

# Introduction to Flash Memory (T1A)

Jim Cooke ([jcooke@micron.com](mailto:jcooke@micron.com))

Staff Architect and Technologist,  
Architecture Development Group

Micron Technology, Inc.

# Agenda

- **The basics of Flash and NAND**
  - Flash cell comparison
  - NAND and NOR attributes and interface comparison
  - Detailed operations
    - Commands, address, and data operations
- **Connecting NAND to a RISC or DSP processor**
- **More NAND Flash device detail**
  - SLC vs. MLC
  - All NAND devices are not created equal
    - Architecture, features, and performance comparisons
- **Performance bottlenecks**
- **ONFI and high-speed NAND introduction**
- **NAND error modes**
  - Program disturb
  - Read disturb
  - Data retention
  - Endurance
  - Wear-leveling
  - ECC fixes almost everything

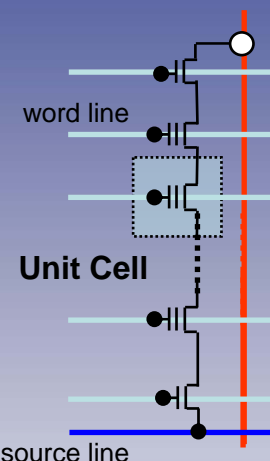
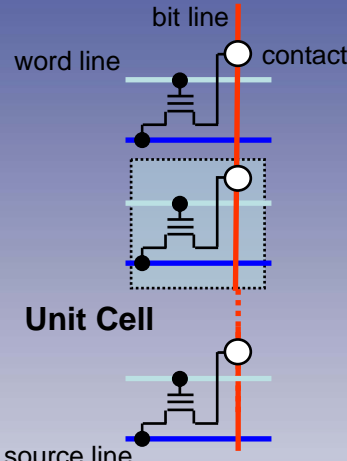
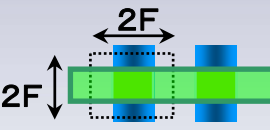
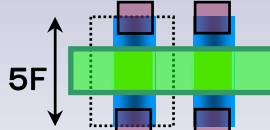


# A Quick Review of Flash Basics

- ▶ Cell differences
- ▶ NAND attributes
- ▶ NAND vs. NOR

# Flash Basics

- Flash data is grouped into blocks, which are the smallest erasable entity
  - Erasing a block sets all bits to “1” or bytes to FFh
- The programming operation changes erased bits from “1” to “0”
  - The smallest entity that can be programmed is a bit
- While NAND cannot inherently perform random access, it is possible at the system level through shadowing

# Flash Memory Cell Comparison

	NAND	NOR
Cell array	 <p>word line</p> <p>Unit Cell</p> <p>source line</p>	 <p>bit line</p> <p>word line</p> <p>contact</p> <p>Unit Cell</p> <p>source line</p>
Layout	 <p>2F</p> <p>2F</p>	 <p>2F</p> <p>5F</p>
Cross-section		
Cell size	<b><math>4F^2</math></b>	<b><math>10F^2</math></b>

- NAND Flash's small cell size enables high density and low cost

# Basic NAND Attributes

- NAND is very similar to a disk drive; it is sector-based (page-based) Flash and is well-suited for storage of sequential data (such as pictures, audio, and files)
  - Like a disk drive, NAND is not well-suited for random access, such as executing code, although random access can be accomplished at the system level by shadowing the data to RAM (similar to what a PC does with BIOS)
  - Like a disk drive, NAND devices have bad sectors or blocks and require management
  - Like a disk drive, NAND requires error correction code (ECC)
  - Unlike a disk drive, it is possible to wear out the NAND cell; with good wear-leveling, this is typically not an issue

# Basic NAND Attributes

- NAND is available in large capacities and is the lowest cost Flash memory available today
- NAND is finding its way into many embedded applications and is used in virtually all removable cards
  - USB cards
  - Memory stick
  - MMC multimedia card
  - SD secure digital
  - CF compact Flash
- Multiplexed interface provides similar pinout over all devices
  - x8 signal pinout has not changed from 64Mb pinout
- x8 devices are used mostly in high capacity (3.3V) consumer applications; the x16 devices are mostly used in embedded (1.8V) applications

# Basic NAND/NOR Comparison

## NAND

- Advantages
  - Fast writes
  - Fast erases
- Disadvantages
  - Slow random access
  - Byte writes difficult
- Applications
  - File (disk) applications
  - Voice, data, video recorder
  - Any large sequential data

## NOR

- Advantages
  - Random access
  - Byte writes possible
- Disadvantages
  - Slow writes
  - Slow erase
- Applications
  - Replacement of EPROM
  - Execute directly from nonvolatile memory



# Flash Memory Comparison

Characteristic	NAND Flash MT29F2G08	NOR MT28F128J3
Random access read	25 $\mu$ s (first byte) 0.03 $\mu$ s each for remaining 2,111 bytes	0.12 $\mu$ s
Sustained read speed (sector basis)	23 MB/s (x8) or 37 MB/s (x16)	20.5 MB/s (x8) or 41 MB/s (x16)
Random write speed	~300 $\mu$ s/2,112 bytes	180 $\mu$ s/32 bytes
Sustained write speed (sector basis)	5 MB/s	0.178 MB/s
Erase block size	128KB	128KB
Erase time per block (typ)	2ms	750ms

**NAND Flash is ideal for file storage, such as data or image files; if code is stored, it must be shadowed to RAM first, as in a PC**

**NOR Flash is ideal for direct code execution (boot code) although it still needs to be shadowed (for speed)**

# Flash Interface Comparison

## ■ NOR Flash

- Random-access interface typically composed of:

- CE# — chip enable
- WE# — write enable
- OE# — output enable
- D15-D0 — data bus
- A20-A0 — address bus
- WP# — write protect

41 pins

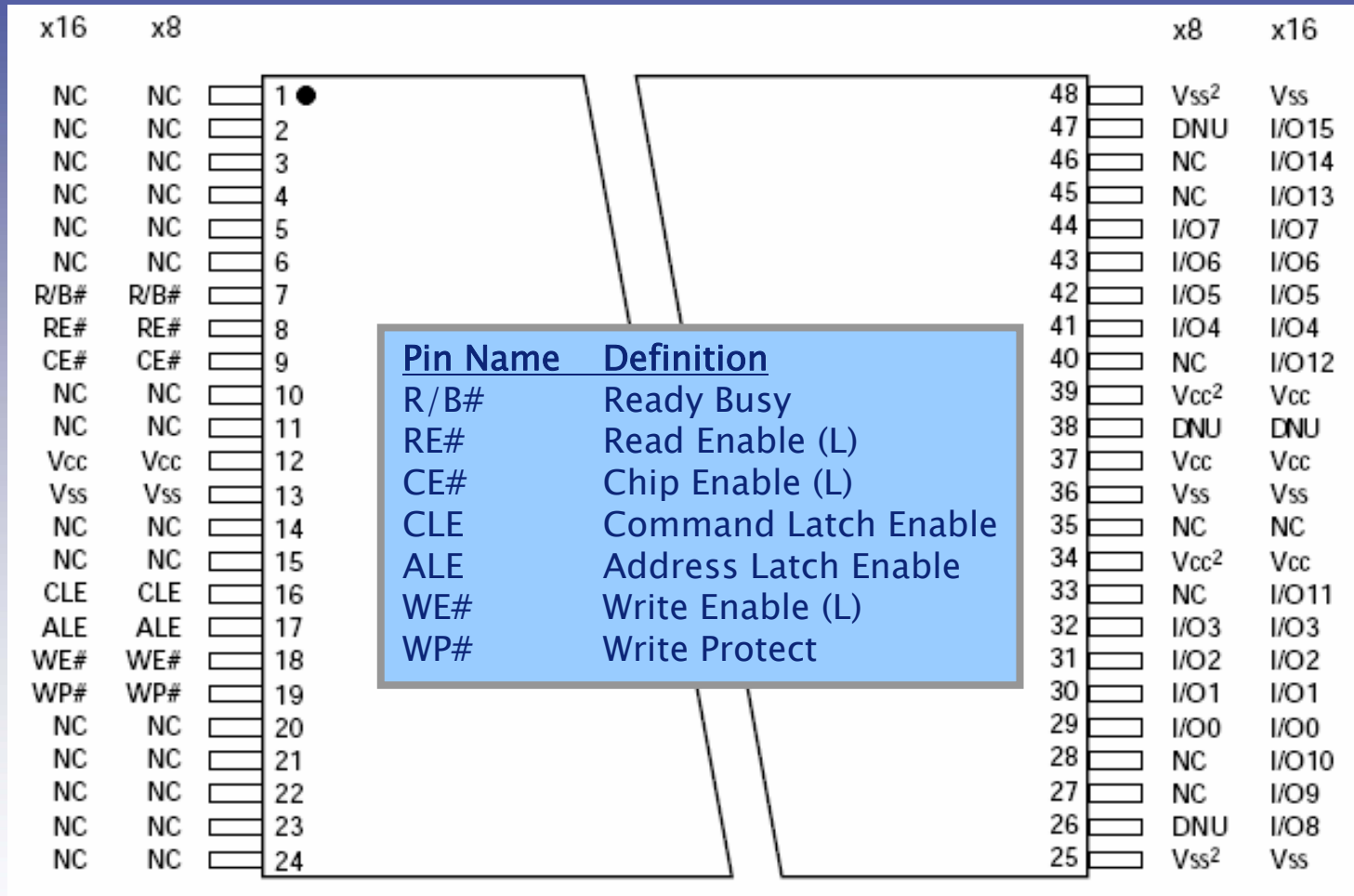
## ■ NAND Flash

- I/O device-type interface composed of:

- CE# — chip enable
- WE# — write enable
- RE# — read enable
- CLE — command latch enable
- ALE — address latch enable
- I/O 7-0 — data bus (I/O 15-0 for x16 parts)
- WP# — write protect
- R/B# — ready/busy

23 pins  
(for x16)

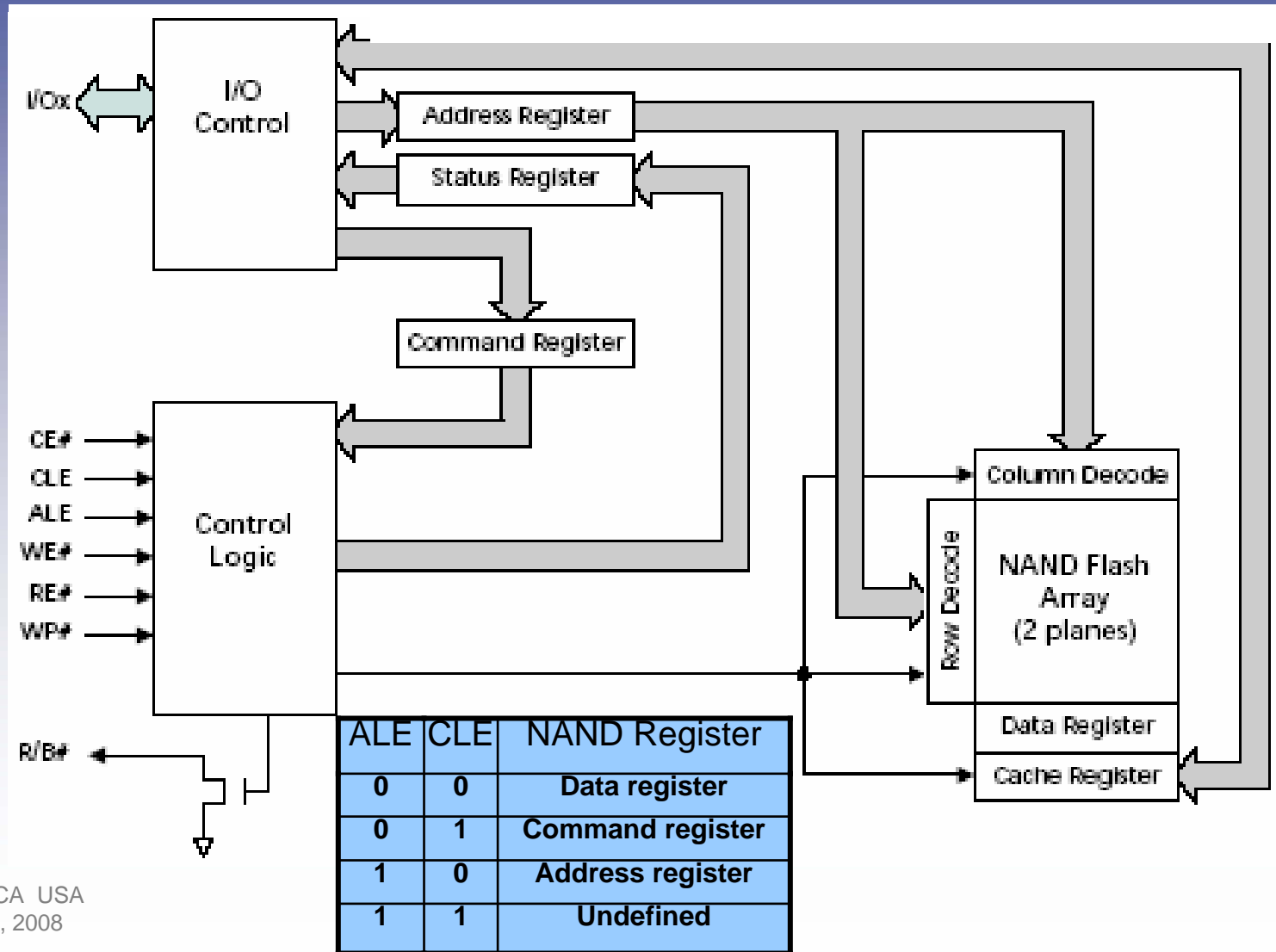
# NAND Flash Physical Interface (TSOP 1)



**Indirect addressing enables no pinout changes among densities**

**Note 2: Additional Vcc and Vss recommended for new PCB designs**

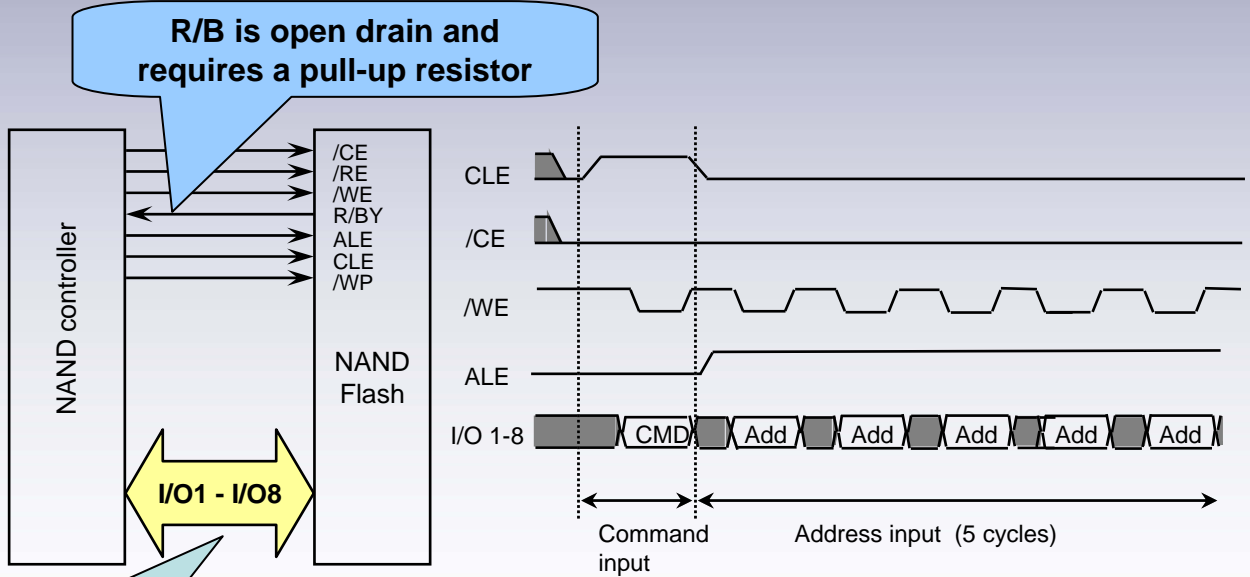
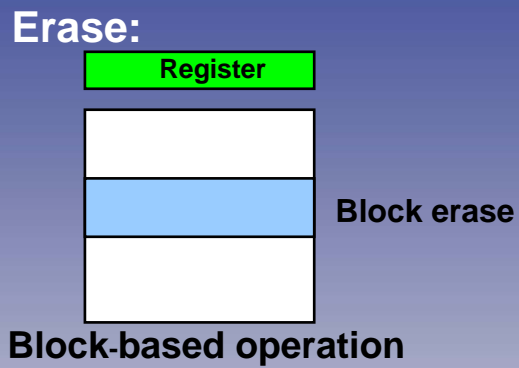
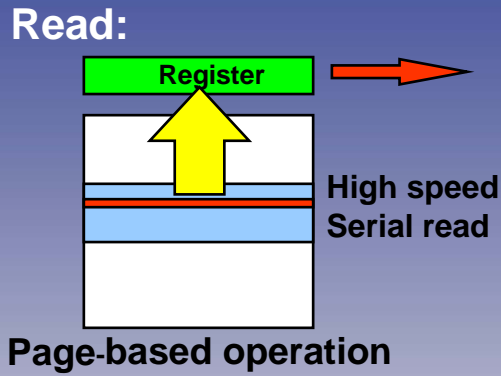
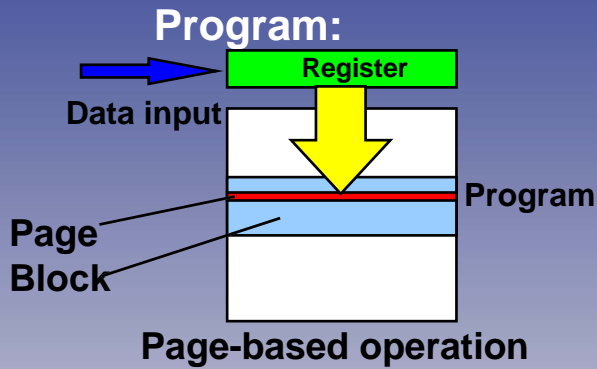
# NAND Block Diagram



# Detailed Operations

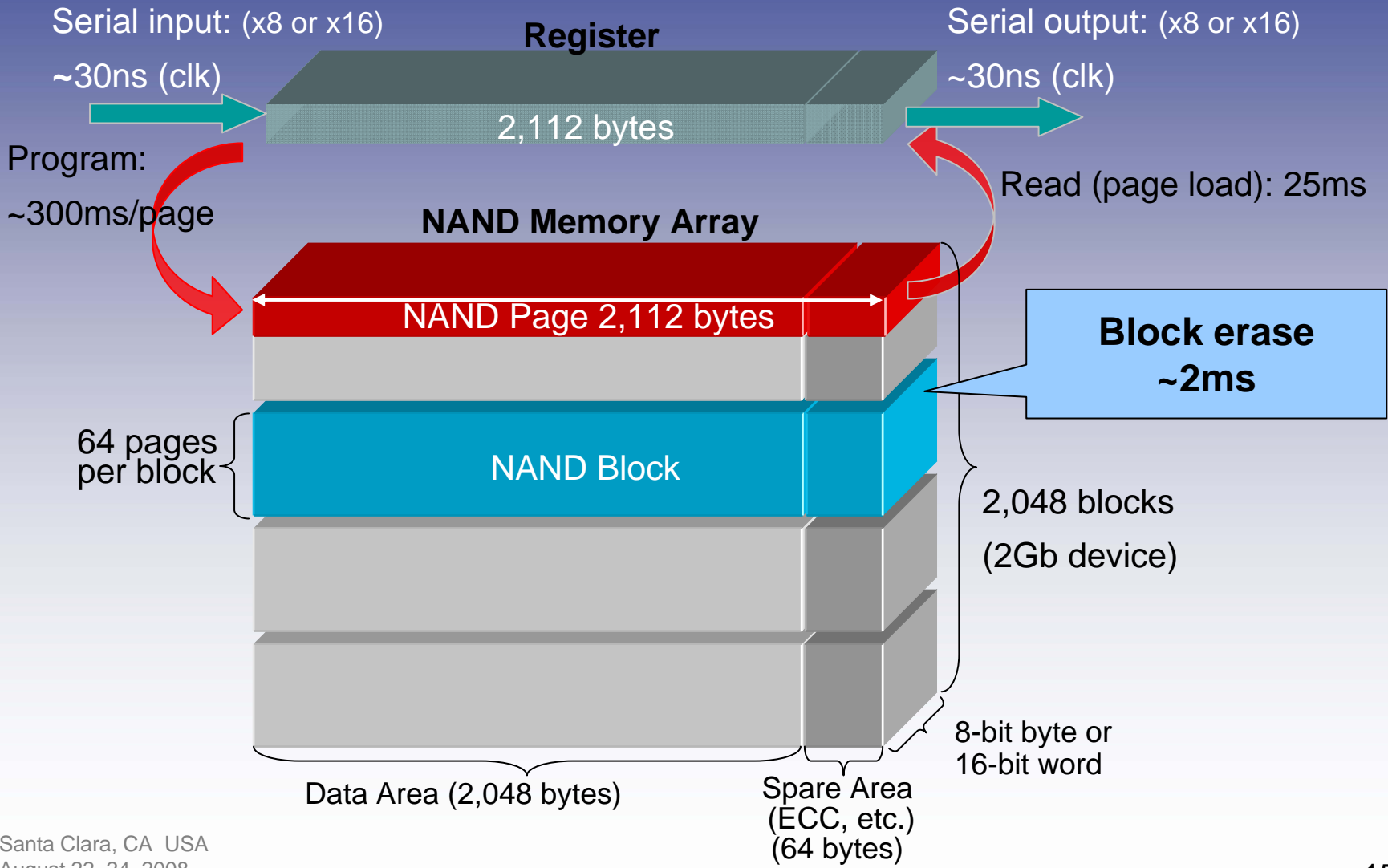
Architecture  
Addressing  
Basic commands

