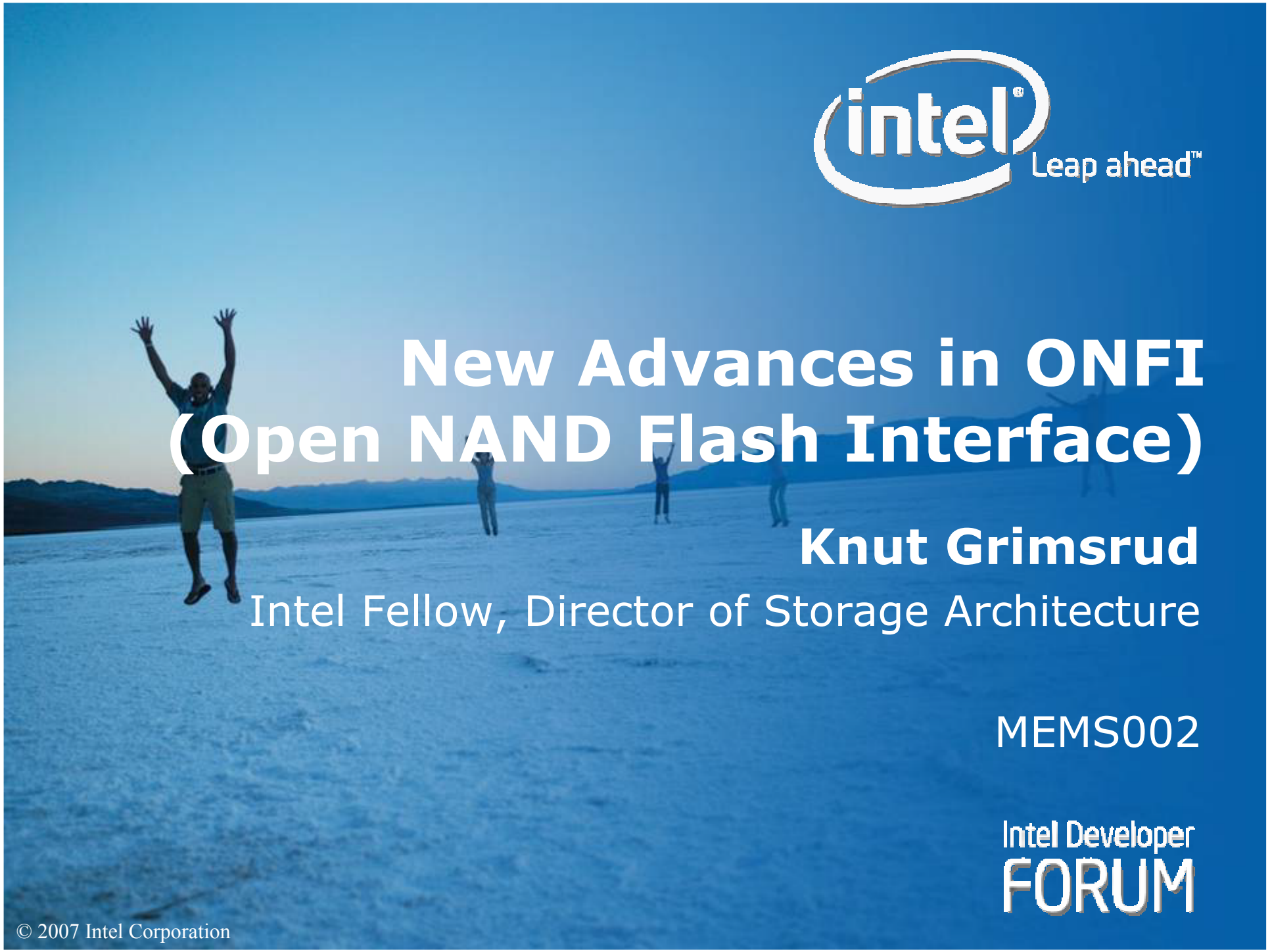

intel
Leap ahead™
Intel Developer
FORUM



New Advances in ONFI (Open NAND Flash Interface)

Knut Grimsrud

Intel Fellow, Director of Storage Architecture

MEMS002

Intel Developer FORUM

Agenda

PART 1

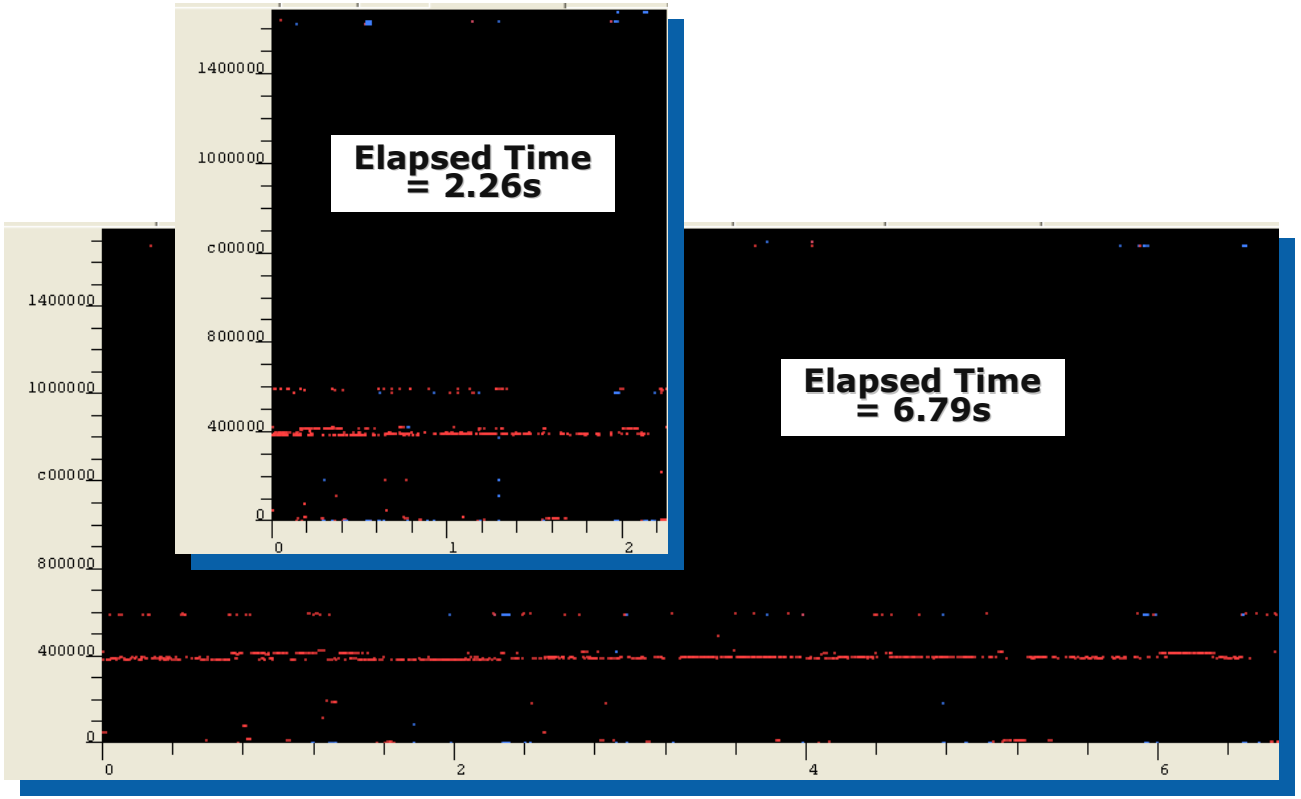
- Flash opportunities in IA
- Further improvement options
- NAND interface performance

PART 2

- ONFI 2.0 overview
- High Speed NAND
- Block Abstracted (BA) NAND
- Connector and module

