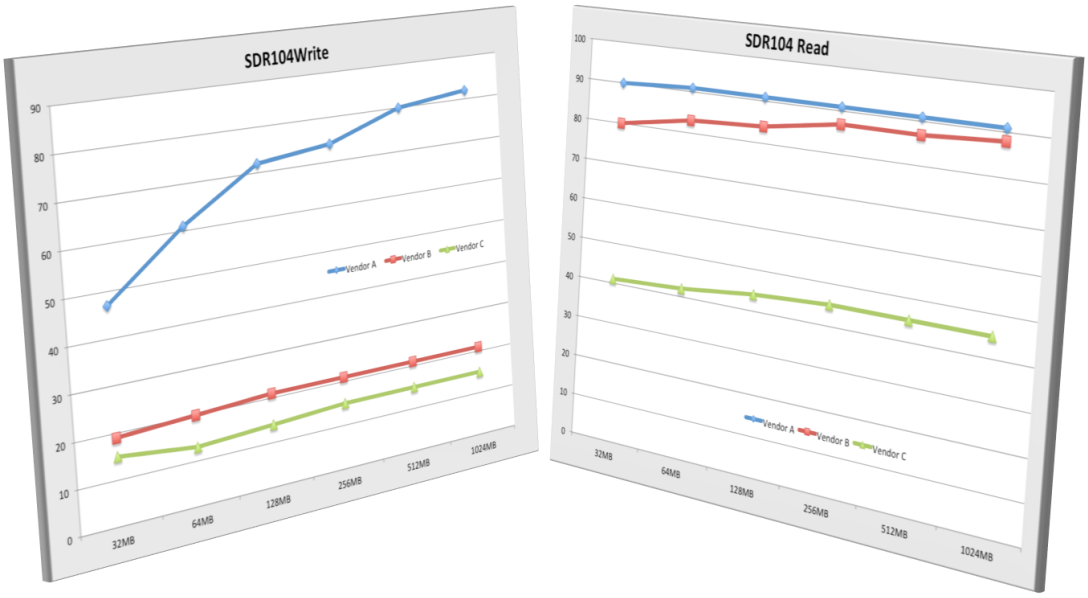
White Paper

Benchmarking Mobile Storage
with the Arasan High Performance HVP

Executive Summary

NAND Flash is the nonvolatile memory used in virtually all mobile devices. (smartphones, tablets, cameras, game controllers). High performance products (Tablets and smartphones) place increasing demands on NAND Flash device capacity, cost and bandwidth. To meet these demands, component and application processor designers must utilize a complex combination of electronic hardware and software. As a result benchmarking NAND Flash at the component and system level is a key element in successful product design.

The Arasan Hardware Validation Platform (HVP) provides a flexible system for IP design and the associated software development and hardware validation and compliance. Arasan HVP are available in multiple configurations for the leading mobile interface standards such as SD/SDIO/eMMC, UFS and MIPI.



Sam W. Beal

Arasan Chip Systems Inc.

Introduction

NAND Flash is the nonvolatile memory behind SSD, mobile computing devices and mobile appliances (such as phones and cameras). Flash technology is the mainstream storage solution for mobile – and mobile is growing dramatically. Some analysts predict tablet shipments will eclipse PC shipments by 2016.