# Basic NAND Flash Operations

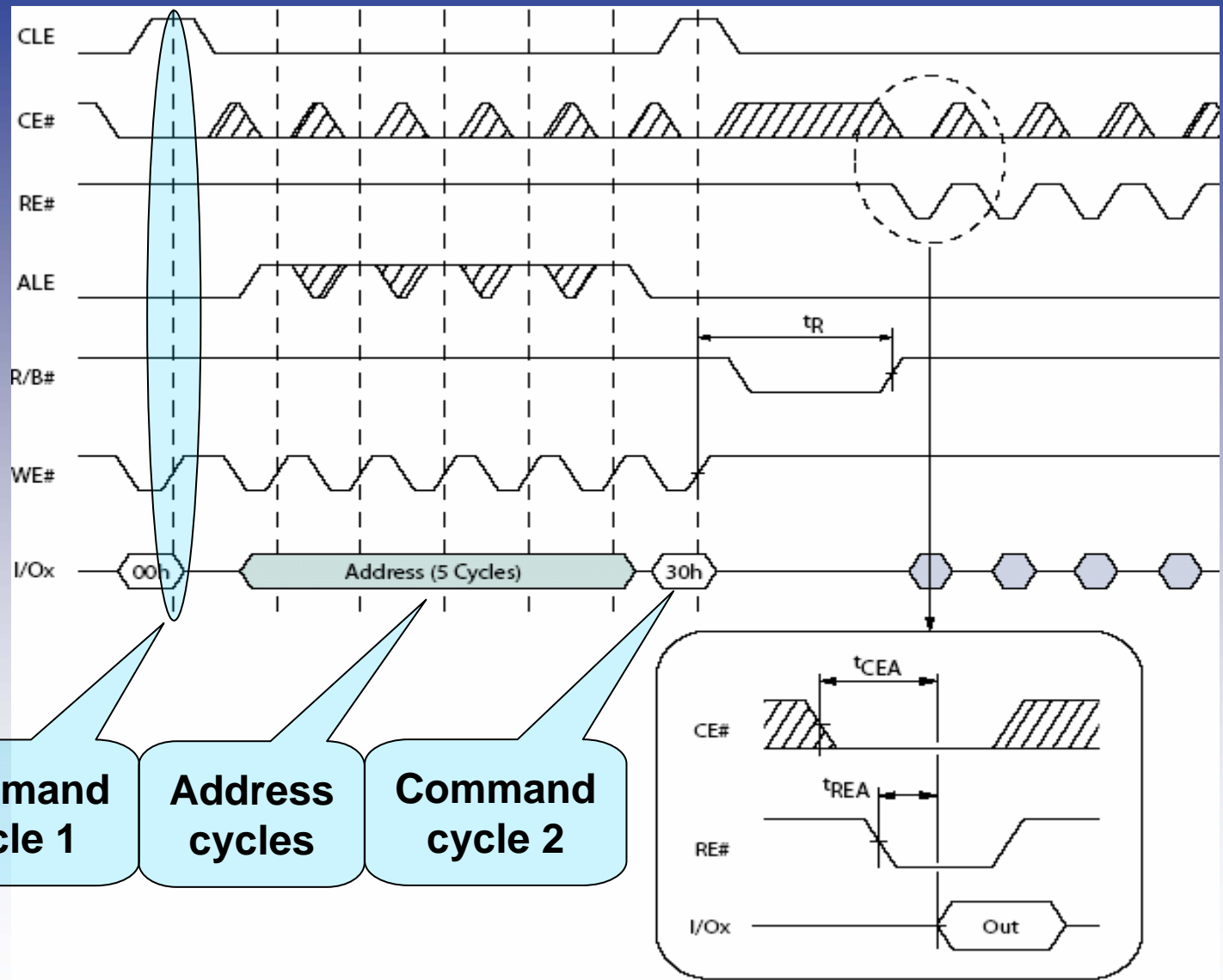


Multiplexed command, address, data protocol

# SLC NAND Flash Memory Diagram



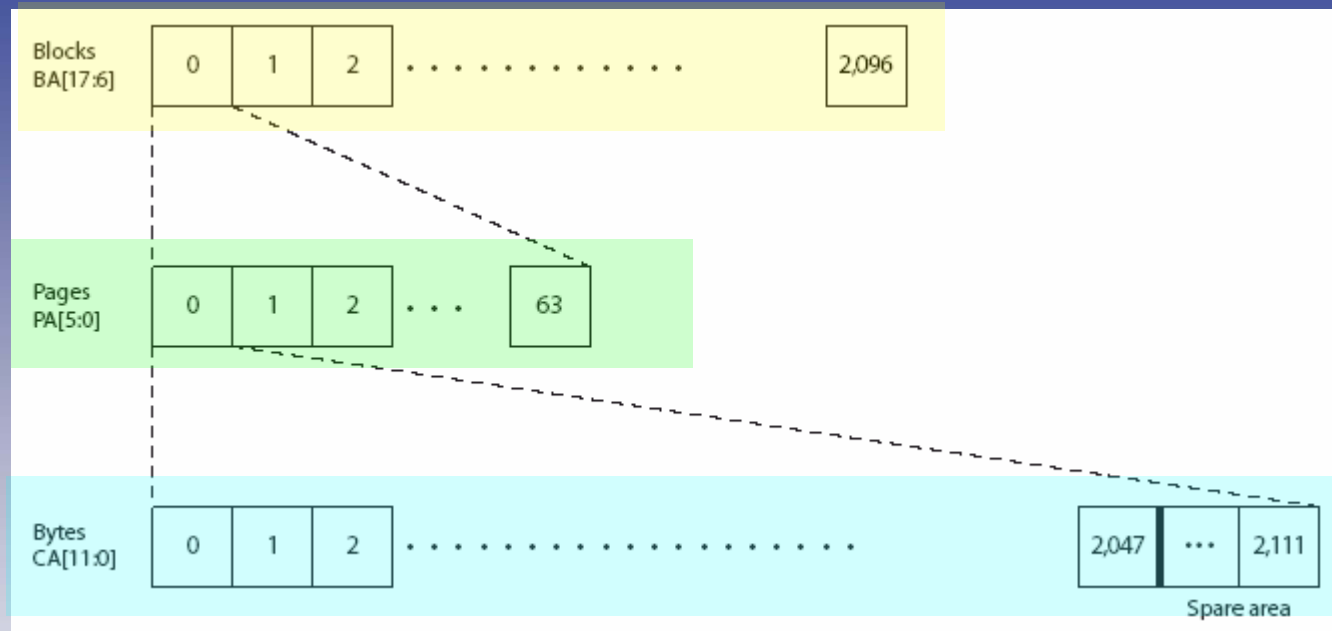
# Basic Access



**Command cycle 1**      **Address cycles**      **Command cycle 2**



# Large-Block NAND Addressing



CAx = column address; RAx = row address.

Cycle	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
First	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Second	LOW	LOW	LOW	LOW	CA11	CA10	CA9	CA8
Third	RA19	RA18	RA17	RA16	RA15	RA14	RA13	RA12
Fourth	RA27	RA26	RA25	RA24	RA23	RA22	RA21	RA20
Fifth	LOW	LOW	LOW	LOW	LOW	LOW	RA29 <sup>1</sup>	RA28

Notes: 1. Die address boundary: 0 = 0 – 2Gb, 1 = 2Gb – 4Gb.

# NAND Command List

## Standard 2Gb (256MB) NAND

Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
BLOCK ERASE	60h	3	No	D0h	No
RESET	FFh	—	No	—	Yes

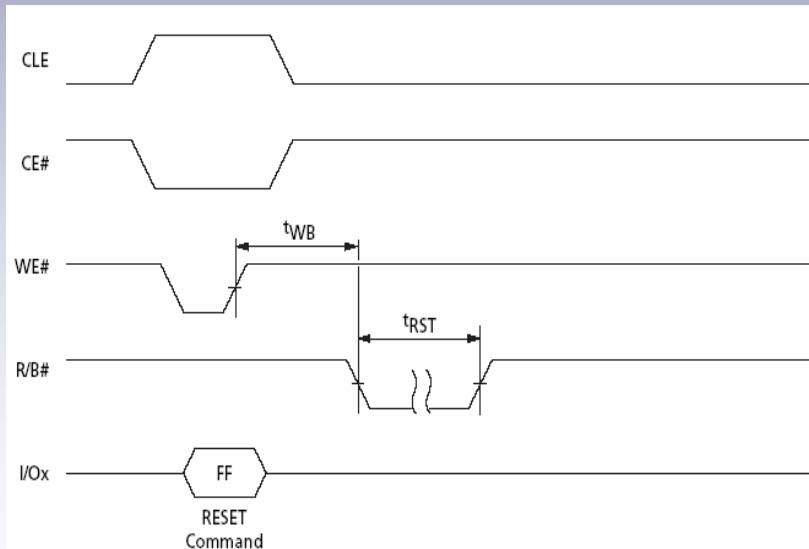
For ease of presentation:

Basic command

Advanced command

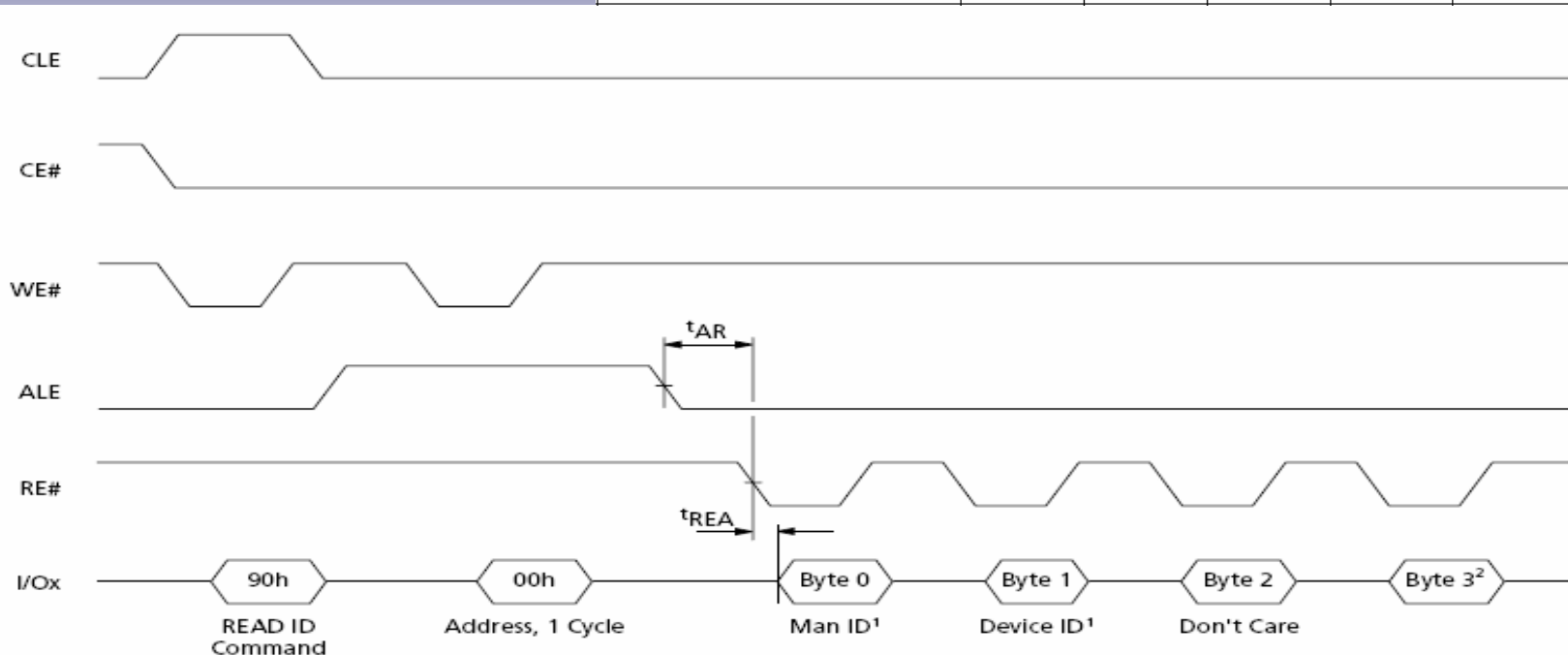
# Reset Operation

Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
BLOCK ERASE	60h	3	No	D0h	No
RESET	FFh	—	No	—	Yes



# Read ID Operation

Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
BLOCK ERASE	60h	3	No	D0h	No
RESET	FFh	—	No	—	Yes

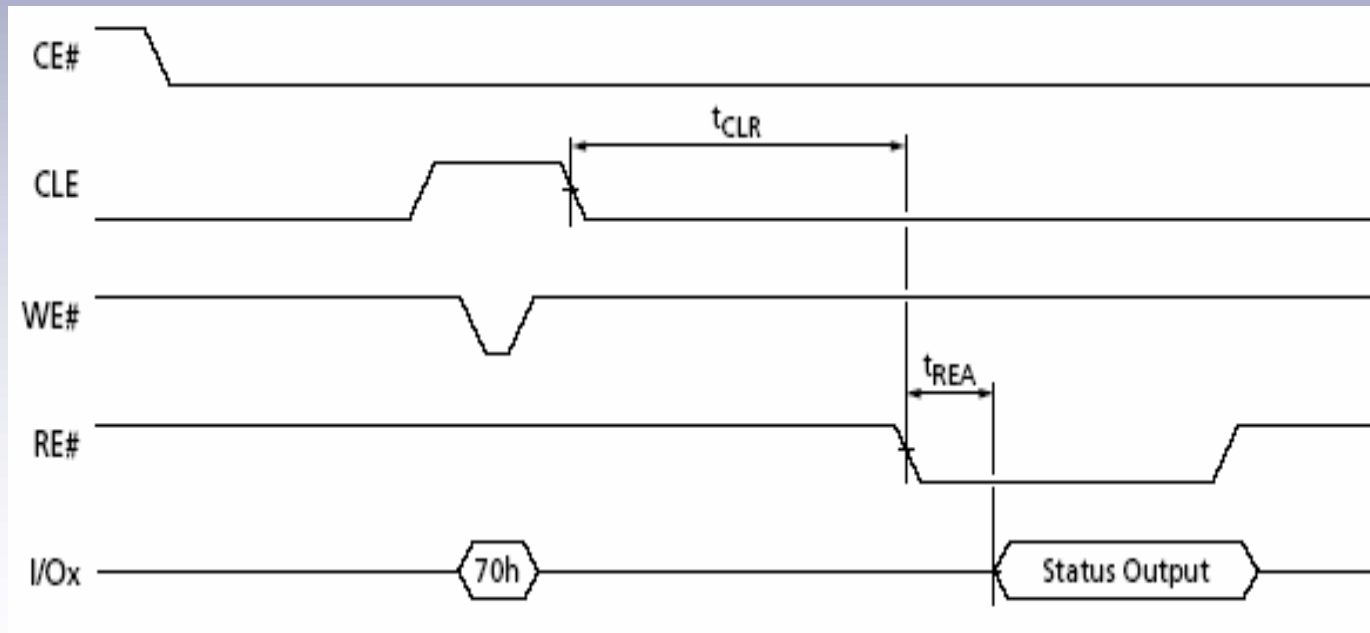


# Device IDs

Density	x8/x16	1.8V/3.3V	# of Die	Byte 0 Manf. ID	Byte 1 Device ID
1Gb	x8	1.8V	1	2Ch	A1h
1Gb	x8	3.3V	1	2Ch	F1h
1Gb	x16	1.8V	1	2Ch	B1h
1Gb	x16	3.3V	1	2Ch	C1h
2Gb	x8	1.8V	1	2Ch	AAh
2Gb	x8	3.3V	1	2Ch	DAh
2Gb	x16	1.8V	1	2Ch	BAh
2Gb	x16	3.3V	1	2Ch	CAh
4Gb	x8	1.8V	2	2Ch	ACh
4Gb	x8	3.3V	2	2Ch	DCh
4Gb	x16	1.8V	2	2Ch	BCh
4Gb	x16	3.3V	2	2Ch	CCh
8Gb	x8	1.8V	4	2Ch	ACh
8Gb	x8	3.3V	4	2Ch	DCh
8Gb	x16	1.8V	4	2Ch	BCh
8Gb	x16	3.3V	4	2Ch	CCh

# Read Status Operation

Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
BLOCK ERASE	60h	3	No	D0h	No
RESET	FFh	—	No	—	Yes

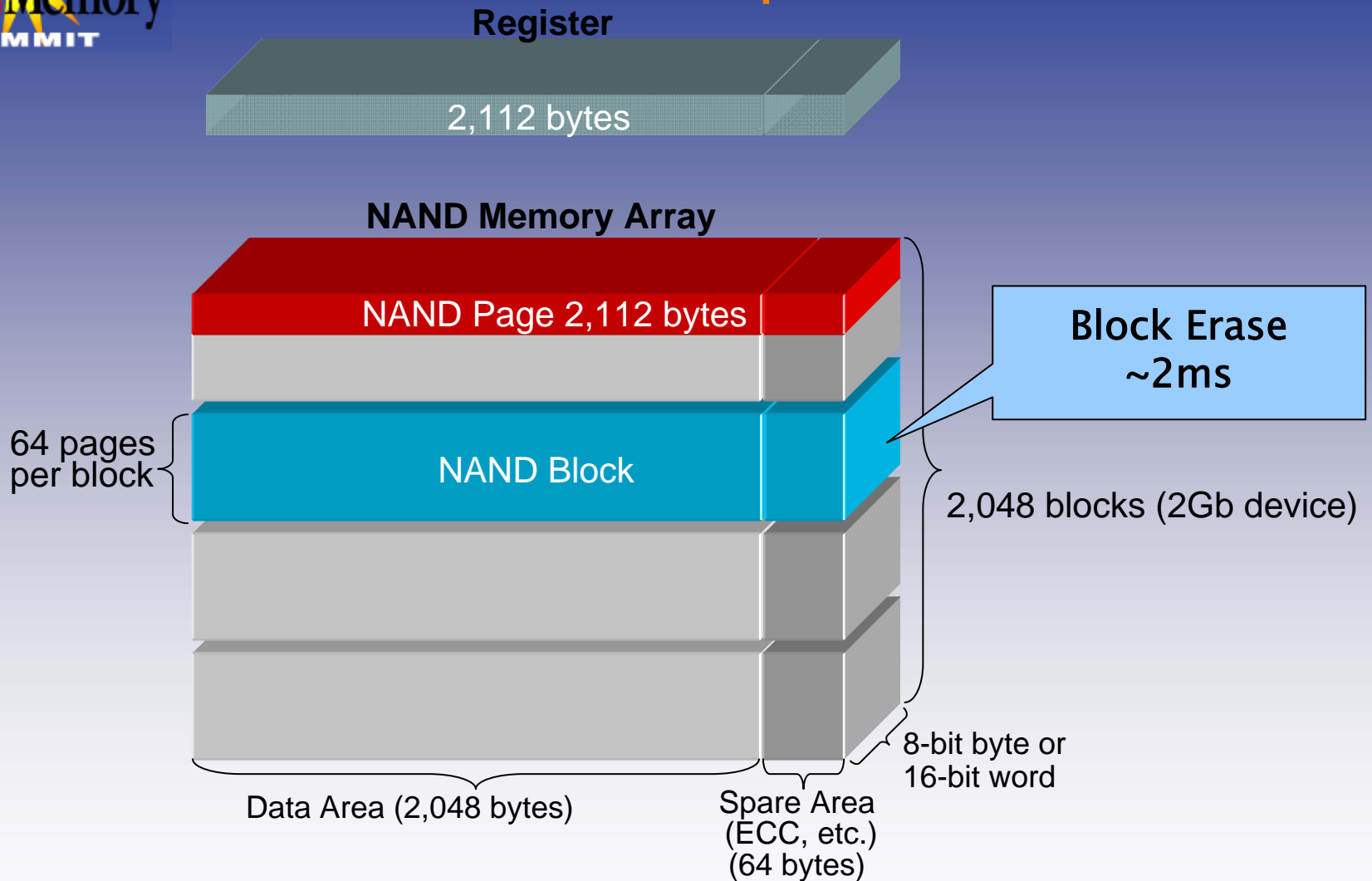


# NAND Flash Read Status Results

SR Bit	Program Page	Program Page Cache Mode	Page Read	Page Read Cache Mode	Block Erase	Definition
0 <sup>1</sup>	Pass/fail	Pass/fail (N)	-	-	Pass/fail	0 = Successful PROGRAM/ERASE 1 = Error in PROGRAM/ERASE
1	-	Pass/fail (N-1)	-	-	-	0 = Successful PROGRAM 1 = Error in PROGRAM
2	-	-	-	-	-	0
3	-	-	-	-	-	0
4	-	-	-	-	-	0
5	Ready/busy	Ready/busy <sup>2</sup>	Ready/busy	Ready/busy <sup>2</sup>	Ready/busy	0 = Busy 1 = Ready
6	Ready/busy	Ready/busy cache <sup>3</sup>	Ready/busy	Ready/busy cache <sup>3</sup>	Ready/busy	0 = Busy 1 = Ready
7	Write protect	Write protect	Write protect	Write protect	Write protect	0 = Protected 1 = Not protected

**Read status typically = E0h when the NAND is ready with no error**

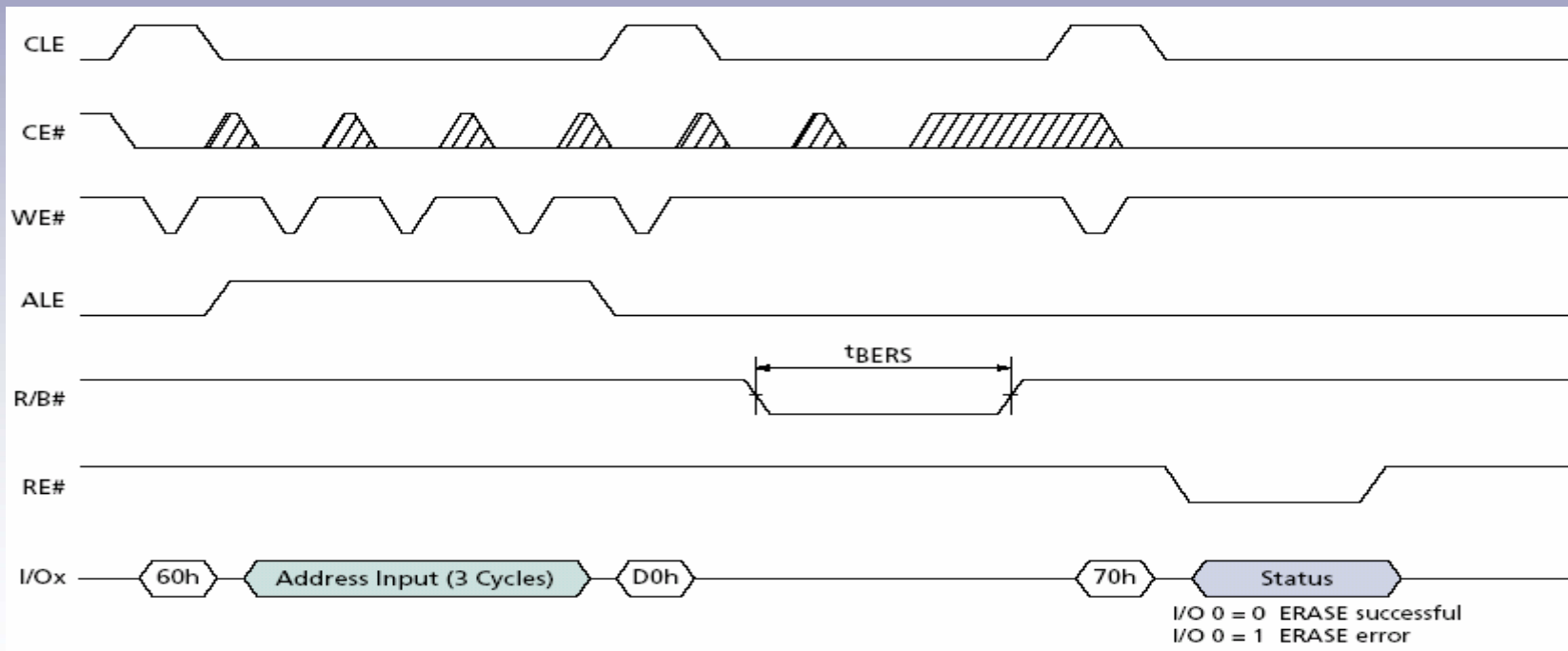
# Block Erase Operation



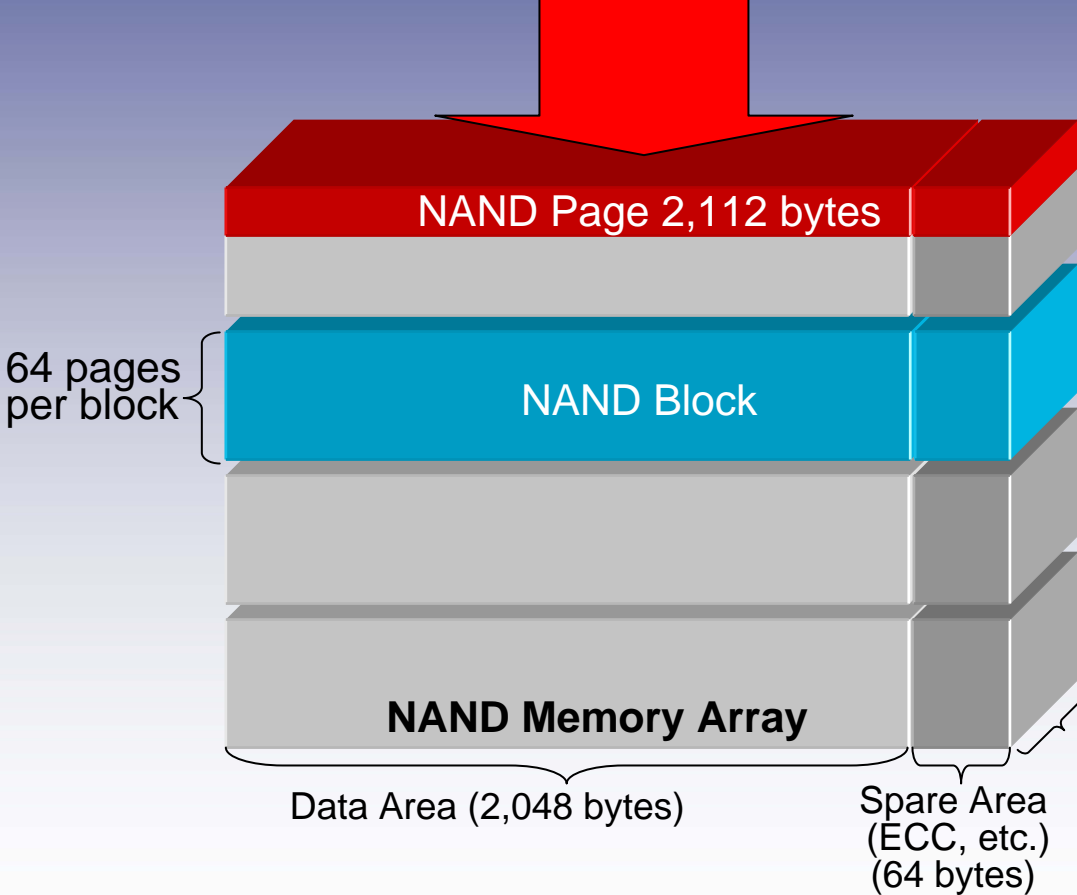


# Block Erase Operation

Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
<b>BLOCK ERASE</b>	<b>60h</b>	<b>3</b>	<b>No</b>	<b>D0h</b>	<b>No</b>
RESET	FFh	—	No	—	Yes



