NAND Flash

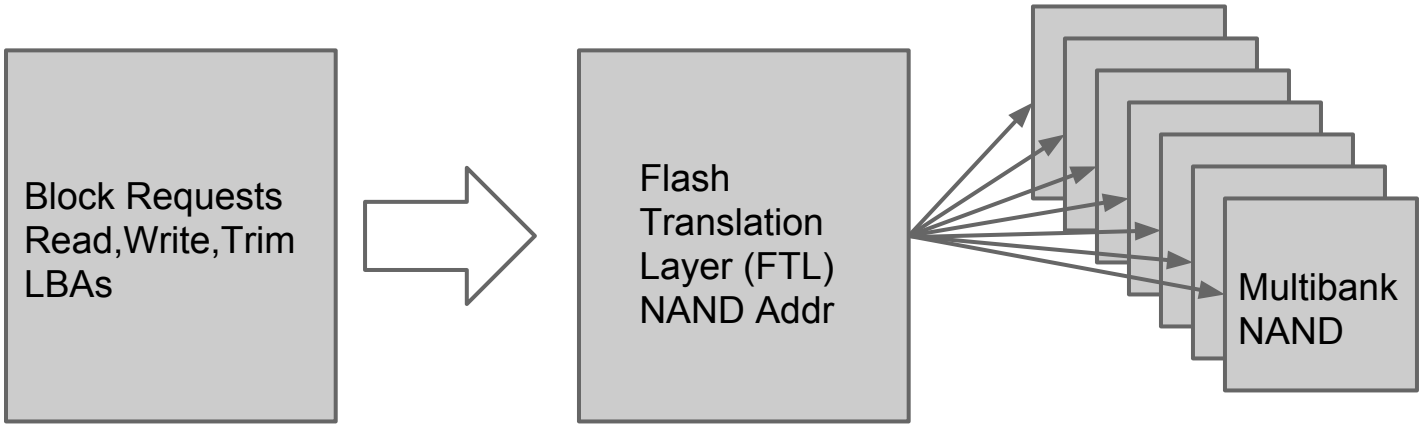
Overview and FreeBSD

http://people.freebsd.org/~imp/bsdcan2014.pdf

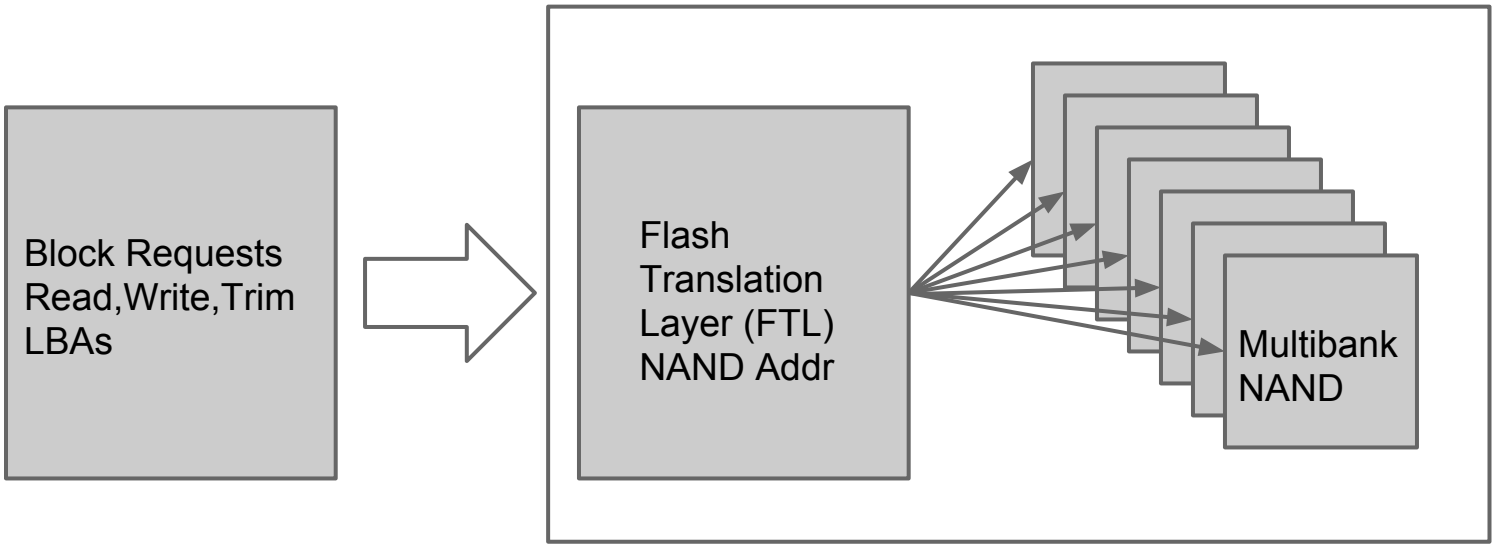
Talk Overview

- NAND Flash Fundamentals
  - Types of NAND
  - Geometry
  - Protocols
  - Storage Devices built on NAND
- FreeBSD NAND
  - Overview of architecture
  - Usage Walkthrough
  - Limitations

NAND Storage Block Diagram



NAND Storage Block Diagram



FreeBSD
Block Layer

CAM -> Controller -> SATA

SSD

Flash Translation Layer

Translate LBA (logical) to NAND Addr(physical)

Manages free space

Manages wear leveling / health

Most IP of flash storage products here

- Hybrid designs (HSSD)
- Flash data center (caching and distributed)

NAND Basics

NAND is a collection of cells.

NAND cell stores data (1-3 bits)

Groups of cells are a page (min read/write)

