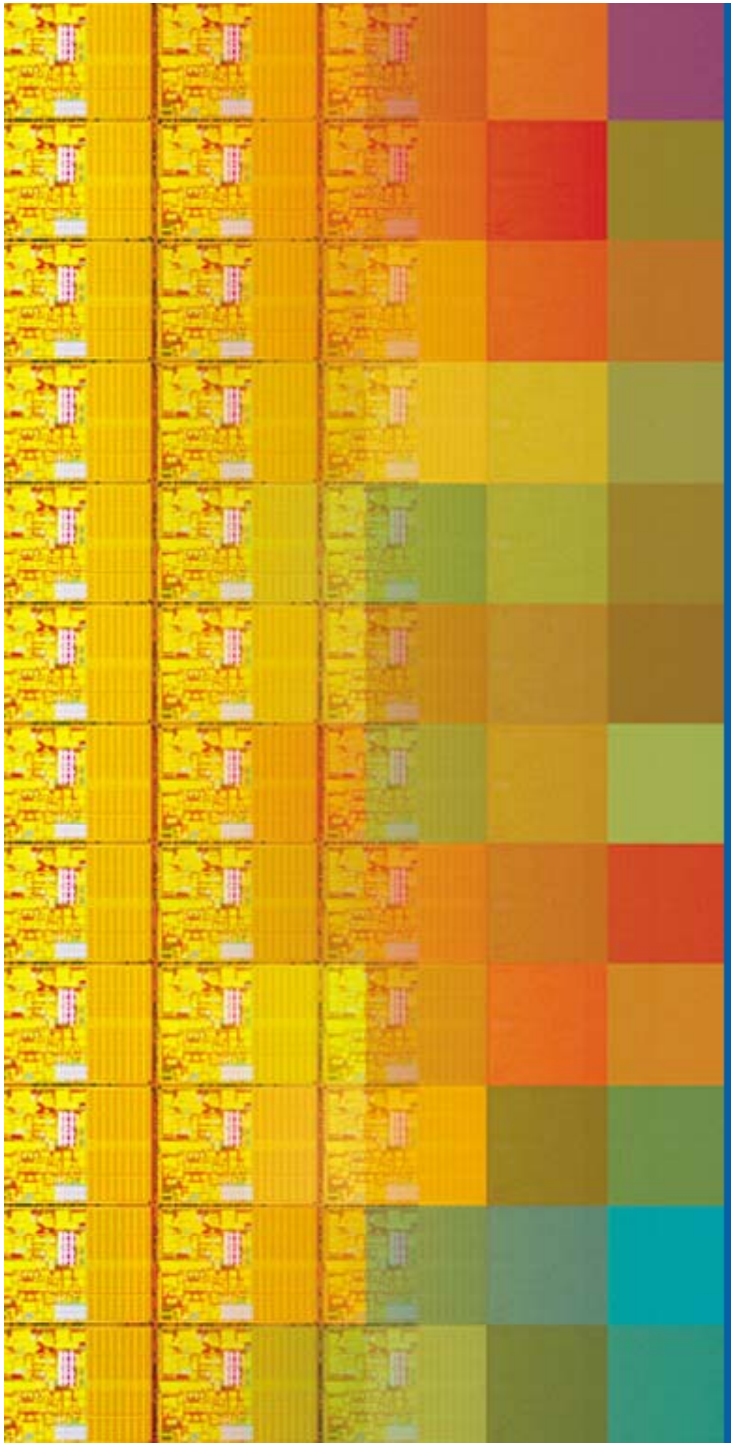


A Standard Interface for NAND Flash

Amber Huffman
Senior Staff Architect, Intel

K.S. Pua
President, Phison

Intel Developer FORUM

Legal Disclaimer

Intel Developer FORUM

Agenda

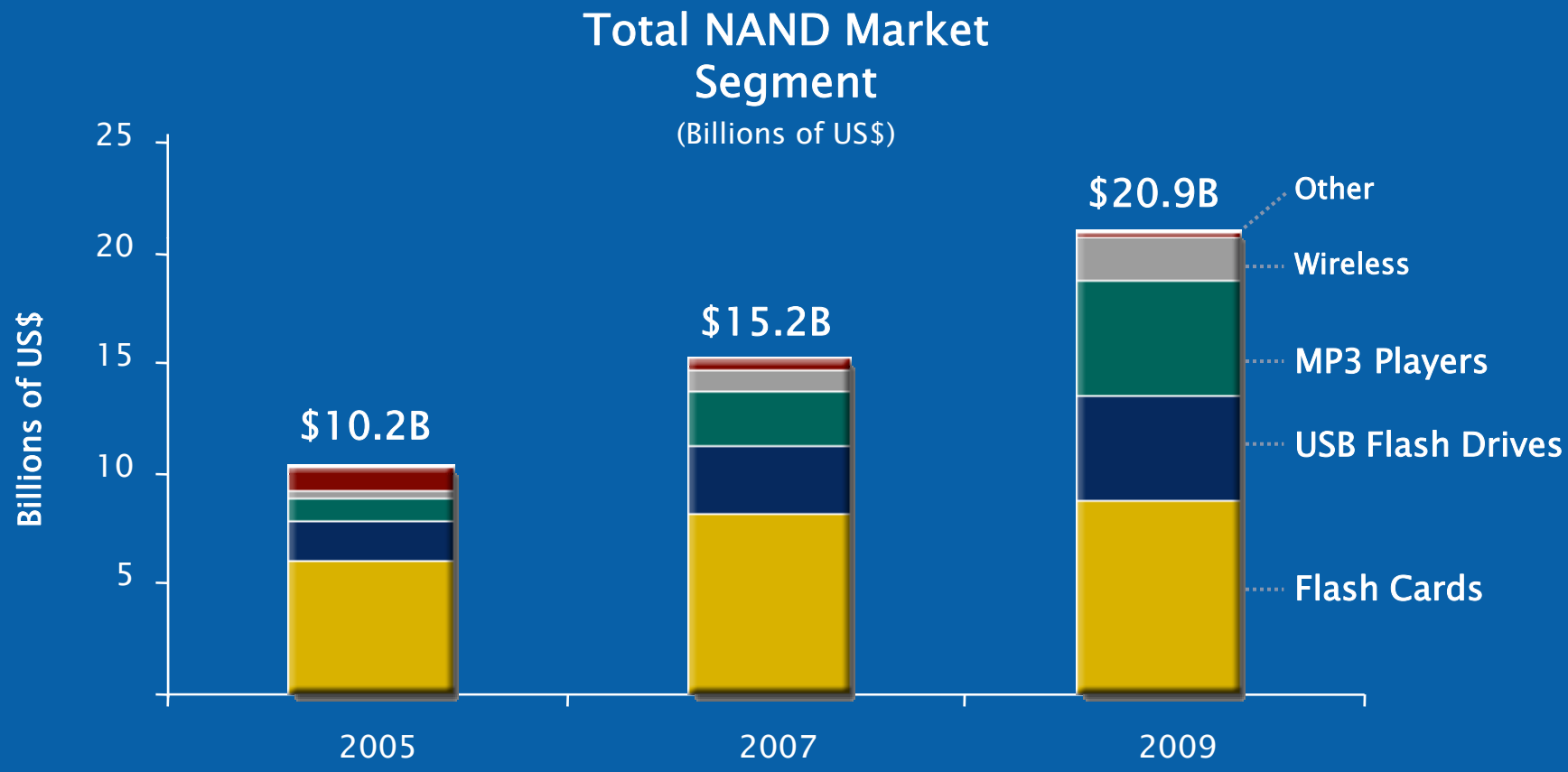
System problems with NAND Today

NAND Flash Product Integration

Open NAND Flash Interface (ONFI) initiative overview

ONFI Technical Preview

NAND Flash Is Becoming a Commodity Memory Product



Source: WSTS, Micron Market Research

Intel Developer FORUM

Intel's NAND Dilemma…

NAND Flash is an increasingly important Intel platform ingredient

- Robson technology highlights increasing role of NAND Flash in Intel's platform plans
- Intel's platforms have longevity and address numerous market segments, which requires support for a range of NAND Flash components
- Intel's stable platform must have the means for conveniently supporting the latest Flash components without having to be revamped