# Program Operation

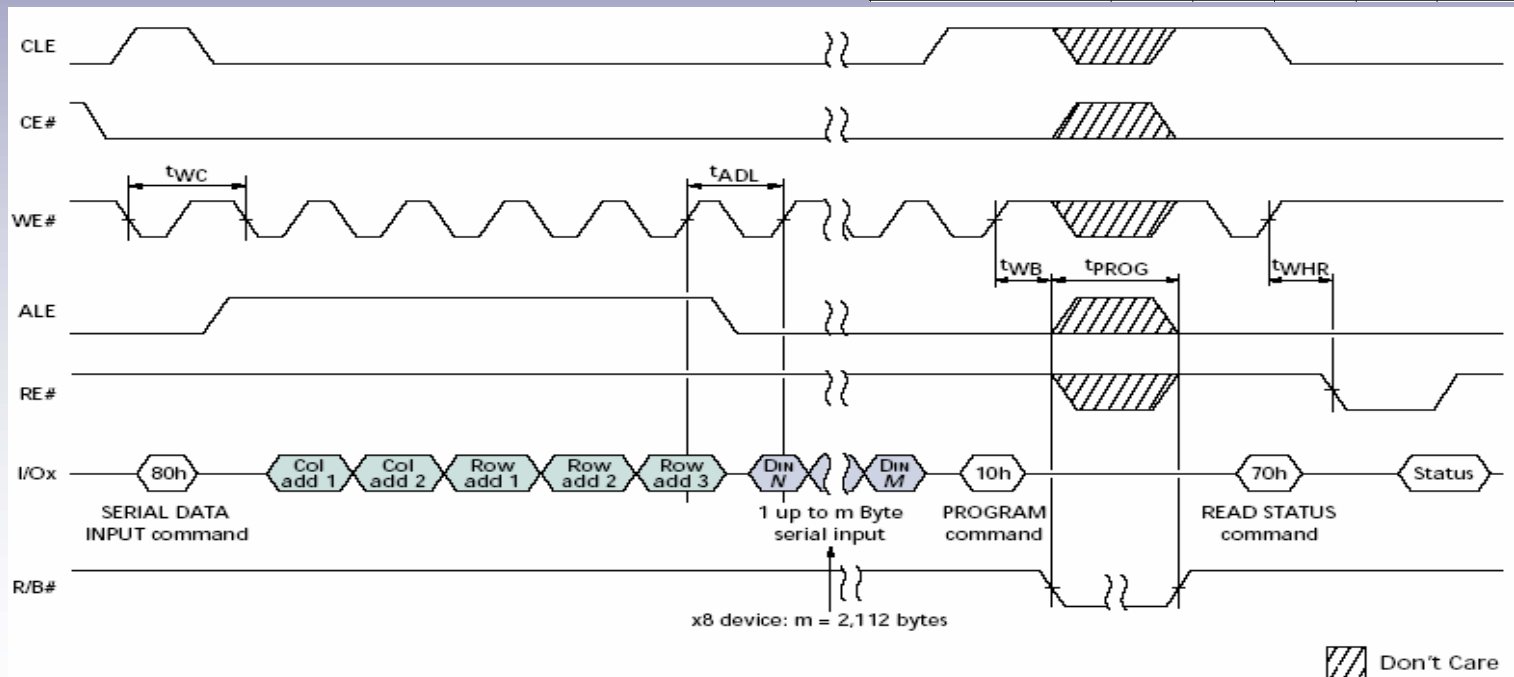


- The programming operation can program only “0” bits
- If you don’t want to program a bit, set it to “1”
- The register is automatically loaded with all “1s” by the 80h command (note the 85h command does not do this)
- After a bit has been programmed to a “0,” if you want to turn it back to a “1,” you must complete a block erase to return the entire block back to all “1s”
- Programming must be done sequentially (within a block)

**Program (tPROG):**  
**~300µs/page**

# Program Operation

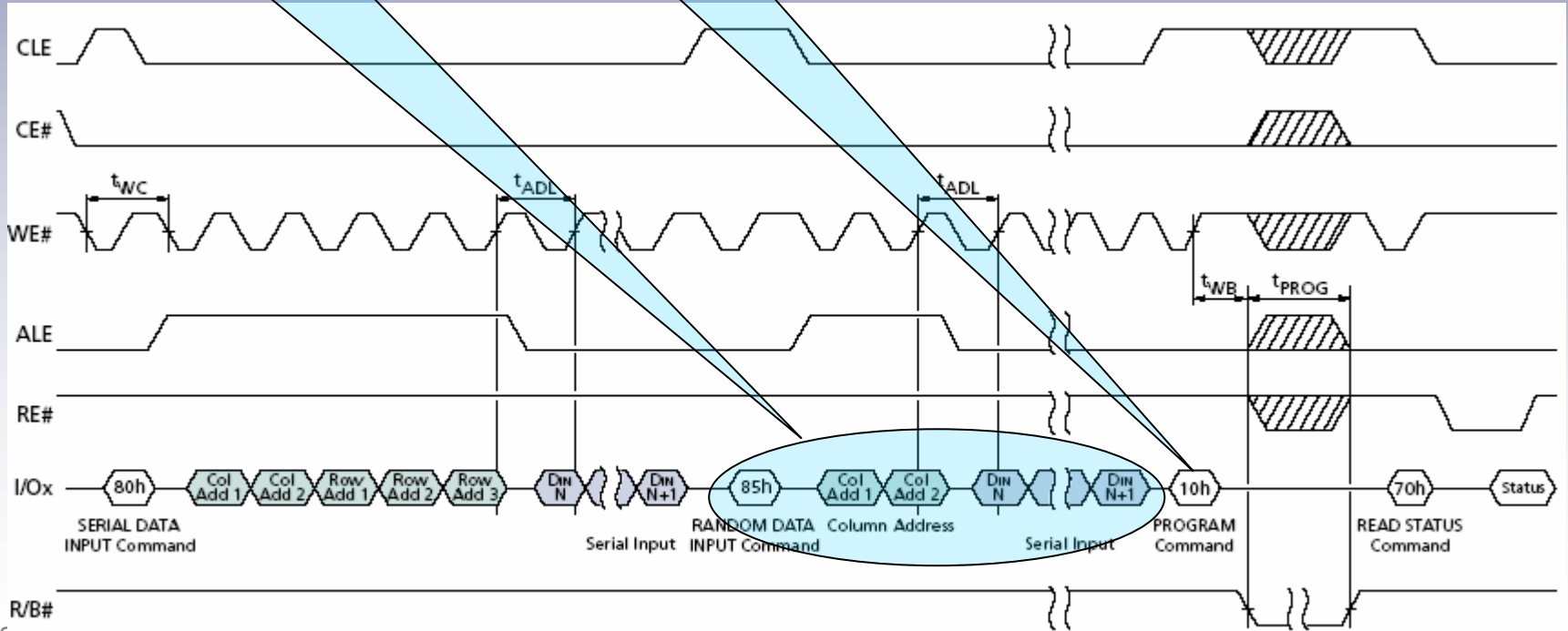
Command	Command Cycle 1	Number of Address Cycles	Data Cycles Required <sup>1</sup>	Command Cycle 2	Valid During Busy
PAGE READ	00h	5	No	30h	No
PAGE READ CACHE MODE START	31h	—	No	—	No
PAGE READ CACHE MODE START LAST	3Fh	—	No	—	No
READ for INTERNAL DATA MOVE	00h	5	No	35h	No
RANDOM DATA READ	05h	2	No	E0h	No
READ ID	90h	1	No	—	No
READ STATUS	70h	—	No	—	Yes
PROGRAM PAGE	80h	5	Yes	10h	No
PROGRAM PAGE CACHE MODE	80h	5	Yes	15h	No
PROGRAM for INTERNAL DATA MOVE	85h	5	Optional	10h	No
RANDOM DATA INPUT	85h	2	Yes	—	No
BLOCK ERASE	60h	3	No	D0h	No
RESET	FFh	—	No	—	Yes



# Random Data Input/Program

You can input as many address and data combinations as you want.

The page is programmed when you issue the 10h confirmation. Each of these counts toward the partial-page programming limit (of 8).

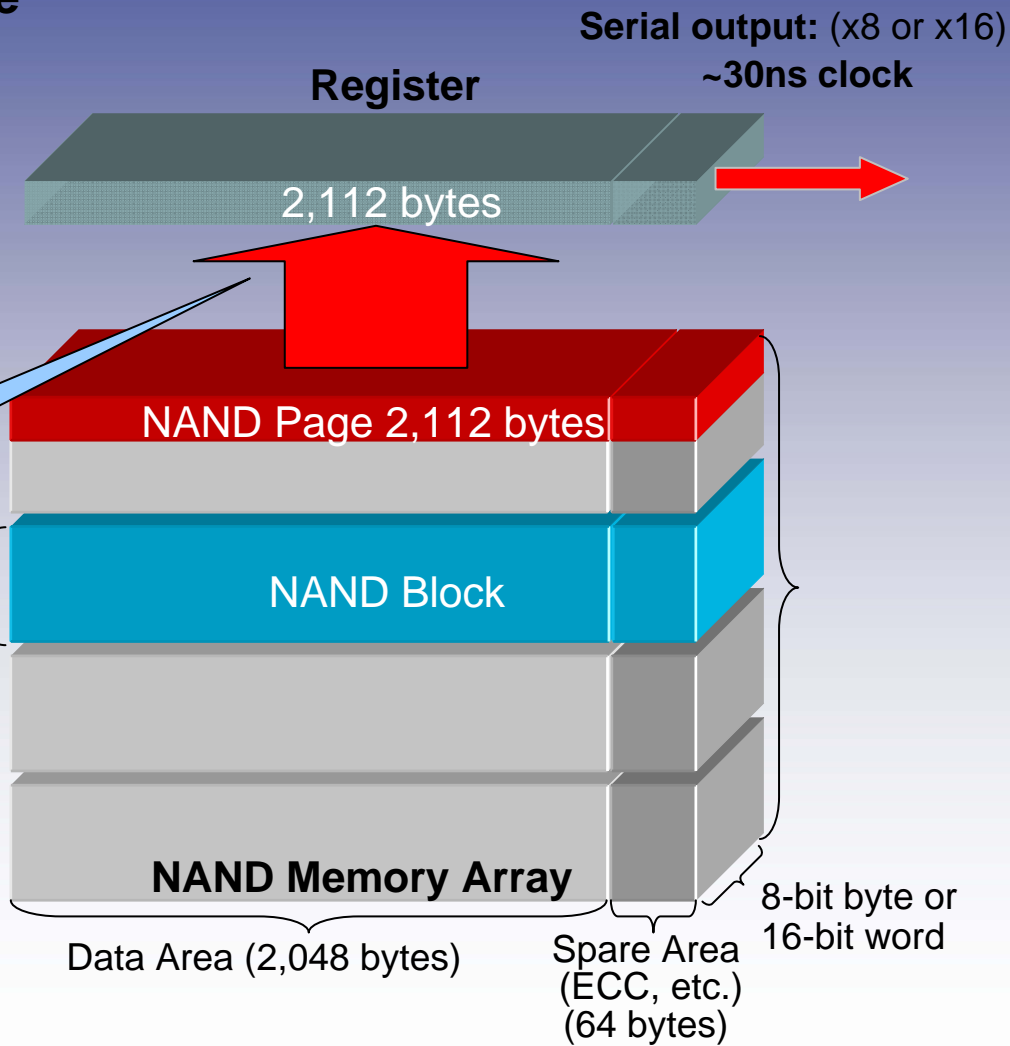


 Don't Care

# Read Operation

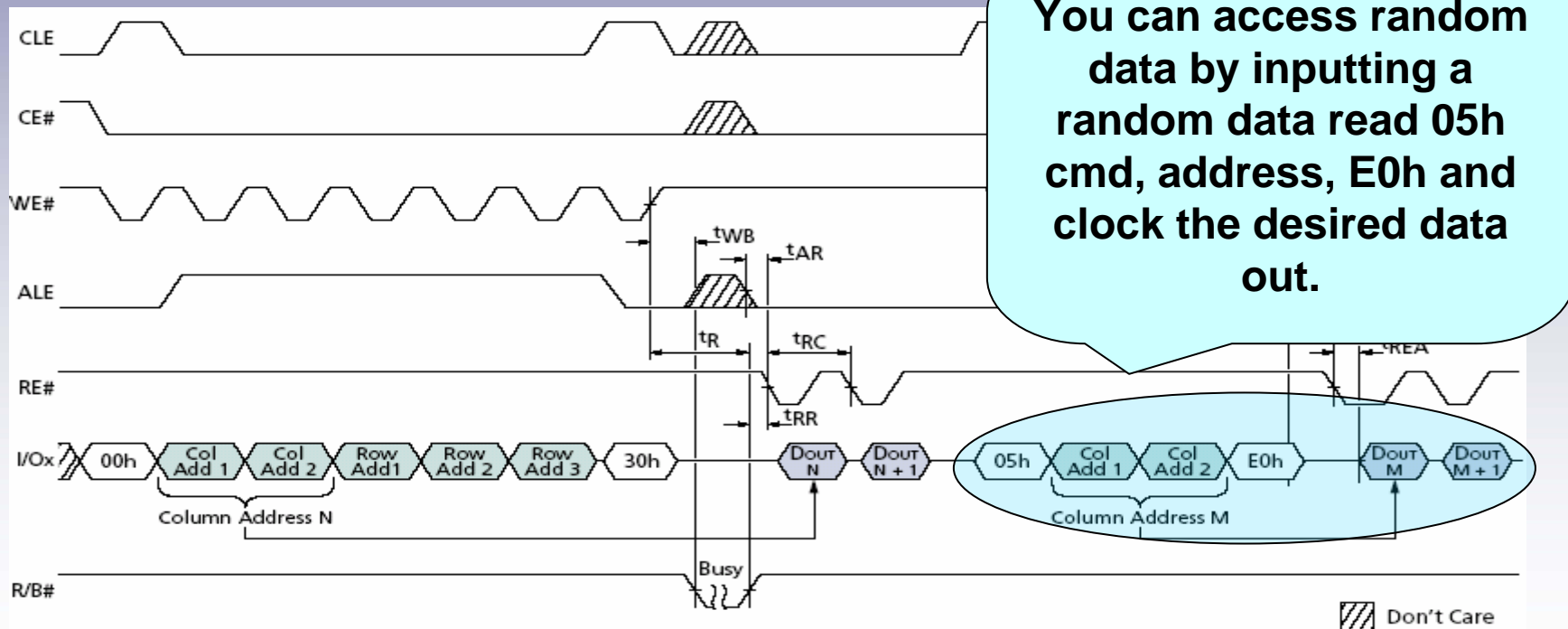
- Read transfers the addressed page from the array to the register
- The column address specifies the first byte out; it can be offset by any amount
- Each clock (RE#) shifts a byte (or word) out

**Read access ( $t_R$ ):**  
~25 $\mu$ s/page



# Random Read Operation

- The RANDOM READ command allows you to specify a new two-byte column address
- Can use the RANDOM READ command to jump around anywhere on the page

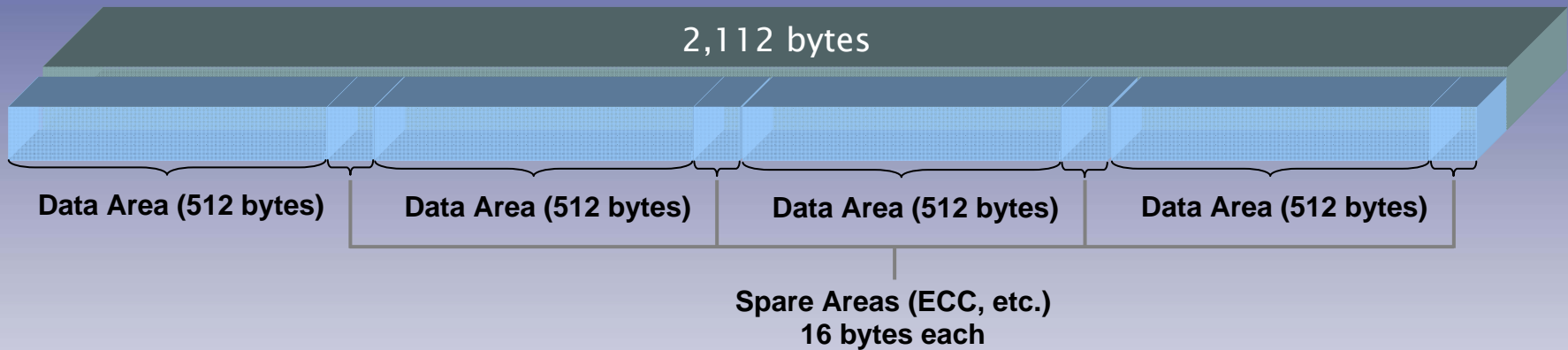


# Partial-Page Programming

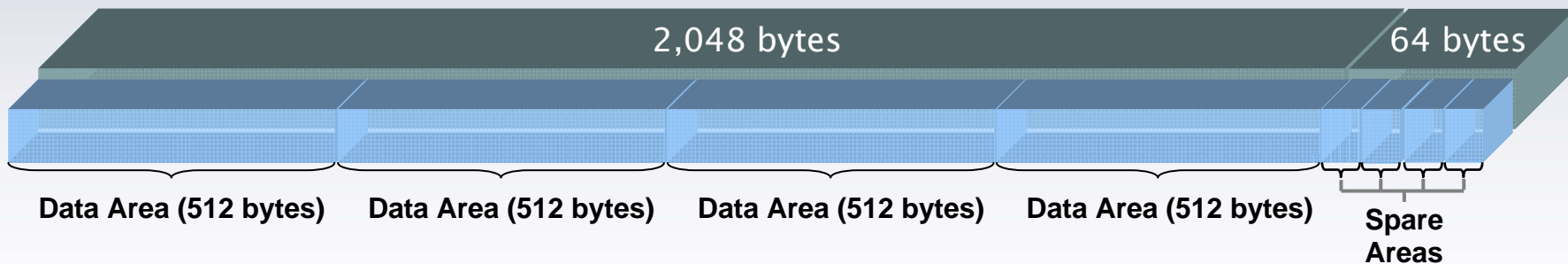
- NOP specifies the number of programming operations that can be executed on the same page
- Pages are programmed in groups due to the large page sizes (SLC only)
  - Typical PC sector size is 512, so four PC sectors fit into one 2K page
  - Programming ECC info separately from the data could require an additional four operations
  - The user can have other info (logical mapping or wear-leveling) in the spare area
- It is best to minimize partial-page programming
  - The number of partial-page program operations is the number of complete programming operations (with confirm 10h) to the same location without an erase
- MLC devices have an NOP of 1

# Methods for Data and Spare Information Placement

## Data and spare information adjacent



## Data and spare information separate





# Connecting NAND to a RISC Processor or DSP That Does not Include a NAND Controller

# Direct Connection to RISC Processor

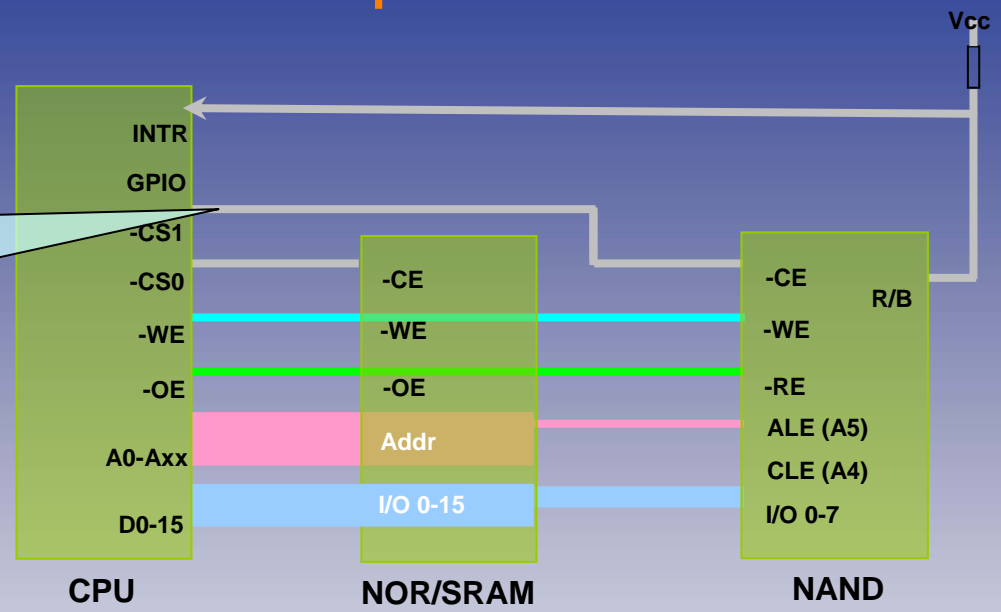
## Memory Mapped NAND Interface

- If microprocessor address 4 is connected to CLE and address 5 is connected to ALE, the NAND can be accessed by a software that uses only three address locations
  - Command register can be accessed by writing to address XX010h
  - Address register can be accessed by writing to address XX020h
  - Data register can be accessed by writing/reading to address XX000h

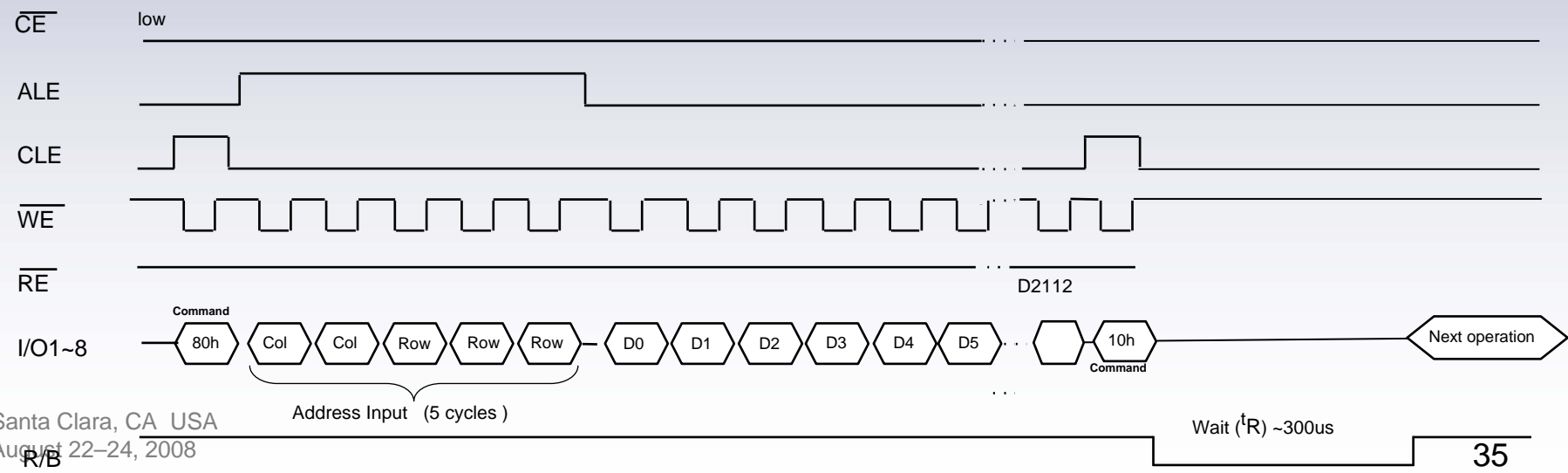
		A7	A6	A5	A4	A3	A2	A1	A0
A5	A4								
ALE	CLE	Memory Address Offset			NAND Register Selected				
0	0	0			Data register				
0	1	1			Command register				
1	0	2			Address register				
1	1	3			Undefined (don't use)				