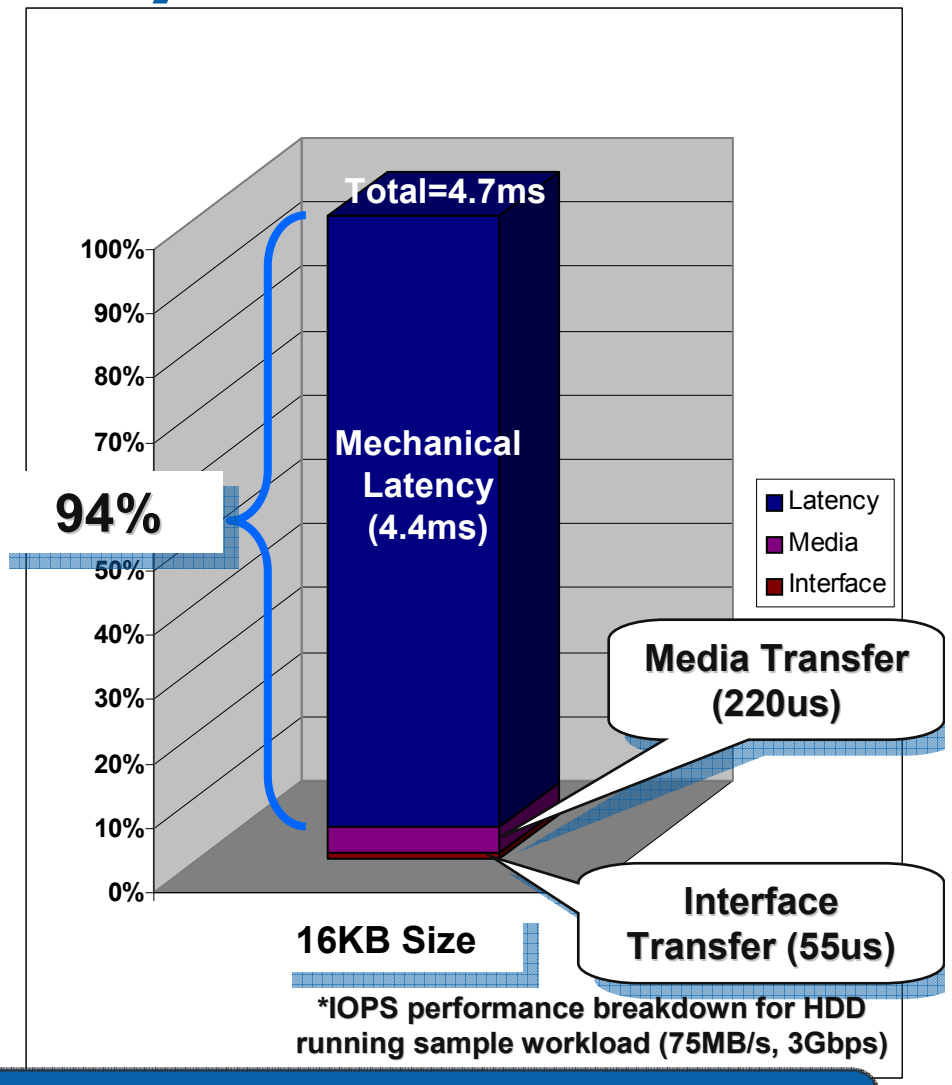
Flash Opportunities in IA

- Flash provides substantial performance, responsiveness, and power savings benefits



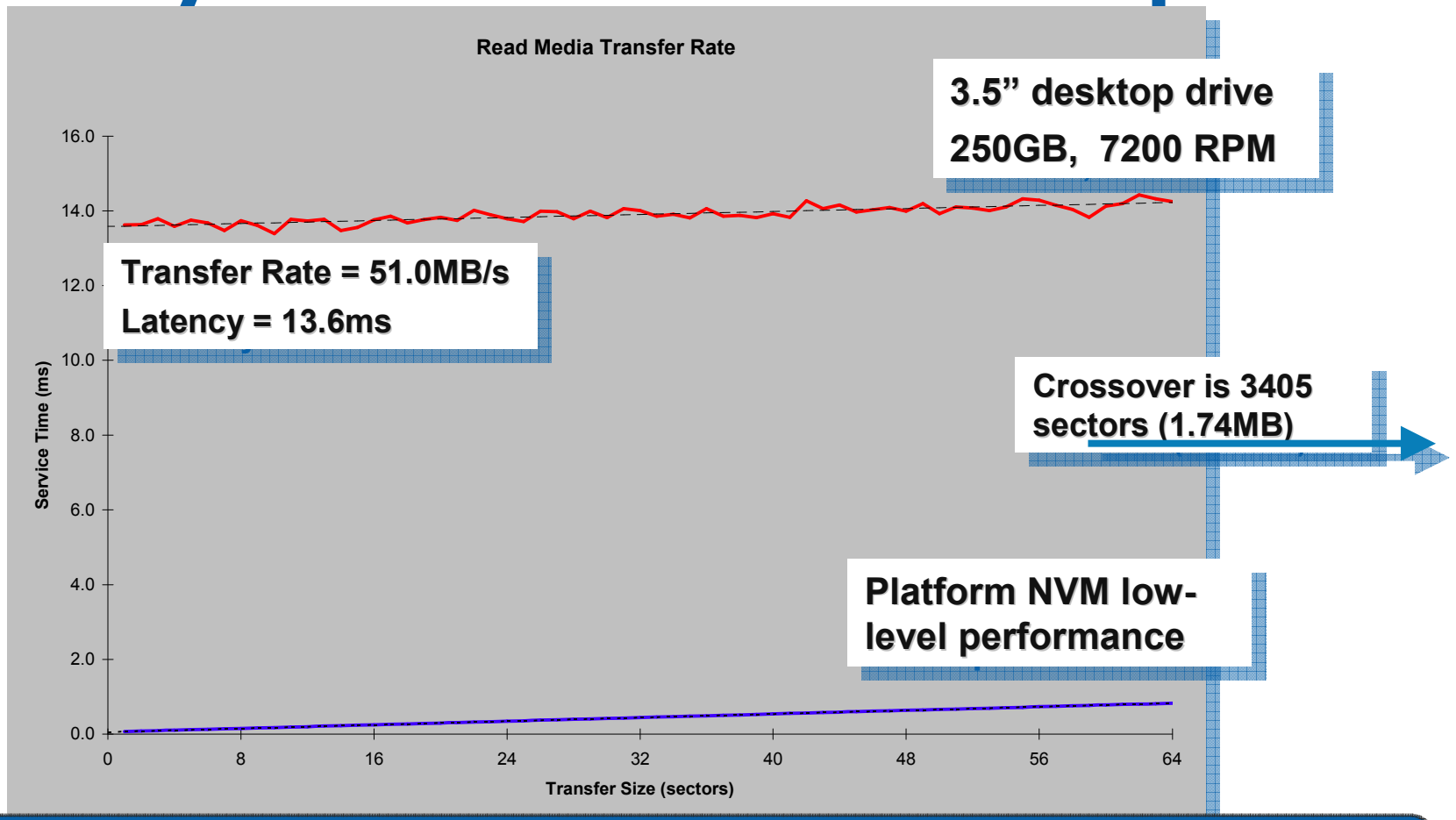
Dispelling Performance Myths

- Raw single-component Flash transfer rates not faster than HDD today
  - Mainstream flash is about 40MB/s reads
  - Mainstream HDD is about 75MB/s reads
- Raw flash access time much faster than HDD
  - Mainstream flash is <100us
  - Mainstream HDD is several milliseconds



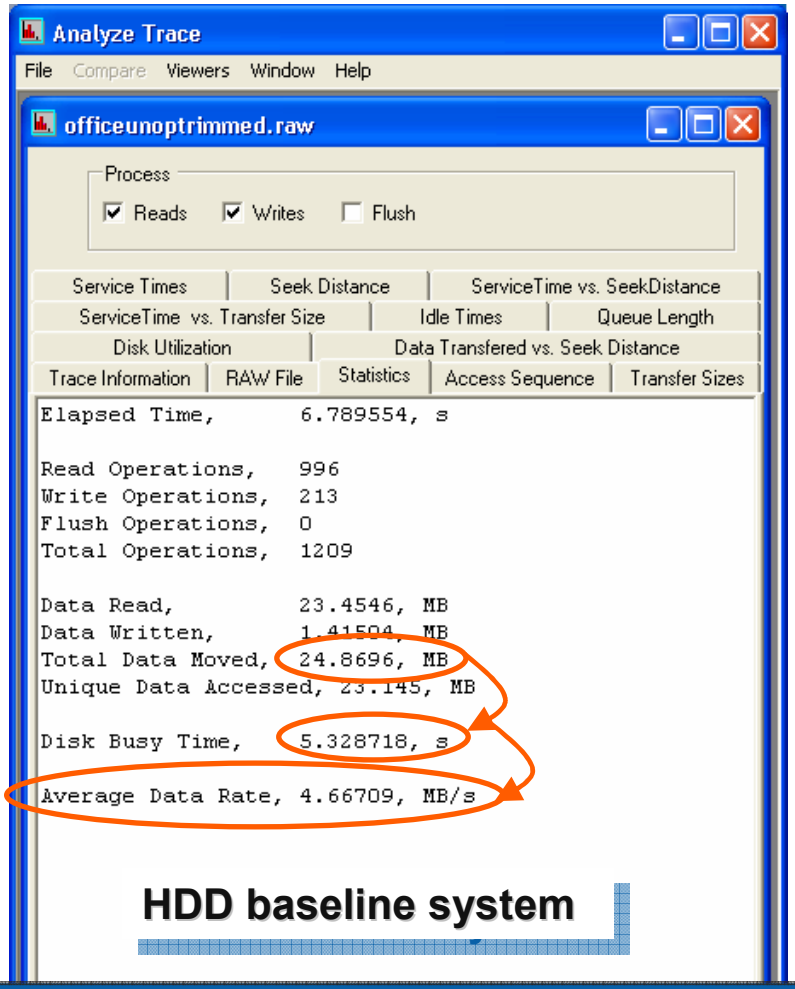
Transfer time accounts for insignificant fraction of actual disk service time. Latency is dominant factor.