Current similar NAND Flash components do not allow for range of NAND to be accommodated in a platform

Similar: Basic Commands

Basic commands *typically* common

– Reset, Read ID, Read, Page Program, Erase, …

More complex operations all over the map

Table 1. Command Sets

Function	1st. Cycle	2nd. Cycle
Read	00h	30h
Read for Copy Back	00h	35h
Read ID	90h	-
Reset	FFh	-
Page Program	80h	10h
Cache Program	80h	15h
Copy-Back Program	85h	10h
Block Erase	60h	D0h
Random Data Input*1	85h	-
Random Data Output*1	05h	E0h
Read Status	70h	

*Samsung K9K4G08U0M datasheet

FUNCTION	1st CYCLE	2nd CYCLE
READ 1	00h	30h
READ FOR COPY-BACK	00h	35h
READ ID	90h	-
RESET	FFh	-
PAGE PROGRAM (start)	80h	10h
COPY BACK PGM (start)	85h	10h
CACHE PROGRAM	80h	15h
BLOCK ERASE	60h	D0h
READ STATUS REGISTER	70h	-
RANDOM DATA INPUT	85h	-
RANDOM DATA OUTPUT	05h	E0h
CACHE READ START	00h	31h
CACHE READ EXIT	34h	-
LOCK BLOCK	2Ah	-
LOCK TIGHT	2Ch	-
UNLOCK (start area)	23h	-
UNLOCK (end area)	24h	-
READ LOCK STATUS	7Ah	-