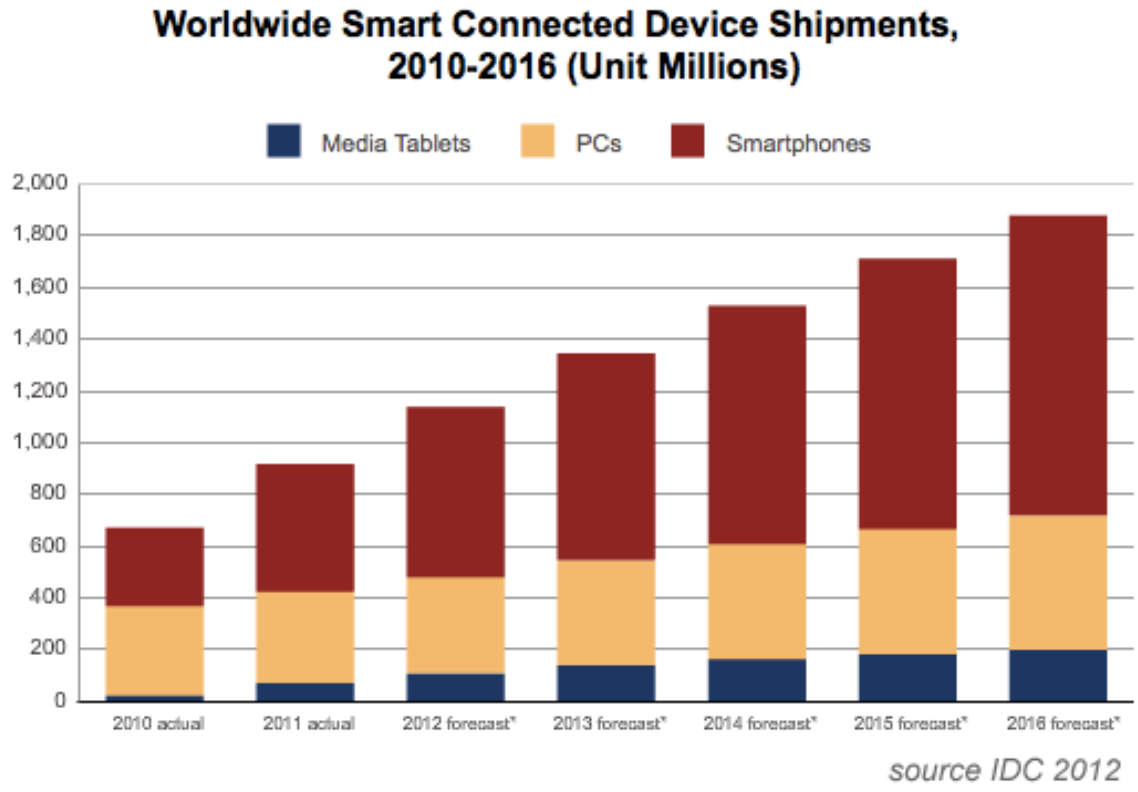


Figure 1. WW Smart Connected Device Shipments

Why Benchmarking matters

Tablet computers and smartphones place increasing demands on NAND Flash device features – capacity, cost and bandwidth. But providing this requires a complex combination of electronic and software development. As a result benchmarking NAND Flash at the component and system level is critical to successful product design.

Benchmark testing spans a wide range of factors such as the device controller implementation, the host interface standard (SD, eMMC, UFS), type of NAND technology (SLC, MLC, TLC), device drivers and file systems and the application requirements, e.g. sequential, random, read-write, etc.

NAND Flash Technology is Tricky

Amplification

NAND Flash cells are read at the page level and erased at the block level. Cells must be erased before programmed. For example, a 16Gb NAND Flash from Micron [1], uses a 4KB page and a 512KB block. Every time data is written to a block the entire block is erased. Internally valid is copied to available blocks. In effect an external write will produce multiple internal writes.

Block Management

Block management accomplishes several tasks related to translation of logical addresses to physical addresses including bad block tagging, dynamic buffering for performance, and garbage collection. This function may be implemented in a Flash Translation Layer (FTL) in firmware or a Flash File System (FLS) running on the host.

Wear Leveling

Wear leveling manages the local to physical mapping so that erase/write cycles are distributed evenly over the entire range of the memory array to optimize the life of the device. Typically this is implemented in controller FTL where the P/E cycle count is known. New approaches based on BER optimization are emerging. [2]

Error Recovery

Some error recovery can be implemented in the controller command logic (with extra P/E cycles) for read / write disturb errors when adjacent (unintended) cells voltage levels shift from I/O activity. Other errors such as voltage shift from alpha particles injection rely on ECC to resolve.