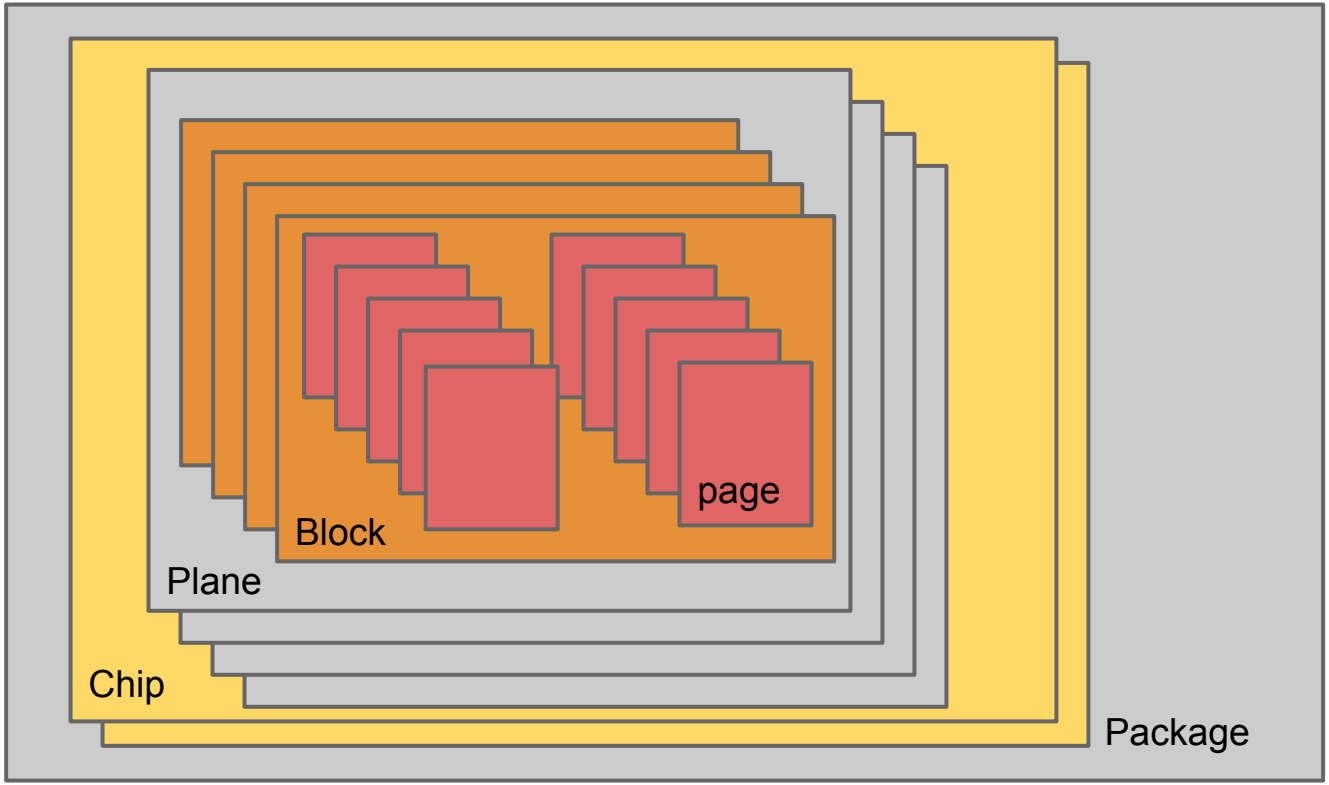
Groups of pages are a block (min erase)