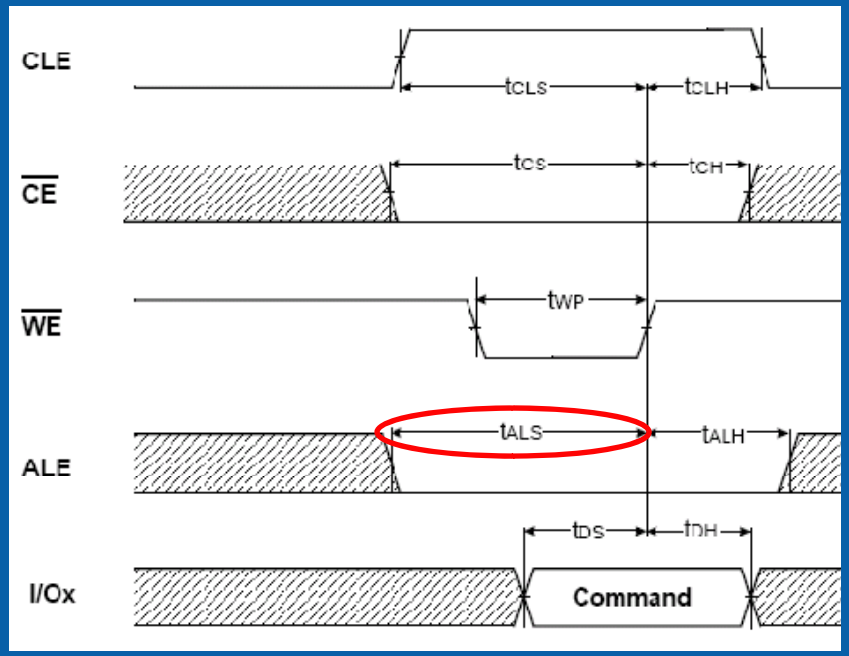
Table 4: Command Set

*Hynix HY27UG084G2M datasheet

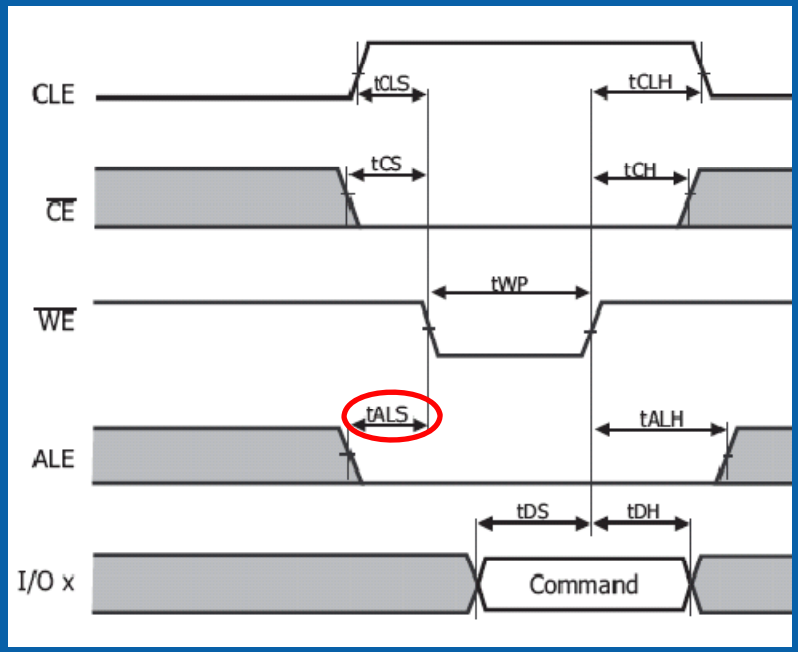
Similar: Timing

Timing requirements are specified differently

Min/max values are not necessarily the same for similar cycle time parts



*Samsung K9K4G08U0M datasheet



*Hynix HY27UG084G2M datasheet

Similar: Status Values

Status values dependent on command

Often the same, but not required to be

–Can you tell that these are the same??

I/O No.	Page Program	Block Erase	Cache Prorgam	Read	Definition
I/O 0	Pass/Fail	Pass/Fail	Pass/Fail(N)	Not use	Pass : "0" Fail : "1"
I/O 1	Not use	Not use	Pass/Fail(N-1)	Not use	Pass : "0" Fail : "1"
I/O 2	Not use	Not use	Not use	Not use	Don't -cared
I/O 3	Not Use	Not Use	Not Use	Not Use	Don't -cared
I/O 4	Not Use	Not Use	Not Use	Not Use	Don't -cared
I/O 5	Ready/Busy	Ready/Busy	True Ready/Busy	Ready/Busy	Busy : "0" Ready : "1"
I/O 6	Ready/Busy	Ready/Busy			
I/O 7	Write Protect	Write Protect			

*Samsung K9K4G08U0M datasheet

*Hynix HY27UG084G2M datasheet

IO	Page Program	Block Erase	Cache Program	Read	Cache Read	CODING
0	Pass / Fail	Pass / Fail	Pass / Fail (N)	NA		Pass: '0' Fail: '1'
1	NA	NA	Pass / Fail (N-1)	NA		Pass: '0' Fail: '1' (Only for Cache Program, else Don't care)
2	NA	NA	NA	NA		-
3	NA	NA	NA	NA		-
4	NA	NA	NA	NA		-
5	Ready/Busy	Ready/Busy	P/E/R Controller Bit	Ready/Busy	P/E/R Controller Bit	Active: '0' Idle: '1'
6	Ready/Busy	Ready/Busy	Cache Register Free	Ready/Busy	Ready/Busy	Busy: '0' Ready': '1'
7	Write Protect	Write Protect	Write Protect	Write Protect		Protected: '0' Not Protected: '1'

Similar: Read ID

The first and second byte are consistently manufacturer and device ID

The number of remaining bytes and what they mean is up in the air

*Samsung K9K4G08U0M datasheet

	Description
1st Byte	Maker Code
2nd Byte	Device Code
3rd Byte	Don't care
4th Byte	Page Size, Block Size, Spare Size, Organization

*Toshiba TH58NVG1S3AFT05 datasheet

	Descripton	I/O8	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	Hex Data
1st Data	Maker Code	1	0	0	1	1	0	0	0	98H
2nd Data	Device Code	1	1	0	1	1	0	1	0	DAH
3rd Data	Chip Number, Cell Type, PGM Page, Write Cache	0 or 1	0	0	0	0	0	0	1	81H or 01H
4th Data	Page Size, Block Size, Redundant Size, Organization	0 or 1	0	0	1	0	1	0	1	95H or 15H
5th Data	Plane Number, Plane Size	0 or 1	1	0	0	0	1	0	0	44H or C4H

Similar Hinders NAND Adoption

To deal with differences, the host must maintain a chip ID table of known devices

- Table contains read/write timings, organization, status bit meanings, etc for each known NAND Flash part

Situation has two major effects:

- Precludes intro of new NAND devices into existing designs
- Makes qualification cycles longer as each NAND device added requires changes to be comprehended