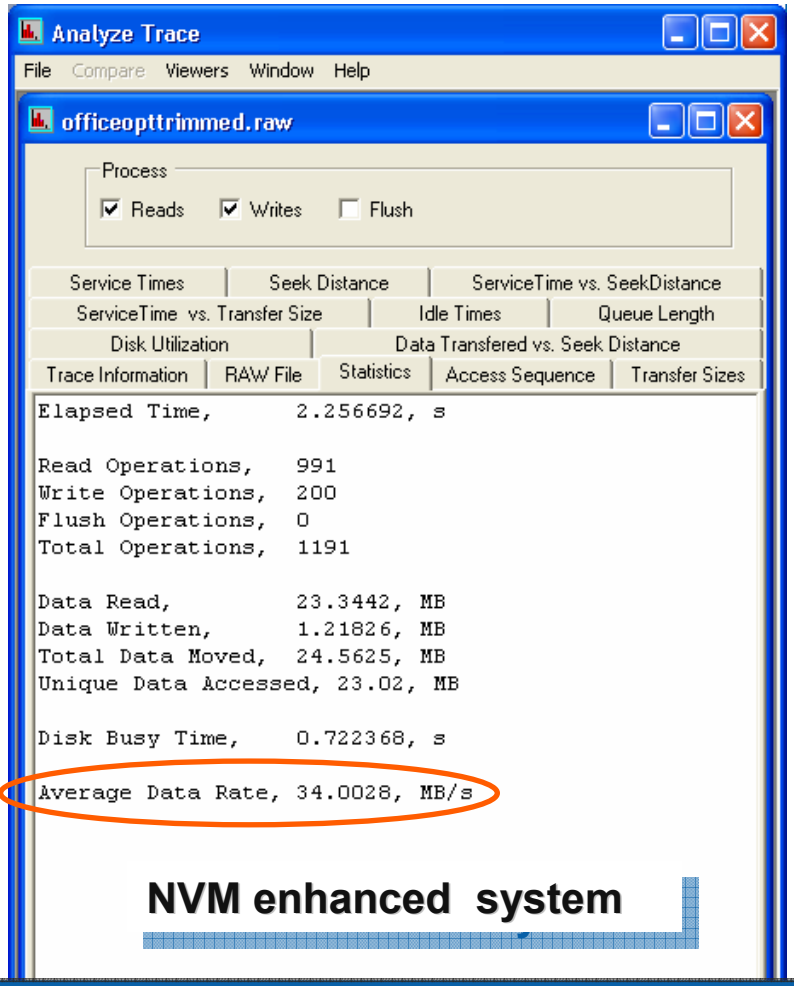
Latency vs Transfer Rate Comparison



Although streaming rate is slightly lower than HDD, realized performance is much better for modest sizes