Groups of blocks are a plane

Groups of planes are a chip

Groups of chips are a package

NAND Hierarchy



NAND Cells

Determines Speed, Reliability and Endurance

Stores 1, 2 or 3 bits per cell

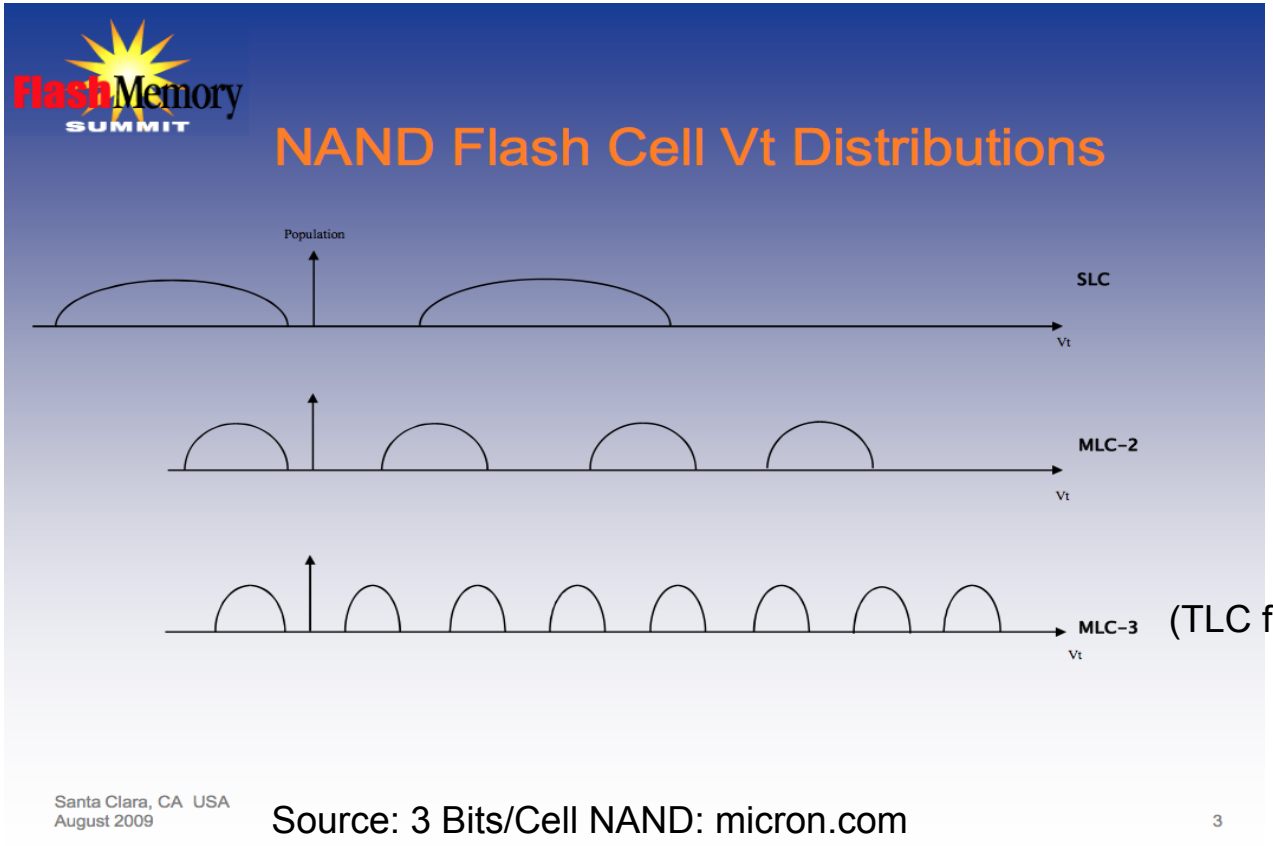
SLC - 1 bit

MLC or MLC-2 - 2 bits

TLC or MLC-3 - 3 bits

Each bit maps to a different NAND page

NAND Cells



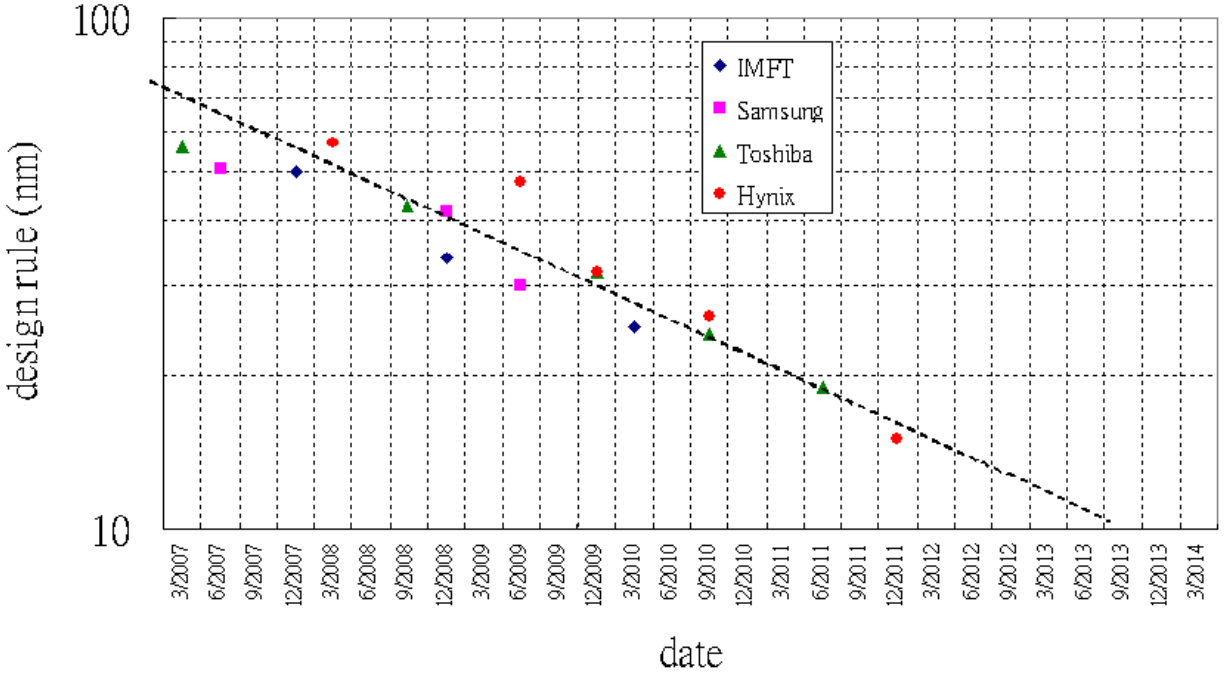
(TLC for this talk)

Source: 3 Bits/Cell NAND: micron.com

NAND Cell comparison

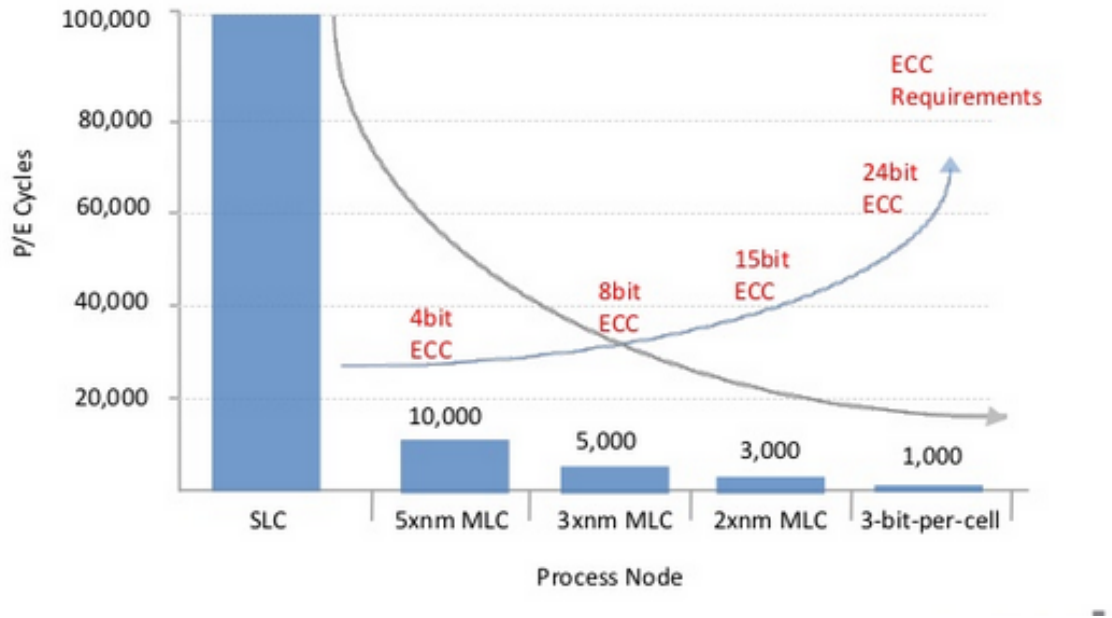
	SLC	MLC	TLC
Read Speed	Fast (25us)	Medium (50us)	Slow (90us)
Endurance	Excellent (100k)	OK (3k)	Poor (500)
Density	Poor	OK	Good
Program	Fast (500us)	OK (1-2ms)	Slow (2-4ms)

NAND Trends (getting smaller)



Source: Guiding Light at English Wikipedia
http://en.wikipedia.org/wiki/File:NAND_scaling_timeline.png

NAND Trends (and crappier)



Source: A new era in embedded flash memory (Anobit)

http://www.slideshare.net/Anobit/a-new-era-in-embedded-flash-memory-anobit-presentation-fms-2011

NAND Error Sources

Retention (left shift)

- Cells lose charge, reducing potential

Read Disturb (right shift)

- Inhibits on reads adds charge to other pages

Program/Erase Cycle (right shift)

- Traps charges
- Damages insulation microscopically

NAND Pages

Minimum read/write unit

Extra bytes for ECC to correct errors

Some NAND can do sub-page reads

Some SLC can do partial pages

MLC/TLC must be programmed in order

Must be erased before programming

1k to 16k (newer generations are larger)

NAND Blocks

Minimum Erase Unit (all pages erased)

64-256 pages per block (more newer)

Basic unit of device/file system log

Much larger than system (512/4k) blocks

Erase time long (2-5ms), especially at EOL

NAND Planes

Multiple planes may allow parallelism

1-4 planes typical

Often consume one or two address bits

NAND Chips and Packages

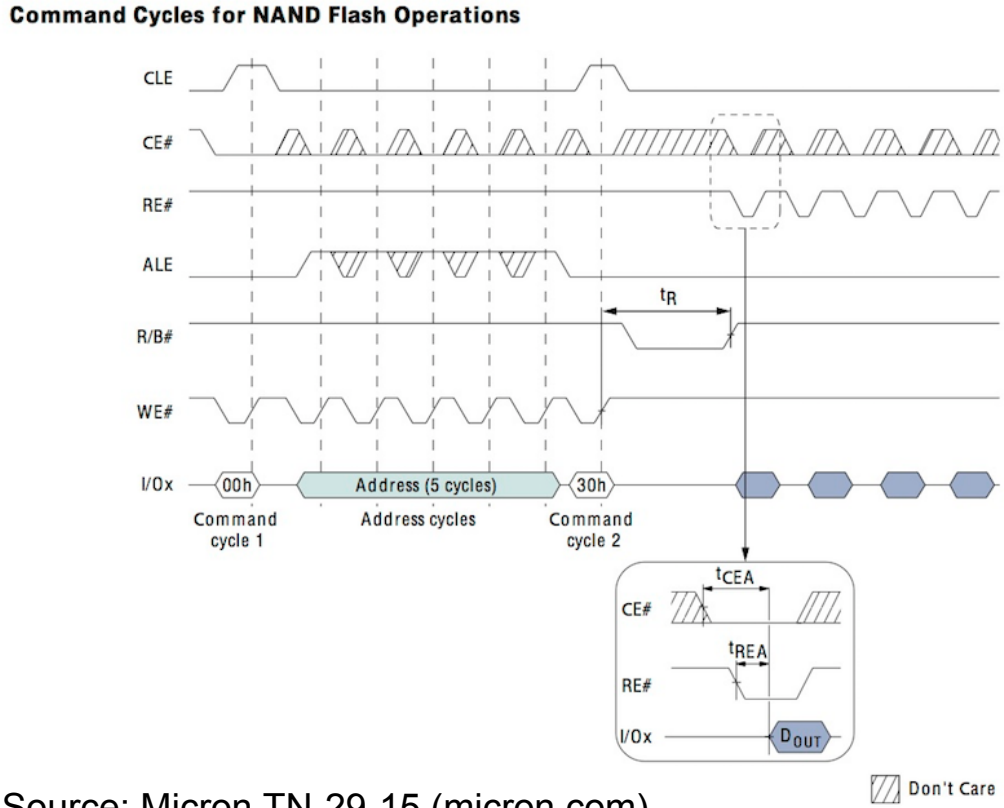
1-4 planes per chip

1-16 chips per package (usually on independent CEs, but sometimes shared)