# Glueless Microprocessor NAND Interface

Assume CS1 address space is 0xFFFF000-0xFFFF0FF



## Program Function





# Glueless Microprocessor NAND Interface

```

Pseudo-Code
SUMMIT Example for PROGRAM:
(All numbers in HEX)
80 -> FFF010 ; CMD = 80
ColL -> FFF020 ; low column
ColH -> FFF020 ; high column
RowL -> FFF020 ; low ROW
RowM -> FFF020 ; Mid ROW
RowH -> FFF020 ; High ROW
D0 -> FFF000 ; Data 0
D1 -> FFF000 ; Data 1

```

```

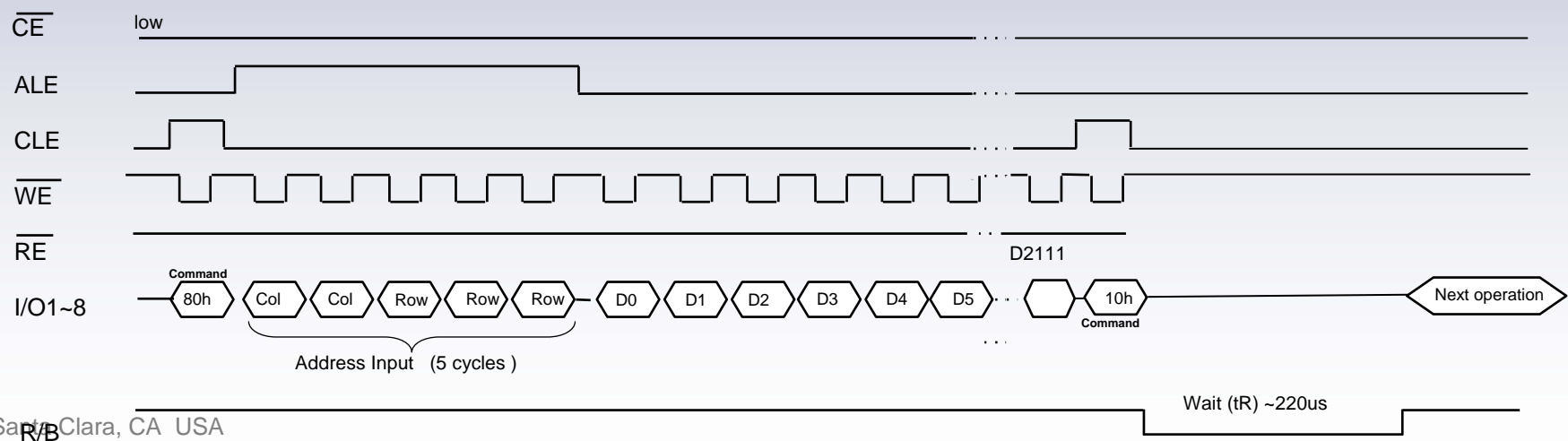
(Complete remaining data)
D2111 -> FFF000 ; Data 2111
10 -> FFF010 ; CMD = 10

LOOP1:
PA -> Acc ; Read status
BIT #6 set ;
JMP NZ LOOP1 ; Jmp if Busy to Loop

; DONE !

```

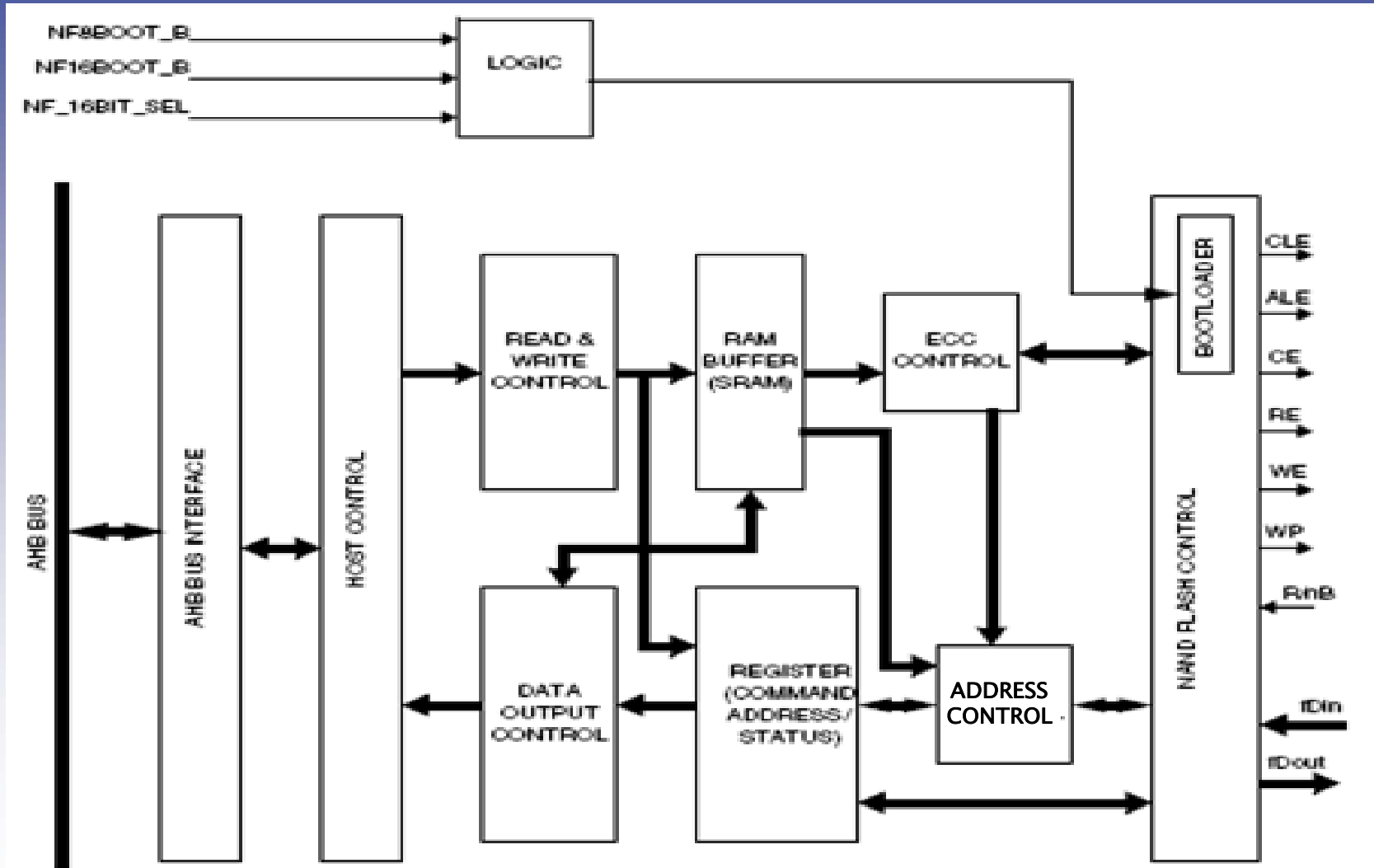
## Program Function



# Processor Support

- Processors with native NAND controller built-in with support for 2K page:
  - Motorola i.MX21 and i.MX31 and others
  - TI Omap 2420 and 2430 and others
  - Other vendors are adding direct-NAND interface; check with your vendor

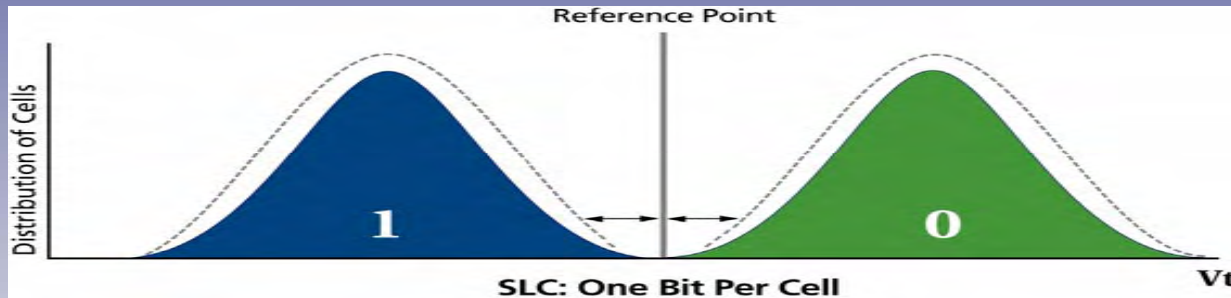
# Native NAND Interface on Freescale i.MX21



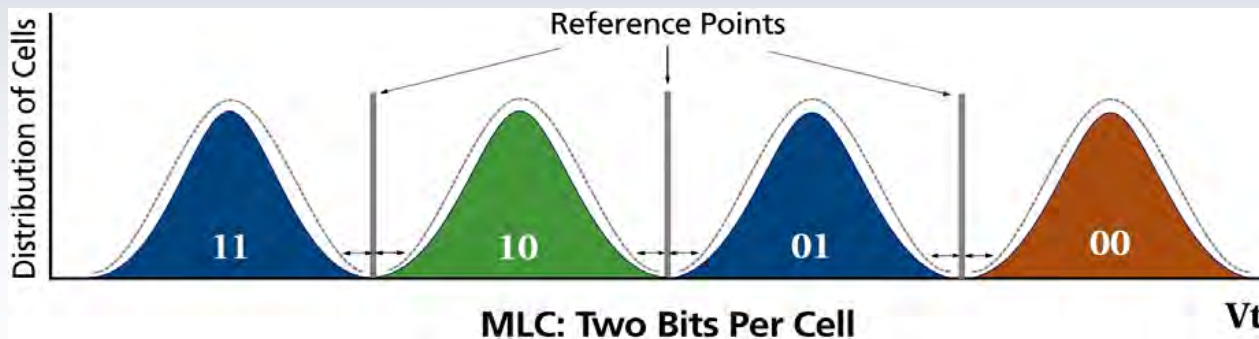
# Single-Level Cell (SLC) vs. Multi-Level Cell (MLC)

# What is the Difference?

- SLC (single-level cell)
  - SLC stores 2 states per memory cell and allows 1 bit programmed/read per memory cell



- MLC (multi-level cell)
  - MLC NAND stores 4 states per memory cell and allows 2 bits programmed/read per memory cell





## SLC vs. MLC

- SLC NAND Flash products offer higher performance and reliability; typical applications include:
  - High performance media cards
  - Solid state drives (SSDs)
  - Many embedded (NAND built inside) designs including:
    - Cell phones (for executing code); MLC will still be considered for high density storage
- Multi-level cell (MLC) NAND Flash will lead in the lowest cost for consumer applications where performance and reliability are not as important; typical applications include:
  - Media players (audio and video)
  - Cell phones (SLC will still be considered for code execution)
  - Consumer media cards (such as USB, SD/MMC, and CF cards)

# SLC Attributes

- Key attributes:
  - Single bit per cell
  - Supports low voltage (1.8V); required for many mobile applications
  - Offered in wide data bus (16 bits) as well as 8-bit
  - Supported by all controllers because SLC generally requires only 1-bit ECC
  - Higher performance
  - Higher reliability

Features	
Bits per cell	1
Voltage	3.3V, 1.8V
Data width (bits)	x8, x16
Architecture	
Number of planes	1 or 2
Page size	2K or 4K bytes
Pages per block	64
Reliability	
NOP (partial-page programming)	4
ECC (per 512 bytes)	1
Endurance (ERASE/PROGRAM cycles)	~100K
Array Operations	
t <sub>R</sub> (Max)	25μs
t <sub>PROG</sub> (Typ)	200–300μs
t <sub>BERS</sub> (Typ)	1.5–2ms

# MLC Attributes

- Key attributes:
  - Two bits per cell; twice the density of similar SLC device
  - Offered only in 3.3V
  - Offered only in x8 data bus
  - Supported only by controllers that include 4-bit (or more ) ECC
  - Compared to SLC NAND:
    - Lower performance
    - Lower reliability
    - Lower price

Features	
Bits per cell	2
Voltage	3.3V
Data width (bits)	x8
Architecture	
Number of planes	2
Page size	2K or 4K bytes
Pages per block	128
Reliability	
NOP (partial-page programming)	1
ECC (per 512 bytes)	4+
Endurance (ERASE/PROGRAM cycles)	~10K
Array Operations	
tR (Max)	50µs
tPROG (Typ)	600–900µs
tBERS (Typ)	3ms

# SLC vs. MLC

SLC		MLC
<b>Features</b>		
1	Bits per cell	2
3.3V, 1.8V	Voltage	3.3V
x8, x16	Data width (bits)	x8
<b>Architecture</b>		
1 or 2	Number of planes	2
2KB or 4KB	Page size	2KB or 4KB
64	Pages per block	128
<b>Reliability</b>		
4	NOP (partial-page programming)	1
1	ECC (per 528 bytes)	4+
~100K	Endurance (ERASE/PROGRAM cycles)	~10K
<b>Array Operations</b>		
25μs	t <sub>R</sub> (Max)	50μs
200–300μs	t <sub>PROG</sub> (Typ)	600–900μs
1.5–2ms	t <sub>BERS</sub> (Typ)	3ms

**SLC is typically offered in lower voltage and wider busses**

**SLC requires less ECC**

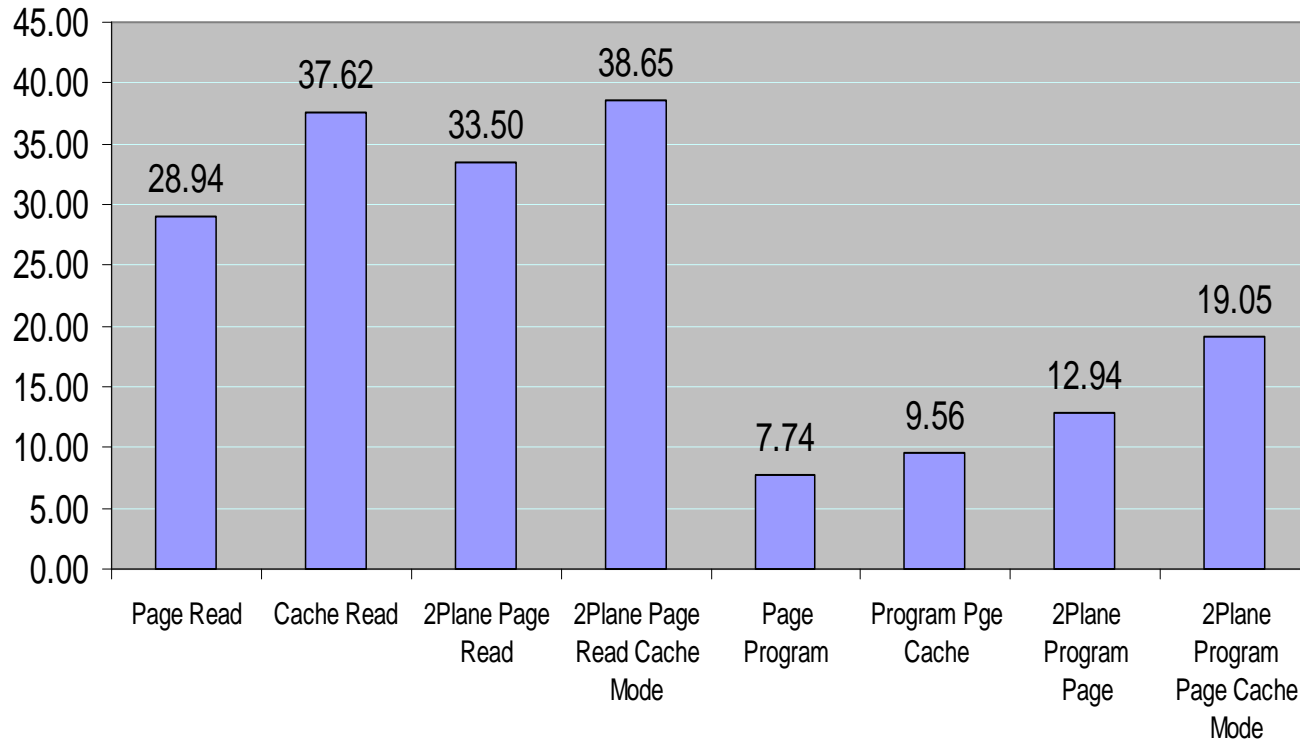
**SLC reliability is 10 times better!**

**SLC performance is ~3 times better**

**MLC density is 2 times that of similar SLC**

# 4Gb SLC Performance

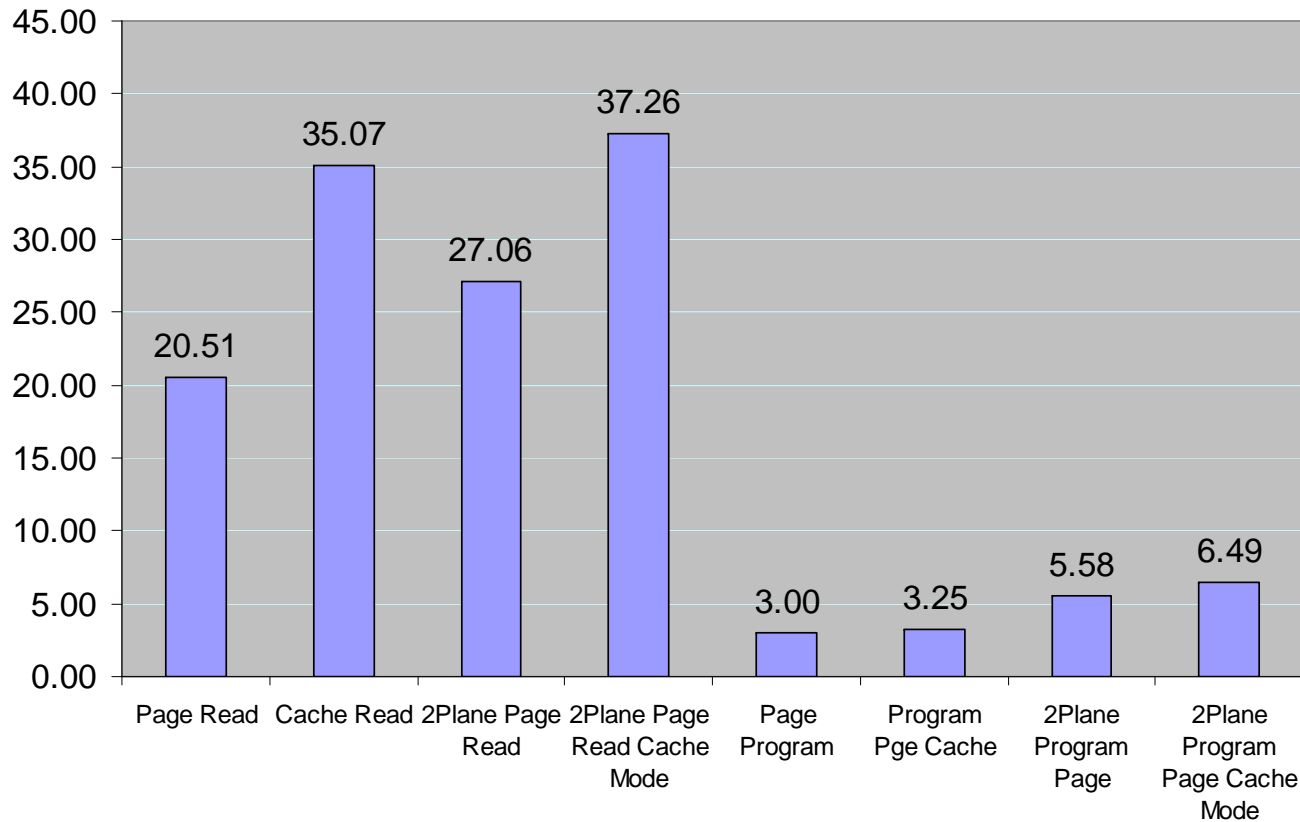
(72nm SLC) 4Gb Performance



Symbol	Time	Units
tR	20	us
tDCBSYR1	3	us
tDCBSYR2	3	us
tRC	25	ns
tRC (C)	25	ns
tRC		ns
tPROG	220	us
tCBSY	3	us
tDBSY	0.5	us
tWC	25	ns
tWC (C)	25	ns
tWC		ns
PS	2112	Byte
NP	64	Pages

# 8Gb MLC Performance

(72nm MLC) 8Gb Performance



Symbol	Time	Units
tR	50	us
tDCBSYR1	7	us
tDCBSYR2	7	us
tRC	25	ns
tRC (C)	25	ns
tRC	25	ns
tPROG	650	us
tCBSY	30	us
tDBSY	0.5	us
tWC	25	ns
tWC (C)	25	ns
tWC	25	ns
PS	2112	Byte
NP	128	Pages

## SLC Requires Less ECC

- While it is possible to implement 1-bit correct (Hamming code) in software, it generally does not provide a high performance solution
- Many microprocessors include NAND controllers that support 1-bit ECC
- Some newer processors are looking to include 4-bit ECC (or more) in their on-chip NAND controllers

# SLC vs. MLC Conclusions

- MLC will always provide the lowest cost per bit
- SLC will always provide the highest performance
- SLC will always provide the highest reliability
- Choose the right NAND device for the application



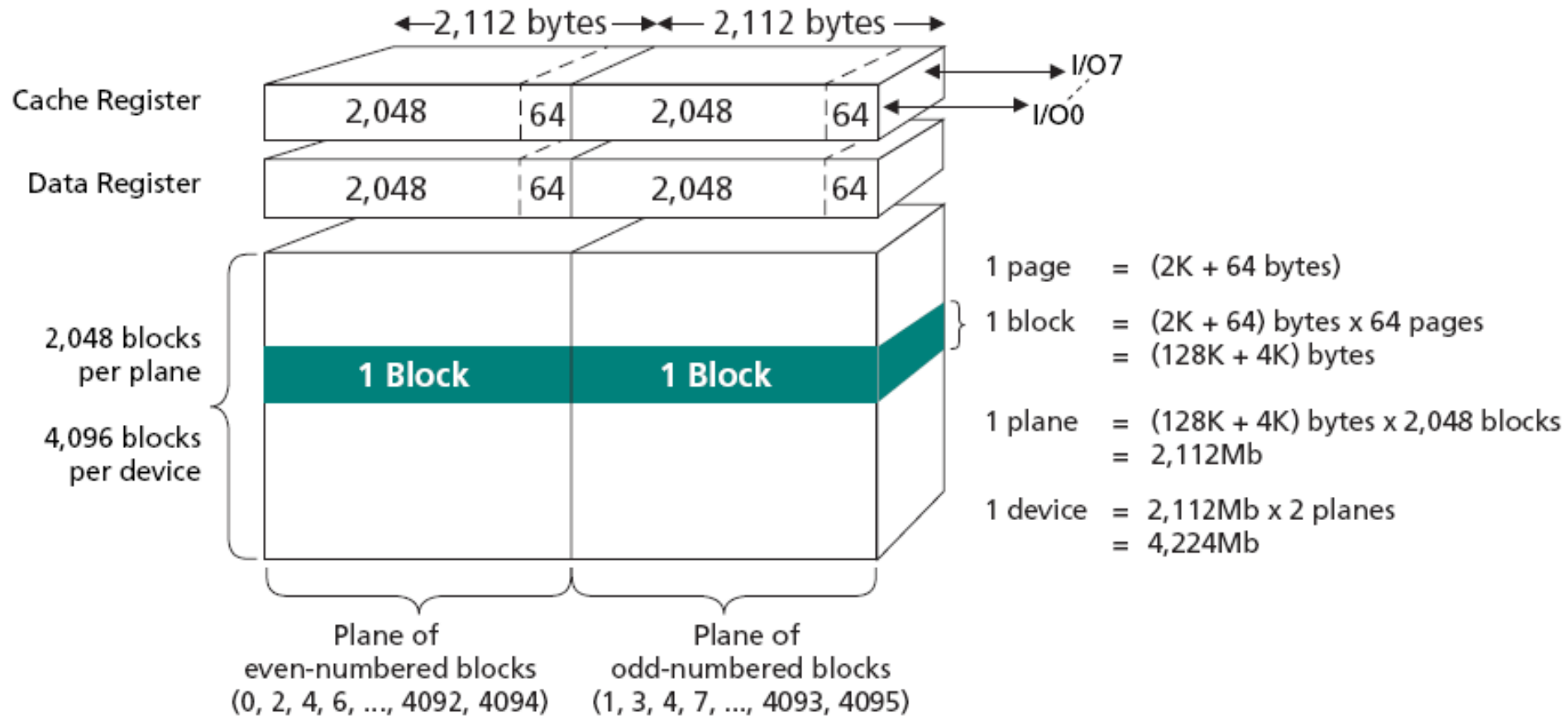
# All NAND Flash Devices Are Not Created Equal