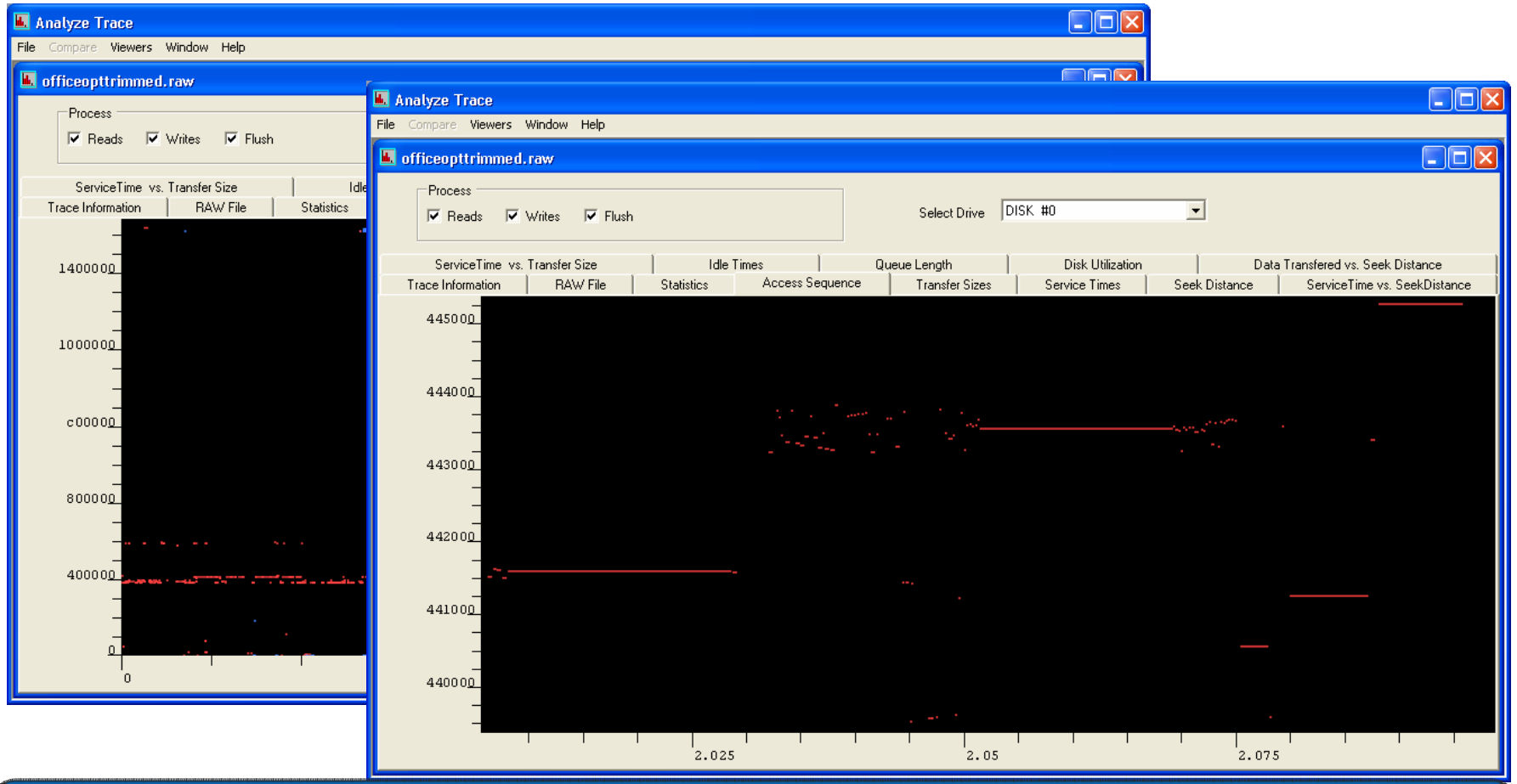
Resulting Flash Platform Impact





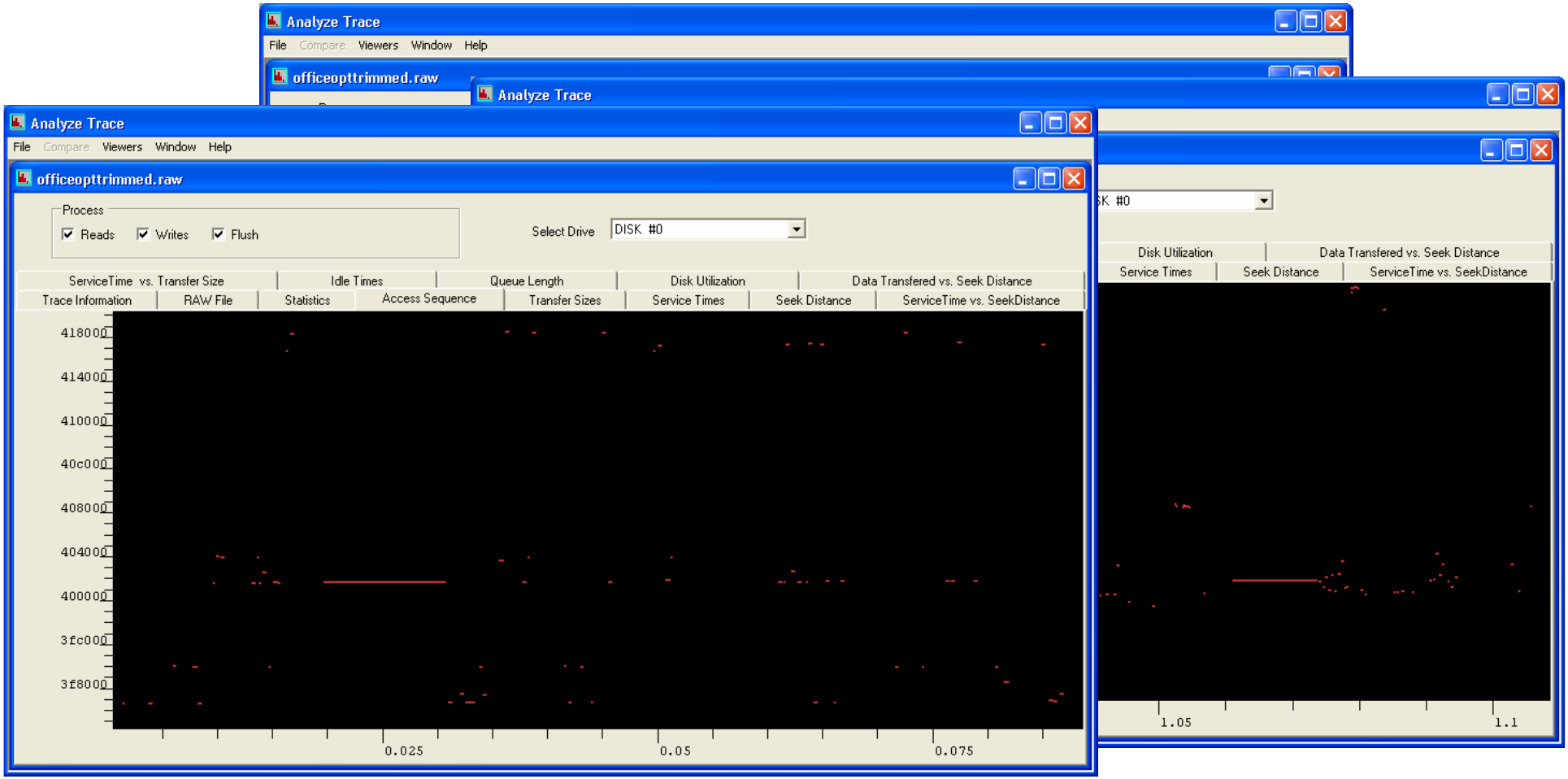
***Effective* data rate for flash solution 7X higher than HDD**

Further Improvement Option 1



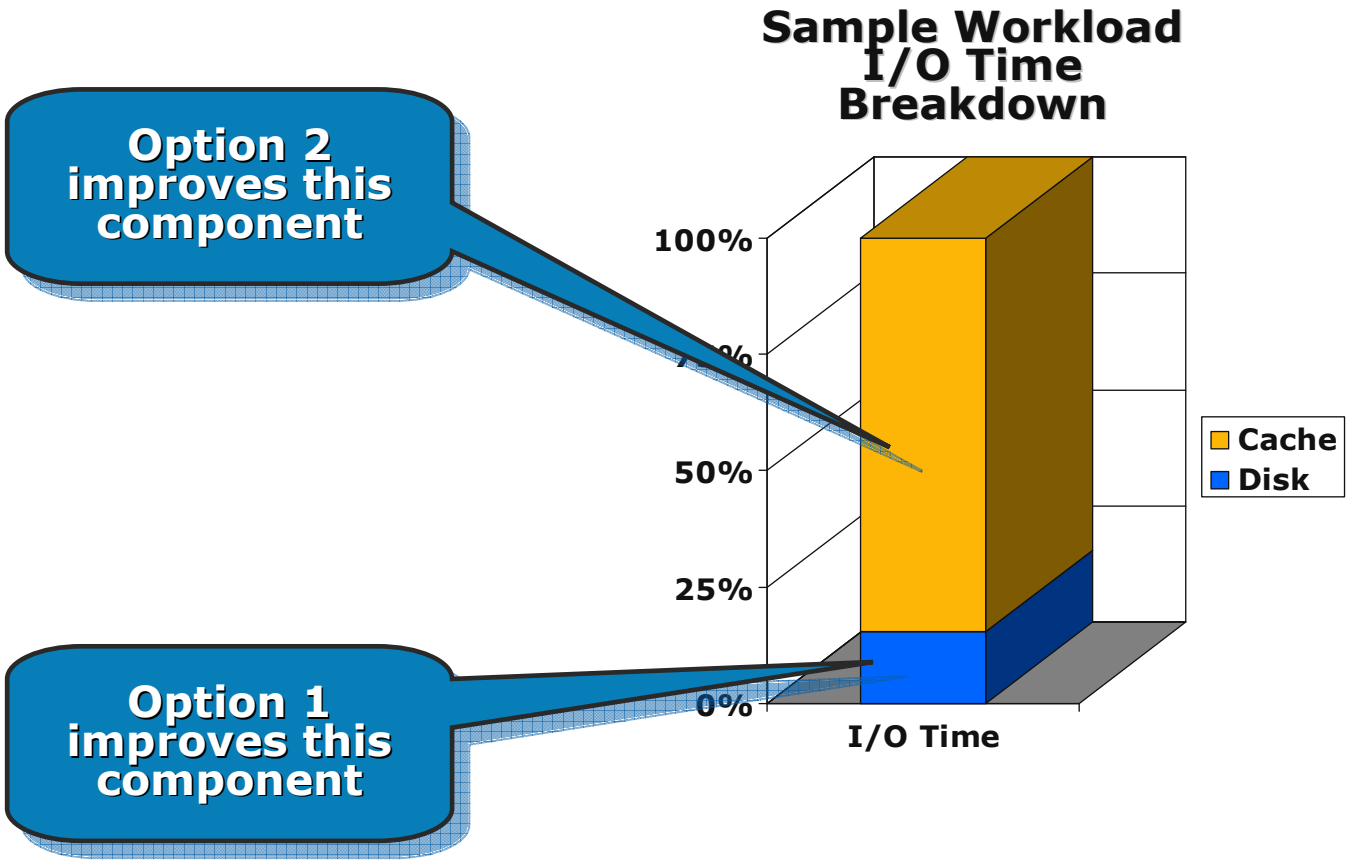
Algorithm improvements and cache size increases can improve performance by further reducing disk accesses

Further Improvement Option 2



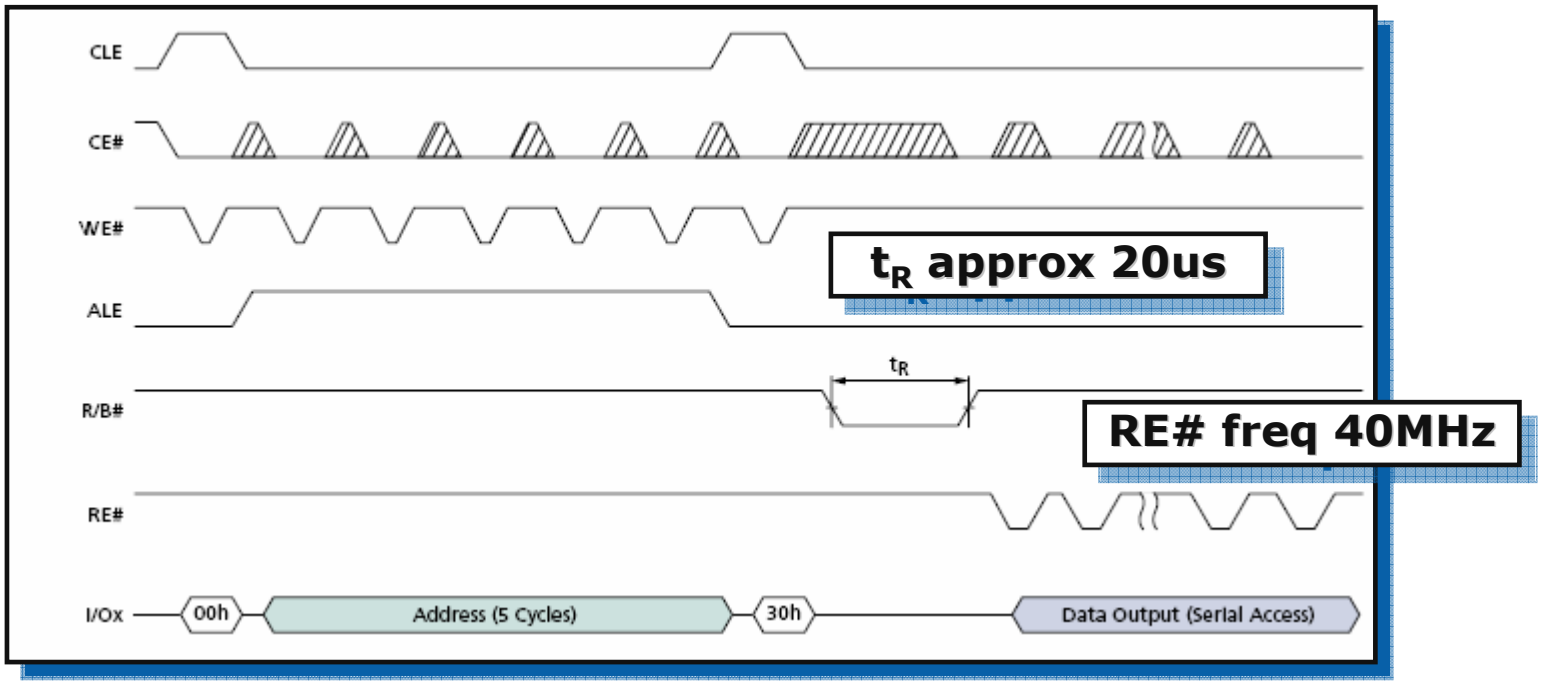
For high hit-rates, improve performance further by decreasing cache hit time