Interfaces to CPU at this level

More CEs usually means more performance

NAND Interface

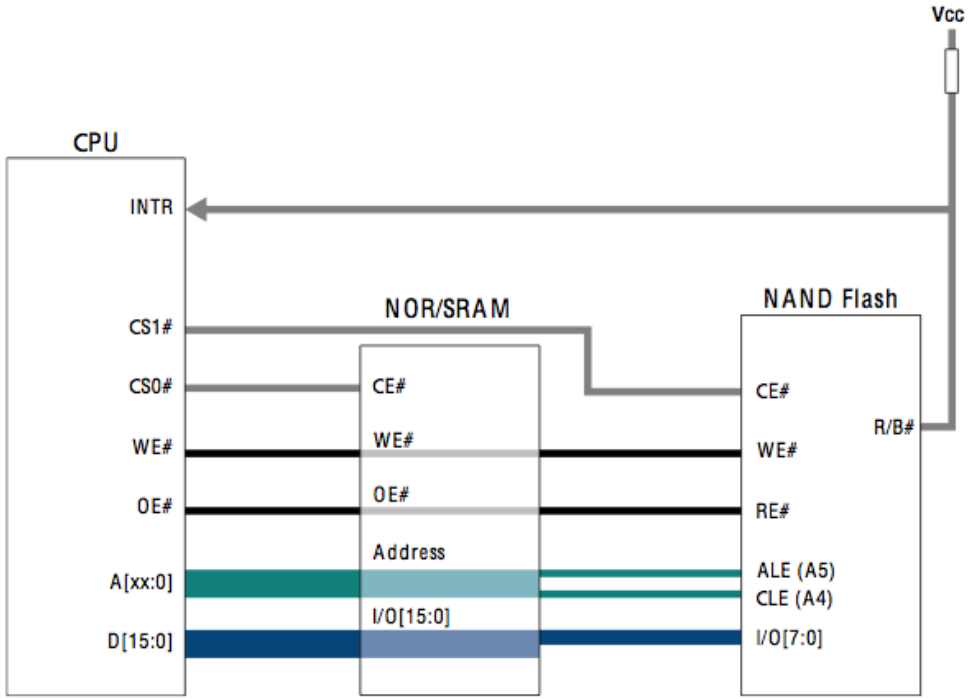


Typical Read operation

tR ~ 30-50 µs
tCEA ~ 45 ns
tREA ~ 35 ns

Source: Micron TN-29-15 (micron.com)

Simple NAND Integration

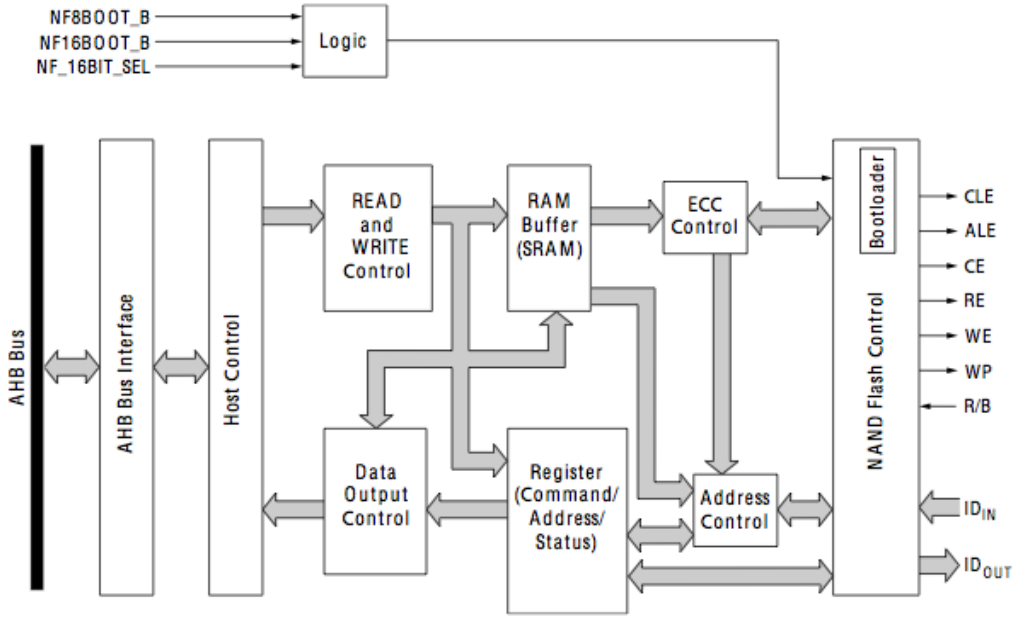


Source: Micron TN-29-19 (micron.com)

Complex NAND Integration

Built-in NAND Flash Interface on Freescale i.MX21 Processor

Go to page 23



Note: Image courtesy of Freescale Semiconductor, Inc.

Source: Micron TN-29-19 (micron.com)

Log Structure

New data appended to end (logical tags)

Compused of many segments in order

Translation between logical to physical needed

Need to GC older, emptier segments

Extra writes generated by system

Log structure terms

WA - Write Ampliphication: Factor by which physical writes are bigger than logical writes

GC - Garbage Collection: Reclaiming blocks that are mostly empty by copying data forward

Log Structure Example

6 extents (B1 - B6)