Similar to the ancient disk drive interfaces that required a list of disk drive types in a BIOS table

Lack of standard impacts platforms supporting a range of NAND

Agenda

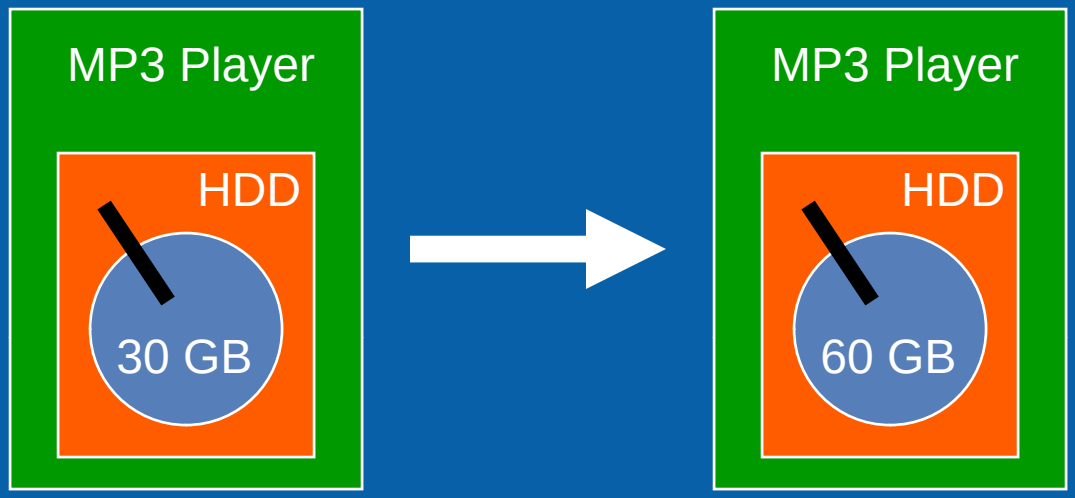
System problems with NAND Today

NAND Flash Product Integration

Open NAND Flash Interface (ONFI) initiative overview

ONFI Technical Preview

Product Update Cycle for HDD

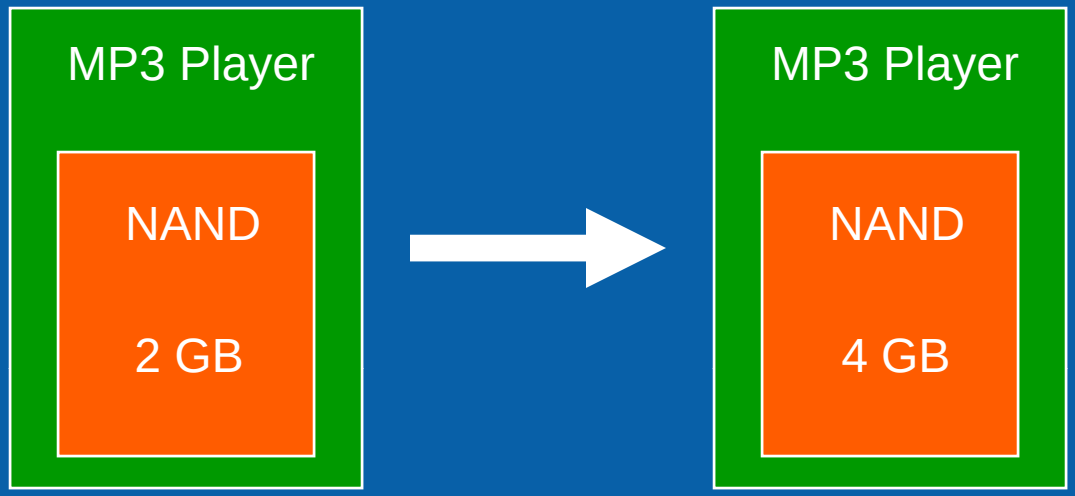


No software changes required

Product-level testing and validation focus

Physical and logical interface standard

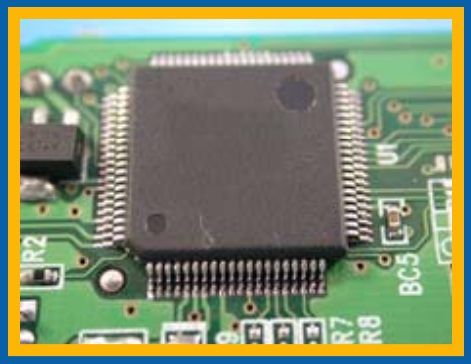
Product Update Cycle for NAND



- Firmware/software changes required
  - Change device ID table to support new component
  - Add support for new commands to maintain or increase performance
- Potentially spin new Mask ROM (1 month delay)
- Complete re-test of firmware and/or software

Inconsistency of NAND Flash

- Higher Performance
  - –Parallel operations
  - –Faster timing
- Higher Density
  - –(Multi-level cell) MLC
  - –Fewer erase cycles
- Process Improvements
  - –70nm, 55nm.....