ECC

Error correction code (ECC) solutions are based on Hamming, Reed-Solomon, BCH [3] and LDPC. Hardware solutions scale in complexity with the NAND block size and the error correction requirements. SInce soft errors scale with shrinking NAND process geometry, the hardware costs of ECC are increasing. Low Parity Density Codes, use statistical algorithms implemented in firmware to reduce block-level BER with less integrated circuit area. [4]

Signal Processing Optimization

Beyond merely correcting errors, signal processing recovers correct bits by analyzing multiple read voltages. Integrating signal processing, ECC and FTL into the device controller, e.g. “managed NAND” by Anobit [5]. NAND device suppliers, e.g. SK Hynix and Micron, are integrating control functionality like ECC and signal processing onto the NAND die or dedicated controller die in the NAND package. [6,21]

Implications

Controller design is intimately tied to the NAND device. The degree is only increasing as NAND technology shrinks and NAND devices grow in complexity and density. Mobile Storage devices may exhibit dramatically different results depending on price and application requirements.

Benchmarking Perspective

There have been extensive benchmarks on SSDs, from both Consumer and Enterprise perspective. To a first approximation, SSDs are drop-in replacements for HDDs and thus the benchmarks use HDD file systems (FAT32, NFS, etc.) and test suites tuned to PC/consumer or enterprise applications, e.g. DiskSim, IOZone, uFlip. IOMeter, PCMark2000 [7,8,9]

The same benchmark environment has been extended to SD cards (Crystal Diskmark, IOTrace, h2benchr/w) [9].

Recently H.Kim [11] published the first detailed usage benchmarks of SD storage in a mobile (Android) environment. Mr. Kim developed a "WebBench" that allowed storage measurement during a range of typical smartphone tasks such as web browsing, app install and launch using several popular apps. The IO characteristics are shown below.

Activity	Write (MB)		Read (MB)	
	Sq	Rn	Sq	Rn
WebBench	41.3	32.2	6.8	0.5
AppInstall	123.1	5.6	0.7	0.1
Email	1.0	2.2	1.1	0.1
Maps	0.2	0.3	0	0
Facebook	2.0	3.1	0	0
RLBench	25.6	16.8	0	0
Pulse	2.6	1.0	0	0

Figure 2. I/O summary developed by H. Kim

He published runtimes for these benchmarks using a range of off-the-shelf SD cards. The results vary significantly across the nine vendor-components selected. Furthermore there was significant application dependence in the vendor results. The methodology also provides insight into app performance independent of the storage solution.

Smartphone and Tablets are aggressively increasing the performance of the processor and functionality of apps. These devices will support

multitasking, video recording and playback at 1080p resolution, wireless streaming to TVs in the home, support for high bandwidth WiFi (600Mbps 802.11n 2-channel and 1Gbps 802.11ac). The memory subsystem (NAND with DRAM cache) will be increasingly tasked to support the MIPS provided by the processor subsystem. Controllers are tightly coupled to a NAND technology in both hardware (logic) and firmware. Optimizing a device for a specific price-performance-application requires optimization of the device controller.

Kim defines four categories for optimization:

1. Better storage media for mobile devices (faster, larger)
2. Firmware and device drivers
3. mobile OS (Android issue since iOS is closed)
4. Application-level changes

Benchmarking Flash File Systems

Embedded Flash File Systems (FFS) can implement many of the FTL functions in software running on the host. Early Apple iOS 4 is one example [9]. The performance of today's mobile solutions preclude software-only solutions. Most FFS implement data compression, bad block management, wear-leveling and garbage collection. Journaling is a useful system feature in newer Linux and Apple OS.

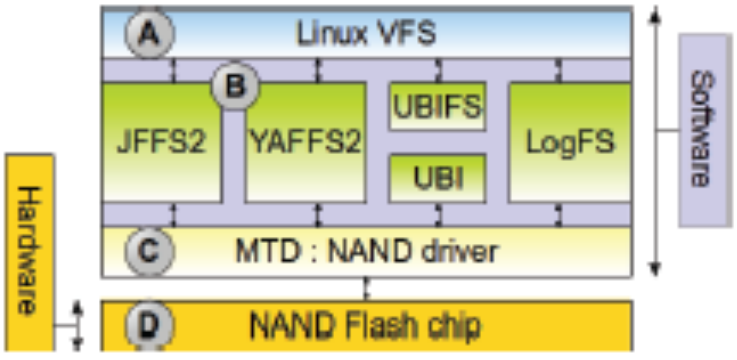


Figure 3. FFS Benchmark Organization[13]

Emulation and Simulation

Some design teams develop hardware emulators [14] and Test beds [15] built with programmable logic FPGAs to allow benchmarking as well as validation functions. On the system side, a software developer can utilize a hardware validation platform for device or block driver development, software utility or application design. SoC designers can use hardware validation platforms to provide a known good device for validating the host interface to the controller. Hardware emulation good but important to not constrain the system with artificial limits (FPGA type, bus interface, etc.)