Sample Workload Breakdown



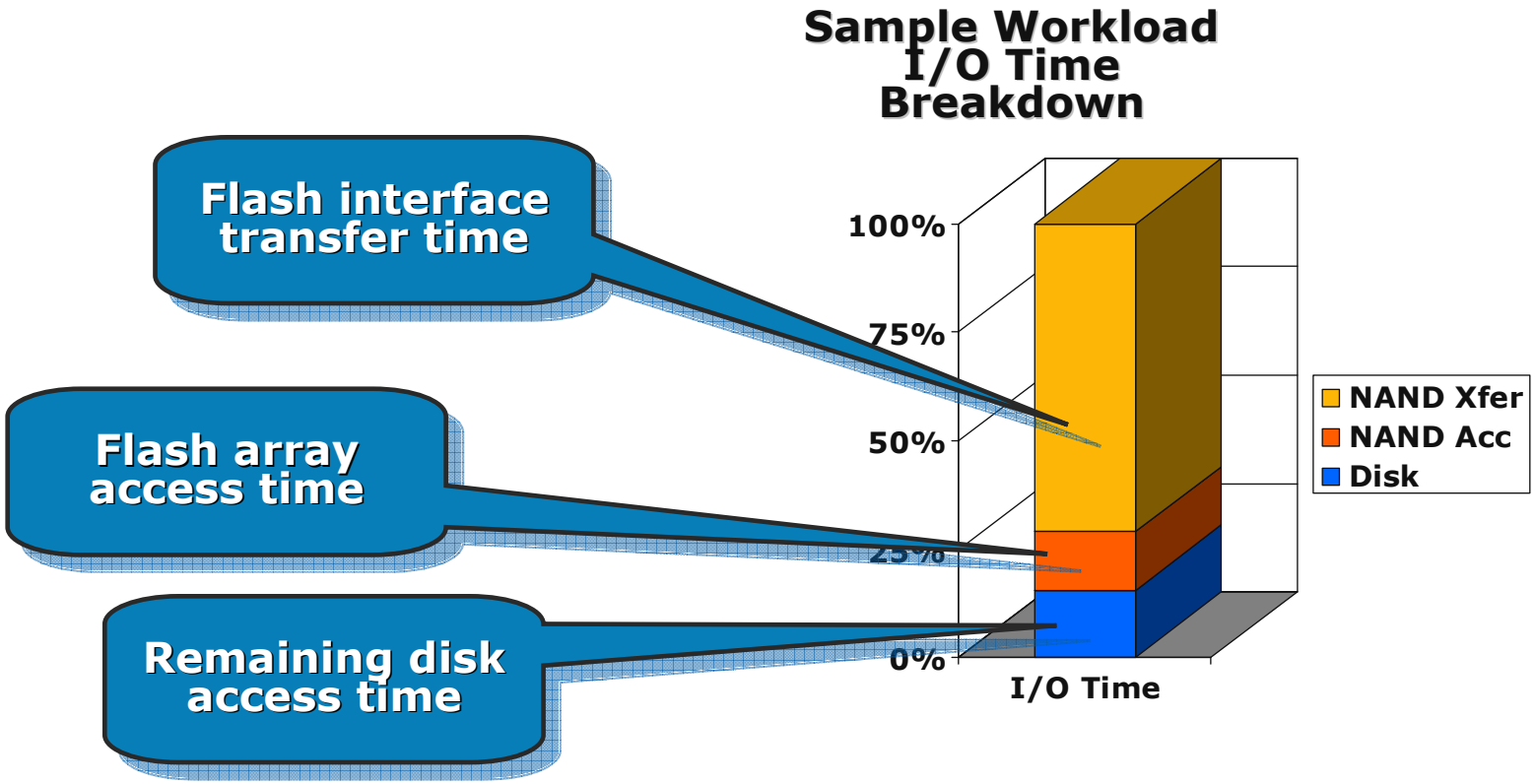
Best approach for further performance improvement is improving cache hit times

Current Flash Performance



- Flash performance consists of 2 primary elements
  - Time to transfer data between the array and the page register (t_R)
  - Time to transfer data between the page register and the host (RE# cycle time)

Sample Workload Additional Breakdown



Largest performance improvement potential from NAND interface improvements

Summary

- NAND Flash shows tremendous promise for accelerating compute applications
  - NAND access latency is strong suit
- Performance can be further improved with two approaches
  - Increase cache hit rate to further reduce disk accesses
  - Increase NAND performance to reduce cache hit times
- Largest improvement potential from increasing NAND Flash interface performance
  - Flash interface is the largest remaining component of the I/O time breakdown

New Advances in ONFI (Open NAND Flash Interface)

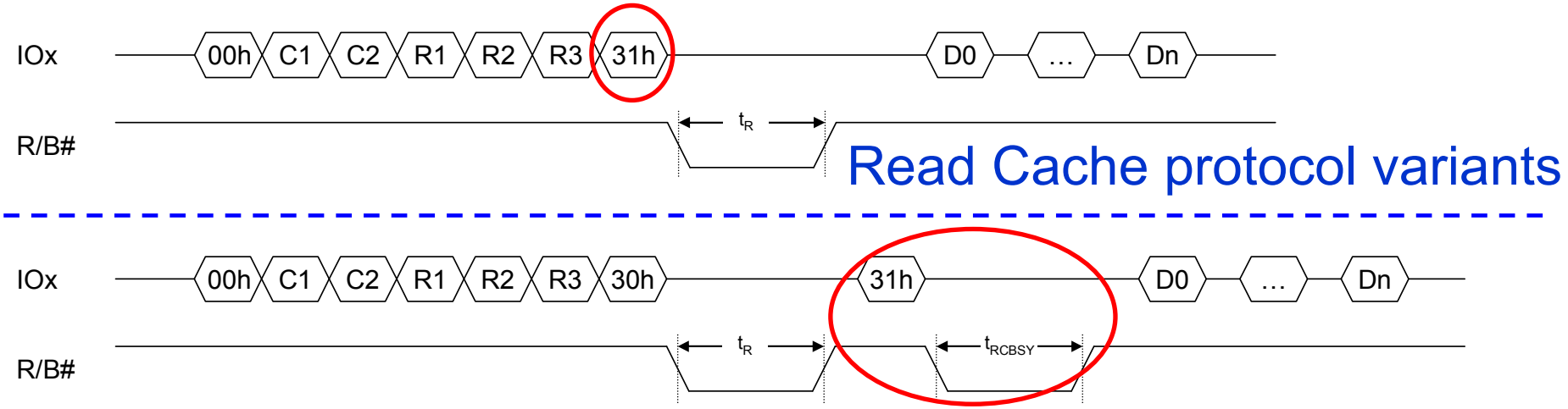
Amber Huffman
Principal Engineer

MEMS002

Intel Developer FORUM

NAND Interoperability Before ONFI

- Prior to ONFI formation in 2006, NAND has been the only commodity memory with no standard interface
- Basic NAND commands are ***similar*** amongst vendors
  - Read, Program, Erase, Reset, Read Status
- Timings vary from vendor to vendor
  - Often supporting a few vendors easy, all is more difficult
- Enhanced NAND commands vary widely