Each extent holds 10 blocks

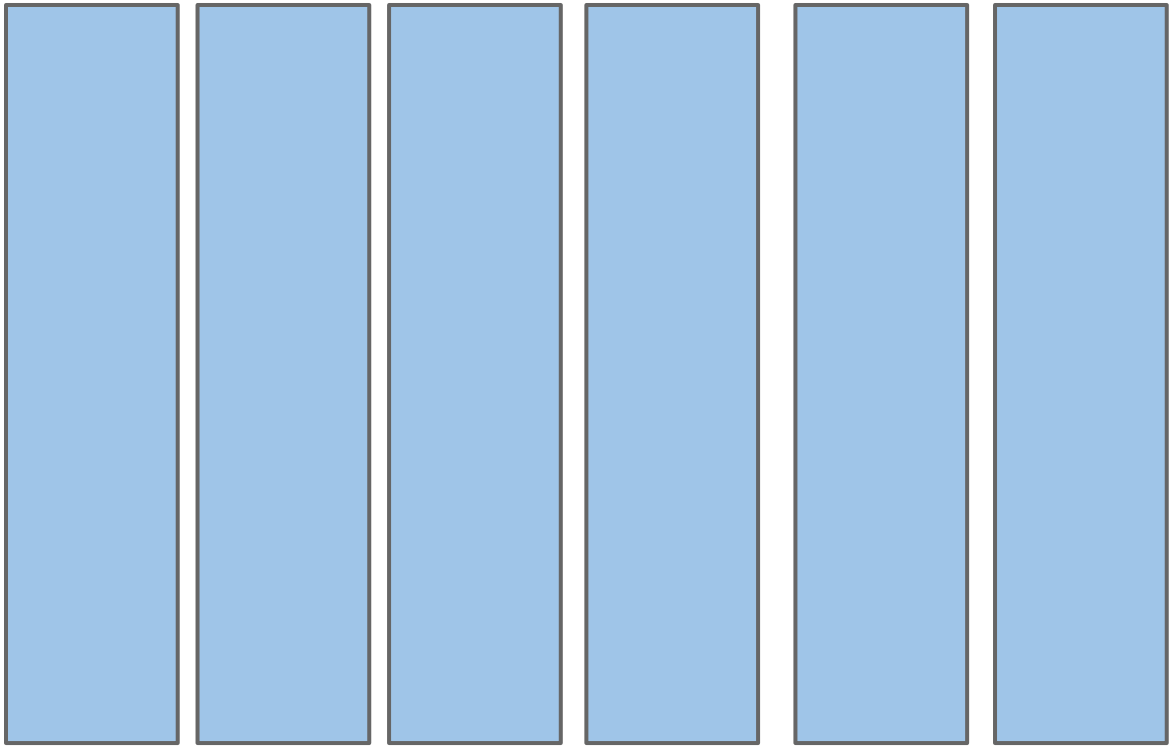
Log capacity 40 LBAs (0-39)