- Differences include:
  - Cell types
  - Architecture
  - Performance
  - Timing parameters
  - Command set
  
- Open NAND Flash Interface (ONFI) drives a standard interface

# Two-Plane Features

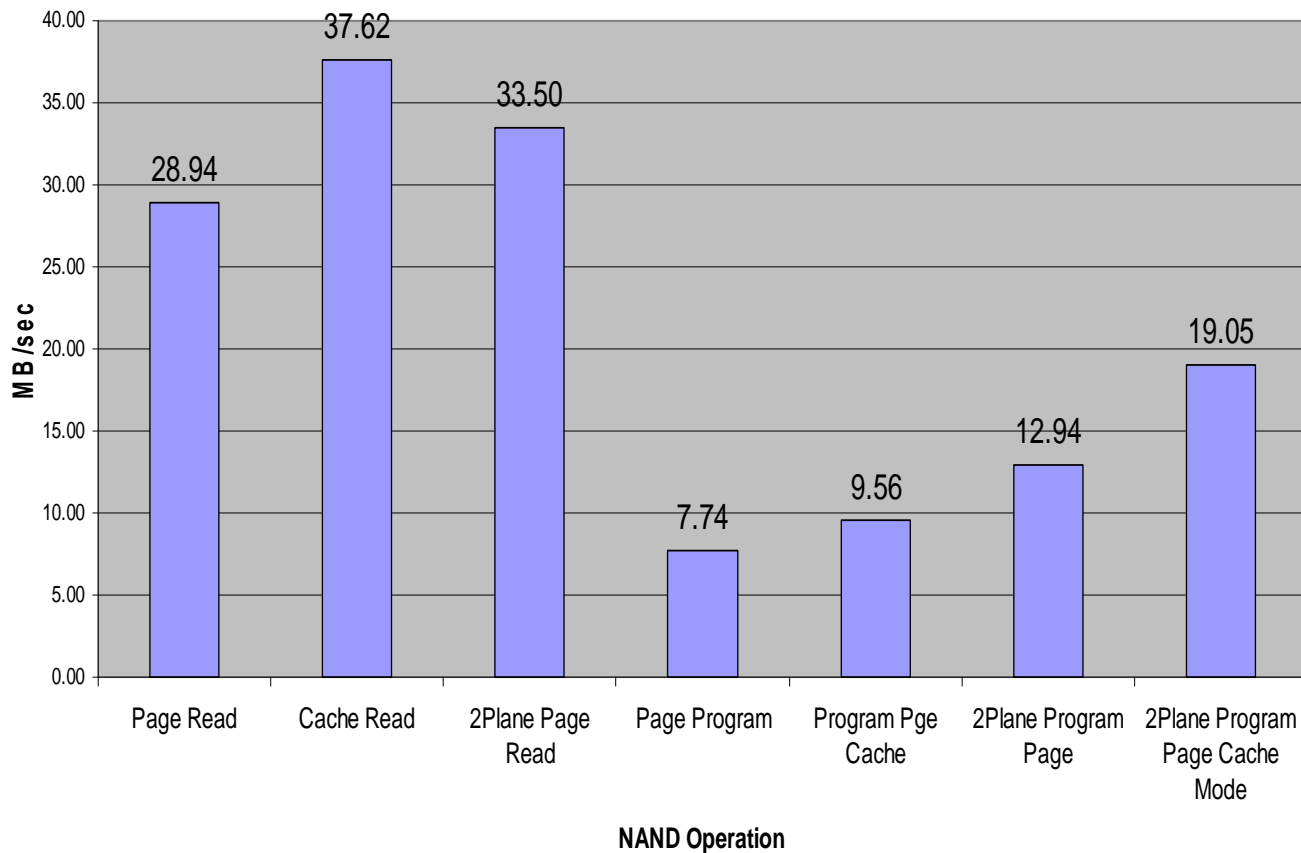
- Device is divided into two physical planes, odd/even blocks
- Users have the ability to:
  - Concurrently access two pages for read
  - Erase two blocks concurrently
  - Program two pages concurrently
- The page addresses of blocks from both planes must be the same during two-plane READ/PROGRAM/ERASE operations

# 4Gb, Two-Plane, 2K-Page SLC NAND Architecture



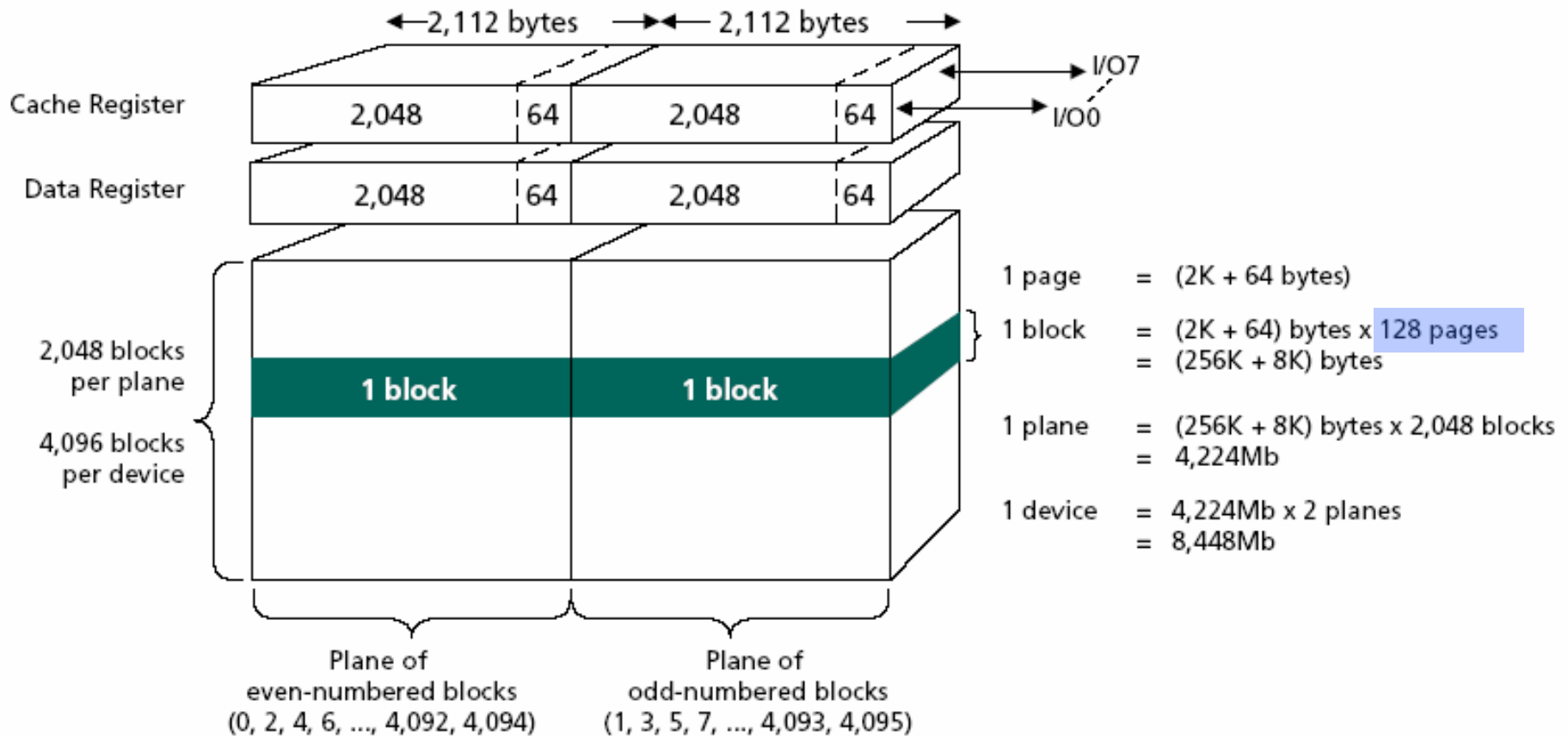
# 4Gb, 2K-Page SLC NAND Performance

Micron (72nm SLC) 4Gb die 2K Page Performance



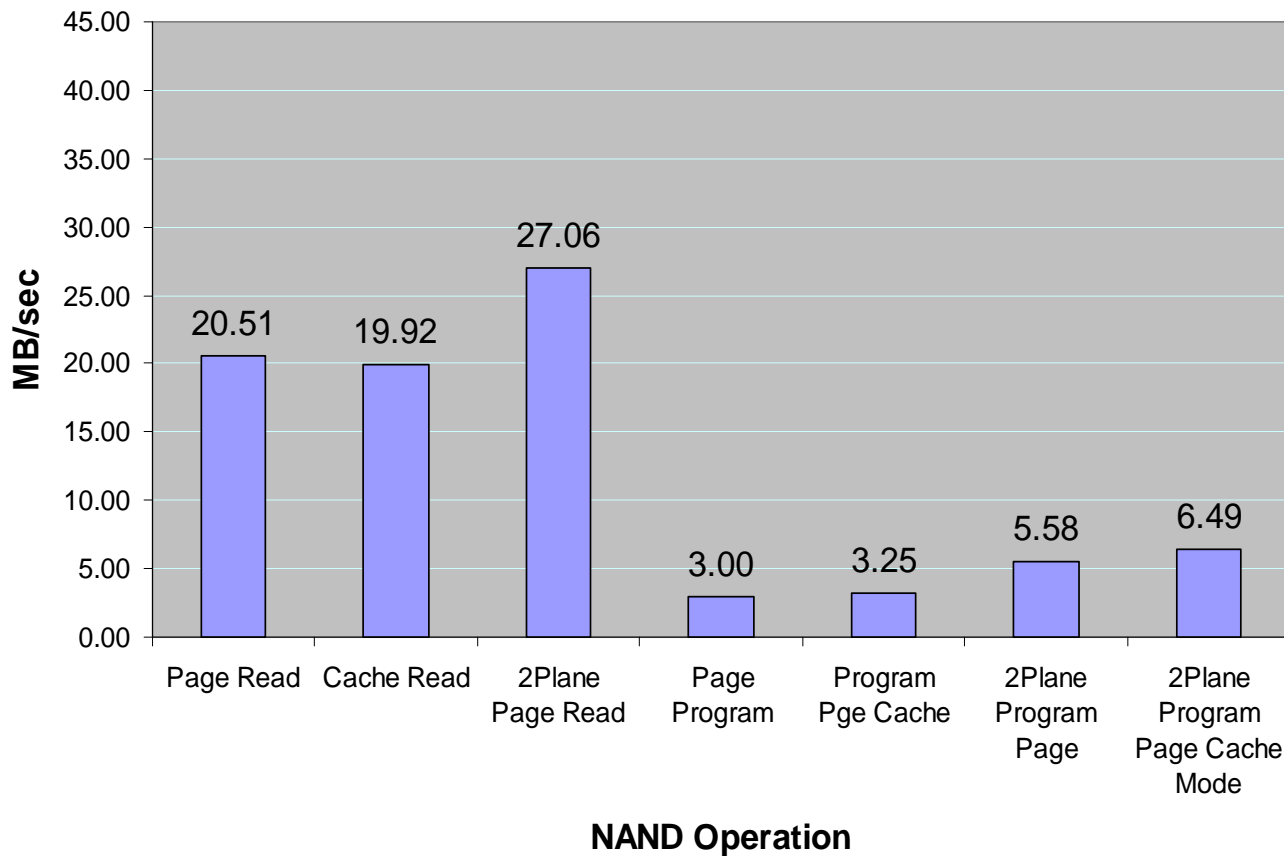
Symbol	Time	Units
tR	20	us
tDCBSYR1	3	us
tDCBSYR2	3	us
tRC	25	ns
tRC (C)	25	ns
tRC		ns
tPROG	220	us
tCBSY	3	us
tDBSY	0.5	us
tWC	25	ns
tWC (C)	25	ns
tWC		ns
PS	2112	Byte
NP	64	Pages

# 8Gb, Two-Plane, 2K-Page MLC NAND Architecture



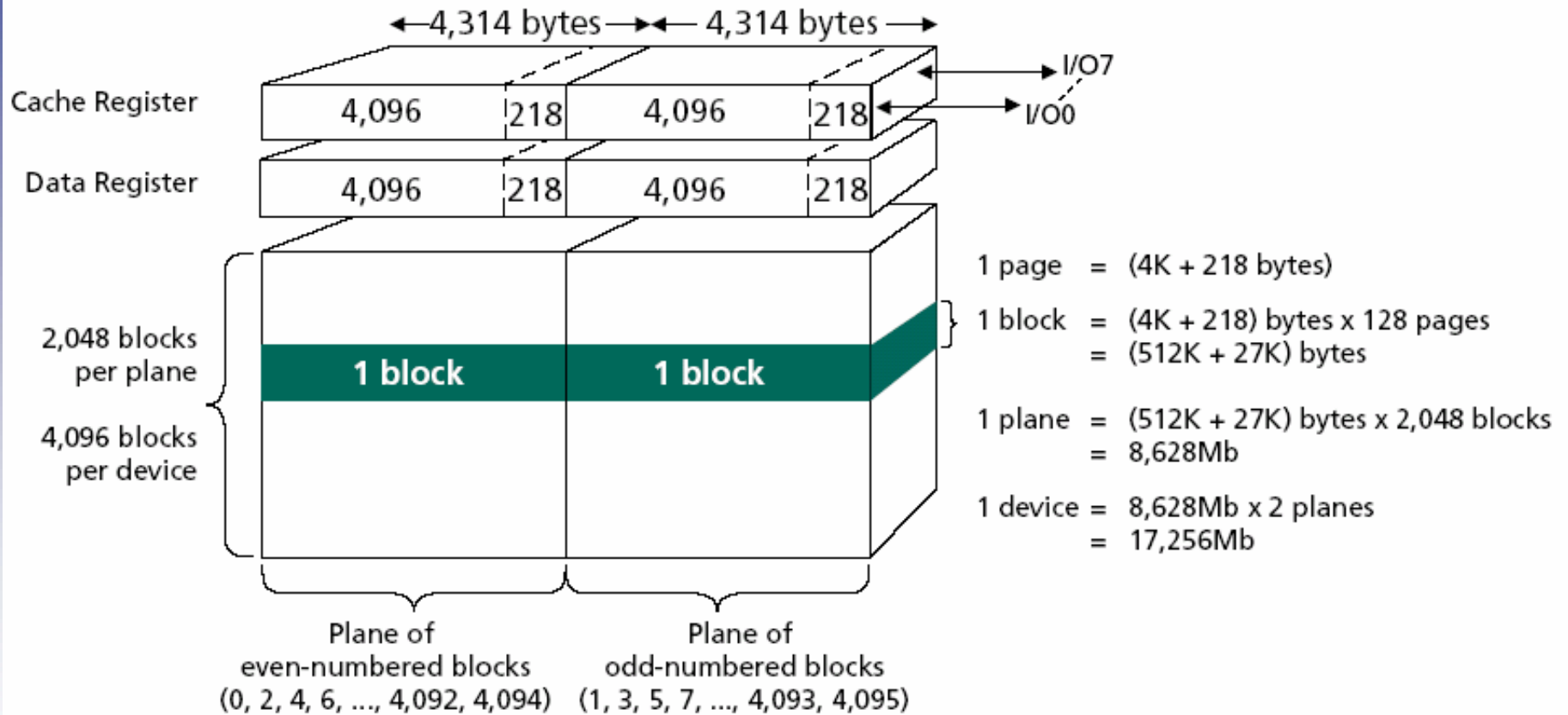
# 8Gb, 2K-Page MLC Performance

Micron (72nm MLC) 8Gb die 2K Page Performance



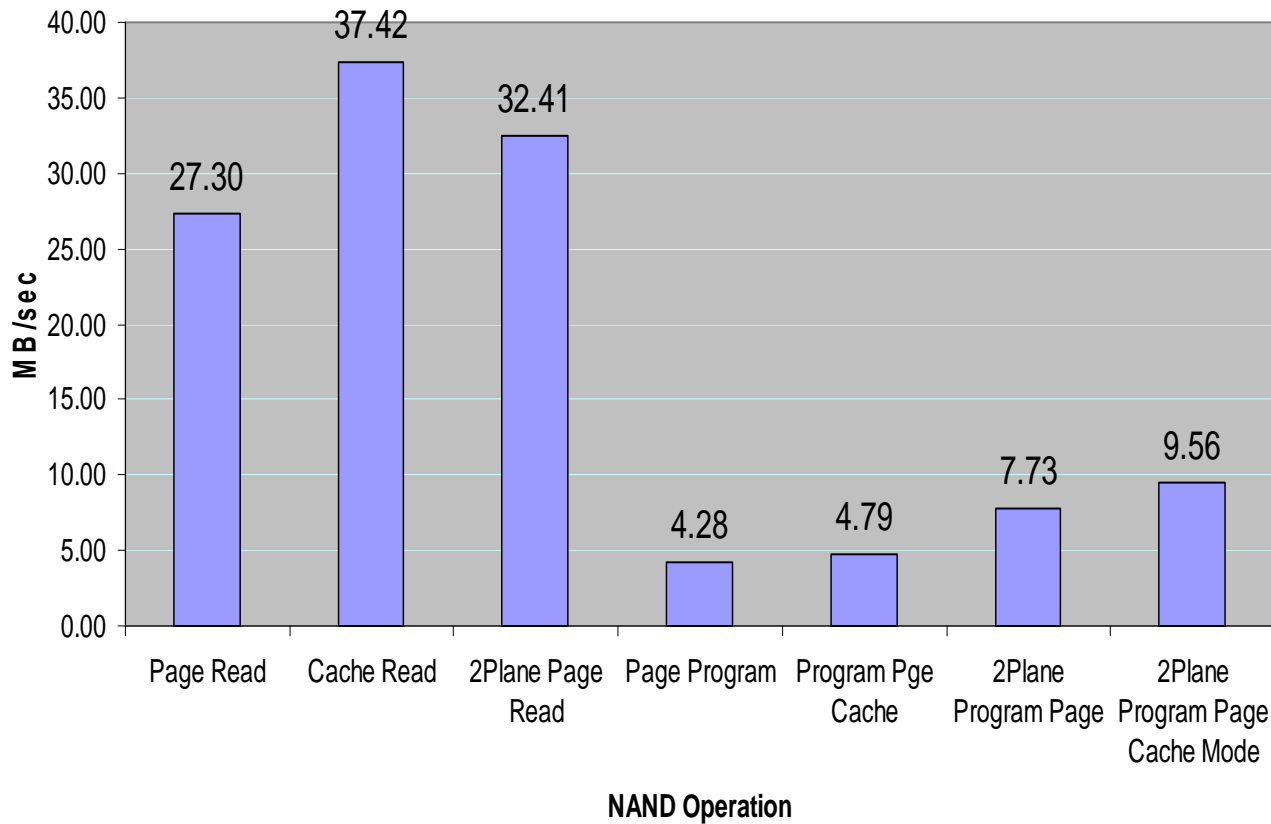
Symbol	Time	Units
tR	50	us
tDCBSYR1	7	us
tDCBSYR2	7	us
tRC	25	ns
tRC (C)	25	ns
tRC	25	ns
tPROG	650	us
tCBSY	30	us
tDBSY	0.5	us
tWC	25	ns
tWC (C)	25	ns
tWC	25	ns
PS	2112	Byte
NP	128	Pages

# 16Gb, Two-Plane, 4K-Page MLC NAND Architecture



# Two-Plane, 4K-Page MLC NAND Architecture

Micron (55nm MLC) 16Gb die 4K Page Performance



Symbol	Time	Units
tR	50	us
tDCBSYR1	7	us
tDCBSYR2	7	us
tRC	25	ns
tRC (C)	25	ns
tRC	50	ns
tPROG	900	us
tCBSY	3	us
tDBSY	0.5	us
tWC	25	ns
tWC (C)	35	ns
tWC	45	ns
PS	4314	Byte
NP	128	Pages



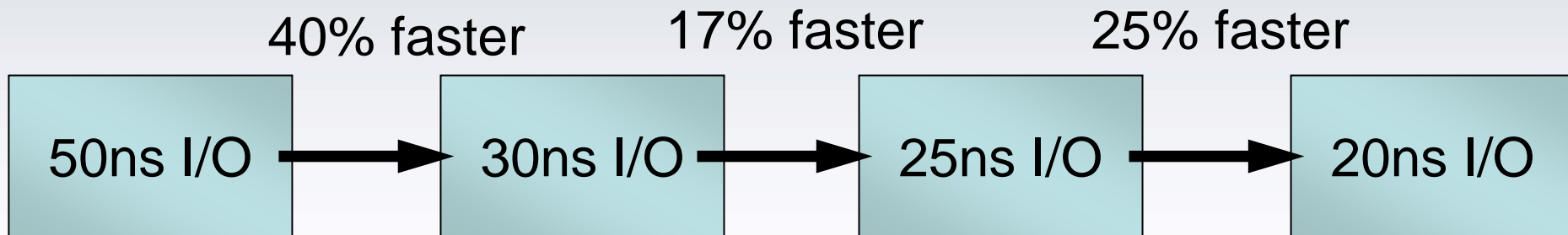
# Performance Bottlenecks

# Read Throughput Limitations in NAND Today

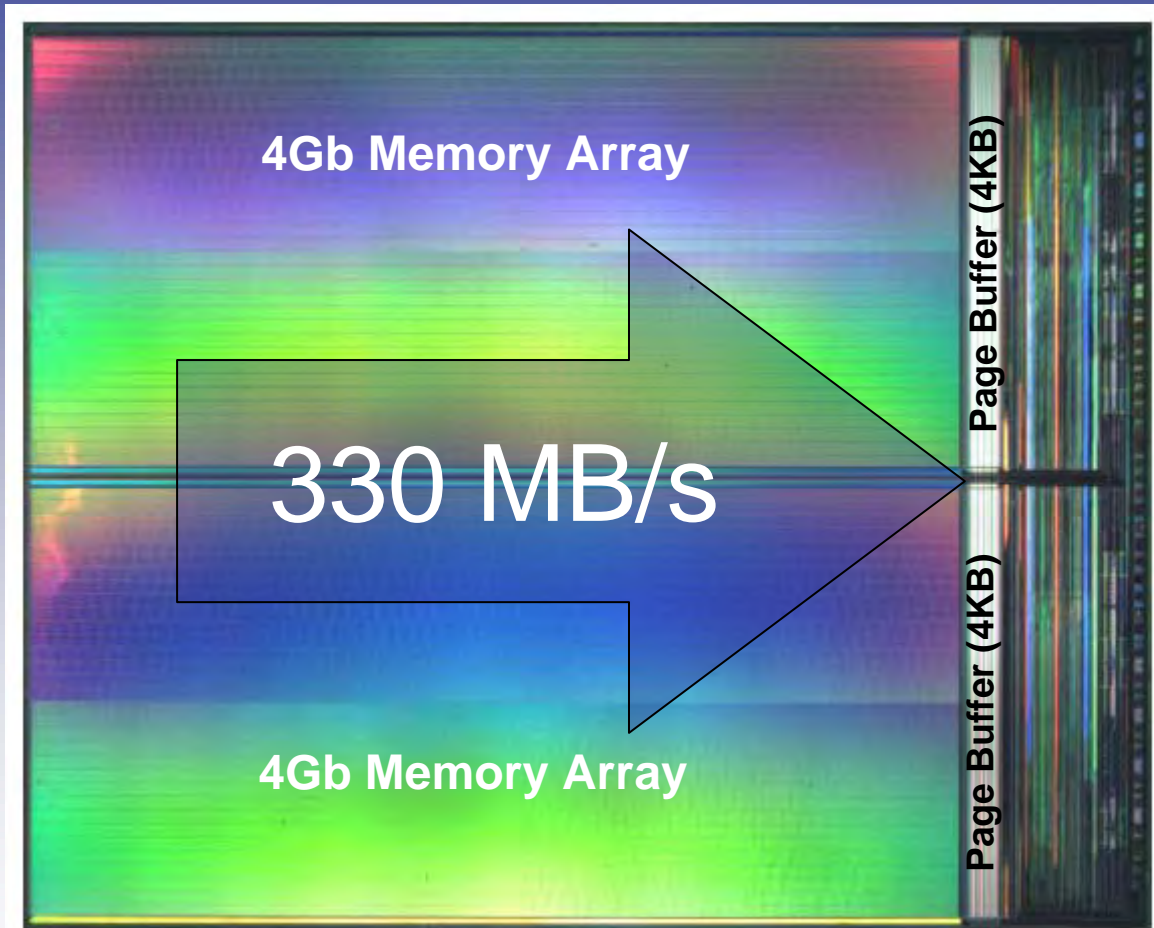
- Read throughput limited by I/O frequency
- I/O time for NAND page ( $t_{RC} = 20\text{ns}$ )
  - 2K page:  $42\mu\text{s}$
  - 4K page:  $86\mu\text{s}$
- NAND array read transfer time
  - SLC:  $t_R$  time is normally 20–25 $\mu\text{s}$  MAX
  - MLC:  $t_R$  time is normally 50 $\mu\text{s}$  MAX
- Today for SLC NAND, the I/O time is 2–4x the array transfer time
- I/O performance must be less than or equal to array performance for maximum sustained read throughput

