wear.

the lifetime of a flash block, complicating the task of summarizing our measurements. The best write performance is obtained just before a block fails; however we hope to rarely if ever operate in this region. The slowest write performance occurs on fresh pages, but may speed up significantly after the first few hundred writes, leading to a sizable difference between expected and worst-case performance.

To address this we report three values for both write and erase: the worst-case latency, seen by the first writes and last erases, mean latency for the first 10000 operations on a block, and the best-case latency as seen by the first erases and last writes. Results are shown in Figures 8 and 9, again compared to manufacturer specifications when available.¹

Experiments were performed to examine the effect of random writes on performance. We note that true random writes are not possible on most flash devices, as the pages within an erase block must be written sequentially in order to avoid disturbing data on previously-written pages. Instead, a random sequence of erase blocks was chosen, and then the first page was written within each block in the sequence, followed by the second in each, etc. No detectable

¹Many test runs for the 4Gbit device showed anomalous write and erase delays, often exceeding the 15ms timeout of the test system; these runs are not included in the calculated results. We are investigating whether these runs reflect true performance of the device, or whether it was due to a malfunction of the test system.

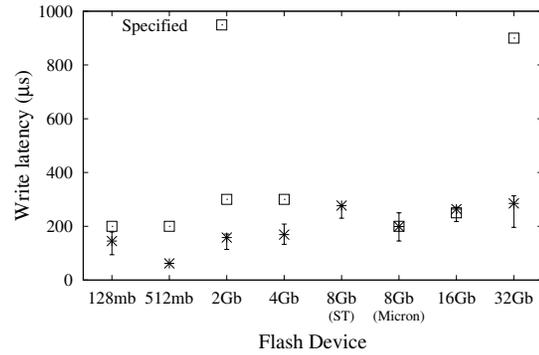


Figure 8: Write latency by device. Values shown are typical (mean of first 10^4 writes to a block), worst-case (mean of first 100 writes), and best-case (mean of last 100 writes before failure).

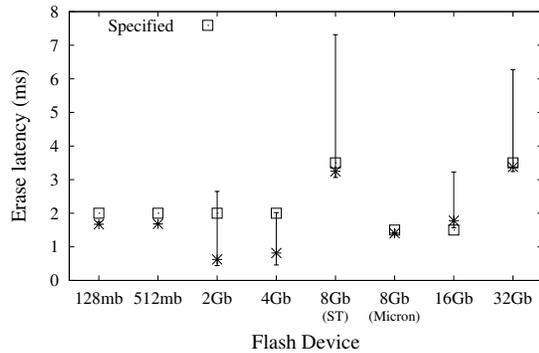


Figure 9: Erase latency by device. Similar to write latency, but the first 100 erasures yield the best case, while the first 10^4 yield the typical value and the last 100 yield the worst-case point.

different in write performance was seen as compared to writing pages sequentially within a single block.

3.4 Additional Testing

Further investigation was performed to determine whether the surprisingly high endurance of the devices tested is typical, or is instead due to anomalies in the testing process. In particular, we varied both program/erase behavior and environmental conditions to determine their effects. Due to the high variance of the measured endurance values, we have not collected enough data to draw strong inferences, and so report general trends instead of detailed results.

Usage patterns: The results reported above were measured by repeatedly programming the first page of a block with all zeroes (the programmed state for SLC flash) and then immediately erasing the entire block. Several devices were tested by writing to all pages in a block before erasing it; endurance appeared to decrease with this pattern, but by no more than a factor of two. Additional tests were performed with varying data patterns, but no difference in endurance was detected.

This result is not unexpected, as we surmise that one way in which erasure or programming fails is when a single cell fails to reach its target state after a certain number of internal program or erase steps. Given some amount of variation between cells, it is not unexpected that changing the state of a larger number of cells would result in a higher chance of failure as cells wear. (We note, however, that repeated erase

cycles with no intervening writes show the same latency increase and similar endurance as erasures with a single intervening page write.)

Environmental conditions: The processes which result in flash failure are exacerbated by heat [13], although internal temperature compensation is used to mitigate this effect [4]. The 16Gbit device was tested at 80° C, and no noticeable difference in endurance was seen. However, at 5° C endurance was seen to drop by a factor of about two. Although not expected, this decrease of endurance at low temperature has also been reported for NOR flash [11].

We note that one of the primary differences between our tests and typical system usage is that cells are erased almost immediately after being programmed. We are curious as to whether endurance would be affected by the passage of time between program and erase or vice versa; however, the long durations required for such tests have precluded their implementation to date.

4. CONCLUSIONS

Many of the results of these tests were expected: read, program (with one exception) and erase times were for the most part slightly lower than the “typical” values specified by the manufacturers, no doubt reflecting a margin to account for variations outside of our test conditions.