Challenges for Controller

When new NAND Flash is released (from the same or a different vendor), the controller vendor needs to do numerous updates

- Look for new Flash Device ID
- Update timing to comprehend new part
- Update ECC to comprehend bit rate
- Other modifications based on device parameters

Another challenge: these changes must be made to a Mask ROM, with small storage area (64KB typical) and updates requiring a 1 month wait for a new IC

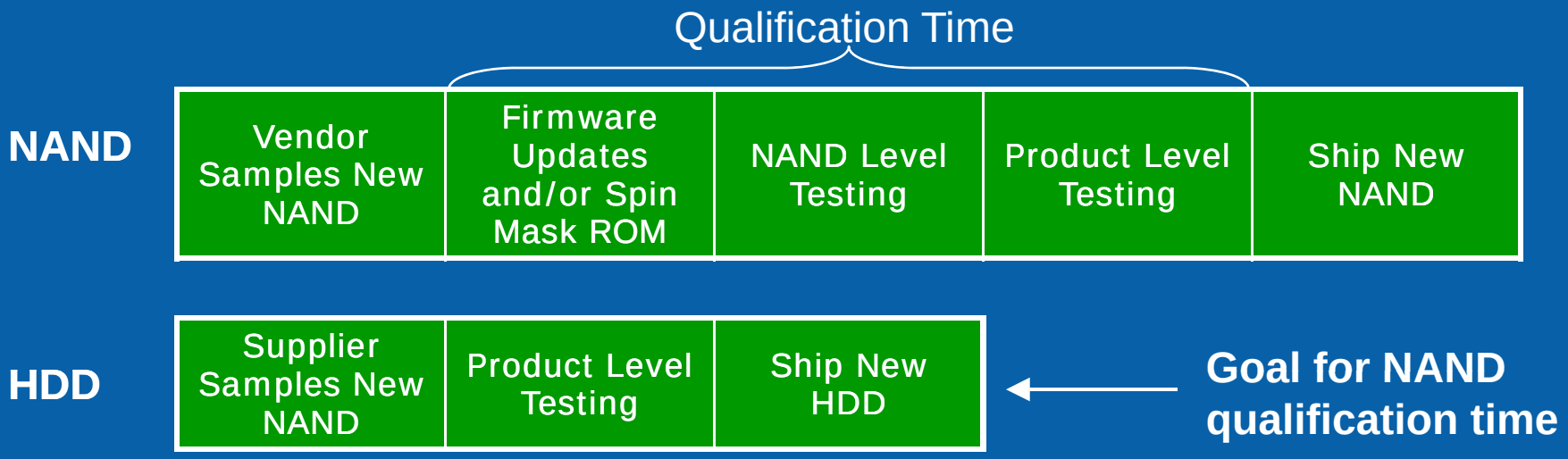
Controllers lag behind Flash updates, and is one of the last to know about changes...

Result of These Challenges

- Inventory control is difficult and array of parts confusing to customers
  - Mask ROMs are too small to support all Flash behavior, must choose subset
  - Several firmware versions must be maintained for every controller to support most NAND in the market
- Any firmware change results in one month delay to the customer
  - Mask ROM change is 1 month for new IC
  - In current environment, cannot anticipate any upcoming changes effectively

NAND Flash Product Integration

Delayed Opportunities…



- NAND Flash requires more qualification time than other commodity memory products
- Lost revenue opportunity that could be rectified with standard interface
  - Impossible today for controller to anticipate and be prepared for upcoming Flash behavior

Lack of standard interface impacts time to market and revenue

Intel Developer FORUM

PHISON Knows What You Need

Agenda

System problems with NAND Today

NAND Flash Product Integration

Open NAND Flash Interface (ONFI) initiative overview

ONFI Technical Preview

Open NAND Flash Interface

Goals of Initiative:

- To develop a standardized NAND Flash interface that allows interoperability between NAND devices
- Accelerate time to market of NAND-based products

Technical Philosophy

ONFI shall ensure no pre-association 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

Initiative Status

- Intel is working with key partners to form the ONFI Workgroup
  - –More details to come imminently
- ONFI specification expected to be released in 2H'06

ONFI is being developed to solve the barriers to the rapid adoption of new NAND products