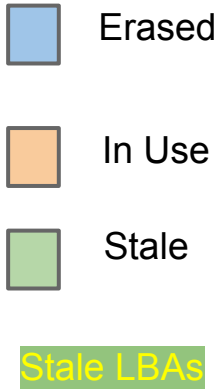
    Implies 2 extents are spare

Metadata omitted for simpicty

Simple writing example to illustrate WA and GC

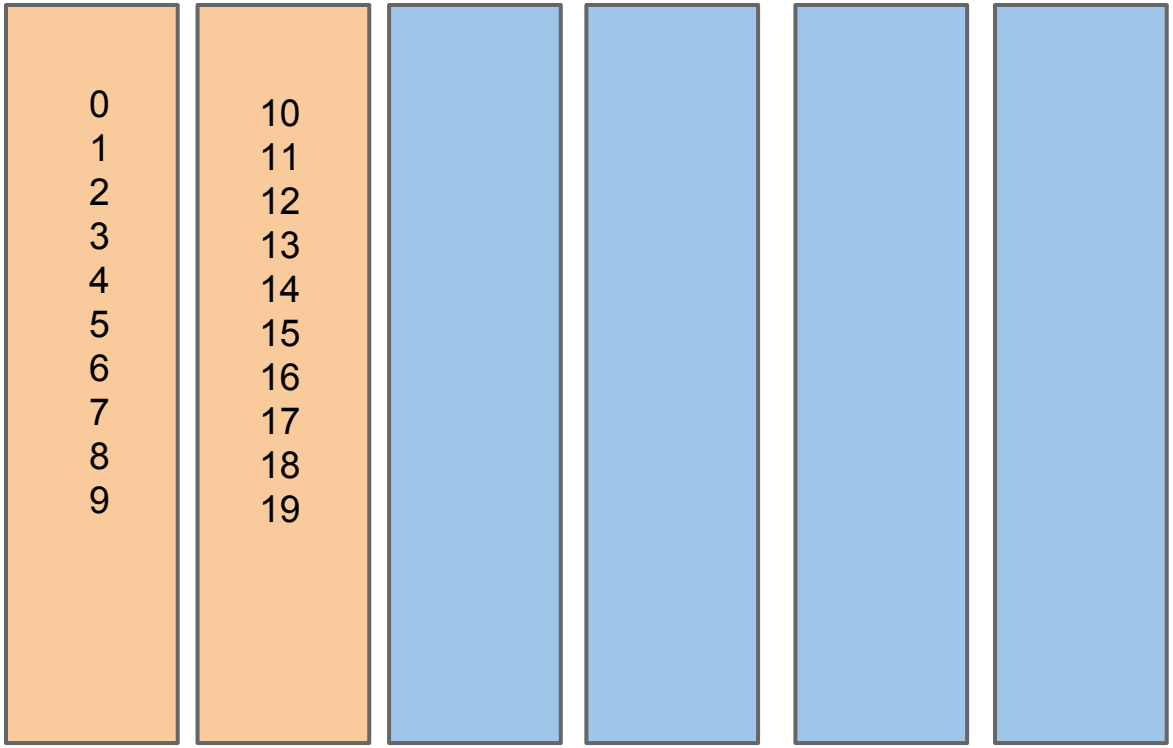
Log Structure Example





Initial state: Empty Log

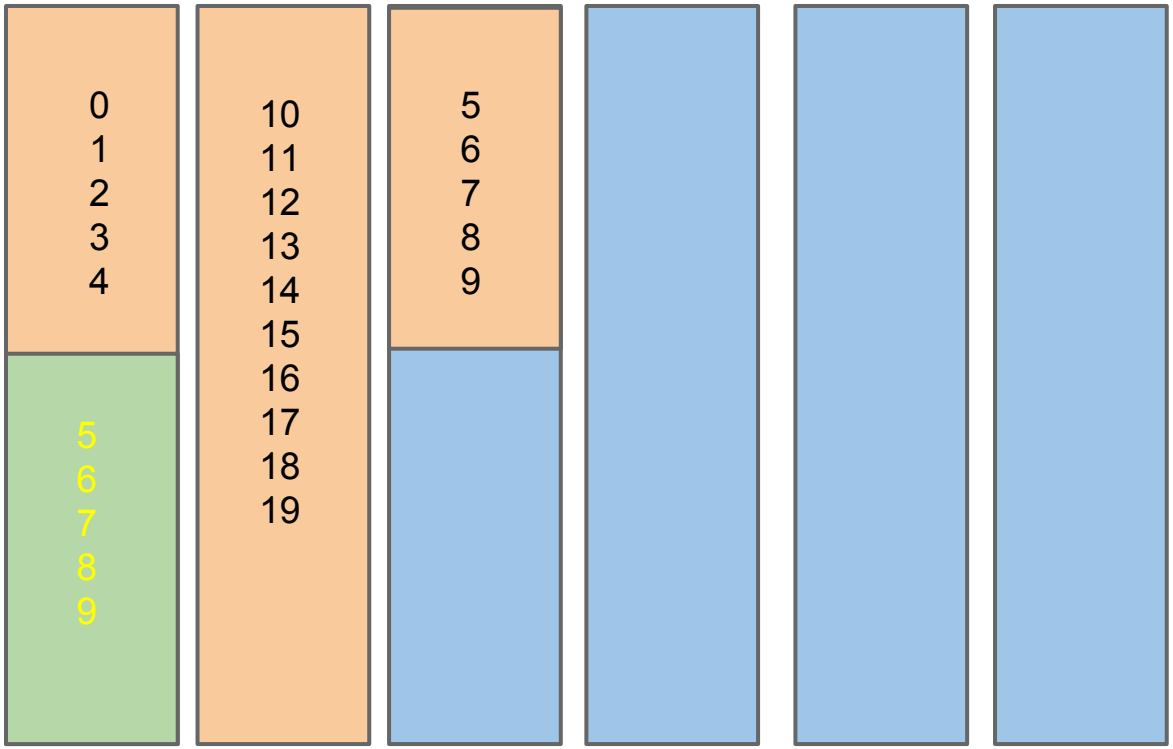
Log Structure Example

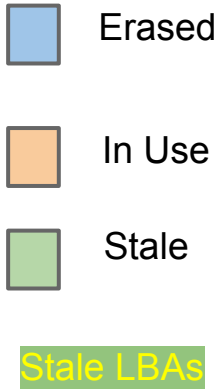


Stale LBAs

Write LBAs 0-19

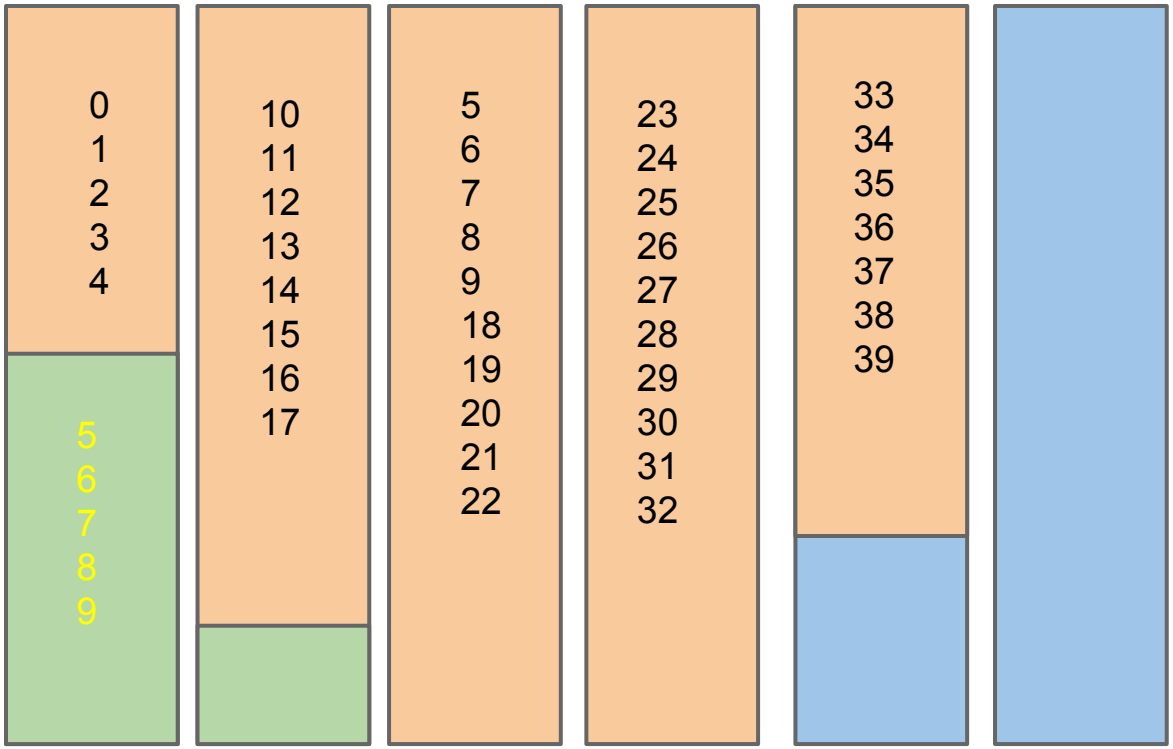
Log Structure Example



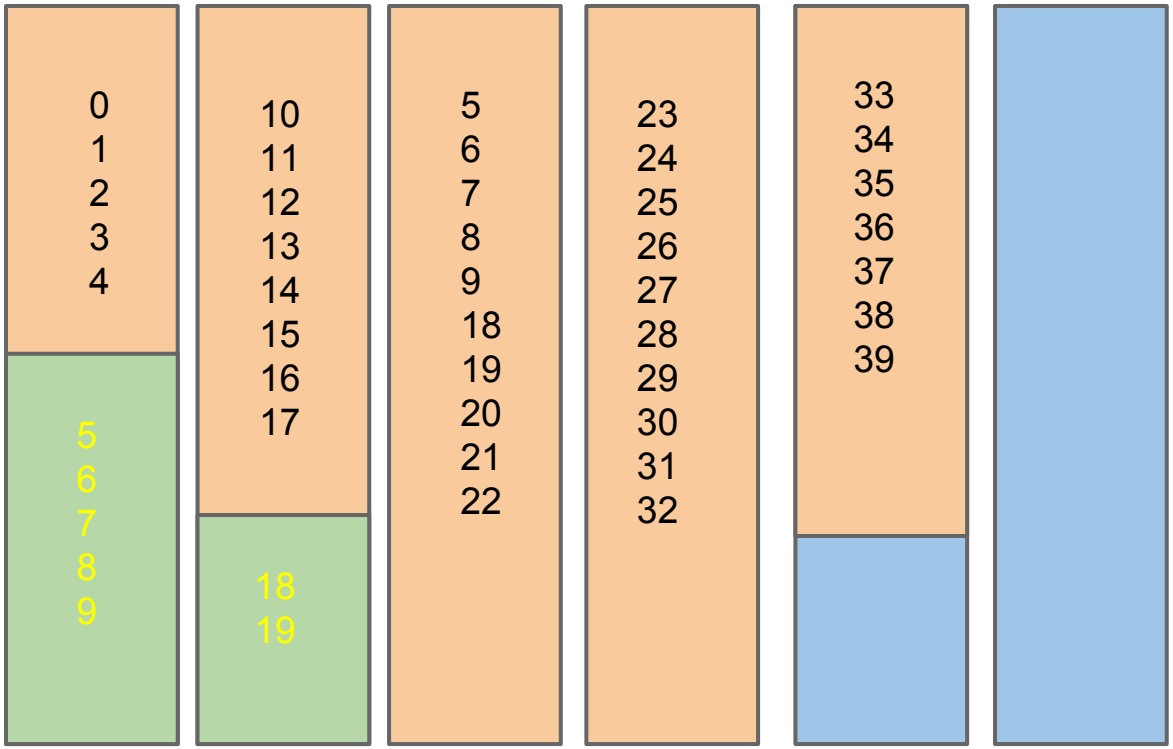


Write LBAs 5-9

Log Structure Example



Log Structure Example



Erased

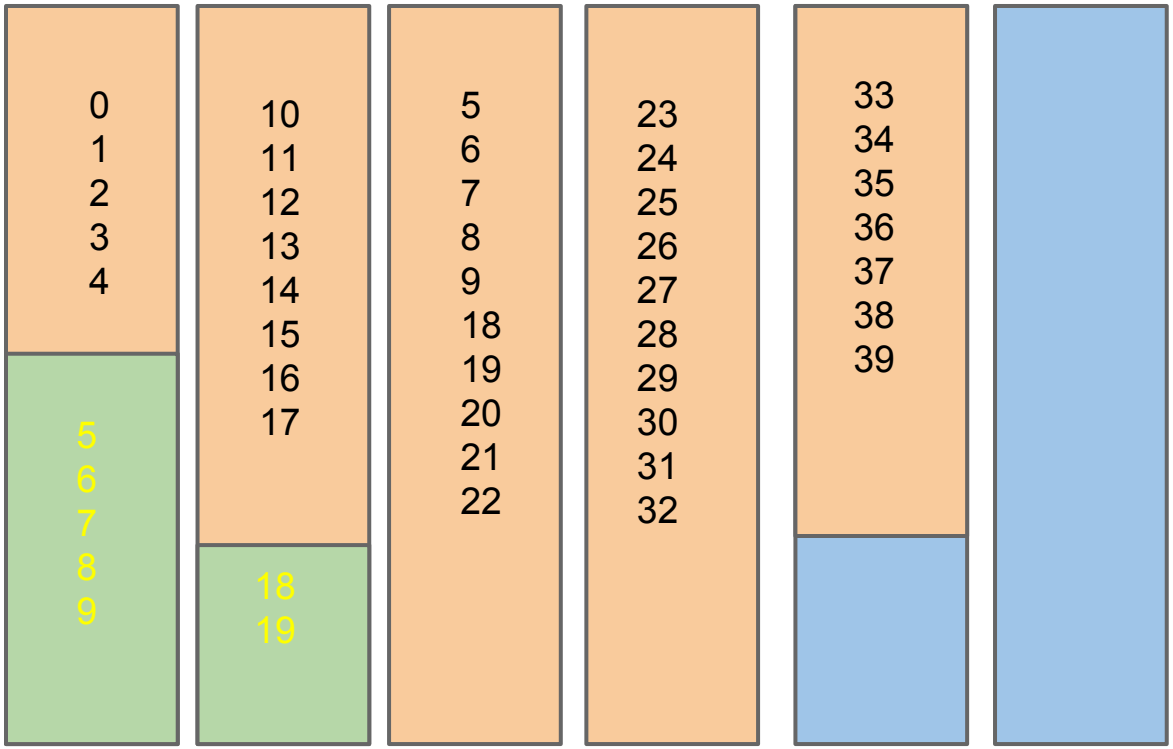
In Use

Stale

Stale LBAs

Want to write LBA 12-29

Log Structure Example



Erased

In Use

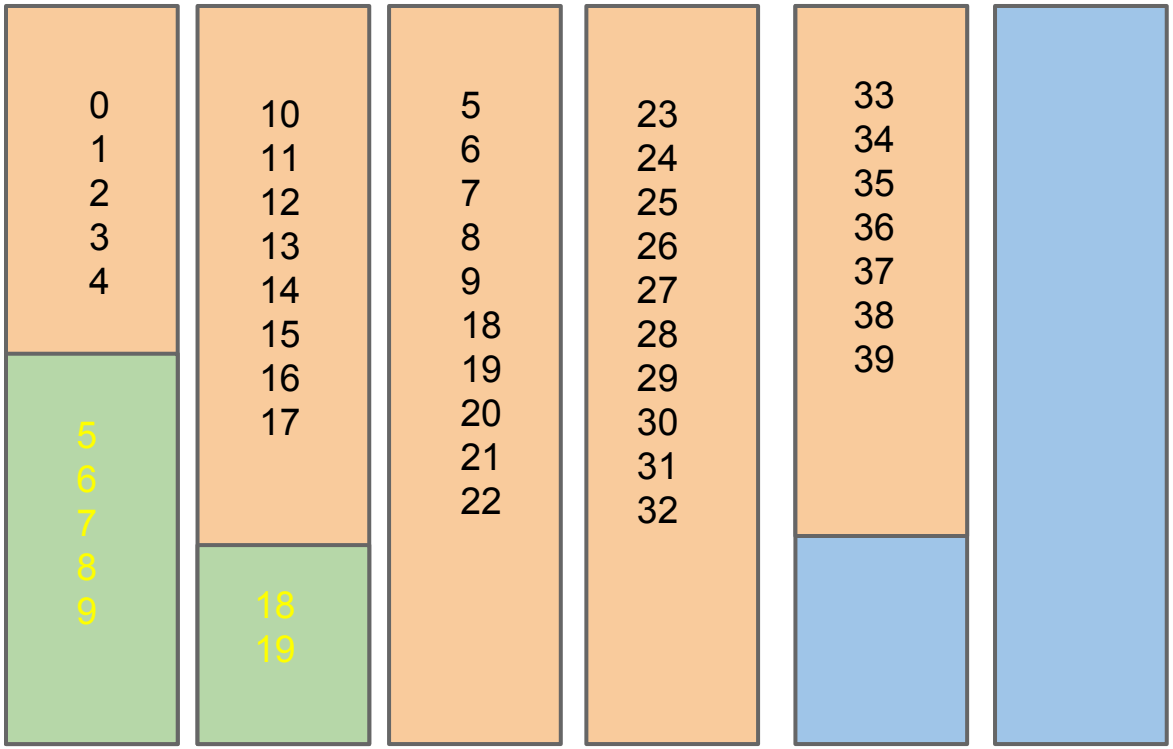
Stale

Stale LBAs

Want to write LBA 12-29

NOT ENOUGH SPACE

Log Structure Example



Erased

In Use

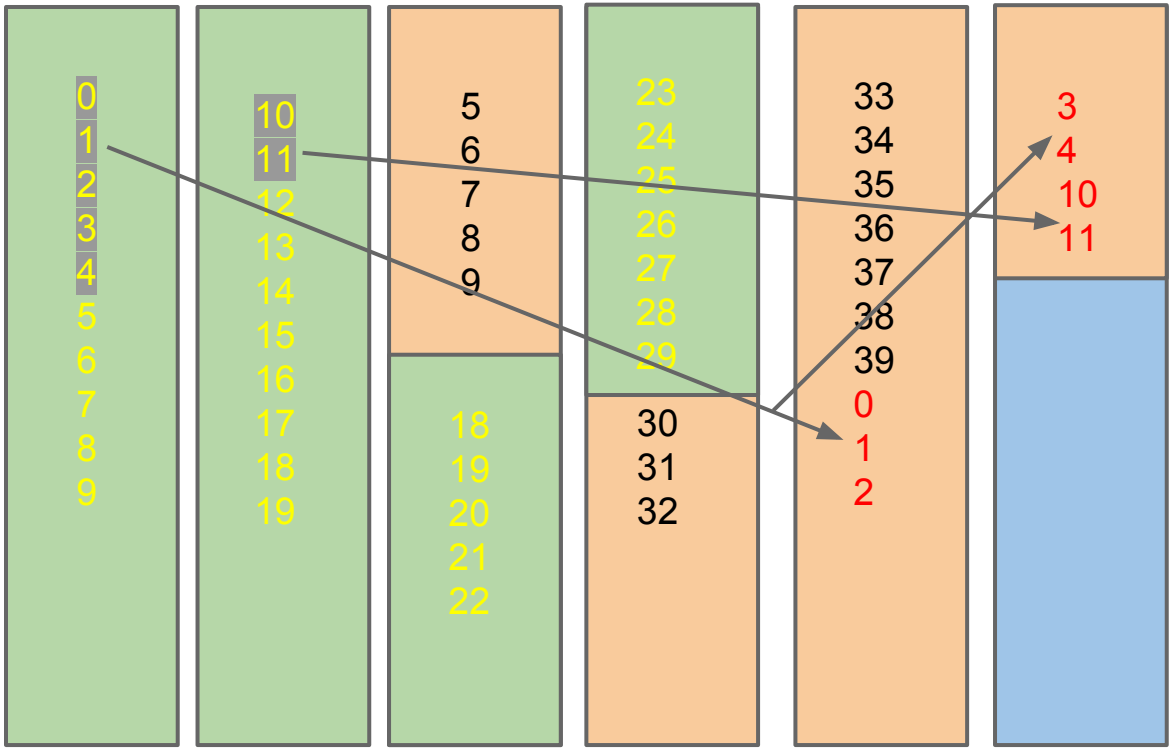
Stale

Stale LBAs

Want to write LBA 12-29

GC Forward 0-4 and 10-11

Log Structure Example



Erased

In Use

Stale

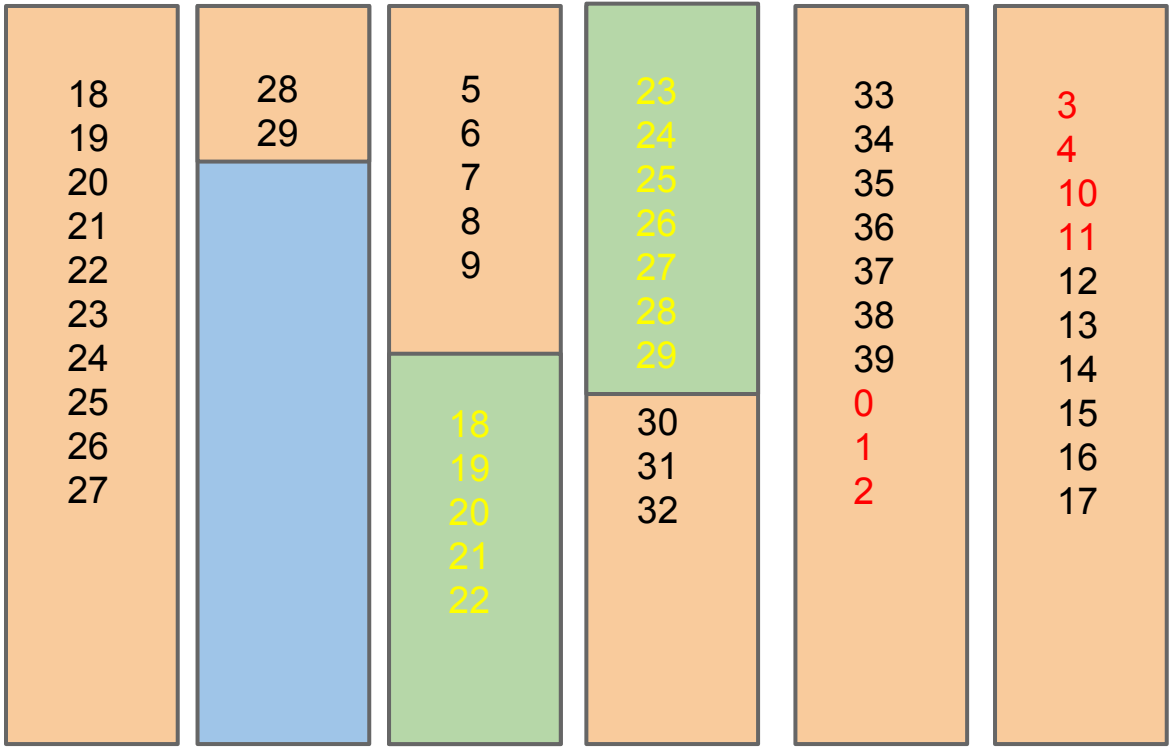
Moved LBAs

Stale LBAs

Want to write LBA 12-29

GC Forward 0-4 and 10-11,
Invalidate (TRIM) 12-29
Head moves to B3

Log Structure Example