OEM integrators can use benchmarking to qualify components. Datasheets are rarely adequate because of the complexity of the interaction described above.

Simulation is better suited for architecture exploration and application tuning. FlashSim [16] is a C++ based tool that incorporates DiskSim. It was developed for SSD benchmarking. It provides control over the FTL (block management, garbage collection and wear leveling) under different application traces. The authors look at throughput and energy consumption in the model. Another SSD based simulator [17] extends analysis with a cycle-accurate model. The hardware model can vary clock frequency, bus interface speed, page size, buffer size, number of channels, data buses and DMA controllers. The model is able to profile CPU operations including execution times and overhead. The model could be applied to portable systems employing SD and embedded Flash.

SD/SDIO/eMMC Host Controller

The **Arasan SD3.0 / SDIO3.0 / eMMC4.5 Host Controller** has an AHB/AXI/OCP processor interface and conforms to SD Host Controller Standard Specification Version 3.00. The controller is provided as licensable IP delivered as RTL with documentation.

The SD3.0/SDIO3.0/eMMC4.5 Host Controller handles SDIO/SD Protocol at transmission level, packing data, adding CRC, Start/End bit, and checking for transaction format correctness. This Host Controller provides Programmed IO method and DMA data transfer method. In programmed IO method, the Host processor transfers data using the Buffer Data Port Register.

The SD3.0/SDIO3.0/eMMC4.5 Host Controller support for DMA can be determined by checking the DMA support in the Capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU. This Host Controller's Host Controller system address register points to the first data address, and data is then accessed sequentially from that address. It supports connection to a single slot and performs multi-block writes and erases the lower access.

The SD3.0/SDIO3.0/eMMC4.5 Host Controller support for DMA can be determined by checking the DMA support in the Capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU. This Host Controller's Host Controller system address register points to the first data address, and data is then accessed sequentially from that address. It supports connection to a single slot and performs multi-block writes and erases the lower access.

Advanced DMA (ADMA2)

High throughput is achieved from advanced DMA based on a scatter-gather strategy. [18]

Arasan Hardware Validation Platform

The **Arasan HVP** is a complete system containing the key features:

- SD 3.0 / SDIO 3.0 / eMMC 4.51 Host Controller in FPGA
- SD/eMMC slot
- Resident stack & drivers & Linux OS
- Benchmark GUI