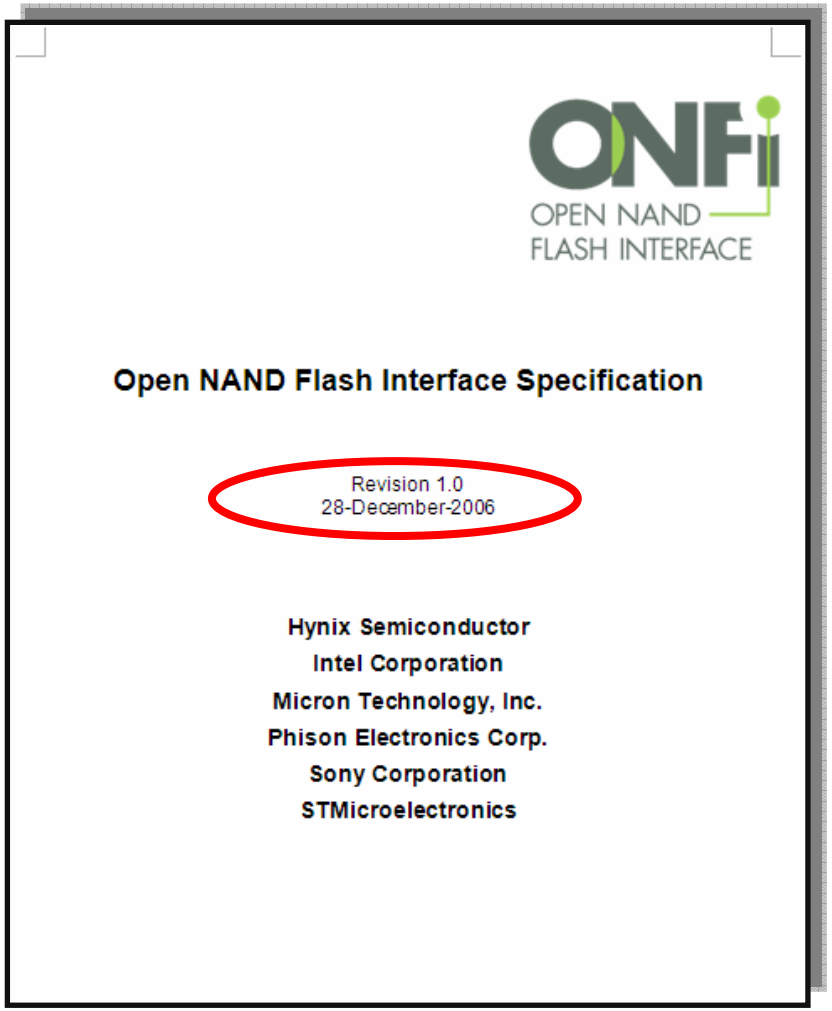
ONFI 1.0 Overview

ONFI 1.0 Defines

- Uniform NAND electrical and protocol interface
  - Raw NAND component interface for embedded use
  - Includes timings, electricals, protocol
  - Standardized base command set
- Uniform mechanism for device to report its capabilities to the host

ONFI 1.0 Status

- Ratified specification in Dec '06
  - Delivered spec in less than 8 months!



ONFI 1.0 establishes a standard interface for NAND

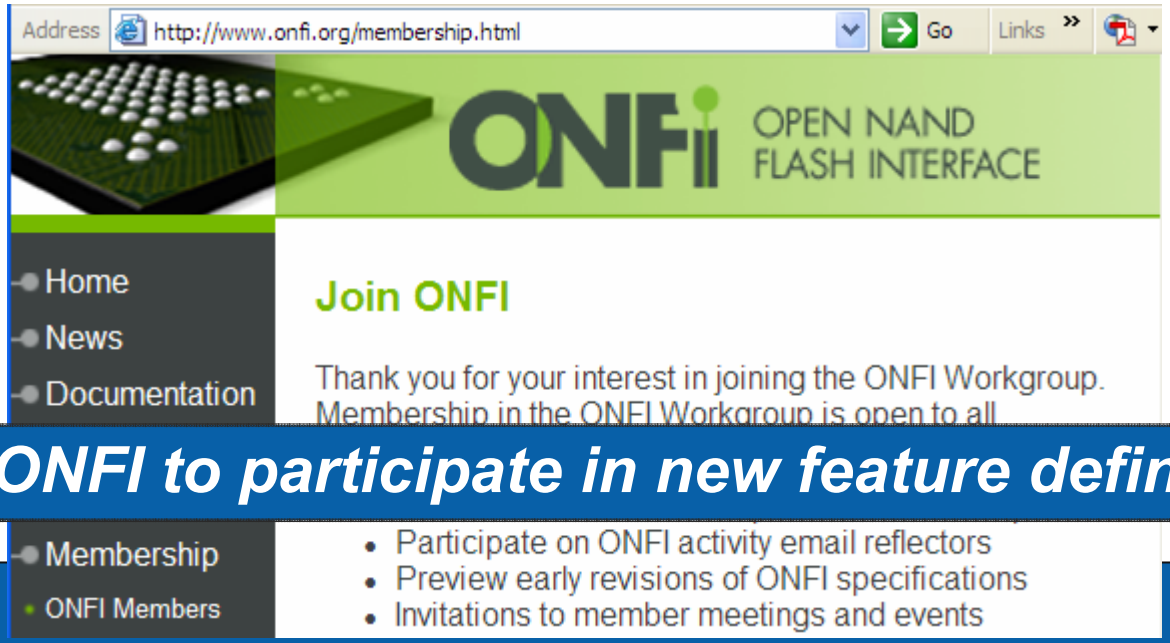
Continually Growing Coalition

Members

A-Data	Alcor Micro	Aleph One
Arasan Chip Systems	ATI	Avid Electronics
BitMicro	Biwin Technology	Cypress
DataFab Systems	DataIO	Denali
FCI	Foxconn	Fusion Media Tech
Genesys Logic	Hagiwara Sys-Com	InComm
Intelliprop	ITE Tech	Jinvani Systech
Kingston Technology	Marvell	Molex
NVidia	Orient Semiconductor	Powerchip Semi.
PQI	Qimonda	Seagate
Shenzhen Netcom	Sigmatel	Silicon Storage Tech
Silicon Motion	SimpleTech	Skymedi
Smart Modular Tech.	Solid State System	Spansion
Super Talent Elec.	Telechips	Testmetrix
Tyco	UCA Technology	WinBond

ONFI Delivering Advanced Features

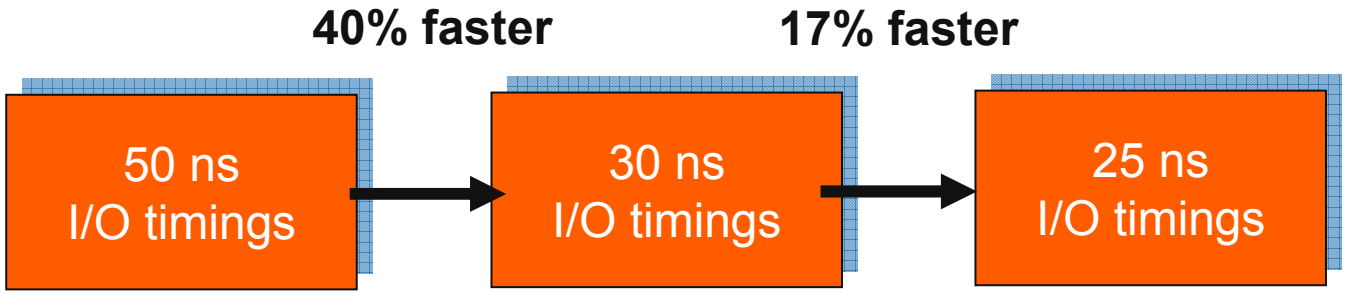
- ONFI is building on the foundation established by the 1.0 specification with significant new features:
  - High speed NAND definition to dramatically improve the interface transfer rate
  - Block abstracted NAND interface to simplify integration of NAND into host platforms
  - Connector definition for insertion of raw NAND modules into build-to-order systems



Join ONFI to participate in new feature definition.

Why is the Legacy Interface Stalling?