Agenda

System problems with NAND Today

NAND Flash Product Integration

Open NAND Flash Interface (ONFI) initiative overview

ONFI Technical Preview

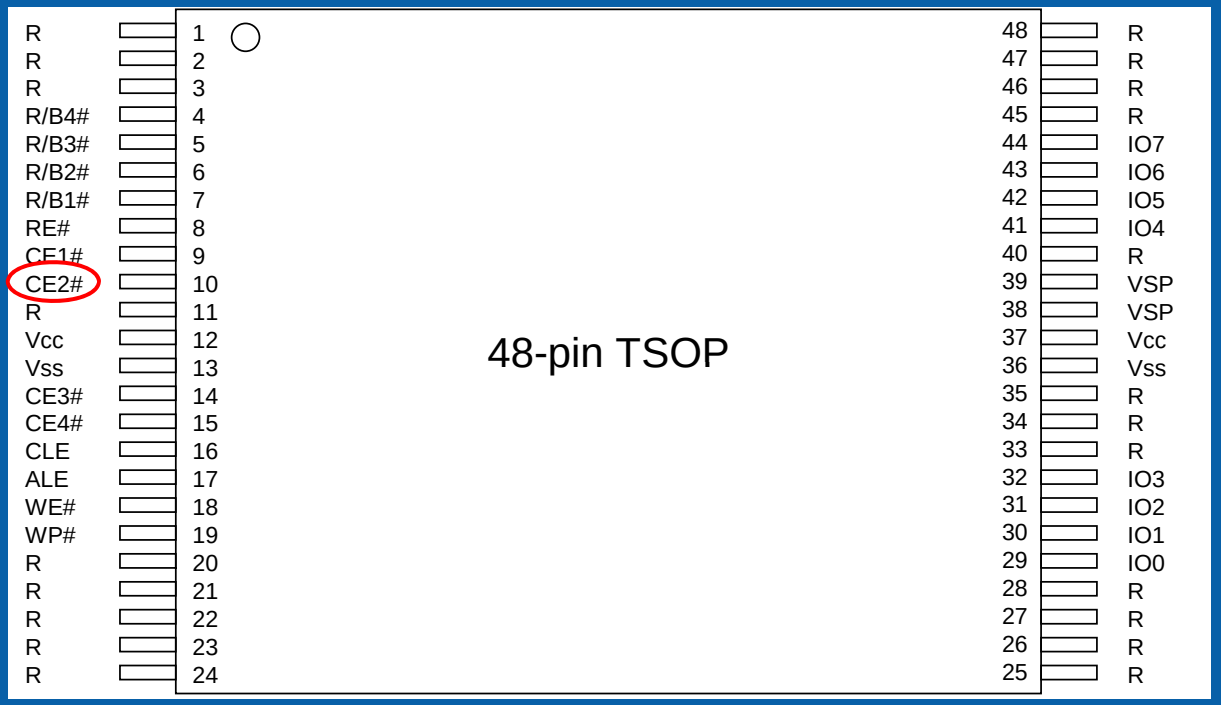
Pin-Out and Package

ONFI defines standard pin-outs for the 48-pin TSOP and 52-pin LGA packages

Pin-out critical for ensuring no board changes required in NAND Flash upgrade

– For example, the host can plan for a new part that will have an additional CE# beforehand

Host can plan for part with additional CE#



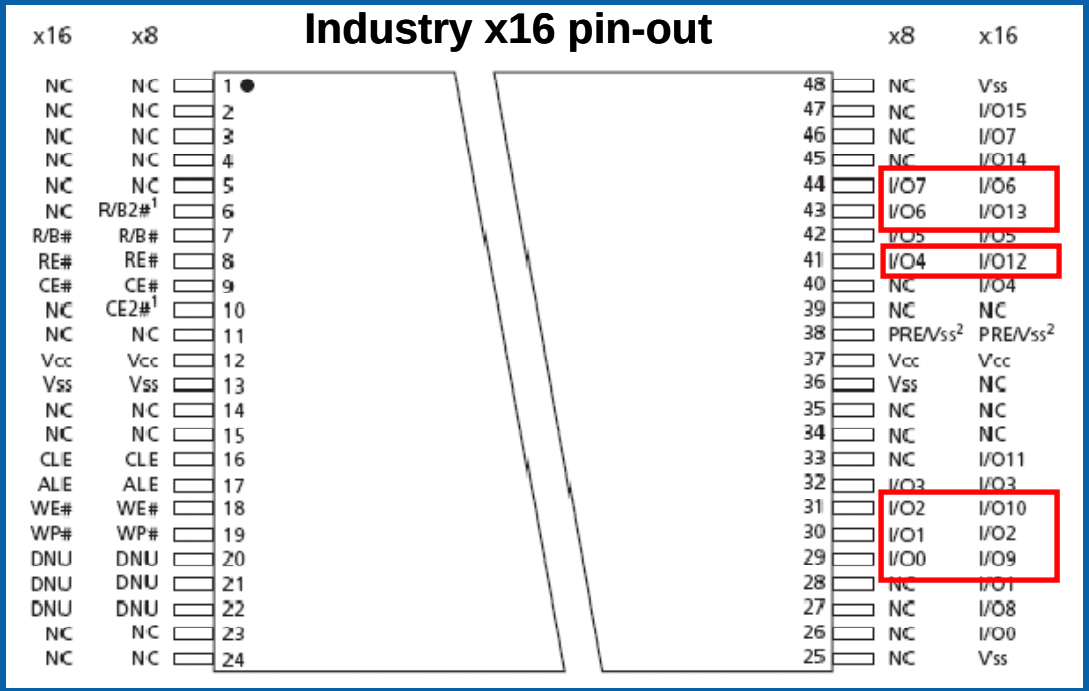
Fixing 16-bit Data Pin-out

The current industry x16 pin-out precludes board designs accepting both x8 and x16 parts easily

– The I/O0 – I/O7 pins are not in the same place!

ONFI is fixing this issue before x16 parts become entrenched (x16 currently is ~ < 1% of volume)

Industry x16 pin-out



6 of 8 data pins are in different locations on the x16 pin-out!

Device Abstraction

ONFI presents a device representation to the host based on independence “levels”