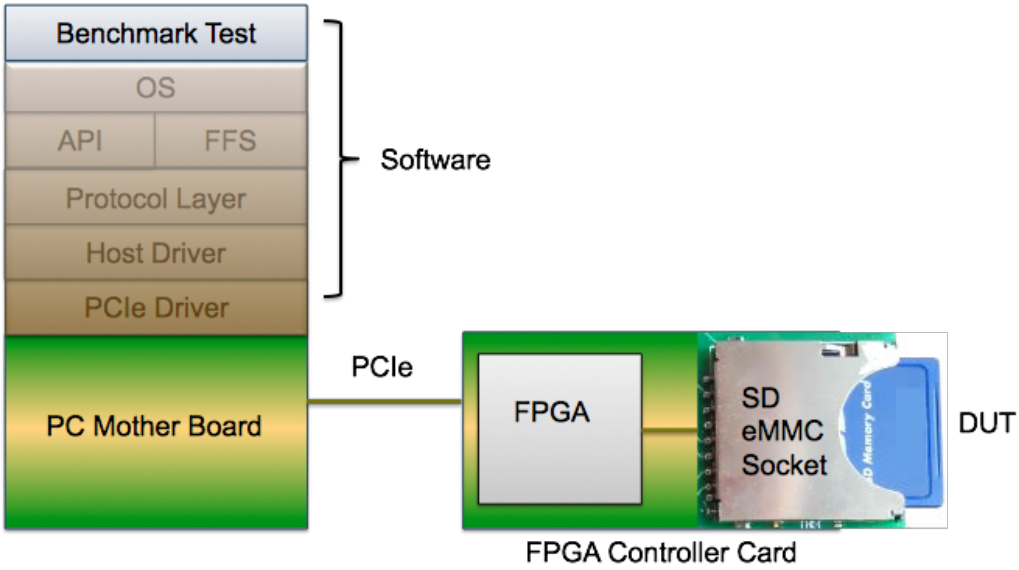


Figure 4. Arasan HVP-SD/eMMC Block Diagram

The system provides a range of benefits to component designers, firmware and file system developers and system architects.

- A platform for evaluating devices & firmware.
- A platform for validating host controller implementation and interface to system bus.
- A platform for application software debug & analysis
- A platform for comparing SD/eMMC devices

InterOp

Arasan participates in “interops” with the HVP such as SDA Interoperability Sessions [18] where SD card suppliers meet to test compatibility and performance. The sessions are open to SDA members. This provides feedback on compliance, compatibility and competitive performance to the attendant device manufacturers.

Max. MB/sec		Company 1	Company 2	Company 3	Company 4	Company 5	Company 6	Company 7	Company 8
SDR104 RD	64GB	47.8	64.4		47.0	37.4		62.3	35.9
	32GB	48.7	49.6		46.8			50.0	
	8GB						FAILE		
SDR104 WR	64GB		11.1		14.5	10.1		16.2	8.8
	32GB	14.4	13.8		13.5			15.0	
	8GB			FAILED			FAILE		
DDR50 RD	8GB			30.1			29.3		36.1
DDR50 WR	8GB			8.9			10.2		5.6

Figure 5. HVP-SD results from past InterOP

Benchmark Methodology

System is initialized with low driver (PCIe) and block driver. A resident set of benchmark apps is provide to write or read data in block sizes from 4KB for random access to 64KB for sequential access tests.