Erased

In Use

Stale

Moved LBAs

Stale LBAs

Want to write LBA 12-29

Erase B1 and B2, do write

Note: Log now starts at B3
Had to write 7 extra blocks
(WA = 72 / 65 = 1.1)

High WA Effects

Higher latency (writes take a long time and block reads)

Reduced endurance (extra writes cause extra wear)

Lower bandwidth (writes block other operations)

---> Poor Performance

Avoiding High WA

Large writes (more efficient FTL)

Enable TRIMs (less GC needed)

Avoid “checker boarding”

Use a log-based FS with segments that match storage devices segments

NAND Management

Wear Leveling

Error correction

Error avoidance

Error recovery

Bad block management

Endurance management

NAND Management and GC

GC triggered to recover space

GC also triggered when NAND is bad

- too high ECC correction rate
- too many reads
- other events (chip failure, charge pump, etc)

Types of NAND Storage Devices

Managed Devices

- Presents logical view to host
- Does all translation and NAND management

Unmanaged Devices

- Presents physical view to host
- Host does some or all NAND management

Managed Devices

SSDs

Thumb Drives

SD Cards

NVMe Cards

Some raw flash parts

Unmanaged Devices

Raw NAND

Some PCIe cards

Many Hybrid NAND parts

- Host does logical to physical translation
- NAND does ECC offload

FreeBSD Support

Most managed devices

- CAM for disks (SSDs, Thumb Drives)
- Custom drivers for RAID cards
- mmcsd stack (SD Cards)

FreeBSD Features

TRIM support

Large block size (UFS)

Log structure (ZFS)

GEOM direct dispatch

Typically devices very good at emulating fast disks

FreeBSD SSD Optimal Performance

Align partitions to large boundary (64k, 128k)

Ensure stripes are similar sized for RAID

Ask drive vendor for optimal block sizes

Use vendor block size for UFS/ZFS

Test variety of settings with specific workload

SSD Danger Zones

Understand read and writes interact

Even relatively small writes can affect reads

- Recall relatively long tPROG

Lots of reads can affect write performance

- Recall read disturb triggers GC

Misaligned (in LBA space) or odd sizes tax FTL

Unmanaged NAND on FreeBSD

nand(4) driver

nandfs(4) driver

SoC specific mid-layer

WITH_NAND=y

Preliminary and Experimental

Questions

Warner Losh

imp@freebsd.org

wlosh@netflix.com

http://people.freebsd.org/~imp/bsdcan2014.pdf