The high endurance values measured—often nearly 100 times higher than specified—were highly unexpected and deserve more study. Further investigation is needed to determine whether such high endurance may be expected under typical system conditions, and whether any special care must be taken to achieve such behavior. If real systems are able to achieve average endurance levels of 10^6 or 10^7 write/erase cycles, then it would appear that many of the concerns raised in the systems community have been misplaced, and that flash endurance may merely become another MTBF parameter, much like mechanical failure in disk drives.

The variation in program and erase performance with wear, although obvious in hindsight, was also unexpected. This has obvious applications in wear leveling algorithms, as it supplies a measurement of a block’s remaining lifetime that—unlike explicit erase count tracking—imposes no additional writes to the device. However, it also has implications for block management on flash devices. If the latency of erasures can be hidden, then repeatedly re-using blocks until they fail may yield improved write performance. However, if system performance is impacted by erase latency, then wear should be distributed as evenly as possible in order to avoid high erase latencies at the end of a block’s lifespan.

Additional experimentation is needed to explore the endurance behavior seen in these experiments. How sensitive are these results to environmental and circuit conditions? Do they hold up across a much wider sampling of devices? And perhaps most importantly, how sensitive are they to system behavior—i.e. usage patterns and wear leveling? Work to date has focused on generating usage patterns which avoid exceeding a fixed endurance threshold for any individual block; however, it appears that this endurance level may be variable, and that it may be more profitable to look for patterns which maximize that endurance, instead.

Our results to date raise more questions than they answer, and we believe that further answers will require closer collaboration between the circuit and device community and

the systems community than may have been present to date. Historically the device community has focused on worst-case behavior, as is appropriate for e.g. memory buses. However, as systems designers we often are concerned with average-case behavior instead. We believe a deeper understanding on both sides, and focused experimentation, will help design higher-performance flash-based systems in the future.

5. REFERENCES

- [1] L. Bouganim, B. JÅšnsson, and P. Bonnet. uFLIP: understanding flash IO patterns. In *Int’l Conf. on Innovative Data Systems Research (CIDR)*, Asilomar, California, 2009.
- [2] P. Huang, Y. Chang, T. Kuo, J. Hsieh, and M. Lin. The Behavior Analysis of Flash-Memory Storage Systems. In *IEEE Symposium on Object Oriented Real-Time Distributed Computing*, pages 529–534. IEEE Computer Society, 2008.
- [3] Hynix Semiconductor, Intel Corporation, Micron Technology Inc., Numonyx, Phison Electronics Corp., Sony Corp., and Spansion. Open NAND Flash Interface Specification, rev. 2.1. Available from www.onfi.org/specifications, Jan. 2009.
- [4] K. Kimura and T. Kobayashi. Trends in high-density flash memory technologies. In *IEEE Conference on Electron Devices and Solid-State Circuits*, pages 45–50, 2003.
- [5] J. Lee, J. Choi, D. Park, and K. Kim. Data retention characteristics of sub-100 nm NAND flash memory cells. *IEEE Electron Device Letters*, 24(12):748–750, 2003.
- [6] J. Lee, J. Choi, D. Park, and K. Kim. Degradation of tunnel oxide by FN current stress and its effects on data retention characteristics of 90 nm NAND flash memory cells. In *IEEE Int’l Reliability Physics Symposium*, pages 497–501, 2003.
- [7] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. In *IPSN/SPOTS*, April 2006.
- [8] K. OÅšBrien, D. C. Salyers, A. D. Striegel, and C. Poellabauer. Power and performance characteristics of USB flash drives. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–4, 2008.
- [9] M. Park, E. Ahn, E. Cho, K. Kim, and W. Lee. The effect of negative V_{TH} of NAND flash memory cells on data retention characteristics. *IEEE Electron Device Letters*, 30(2):155–157, 2009.
- [10] M. Sanvido, F. Chu, A. Kulkarni, and R. Selinger. NAND flash memory and its role in storage architectures. *Proceedings of the IEEE*, 96(11):1864–1874, 2008.
- [11] R. Saripalli. Maximizing endurance of MSC1210 flash memory. Technical Report Application Report SBAA091, Texas Instruments, 2003.
- [12] N. Shibata, H. Maejima, K. Isobe, K. Iwasa, et al. A 70 nm 16 gb 16-Level-Cell NAND flash memory. *IEEE Journal of Solid-State Circuits*, 43(4):929–937, 2008.
- [13] H. Yang, H. Kim, S. Park, J. Kim, et al. Reliability issues and models of sub-90nm NAND flash memory cells. In *Solid-State and Integrated Circuit Technology (ICSICT)*, pages 760–762, 2006.