Figure 6. FPGA Device board with SD slot visible

Results

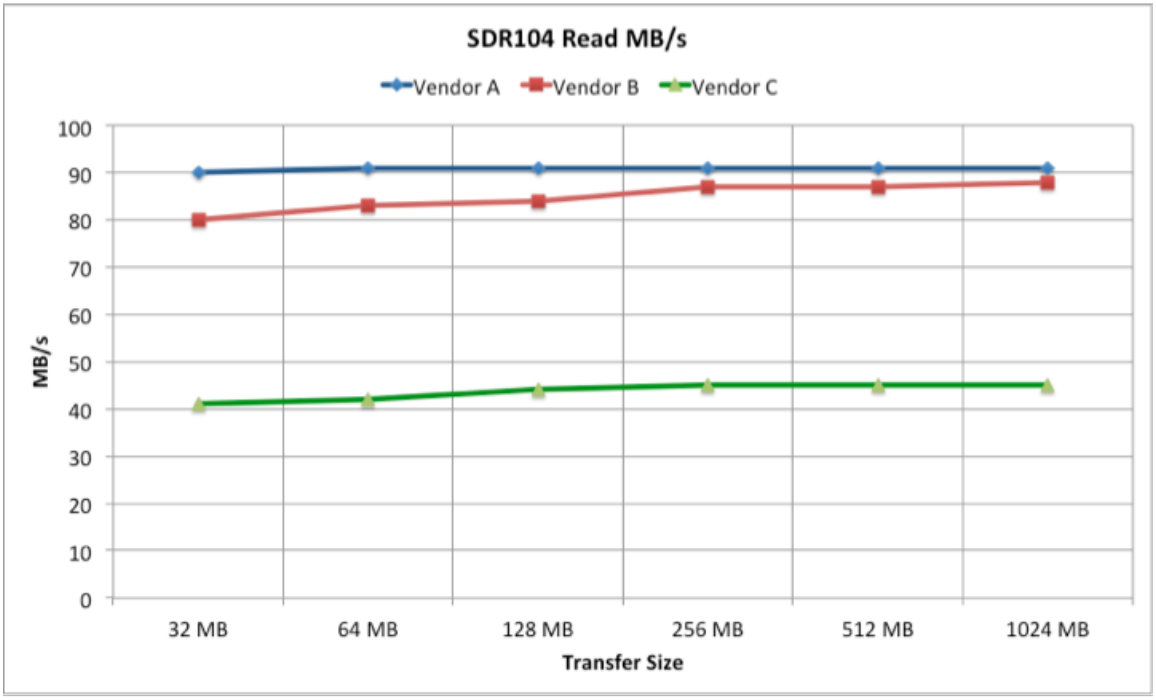


Figure 7. Benchmark Results for SDR104 Read

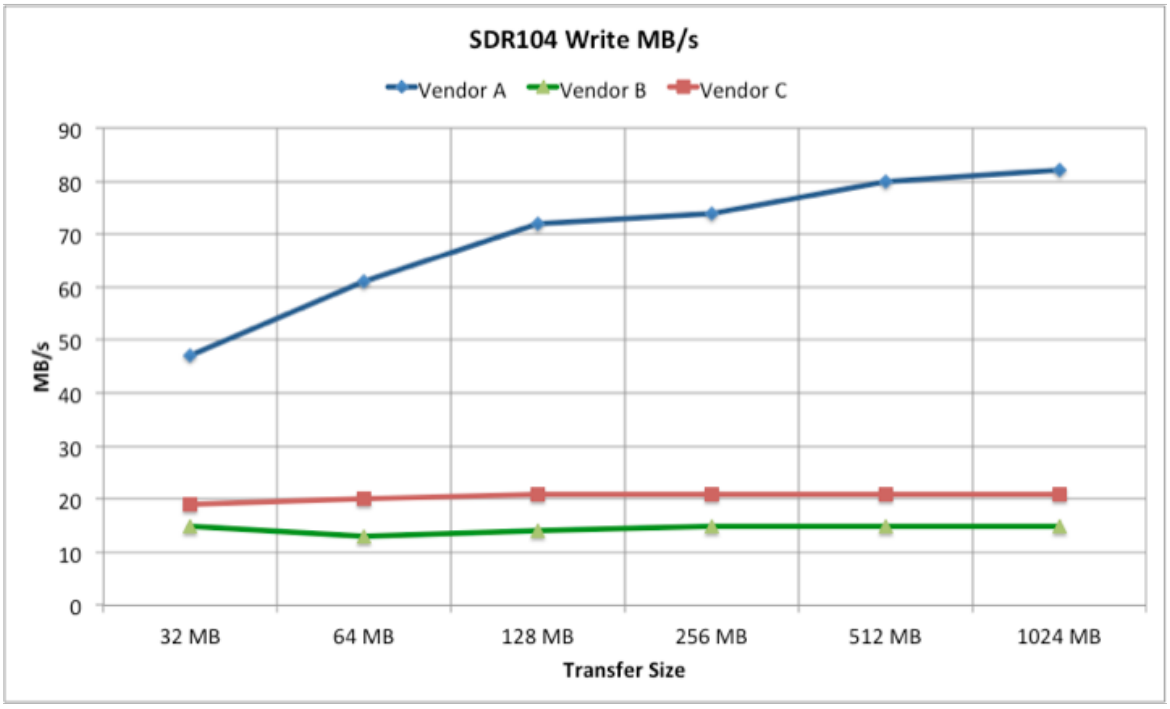


Figure 8. Benchmark Results for SDR104 Write

Results

Results for three different SD cards are shown in Figures 8 &9. A maximum read bandwidth of 90MB/s (90% of the theoretical maximum) was achieved by one vendor. A second vendor came in at ~80%. The third device was substantially lower. The vendor with the highest read bandwidth also achieved the highest write bandwidth, but the results were proportional to the transfer size.

Implications for Controller Design

Providing high performance non-volatile storage at low cost/bit comes at a price – in complexity of the device controller. When things become more complex, the interaction between the controller developer and the NAND device designer must increase. Some devices (e.g. Samsung 64Gb MLC) provide additional features such as soft data decision read mode [20] to support sophisticated ECC. There is also a trend to embedding the most complex aspects of the controller directly on the NAND device die. [6].