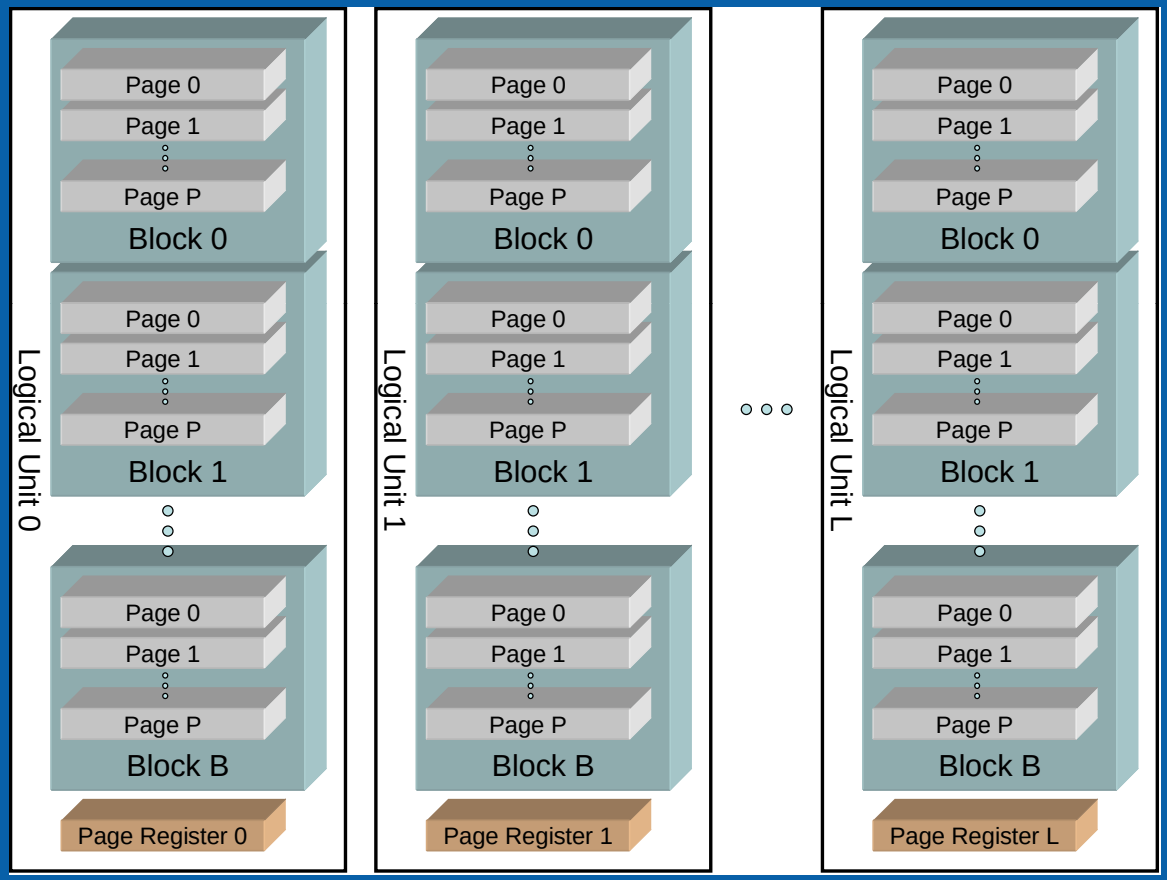
- ONFI has no notion of number of die or number of planes, etc.

Target: Completely independent unit with its own chip enable (CEx#)

- Logical Unit (LUN): Logically independent unit, shared chip enable
  - Interleaved Addressing: Dependent operations allowed for different blocks with same page address

Targets and LUNs

- Each target may support multiple logical units
- Each LUN may support interleaved addressing for increased parallelism
- Actual implementation abstracted



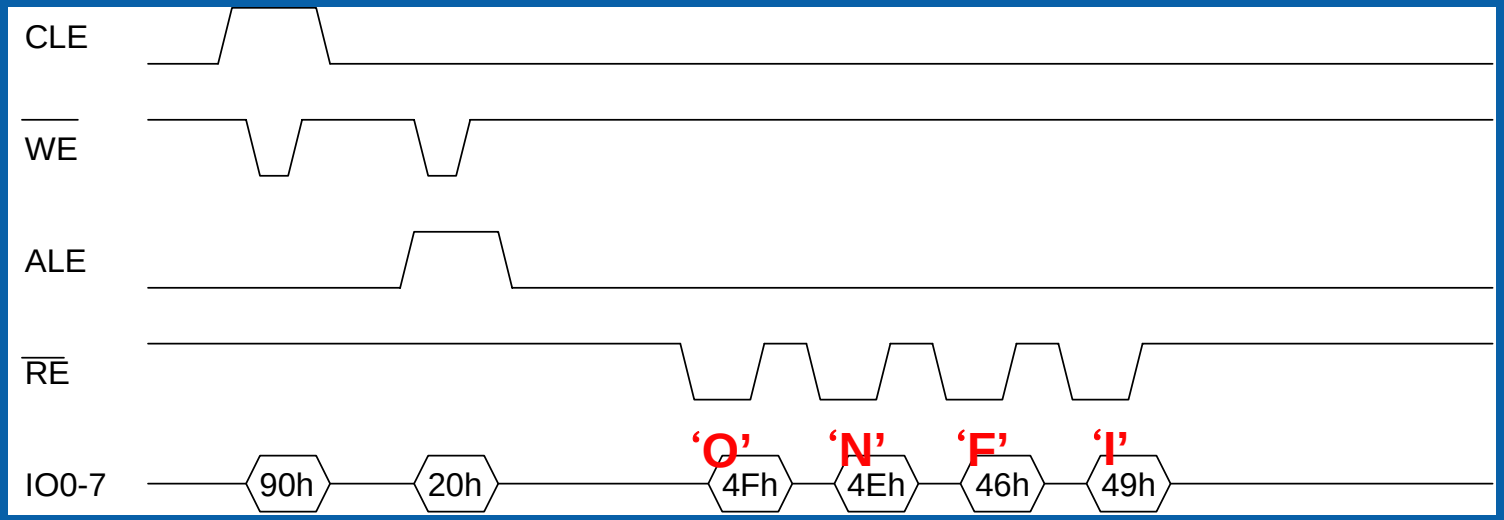
Target X, using CEx#

Determining ONFI Support

Read ID is used by Flash parts today to report device ID for use in chip ID table lookup

ONFI support is shown by responding to Read ID for address 20h with ASCII ‘ONFI’

Support for vendor specific interface and ONFI allowed by changing address cycle to 20h from 0h



Self Reporting Capabilities

Each target describes its features and capabilities through a parameter page

Blocks of the parameter page are devoted to:

- Revision information
- Device features
- Manufacturer information
- Memory organization
- Timing parameters
- Vendor specific