# I/O Throughput Cannot Scale

- NAND Flash interface is asynchronous
- NAND timing parameters cannot scale indefinitely to faster speeds
- As tRC decreases, it becomes difficult for controllers to latch data output from the NAND
- As tWC for data input decreases, time to process command and address cycles does not decrease



# SLC Read Array/Bus Performance Mismatch



**NAND array reads are parallel and very fast**

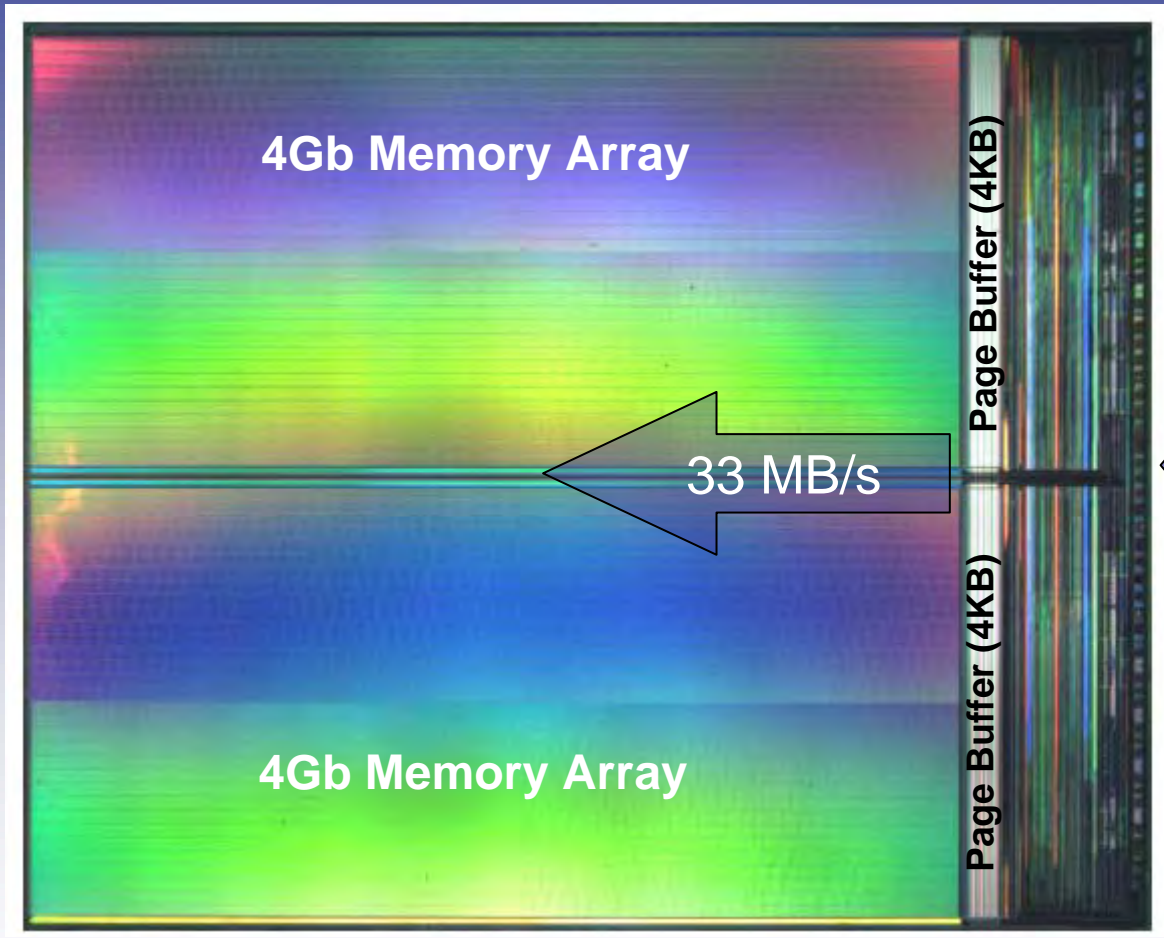
- Read array bandwidth is greater than 330 MB/s (8KB read in 25 $\mu$ s)

40 MB/s

**Interface speed is the limiting factor**

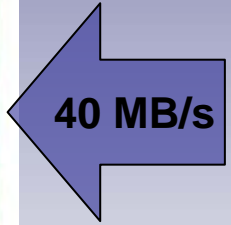
- Read bus bandwidth is only 40 MB/s (25 $\mu$ s clock)

# SLC Program Array/Bus Performance



Interface speed is no longer the limiting factor

- Bus bandwidth is 40 MB/s (25 $\mu$ s)



Program performance is not so impressive

- Array program bandwidth is 33 MB/s (8KB programmed in 250 $\mu$ s)

# ONFI and High-Speed NAND Introduction

# What is ONFI

- ONFI = Open NAND Flash Interface
- Includes NAND vendors, enablers, and customers
- Purpose is to standardize the NAND Flash interface
  - Packages
  - Timing parameters
  - Addressing
  - Command set
  - Device behavior
- Benefits
  - NAND devices self-describe their capabilities to controllers
  - Reduces time to qualify NAND devices at enablers and OEMs



# NAND Flash Inconsistencies Without a Standard

- Device identification using read ID
- Array architecture and addressing
- Command set
- Timing parameters
- ECC and endurance
- Factory-marked bad blocks
- Device behavior and status



# ONFI Technical Philosophy

- ONFI shall ensure no preassociation with NAND Flash at host design is required
  - Flash must self-describe features, capabilities, timings, etc., through a parameter page
  - Features that cannot be self-described in a parameter page (like number of CE#) shall be host discoverable
- ONFI should leverage existing Flash behavior to the extent possible
  - Intent is to enable orderly and TTM transition, so highly divergent behavior from existing NAND undesired
  - Where prudent for longevity or capability need, existing Flash behavior shall be modified or expanded
- ONFI needs to enable future innovation

# The NAND Interface Today

- ONFI 1.0 has standardized today's NAND interface
  - Consistent and easier for controller designers to identify and use NAND features
- ONFI 1.0 introduced timing mode 5 for faster I/O throughput
  - New standard for NAND interface performance
  - $t_{RC} / t_{WC} = 20\text{ns}$

# Goals of a High-Speed NAND Interface (ONFI 2.0)

- Keep transition to high-speed interface simple
  - Keep and/or redefine original NAND signals to provide high-speed signaling without disrupting the NAND protocol and command set
  - Provide backward compatibility to asynchronous NAND interface to make device identification simple
- Increase I/O throughput with room to grow
  - Remove tRC latching limitation by adding a bidirectional source-synchronous strobe (DQS)
  - Remove tWC command and address cycle limitation by decoupling command and address processing from the data input rate
- Ensure a graceful transition from standard NAND to high-speed NAND

# Scalable I/O Performance

- A scalable interface is needed for more than read I/O throughput
- As more NAND devices are added to the bus, it is possible for even slower MLC devices to max the I/O bus bandwidth

- A fast NAND Flash interface is possible by:
  - Adding bidirectional source-synchronous DQS
  - Providing scalable DDR data I/O interface
  - Optimizing the signaling to allow enough time to process command and address cycles
  - Minimizing NAND pin capacitance





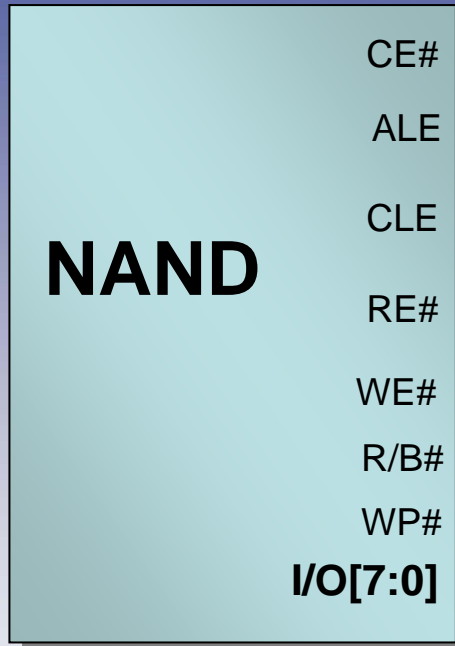
# Key Feature Comparison

Feature	Standard NAND	HS-NAND
“Standard” asynchronous interface	Yes	Yes
Synchronous interface	No	Yes
NAND command protocol	Standard	Standard
tRC	≥ 25ns (SDR)	6ns (DDR)
tWC	≥ 25ns (SDR)	6ns (DDR)
Standardized	ONFI 1.0	ONFI 2.0
Scalable to higher performance	No	Yes
Error correction requirements	2	8
Page size	2KB + 64B	4KB +224B
Block size	64 pages	128 pages
Cache mode	Some	Yes
Vil/Vih and Vol/Voh	CMOS	CMOS
VCCq	3.3V	1.7V to 1.95V
VCC	3.3V	2.7V to 3.6V
Parameter page	Some	Yes
Package	TSOP	BGA

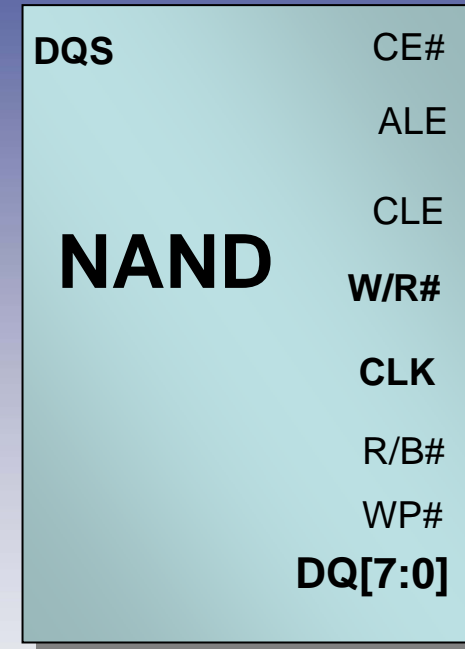
**A natural extension to standard NAND**

# Backward-Compatible ONFI 2.0 Interface

## Asynchronous NAND Interface



## Synchronous NAND Interface

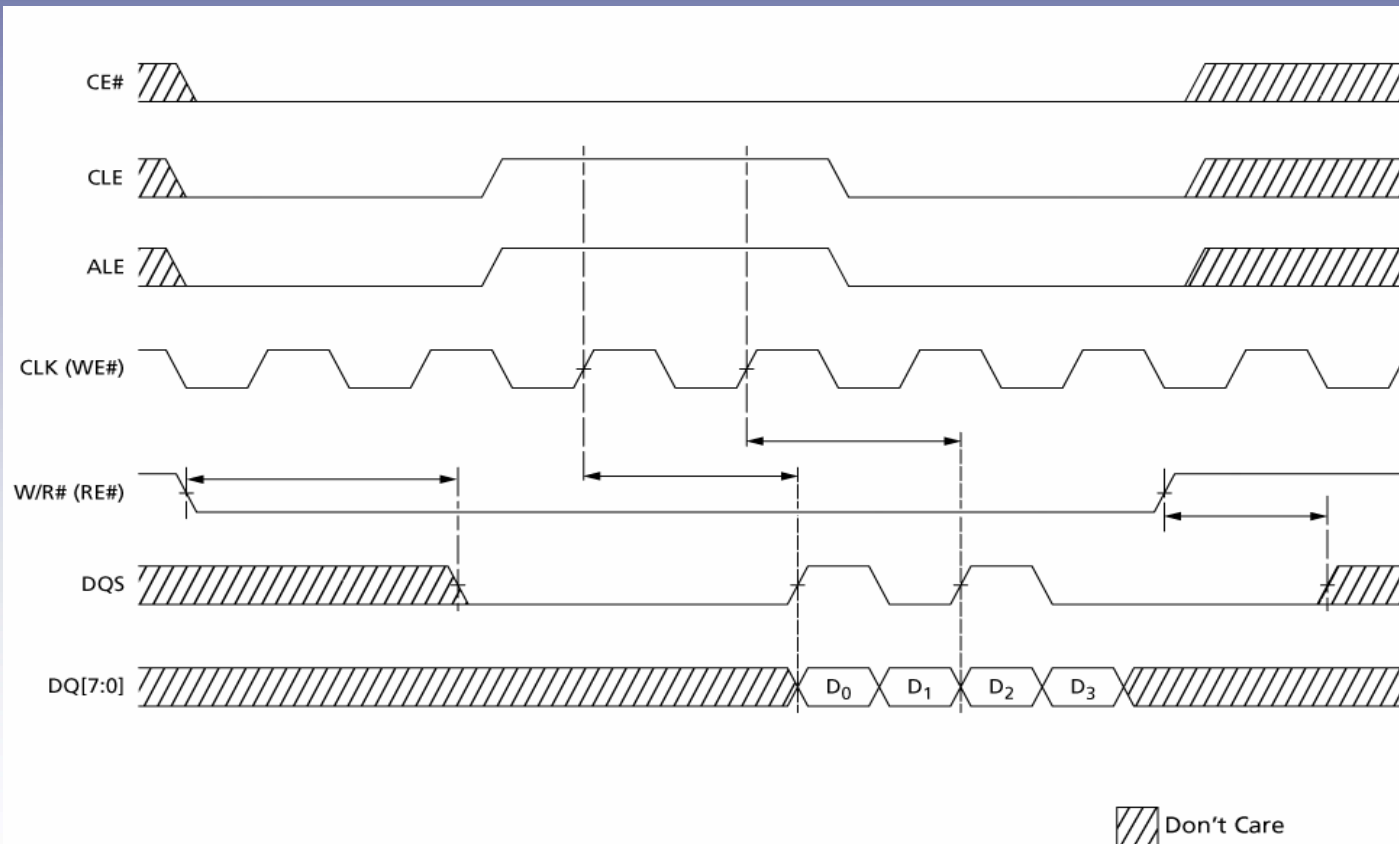


- High-speed-capable NAND Flash devices power on using the asynchronous interface for backward compatibility
- Set features enable source-synchronous interface
- WE# becomes a fast CLK
- RE# handles data direction by becoming W/R# (Write/Read#)
- I/O[7:0] renamed to DQ[7:0] (name change only, functionally identical)
- DQS, a new bidirectional signal, is enabled

# High-Speed NAND Signal Description

Signals		Description
Async	Sync	
WE#	CLK	<ul style="list-style-type: none"> <li>Free-running and used to latch command and address cycles</li> <li>During idle CLK, may be stopped to save power</li> </ul>
RE#	W/R#	<ul style="list-style-type: none"> <li>Controls direction of DQ bus and DQS               <ul style="list-style-type: none"> <li>W/R# = "1": Data input</li> <li>W/R# = "0": Data output</li> </ul> </li> </ul>
—	DQS	<ul style="list-style-type: none"> <li>During data phase, each DQS rising and falling edge corresponds to a data byte               <ul style="list-style-type: none"> <li>DQS is center-aligned for data input</li> <li>DQS is edge-aligned for data output</li> </ul> </li> </ul>
ALE/CLE	ALE/CLE	<ul style="list-style-type: none"> <li>For synchronous mode:               <ul style="list-style-type: none"> <li>ALE / CLE = "11": Data transfer</li> <li>ALE / CLE = "00": Bus idle</li> </ul> </li> </ul>

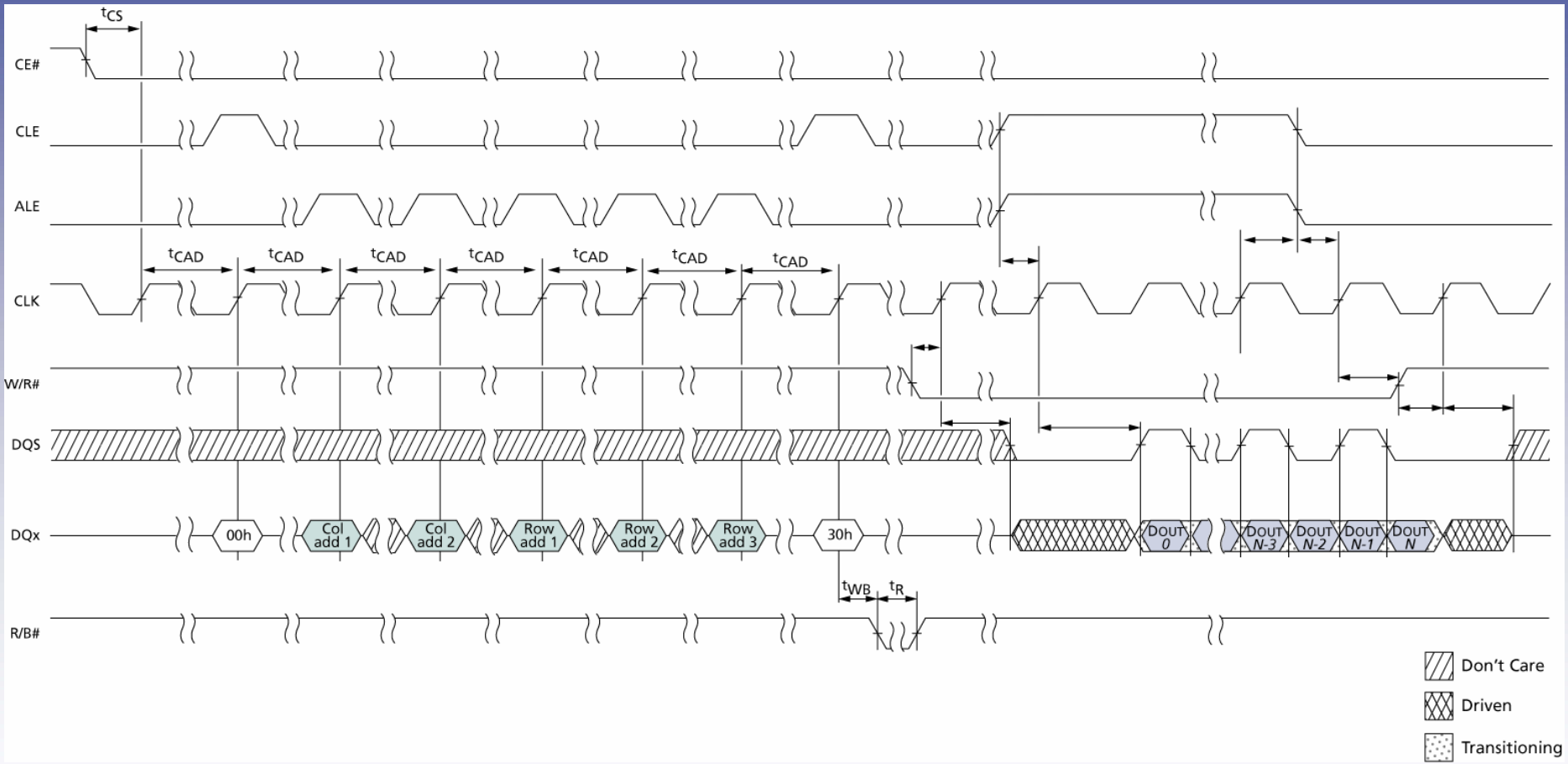
# Source-Synchronous Data Output Signaling Example







# Source-Synchronous Page Read Example



# Low-Power Signaling

- As process geometry shrinks, it becomes more difficult for controllers to stay with 3.3V I/O
  - Many applications today use 1.8V signaling
  - Many high-speed interfaces today use smaller voltage swings so signals can transition faster
    - Example: Full-speed USB – 12 Mbit/s at 3.3V, High-speed USB – 480 Mbit/s at 400mV
- NAND Flash today requires the array and I/O to operate at the same voltage
  - $V_{cc} = 2.7\text{--}3.6\text{V}$
  - $V_{cc} = 1.7\text{--}1.95\text{V}$
- NAND Flash array operations perform best when  $V_{cc} > 1.8\text{V}$ , providing faster program, read, and erase times
- By splitting the array voltage ( $V_{cc}$ ) from the I/O voltage ( $V_{ccQ}$ ), it is possible to get fast array operations *and* faster, lower-power I/O signaling
- Potential high-speed voltage configurations
  - $V_{cc} = 2.7\text{--}3.6\text{V}$ ,  $V_{ccQ} = 2.7\text{--}3.6\text{V}$
  - $V_{cc} = 2.7\text{--}3.6\text{V}$ ,  $V_{ccQ} = 1.7\text{--}1.95\text{V}$

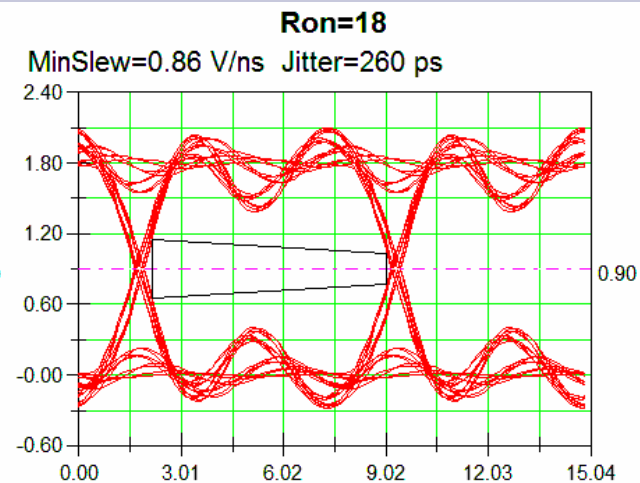
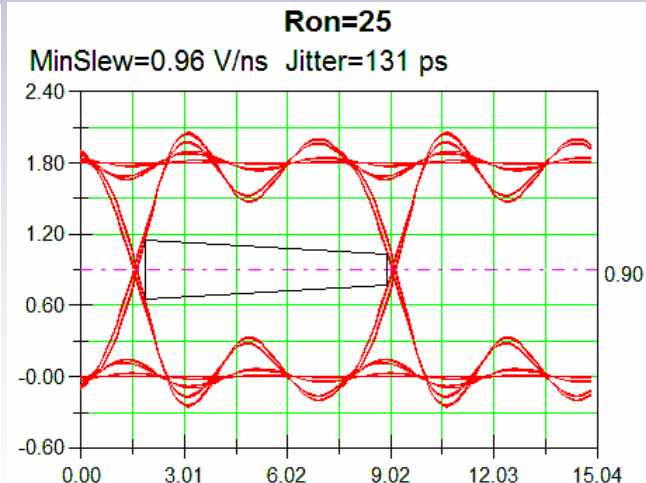
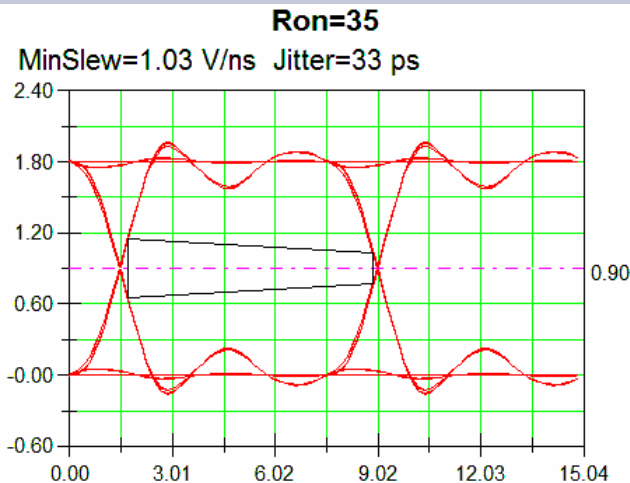
# High-Density Scalability