- **Issue 1:** The legacy interface requires that the NAND process commands in a single cycle directly impacting the write cycle time
  - Example: Reads require the NAND to process two commands and five addresses within seven cycles, followed by assertion of busy in 100 ns
- **Issue 2:** NAND timing is not source synchronous, making it difficult for the host to know where the data is valid at higher speeds
  - Supporting different configurations (e.g. single die vs quad die package) makes it difficult to latch data cleanly at higher speeds
- These issues are reflected by the slowdown in NAND timing improvements



Delivering Higher Speed

The Path to Higher Speed

Step 1: Source synchronous
- Add source synchronous data strobes

Step 2: Learn from DRAM
- The lessons of DDR can take us far

Step 3: Easy transition
- Break apart the command phase and data phase

ONFI Interface Rate Roadmap	
Legacy	40 MB/s
Gen1	~ 133 MB/s
Gen2	~ 266 MB/s
Gen3	400 MB/s +

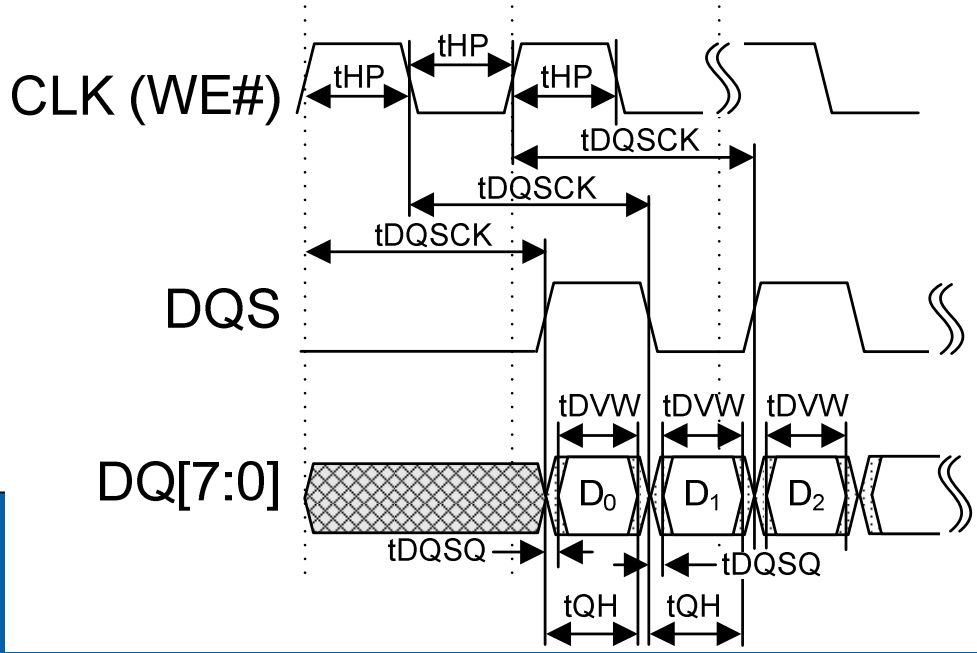
Going Source Synchronous

- I/O[7:0]: Data/address bus
  - Changed name to **DQ** to align with DRAM DDR naming conventions
- DQS: Data strobe
  - Only new signal for first generation of high speed
  - Strobe is used to indicate where data should be latched
- WE#: Write enable becomes source synchronous CLK
  - CLK is used for all interface transfers
- RE#: Read enable becomes direction signal, W/R#
  - No longer used to latch read data
  - Indicates owner of the DQ bus and the DQS signal

Symbol		Type	Description
Traditional	**Source synchronous**		
I/O[7:0]	DQ[7:0]	I/O	Data inputs/outputs
—	DQS	I/O	Data strobe
WE#	CLK	Input	Write enable => Clock
RE#	W/R#	Input	Read enable => Write / Read# direction

Adopting DDR Protocol

- High speed NAND uses a DDR protocol
- DQS identifies start of data byte on the DQ bus
  - Data is latched on each edge of DQS (rising and falling)
- Value of having a data strobe:
  - Eliminates the uncertainty of the clock insertion delay across vendors
  - Makes the design more robust to noise since the strobe and the data are impacted by noise events together
  - Easier to deal with different loading (single-die vs quad-die)



VccQ and Lower Power

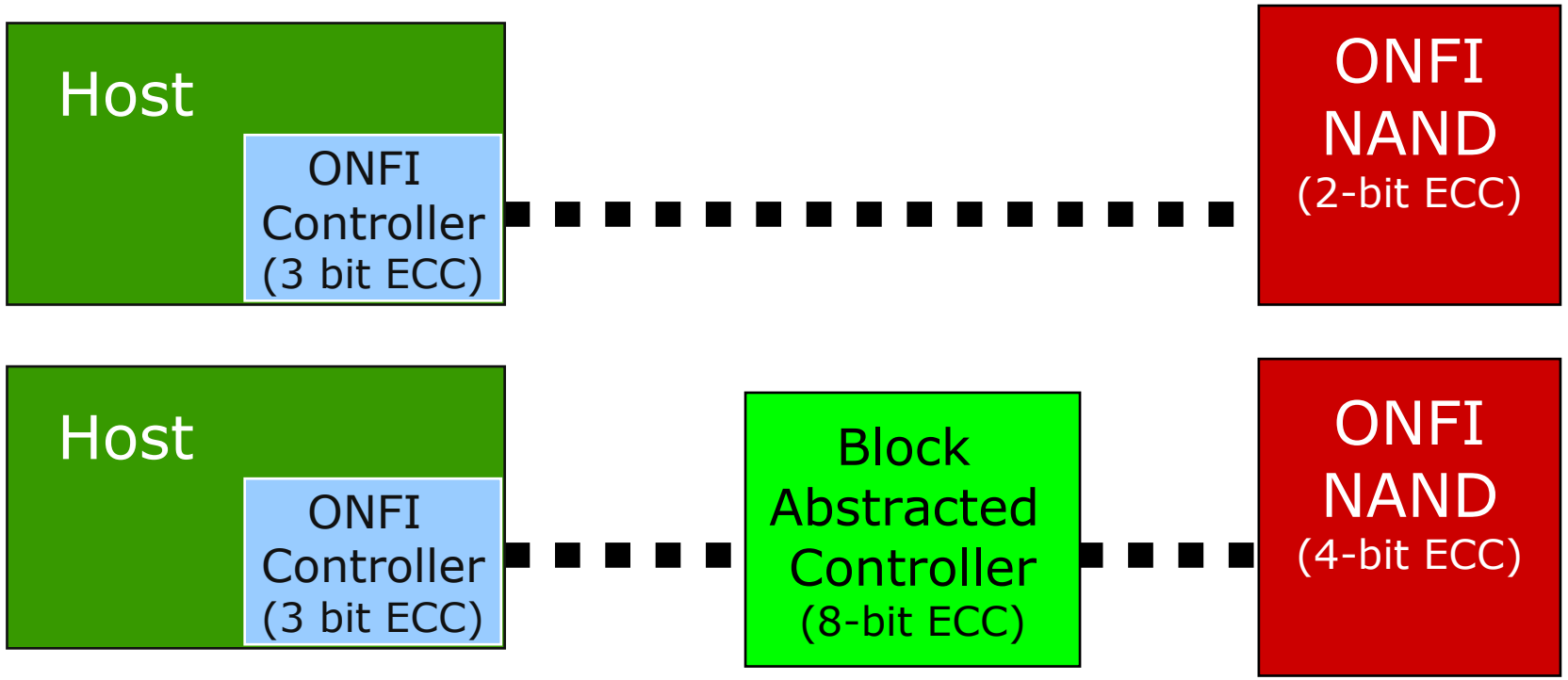
- With increased interface speed, comes increased power consumption
  - NAND is targeted at low power applications, important to optimize for power
- Solution: Scale the I/O voltage (VccQ) lower
  - For a CMOS based I/O buffer, most of the power consumption is from the driver swinging the output from 0V to VccQ
  - The power consumption per single data lane is governed by P = C * V * V *f
- For an 8-bit data bus, lowering VccQ to 1.8V can save over **600 mW** of power for worst case I/O patterns!
- Recommend scaling NAND VccQ along the lines of DRAM
  - DDR2 = 1.8V VccQ, DDR3 = 1.5V VccQ

8-bit I/O Power

	50 MHz	100 MHz	150 MHz	200 MHz
VccQ = 3.3V	218 mW	435 mW	653 mW	871 mW
VccQ = 1.8V	64 mW	129 mW	194 mW	259 mW
VccQ = 1.5V	45 mW	90 mW	135 mW	180 mW