Memory Organization Block of Parameter Page

- Number of data bytes per page
- Number of redundant bytes per page
- Number of pages per block
- Number of blocks per logical unit (LUN)
- Number of logical units
- Number of address cycles
- Number of bits per cell
- Block endurance
- Number of programs per page
- Recommended bits of ECC correction
- Interleaved addressing

Command Set Overview

Command	O/M	1st Cycle	2nd Cycle	Acceptable while Accessed LUN is Busy	Acceptable while Other LUNs are Busy
Read	M	00h	30h		Y
Change Read Column	M	05h	E0h		Y
Change Read Column Enhanced	O	06h	E0h		Y
Read Cache	O	31h			
Read Cache End	O	3Fh			
Block Erase	M	60h	D0h		Y
Read Status	M	70h		Y	Y
Read LUN Status	O	78h		Y	Y
Page Program	M	80h	10h		Y
Page Cache Program	O	80h	15h		Y
Change Write Column	M	85h			Y
Read ID	M	90h			
Read Parameter Page	M	ECh			
Get Features	O	EEh			
Set Features	O	EFh			
Reset	M	FFh		Y	Y

Timing Requirements

- NAND contains a lot of timing requirements and hence timings
- Reporting each and every timing value leads to validation challenge
  - Requires validation of all combinations
- To make timing information useful, organized into timing modes

Parameter	Description
tADL	Minimum ALE to data loading time
tALH	Minimum ALE hold time
tALS	Minimum ALE setup time
tAR	Minimum ALE to RE# delay
tBERS	Maximum block erase time
tCEA	Maximum CE# access time
tCH	Minimum CE# hold time
tCHZ	Maximum CE# high to output hi-Z
tCLH	Minimum CLE hold time
tCLR	Minimum CLE to RE# delay
…	…
tREH	Minimum RE# high hold time
tRHOH	Minimum RE# high to output hold
tRHW	Minimum RE# high to WE# low
tRHZ	Maximum RE# high to output hi-Z
tRLOH	Minimum RE# low to output hold
tRP	Minimum RE# pulse width
tRR	Minimum Ready to RE# low
tRST	Maximum device reset time
tWB	Maximum WE# high to R/B# low
tWC	Minimum write cycle time
tWH	Minimum WE# high hold time
tWHR	Minimum WE# high to RE# low
tWP	Minimum WE# pulse width

Timing Modes

- Timing modes define vast majority of required host timings as one “set”
- Three parameters are specified separately in RPP
  - –Max page read time
  - –Max block erase time
  - –Max page program time
- Timing modes supported reported in timing parameters block of RPP

Parameter	Mode Example		Unit
	30		ns
	Min	Max	
tADL	100		ns
tALH	5		ns
tALS	10		ns
tAR	10		ns
tCEA		25	ns
tCH	5		ns
tCHZ		20	ns
tCLH	5		ns
tCLR	10		ns
tCLS	15		ns
…	…	…	ns
tWB		100	ns
tWC	30		ns
tWH	10		ns
tWHR	60		ns
tWP	15		ns

Interleaved Operation

Interleaving may be used to complete the same operation on additional blocks on a per logical unit basis to enhance performance

- Concurrent interleaving: Operations to all of the blocks is issued at the same time and then executes in parallel
- Overlapped interleaving: Operations may be issued independently, allows host to determine later to do an additional operation

Interleaved operations may be reads, programs, or erases

When using interleaving, the lowest order bits of the block address may be modified.

- The rest of the address must be the same as the other operations being issued on that LUN.

Defect Mapping & Enumeration

An invalid block is indicated by a 00h value in the first or last page of a block

- Ensures robustness in marking bad pages in face of recoverable bit errors

The host uses this information to create its initial bad block table

```
// For each LUN maps defects
for (i = 0; i < NumLUNs; i++) {

  // For each block within this LUN, map defects
  for (j = 0; j < BlocksPerLUN; j++) {
    Defective=FALSE;

    // If a 00h value is in the first page, this block is defective
    ReadPage(lun=i; block=j; page=0; DestBuff=Buff);
    for (column=0; column<PageSize+RedundantBytes; column++) {
      if (Buff[column] == 00h)
        Defective=TRUE;
    }

    // If a 00h value is in the last page, this block is defective
    ReadPage(lun=i; block=j; page=PagesPerBlock-1; DestBuff=Buff);
    for (column = 0; column < PageSize+RedundantBytes; column++) {
      if (Buff[column] == 00h)
        Defective=TRUE;
    }

    // If the block was defective, then keep track of this
    if (Defective)
      MarkBlockDefective(lun=i; block=j);
  }
}
```

ONFI Technical Highlights

- ONFI support is identified via Read ID, the standard NAND chip ID command
- NAND devices report their capabilities using the Read Parameter Page
- ONFI standardizes the base subset of commands required to be supported by all NAND devices
- ONFI supports increased performance through parallelism made possible by multiple LUNs and interleaved addressing
- ONFI standardizes the pin-out and packaging to ensure no PCB changes are required for a new NAND part

ONFI provides the solid technical base necessary to confidently build NAND-based products

Summary

Lack of standard makes it impossible for platforms to support a range of NAND components, including components introduced at a later date

Lack of standard NAND Flash interface impacts time to market and revenue

ONFI is being developed to solve the barriers to the rapid adoption of new NAND products

ONFI enables NAND feature self- identification, and standardizes command set, pin-out, and packaging

Please fill out the Session Evaluation Form.

Additional sources of information on this topic:

Initiative updates available at: www.onfi.org

Session presentation available on IDF web site – when prompted enter:

Username: idf

Password: SPR2006

Thank You!

Intel Developer FORUM

Backup Slides


intel
Leap ahead™