- By providing multiple output drive strength settings, many NAND devices can share the I/O bus while maintaining I/O throughput
- Example: 133 MT/s data throughput

35 $\Omega$  driver, 4 NAND die

25 $\Omega$  driver, 8 NAND die

18 $\Omega$  driver, 16 NAND die

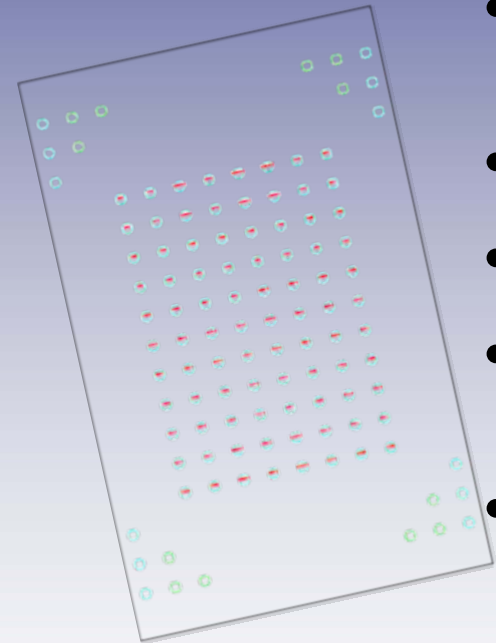


# High-Speed NAND Packaging

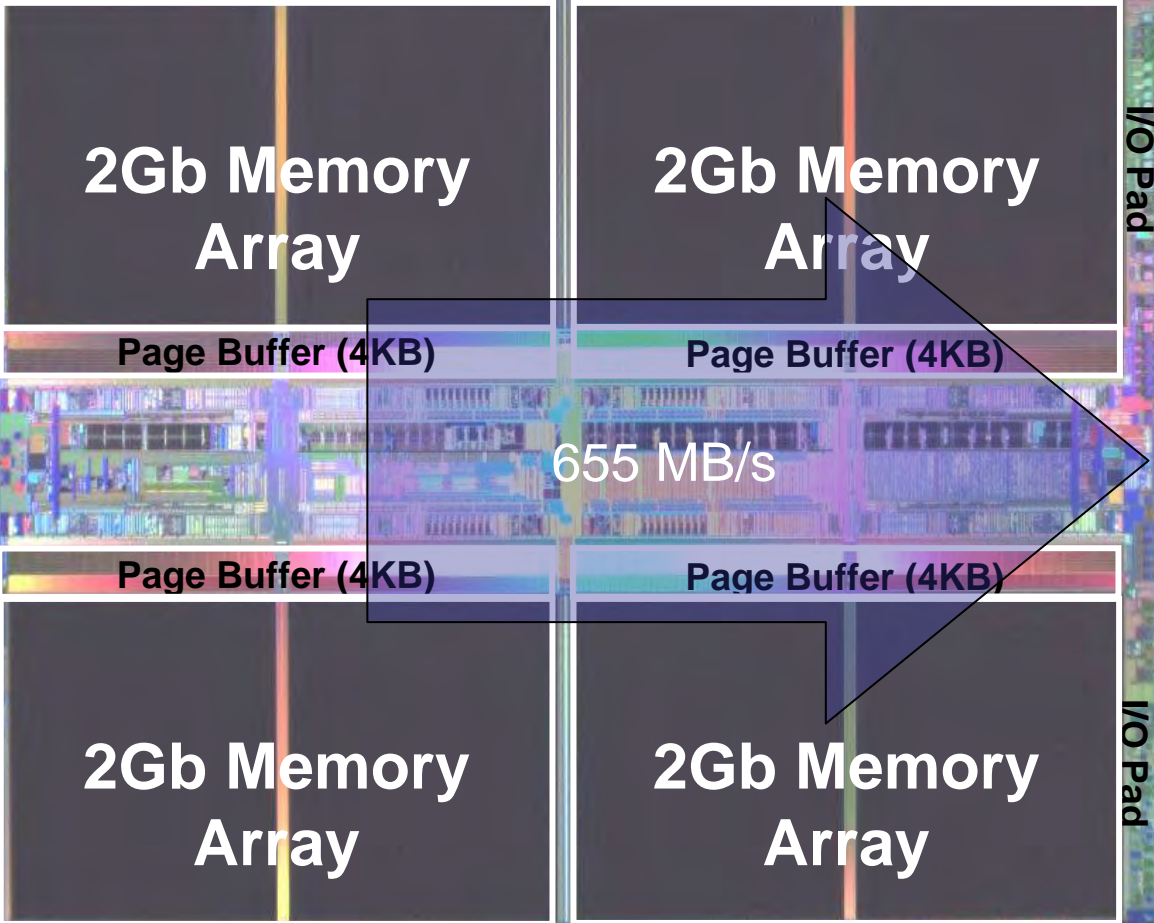
- High-speed-capable packages receive
  - DQS signal
  - Some Vcc changes to VccQ
  - Some Vss changes to VssQ
- The following packages will be transitioned to high-speed NAND
  - 48-pin TSOP
  - 63-ball BGA

# Introducing a New BGA Package

- ONFI 2.0 will introduce a new BGA package
  - Accommodates high-speed *and* asynchronous-only NAND Flash devices
  - Dual x8 interface
  - More power/ground balls for lower noise
  - Signals arranged for excellent signal integrity
  - 1mm ball spacing for low cost PCB assembly
  - Accommodates ever-increasing NAND densities with two package outline options

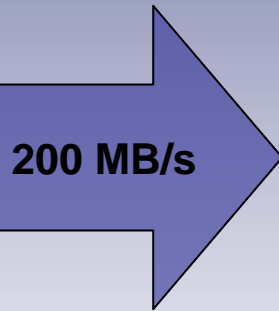


# High-Speed NAND Read Array/Bus Performance



Array program bandwidth is:

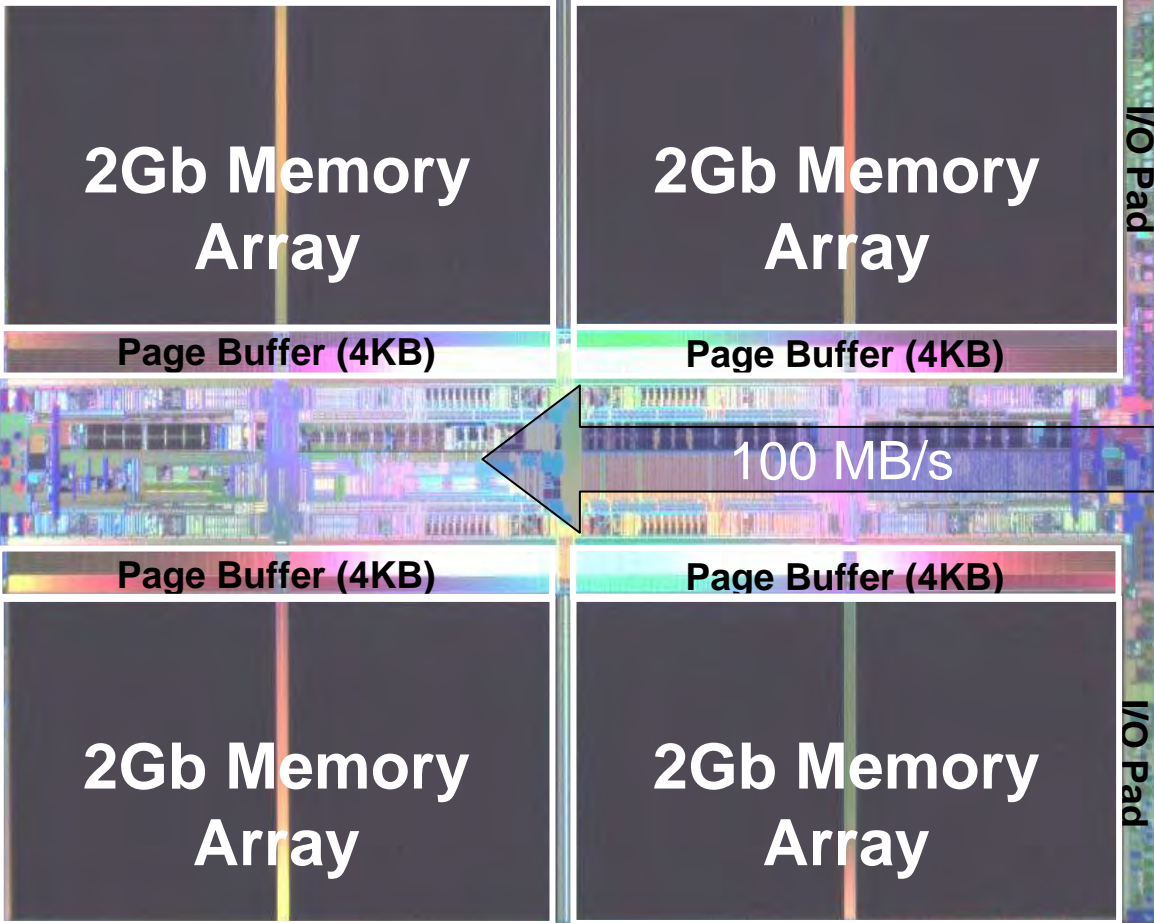
- 655 MB/s (16KB read in 25 $\mu$ s)
- or
- 163 MB/s (4KB read in 25 $\mu$ s)



Interface speed is well matched

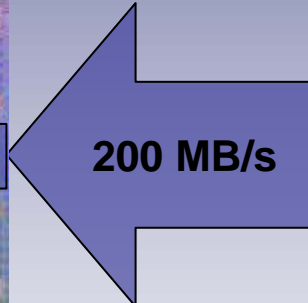
- Bus bandwidth is 200 MB/S (10ns [DDR])

# High-Speed NAND Program Array/Bus Performance



Interface speed is no longer a limitation to programming

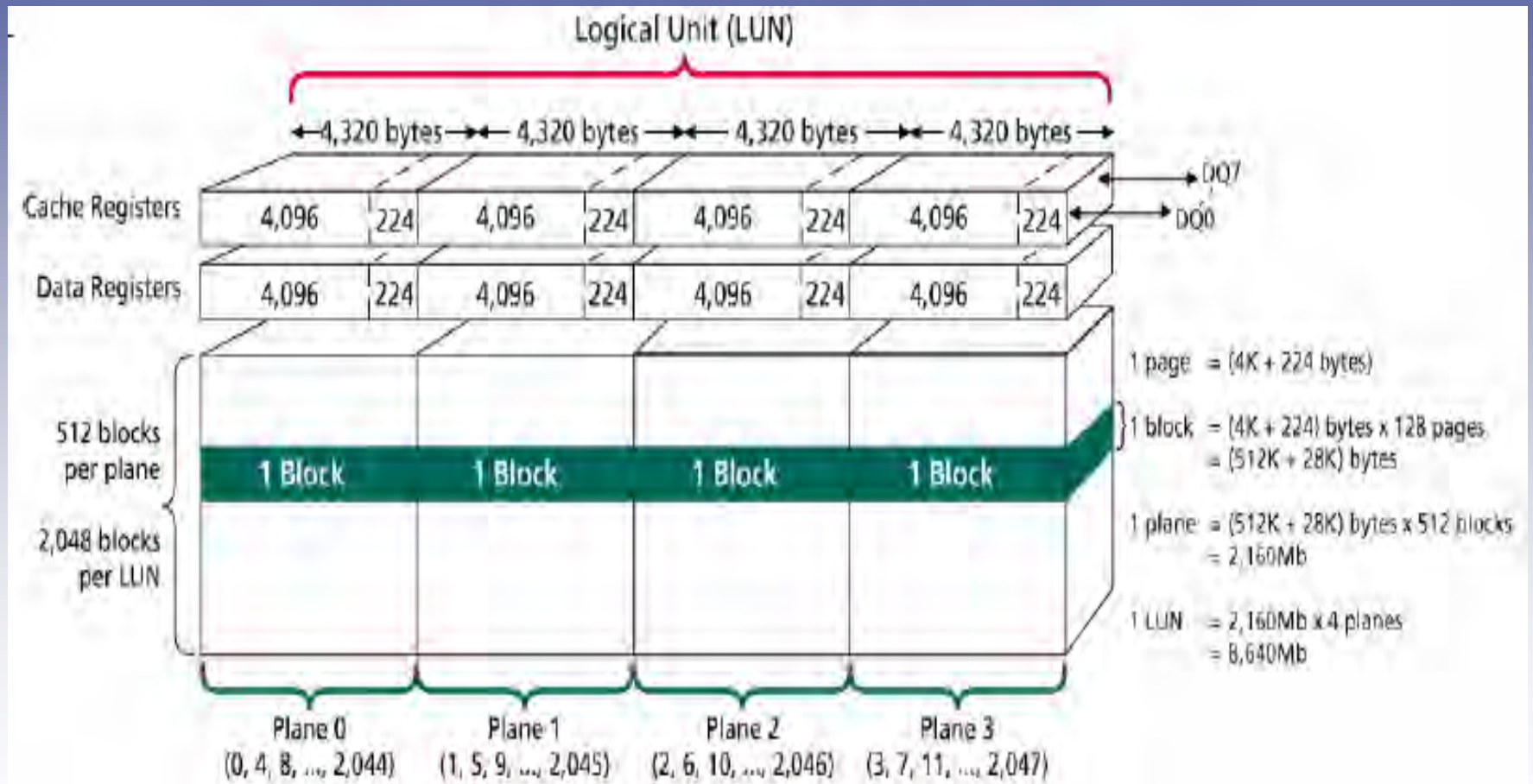
- Bus bandwidth is 200 MB/s (10ns [DDR])



Program performance is very impressive

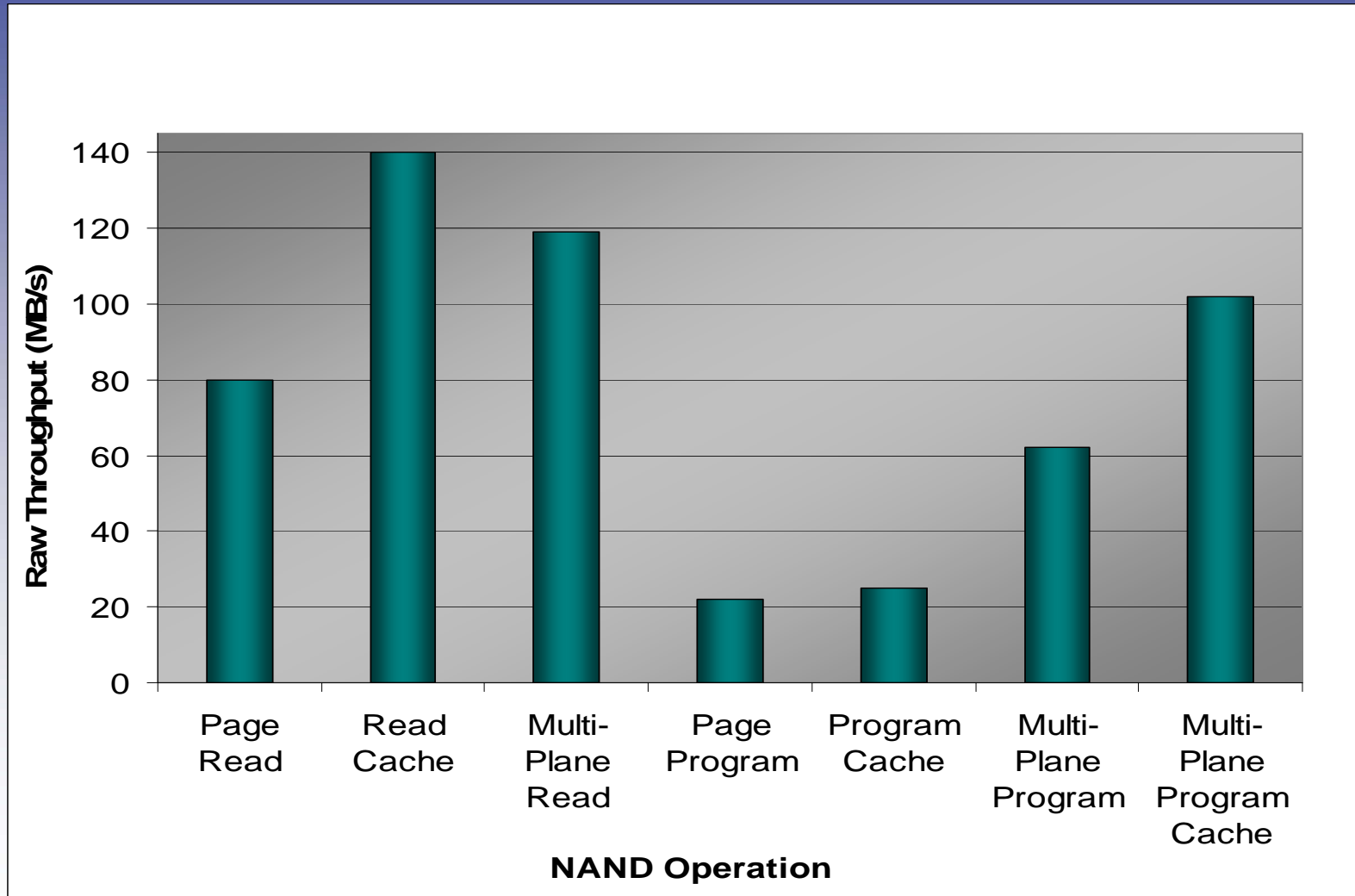
- Array program bandwidth is 100 MB/s (16KB programmed in 160μs)

# Four-Plane, 4K-Page, SLC High-Speed NAND Architecture



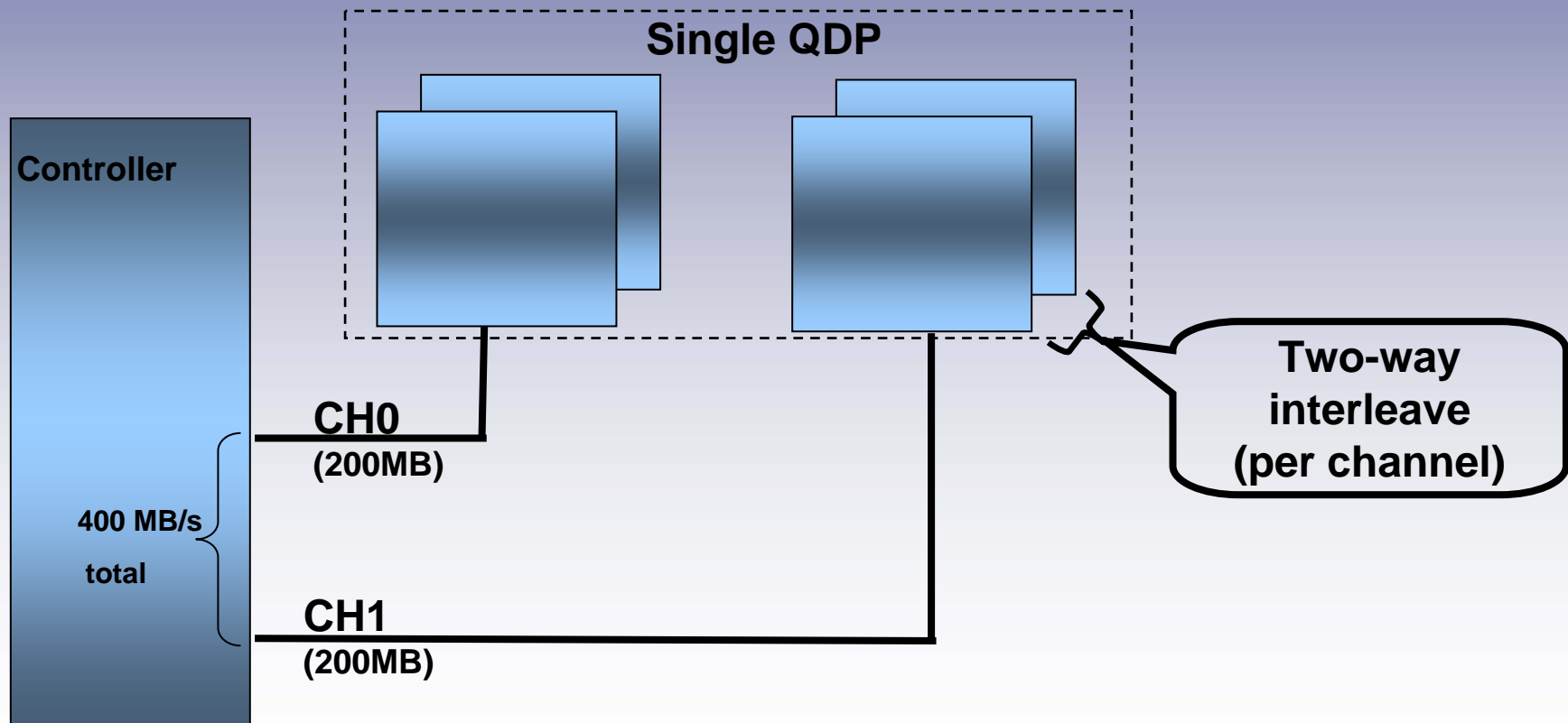


# Four-Plane, 4K-Page SLC NAND Architecture (High-Speed NAND)



# HS-NAND Solution for High Performance

- Micron can achieve 400 MB/s of programming performance using a single HS-NAND package (4 die total)
  - Two channels, two-way interleave (100 MB/s per die)
  - This provides a minimum density of 4GB