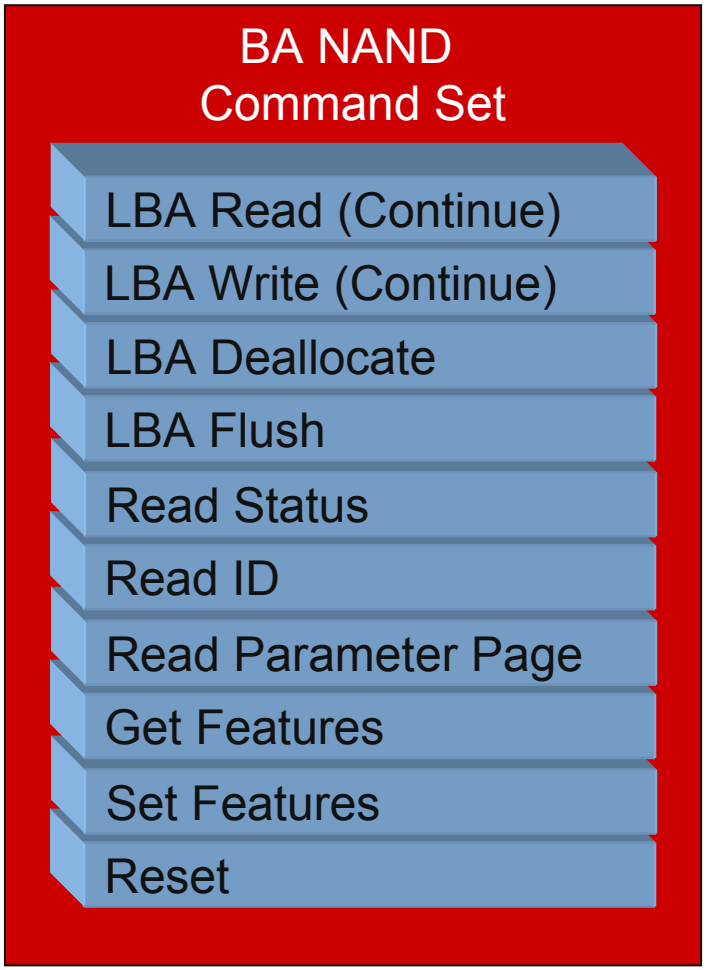
Block Abstracted NAND to enable Broader NAND Use

- NAND may have ECC or other management requirements that are beyond the host's capabilities
- Block Abstracted NAND allows a controller to be inserted in the middle that abstracts some of the complexities of NAND



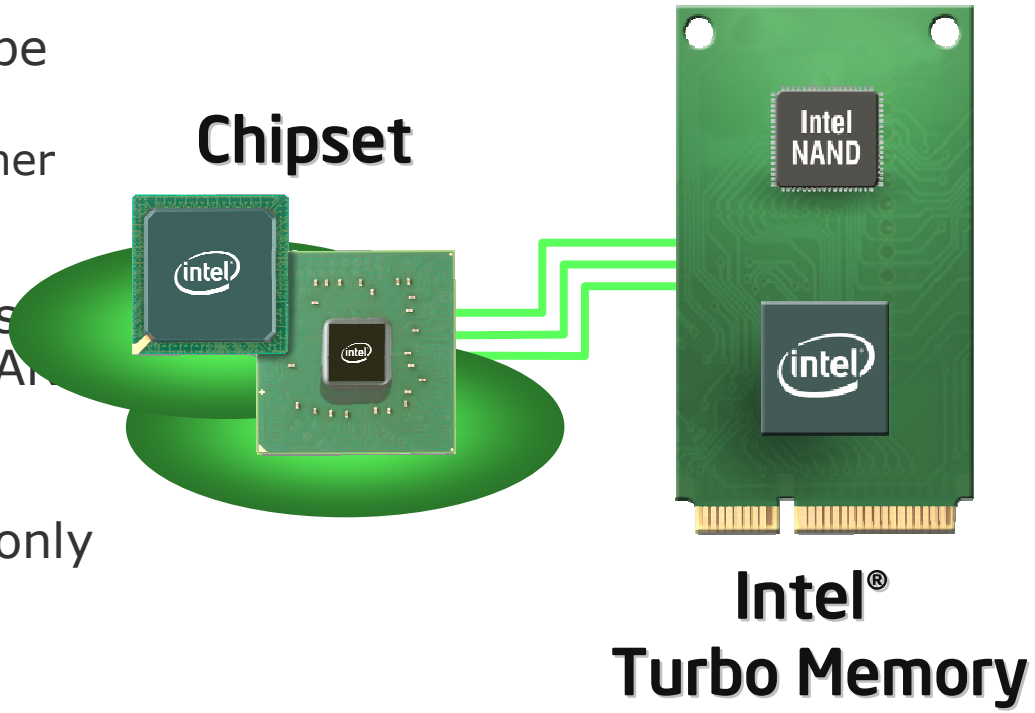
Block Abstracted Details

- Block abstracted uses the same physical interface as "raw" NAND
  - May also use high speed interface
- The command set abstracts the NAND to look more like a hard drive
  - Uses LBAs rather than NAND pages
- Block abstracted NAND controller manages bad blocks, wear levels, performs ECC, etc
- All the vagaries of NAND management may be avoided by the host



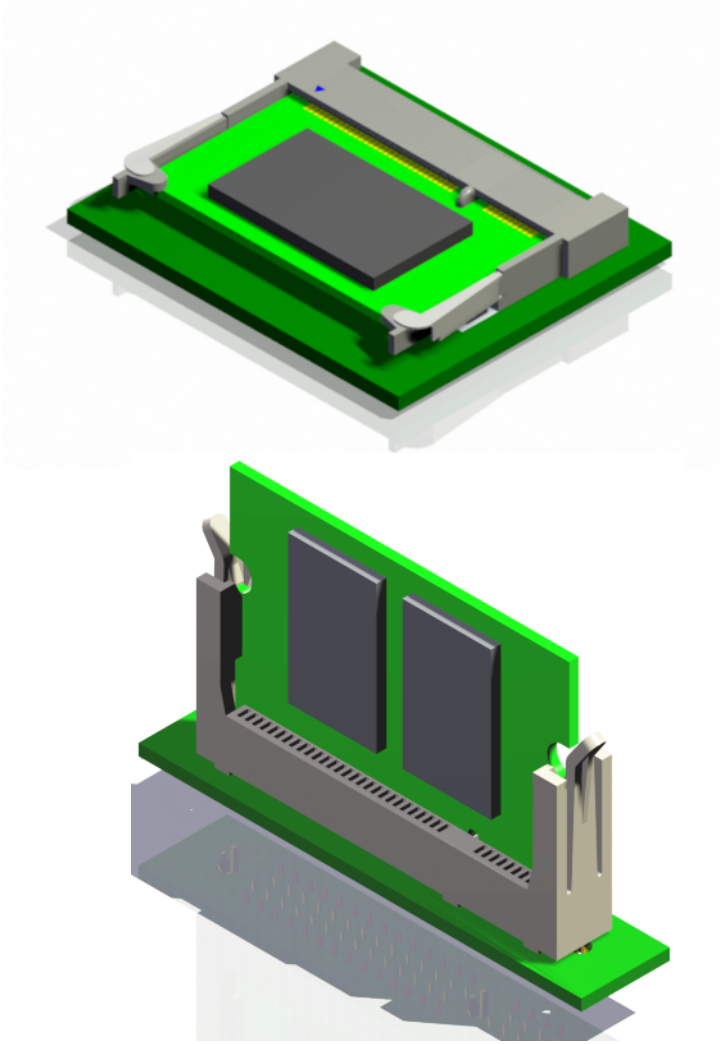
NAND in the Platform

- NAND in the platform has started with modules plugged in on PCIe
- As NAND becomes more prevalent, the controller will be integrated with the platform
  - Down on motherboard or higher levels of integration
- OEMs want to offer customers capacity/feature choice, so NAND will remain on a module
- **Issue:** How to plug a NAND-only module into a PC platform?
  - NAND does not talk PCIe*



Connector for NAND-only Modules

- To offer capacity choice, ONFI is defining a standard connector
  - Enables OEMs to sell NAND on a module
  - Like an unbuffered and unregistered DIMM
- The ONFI connector effort is leveraging existing DRAM standards
  - Avoids major connector tooling costs
  - Re-uses electrical verification
  - Ensures low cost with quick time to market
- Both right-angle and vertical entry form factors are being delivered



Summary

- ONFI 1.0 has established a standard interface for NAND
- ONFI 2.0 is adding significant new features on this foundation
  - High speed NAND definition to dramatically improve the interface transfer rate
  - Block abstracted NAND interface to simplify integration of NAND into host platforms
  - Connector definition for insertion of raw NAND modules into systems for late-binding configurations
- Join the ONFI Workgroup to get involved in these exciting new development activities!

Additional sources of information on this topic:

- More web based info: www.onfi.org

This Session presentation (PDF) is available from www.intel.com/idf web site under Technical Training. Some sessions will also provide Audio-enabled presentations after the event.

Please fill out the Session Evaluation Form

Thank You for your input, we use it to improve future Intel Developer Forum events

Save your date for IDFs this Fall:

San Francisco, USA 2007 September 18 -20

Taipei, Taiwan, 2007 October 15 -16


intel
Leap ahead™