Micron's EZ-NAND or ClearNAND package embeds a controller IC in a NAND component package. The controller provides ECC and command queue features that improve multi-tasking by optimizing the logic to physical address. [21]

This interdependence percolates up through the driver, file system, the host interface and the final application layer. Everything in a storage system is a candidate for optimization, but raw performance at the bottom (physical device layer) establishes the floor.

Benchmarking can provide insight, feedback and validation. A benchmark system based on Arasan's high performance HVP allows a unrestricted analysis.

Looking Forward: Mobile Storage Roadmap

NAND Flash is following an aggressive technology scaling roadmap. Similarly the interface standards from the SD Association, JEDEC and UFSA are evolving to provide higher performance.

SD 2.0	SD 3.0	SD 4.0	removeable
25MB/s	104MB/s	1.56 / 3.12 Gbps	
2007	2011	2012	
eMMC 4.4	**eMMC 4.5**	**UFS 1.1**	**embedded**
50MB/s	100MB/s	300MB/s	
2009	2011	2013	

Table 1. Performance Roadmap and Year of First Production

Summary

NAND Flash Memory usage is growing rapidly in two major markets: mobile consumer and SSD. SSD further segments into Enterprise class and consumer peripheral. Each of these segments has special requirements that in turn affect the type of NAND Flash device technology, the device controller functionality and the software interface to the application layer

Designing mobile storage solutions with NAND Flash memory requires subtle software and complex electronic design. Benchmarking is an effective and necessary piece of a development solution - from device level to system level. You don't want to build your own evaluation environment unless you have to.

Arasan has demonstrated an SD 3.0 Hardware Validation Platform (HVP) running at more than 90MB/s sustained read and more than 58MB/s sustained write using a Linux storage device benchmark. This represents ~90% of the theoretical maximum performance of 104MB/s defined in the SD 3.0 standard.

References

1. Micron MT29Fxx NAND Family Datasheet
2. Pan, Dong, Zhang, “Error Rate-Based Wear-Leveling for NAND Flash Memory”, 2012 (to be published in IEEE Transactions on VLSI Systems).
3. Subbiah, Viswanath, “Evolving throughput driven architecture for error correction in NAND Memory”, Arasan Chip Systems White Paper 2010
4. Wang, Courtade, Shankar, Wesel, “Soft Information for LDPC Decoding in Flash”. Global Telecommunications Conference 2011
5. Flash Memory Summit, Santa Clara, CA. 2011
6. Shim, Lee, Kim et. al., “Highly Reliable 26nm 64Gb MLC E2NAND”, 2011 Symposium on VLSI Technology
7. Polte, Simsa, Gibson, “Comparing Performance of Solid State Devices and Mechanical Disks”, 2008
8. Jung, Wilson, Donofrio, et. al., “NANDFlashSim Intrinsic Latency Variation Aware NAND Flash Memory System Modeling”, 2012
9. Bouganim, Jonsson, Bonnet, “uFLIP: Understanding Flash IO Patterns”, 4th Biennial CIDR, Jan 2009
10. See http://tomshardware.com
11. Revisiting Storage for Smartphones – H. Kim, 2012
12. See http://esec-lab.sogeti.com/post/Low-level-iOS-forensics for example
13. Olivier, Boukhobza, Senn, “On Benchmarking Embedded Linux Flash File Systems”, IEEE 2012
14. , Park, Sho, Lee, Park. (Samsung), “Co-Validation Environment for Memory Card Compatibility”, IEEE 2004
15. Grupp, Caulfield, Coburn “Beyond the datasheet: Using Testbeds”, IEEE 2010
16. Kim, Tauras, Gupta, Urgaonkar , “Flashsim: A simulator for Flash-based SSD”, IEEE 2009
17. Jung, Jung, Song, “Architectural Exploration of Flash Memory Storage Controller”, IEEE 2011
18. Simplified Host Controller Spec – SD Organization, SDCard.org
19. http://SDcard.org - Fall Interoperability Session, Oct 30, Tokyo, Japan
20. Kim, Ryu, Lee, et. al. “A 21nm High Performance 64Gb MLC NAND Flash Memory (Samsung)”, IEEE Journal of Solid State Devices, 2012
21. C. Lay, System Benefits of EZ-NAND / Clear-NAND FMS 2011