# ONFI 2.0 Summary

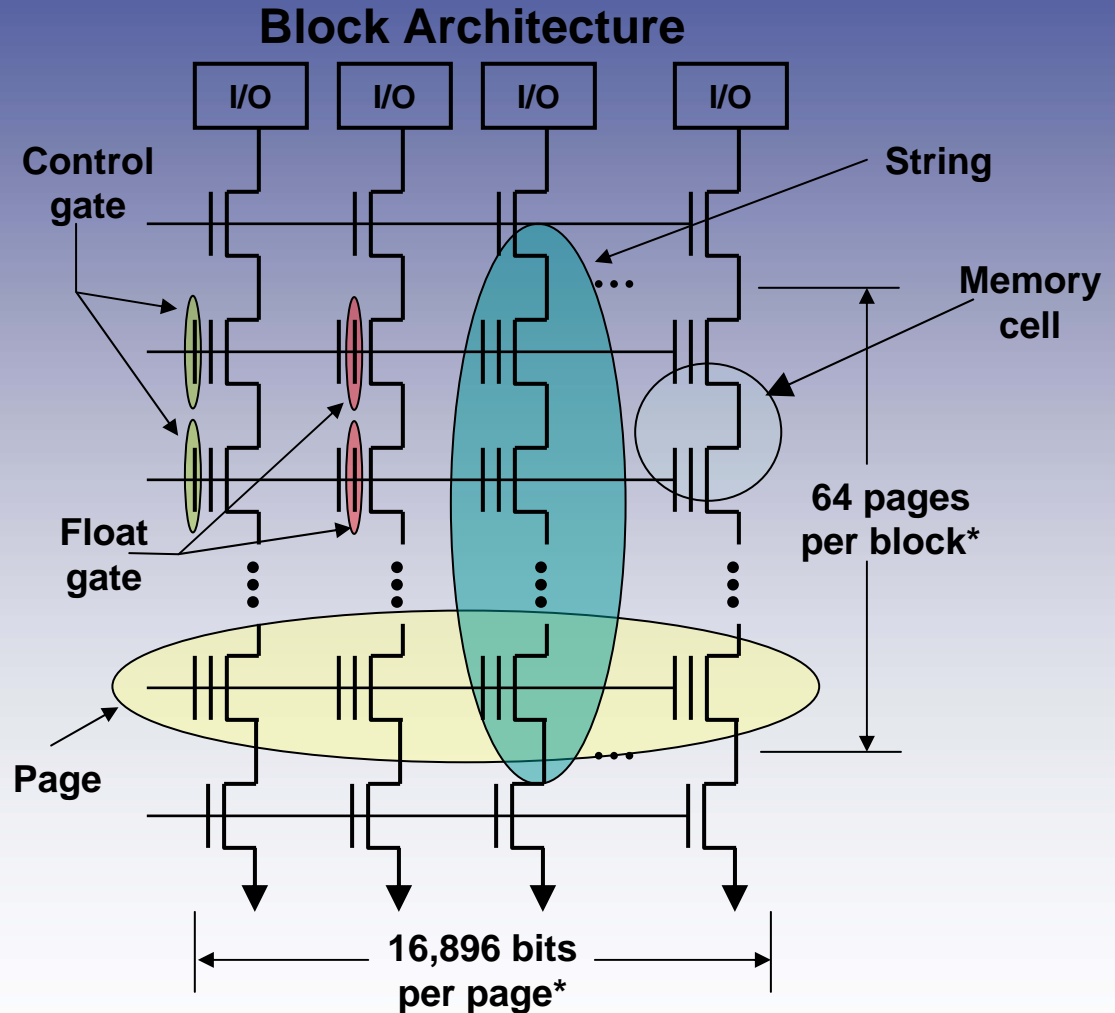
- Fast source-synchronous interface
- Backward compatible with ONFI 1.0
  - Asynchronous interface support
  - ONFI protocol compatible
  - Self-identification of NAND features through parameter page
- Low-power DDR I/O
- Scalability for high-density applications
- New industry standard BGA package
- For more details on ONFI, visit <http://www.onfi.org/>

# NAND Error Modes

- Program disturb
- Read disturb
- Data retention
- Endurance

# Let's Get Orientated

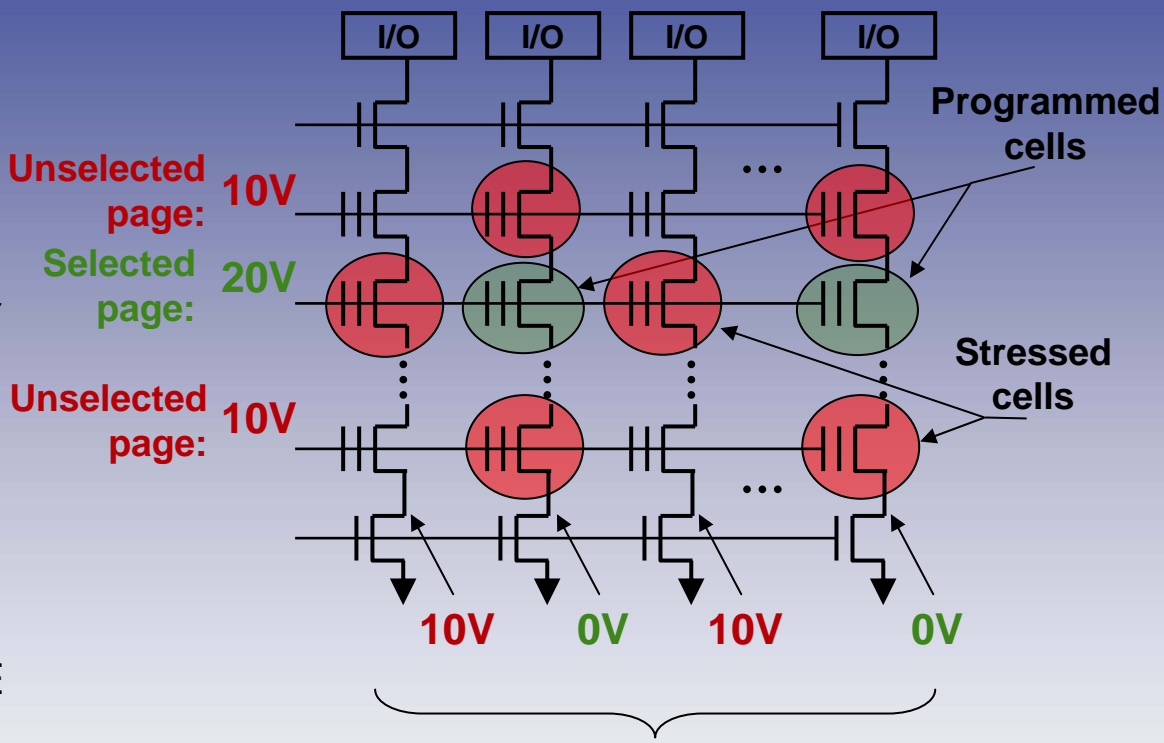
- NAND architecture is based on independent *blocks*
- Blocks are the smallest erasable units
- Pages are the smallest programmable units
  - Partial pages can be programmed in some devices



\* Typical for 4Gb SLC

# Program Disturb

- Cells *not* being programmed receive elevated voltage stress
- Stressed cells
  - Are always in the block being programmed
  - Either can be on pages *not* selected or in a selected page, but not supposed to be programmed
- Charge collects on the floating gate causing the cell to appear to be weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels
- Disturbed bits are effectively managed with error correction codes (ECC)
- Partial-page programming accelerates disturbance



Strings being programmed are grounded; others are at 10V

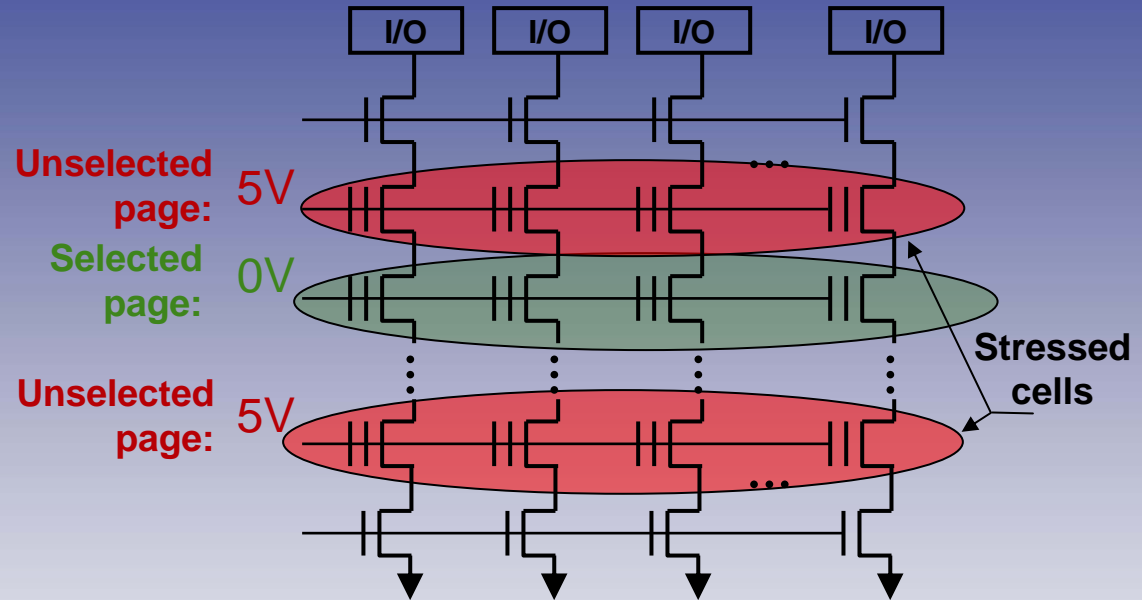
Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

# Reducing Program Disturb

- Program pages in a block sequentially, from page 0 to page 63 (SLC) or 127 (MLC)
- Minimize partial-page programming operations (SLC)
- It is mandatory to restrict page programming to a single operation (MLC)
- Use ECC to recover from program disturb errors

# Read Disturb

- Cells *not* being read receive elevated voltage stress
- Stressed cells are
  - Always in the block being read
  - Always on pages not being read
- Charge collects on the floating gate causing the cell to appear to be weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels
- Disturbed bits are effectively managed with ECC



Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

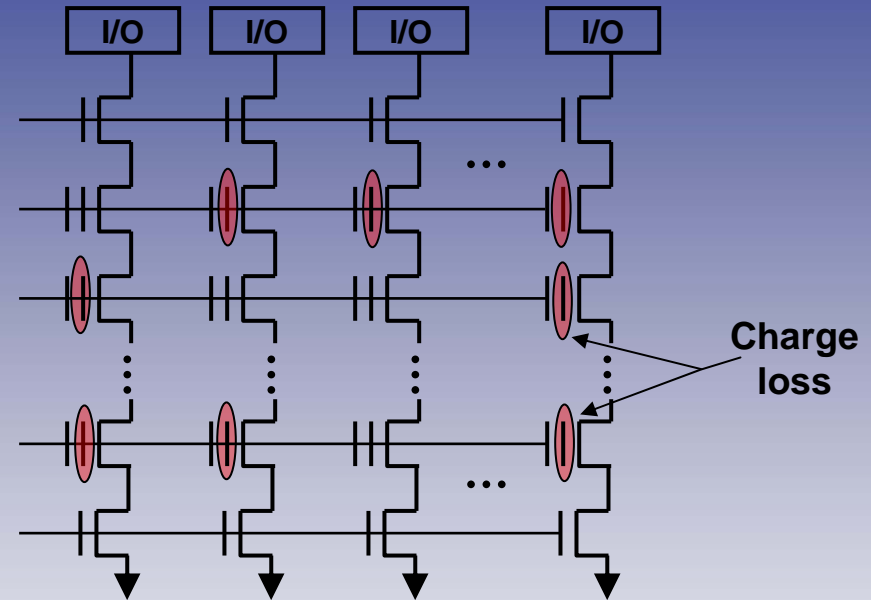


# Reducing Read Disturb

- Rule of thumb for excessive reads per block between ERASE operations
  - SLC – 1,000,000 READ cycles
  - MLC – 100,000 READ cycles
- If possible, read equally from pages within the block
- If exceeding the rule-of-thumb cycle count, then move the block to another location and erase the original block
- Establish ECC threshold to move data
- Erase resets the READ DISTURB cycle count
- Use ECC to recover from read disturb errors

# Data Retention

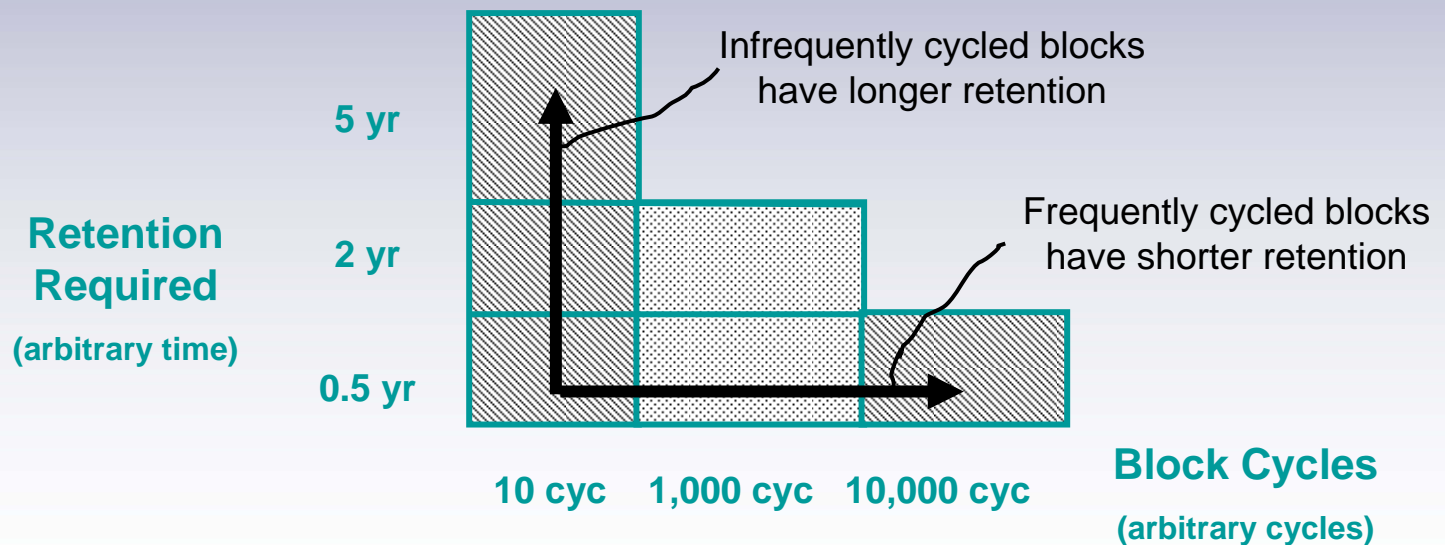
- Charge loss/gain occurs on the floating gate over time; device threshold voltage trends to a quiescent level
- Cell is undamaged; block can be reliably erased and reprogrammed



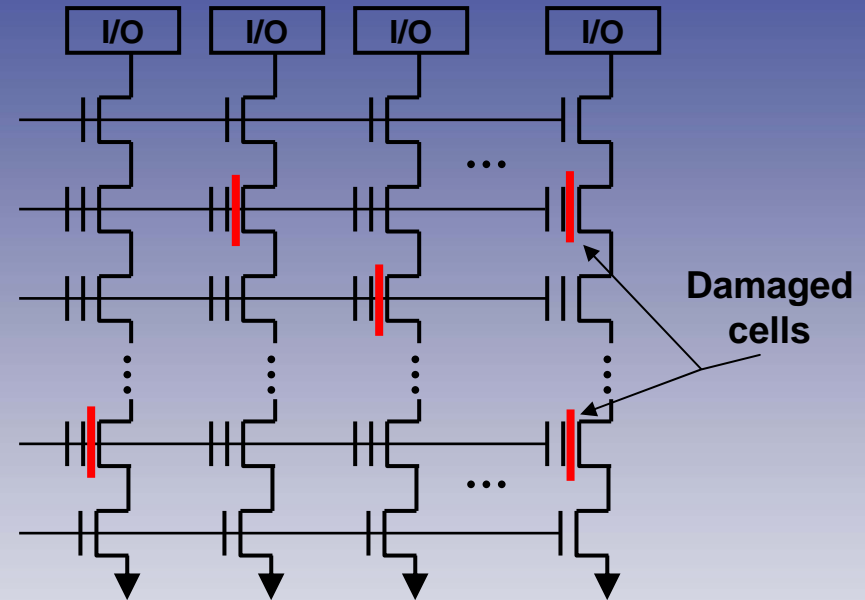
Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

# Improving Data Retention

- Limit PROGRAM/ERASE cycles in blocks that require long retention
- Limit READs to reduce read disturb
- Review JEDEC (JESD47) standard



- PROGRAM/ERASE cycles cause charge to be trapped in the dielectric
- Causes a permanent shift in cell characteristics—not recovered by erase
- Observed as failed program or erase status
- Blocks that fail should be retired (marked as bad and no longer used)



Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

# Endurance Recommendations

- Always check pass/fail status (SR0) for PROGRAM and ERASE operations
  - Note: READ operations do not set SR0 to fail status
- If fail status after PROGRAM, move all block data to an available block and mark the failed block bad
- Use ECC to recover from errors
- Write data equally to all good blocks (wear-leveling)
- Protect block management/meta data in spare area with ECC

# Wear-Leveling

- Wear-leveling is a plus on SLC devices where blocks can support up to 100,000 PROGRAM/ERASE cycles
- Wear-leveling is imperative on MLC devices where blocks typically support fewer than 10,000 cycles
- If a block was erased and reprogrammed every minute, the 10,000 cycling limit would be exceeded in just 7 days!

$$60 \times 24 \times 7 = 10,080$$

- Rather than cycling the same block, wear-leveling involves distributing the number of blocks that are cycled

## Wear-Leveling (continued)

- An 8Gb MLC device contains 4,096 independent blocks
- Using the previous example, if the cycles were distributed over 4,096 blocks, each block would be programmed fewer than 3 times (vs. 10,800 cycles if the same block is cycled)
- If perfect wear-leveling was performed on a 4,096-block device, a block could be erased and programmed every minute, every day for 77 years!

$$\frac{10,000 \times 4,096}{60 \times 24} = \frac{40,960,000}{1,440} = 28,444 \text{ days} = 77.9 \text{ years}$$

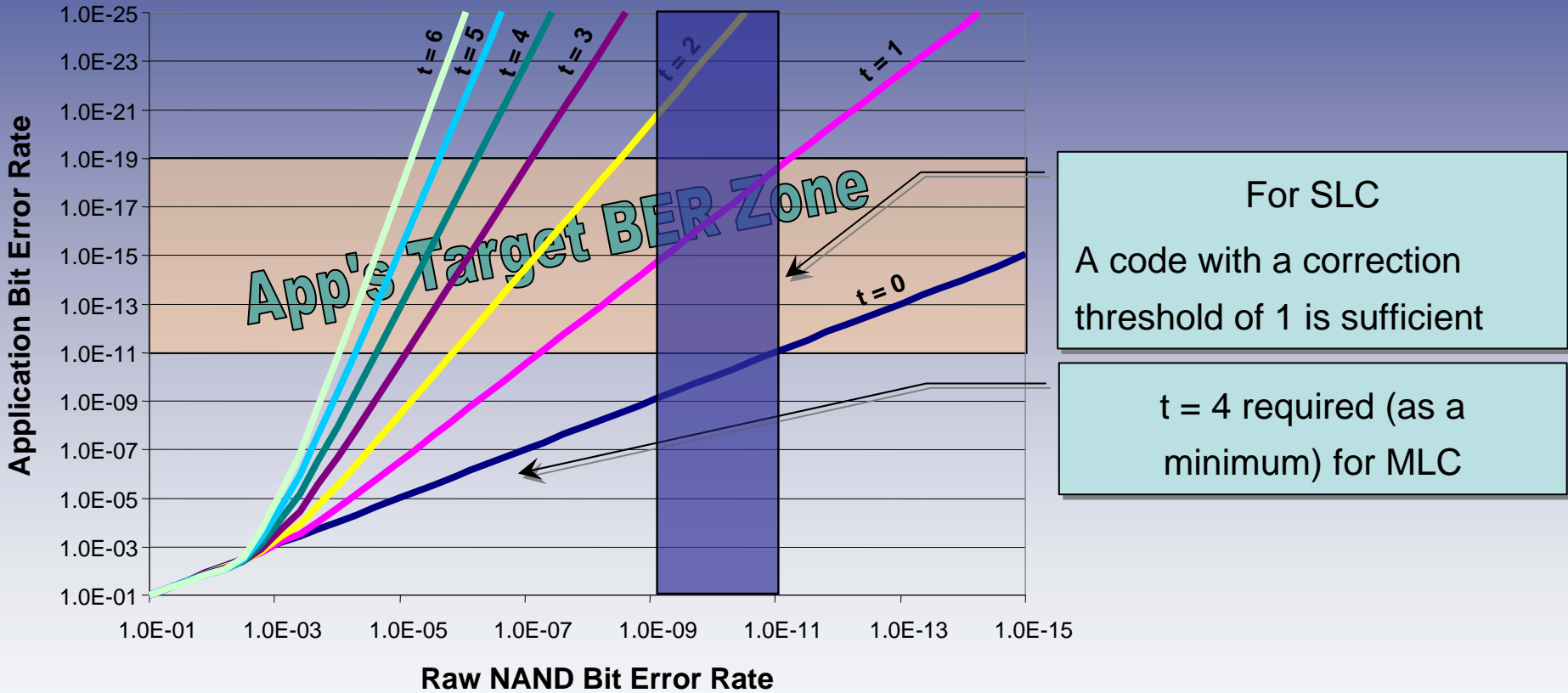
- Consider static vs. dynamic wear-leveling

# ECC Can Fix Everything (well, almost)

- Understand the target data error rate for your particular system
- Understand the use model that you intend for your system
- Design the ECC circuit to improve the raw bit error rate (BER) of the NAND Flash, under *your* use conditions, to meet the system's target BER



# ECC Code Selection is Becoming Even More Important



- As the raw NAND Flash BER increases, it becomes more important to match the ECC to the application's target BER

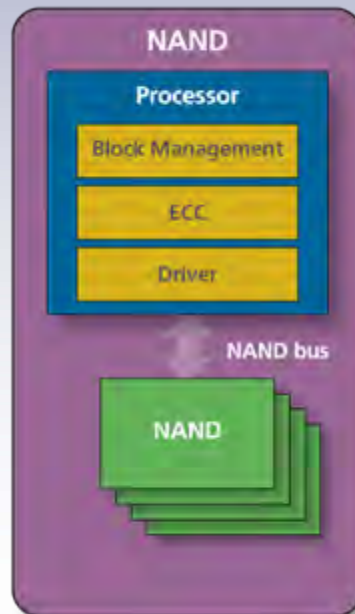
## Another Option: e-MMC™ Embedded Memory

- The complexities of future MLC require increased attention; the ECC algorithm, for example, is becoming more and more complex, moving from 4+ bits to 8+ bits in the future
- A managed interface addresses the complexities of current and future NAND Flash devices; this means the host does not need to know the details of NAND Flash block sizes, page sizes, planes, new features, process generation, MLC vs. SLC, wear-leveling, ECC requirements, etc.
- e-MMC™ embedded memory is the next logical step in the NAND Flash evolution for embedded applications because it turns a program/erase/read device with bad blocks and bad bits (NAND Flash) into a simple write/read memory

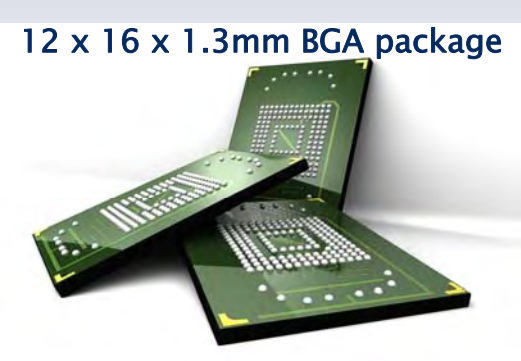
# Micron Solution: e-MMC Embedded Memory (Managed NAND)

- MLC NAND + MMC 4.3 controller in one device
- High-speed solution:
  - Host selectable x1, x4, and x8 I/Os
  - 52 MHz clock speed (MAX) – 416 Mb/s data rate (MAX)
- Fully backward compatible with previous MMC systems
- ECC, wear-leveling, and block management (built in)

**Direct NAND Interface**



**Managed NAND**



# Error Mode Conclusions

- NAND Flash is the lowest cost, nonvolatile memory available today
- Complexities of MLC NAND require increased hardware and software design
- All these complexities are addressed through the use of the controller included with eMMC embedded memory

# Reference Material

- Micron presentations and webinars:  
<http://NAND.com>
- Micron documentation (specifications, technical notes, and FAQs):  
<http://www.micron.com/products/nand/>
- The Error Correcting Codes (ECC) Page:  
<http://www.eccpage.com/>
- Standards
  - MultiMediaCard Association (MMCA):  
<http://www.mmca.org/>
  - JEDEC:  
<http://www.jedec.org/>
  - Open NAND Flash Interface (ONFI) Workgroup:  
<http://www.onfi.org/>
  - SD Card Association (SDA):  
<http://www.sdcard.org/>



# Thank You

©2008 Micron Technology, Inc. All rights reserved. Products are warranted only to meet Micron's production data sheet specifications. Information, products and/or specifications are subject to change without notice. All information is provided on an "AS IS" basis without warranties of any kind. Dates are estimates only. Drawings not to scale. Micron and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners.

Santa Clara, CA USA  
August 22–24, 2008