ICAC TRAINING & TECHNICAL ASSISTANCE

Solid State Devices and Forensics

Gary C. Kessler
April 21, 2015

iCAC
Internet Crimes Against Children
Training & Technical Assistance Program

NCJTC
National Criminal Justice Training Center
Fox Valley Technical College

Webinar Information

This webinar is supported by grant 2013-MC-FX-K104, provided by the Office of Juvenile Justice and Delinquency Prevention (OJJDP), and is brought to you by the ICAC Training & Technical Assistance Program. Points of view or opinions in this document are those of the author and do not necessarily represent the official position or policies of the U.S. Department of Justice or Fox Valley Technical College.

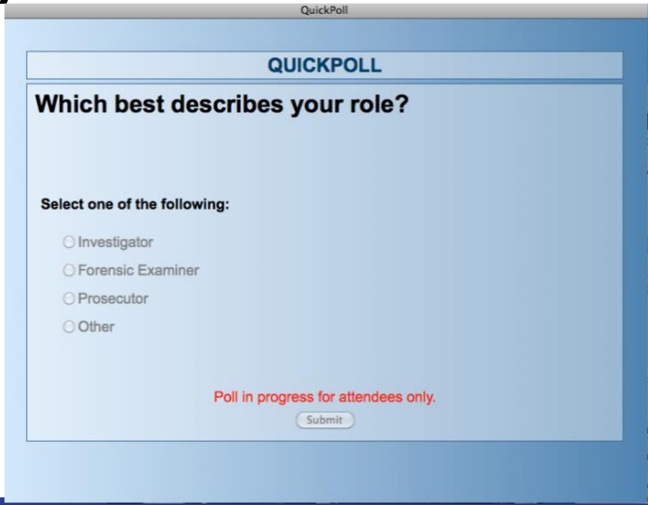
ICAC Training & Technical Assistance is a program of the Fox Valley Technical College-National Criminal Justice Training Center (NCJTC).

During the Webinar

- All attendees will be muted.
- If you desire to ask a question, please use the questions section of the GoToWebinar dialogue box, typically on the right side of your screen.
- Please do not raise your hand for questions, we can not unmute you.
- Questions will either be answered directly by a panelist or asked to the presenter who will answer.

Poll Questions

- Poll questions may be asked during the webinar. They are asked so we can better understand the audience and provide the most useful information to you.
- As they will only be open a short period of time, please respond promptly.



Post Webinar Information

- At the conclusion of the webinar, a short survey will appear. We ask that you complete the survey in an effort to gather information to better serve the community in preparation for future webinars. Please complete it before signing off.

- You will receive a link to access our law enforcement only webinar library where you can view the recording and access related webinar material. Due to the sensitivity of some of the material you must be a registered law enforcement member of the *NCJTC.org* or *ICACTaskforce.org* websites. If you are not currently a member, you will need to register for access at www.ncjtc.org.

Author Contact Information

Gary C. Kessler, Ph.D., CCE, CCFP, CISSP
Gary Kessler Associates
Ormond Beach, FL

mobile: +1 802-238-8913
office: +1 386-226-7947
e-mail: gck@garykessler.net
gary.kessler@erau.edu
Skype: *gary.c.kessler*

http://www.garykessler.net

North Florida ICAC/Volusia Co. Sheriff
Embry-Riddle Aeronautical University

Overview

- **GOAL:** *Introduce solid state device technology and assist the forensic examiner to understand SSD's impact on forensics and digital evidence*

- Introduction to solid state devices
- SSD terms, concepts, and operation
- Impact on digital forensics
- Conclusion

Acronyms and Abbreviations

µs	Microsecond; millionths (10^{-6}) of seconds
16LC	16-level cell
b	Bit
B	Byte (8 bits)
CHS	Cylinder, head, sector
exFAT	Extended File Allocation Table
ext3/ext4	Third/Fourth extended file system
F2FS	Flash-Friendly File System
FAT32	File Allocation Table (32-bit) file system
FTL	Flash Translation Layer
GB	Gigabyte; billions (2^{30}) of bytes
GC	Garbage collection
HDD	Hard disk drive
HFS	Hierarchical File System (*aka* Mac OS Standard)
HFS+	HFS Plus (*aka* Mac OS Extended)
I/O	Input/output
KB	Kilobyte; thousands (2^{10}) of bytes
LBA	Logical Block Addressing
MBps	Megabytes per second; millions (2^{20}) of bytes
MLC	Multi-level cell
ms	Millisecond; thousandths (10^{-3}) of seconds
NAND	Not AND
NTFS	New Technology File System
OP	Over provisioning
OS	Operating system
P/E	Program/erase
PBA	Physical block addressing
RAID	Redundant array of independent (inexpensive) disks
SATA	Serial Advanced Technology Attachment
SSD	Solid state device (*or* disk)
SLC	Single-level cell
TLC	Tri-level cell
TSOP	Thin small outline package
UFS	Unix File System
V	Volts
YAFFS	Yet Another Flash File System 2

Solid State Devices

Poll question #2

Why Study SSDs?

Really, Why Study SSDs?

- No, the sky is ***not*** falling...
  - But things are changing and we will need to wear helmets!
- Solid state devices (SSDs) operate very differently from traditional spinning hard disk drives (HDDs) and this will affect our forensic analysis
  - Unallocated space will not yield reliable and complete information
  - Inferences cannot be drawn about a wiped drive
  - Data loss can occur on an SSD without a command by the computer, arguably causing loss of evidence

Why Do You Need to Know This?

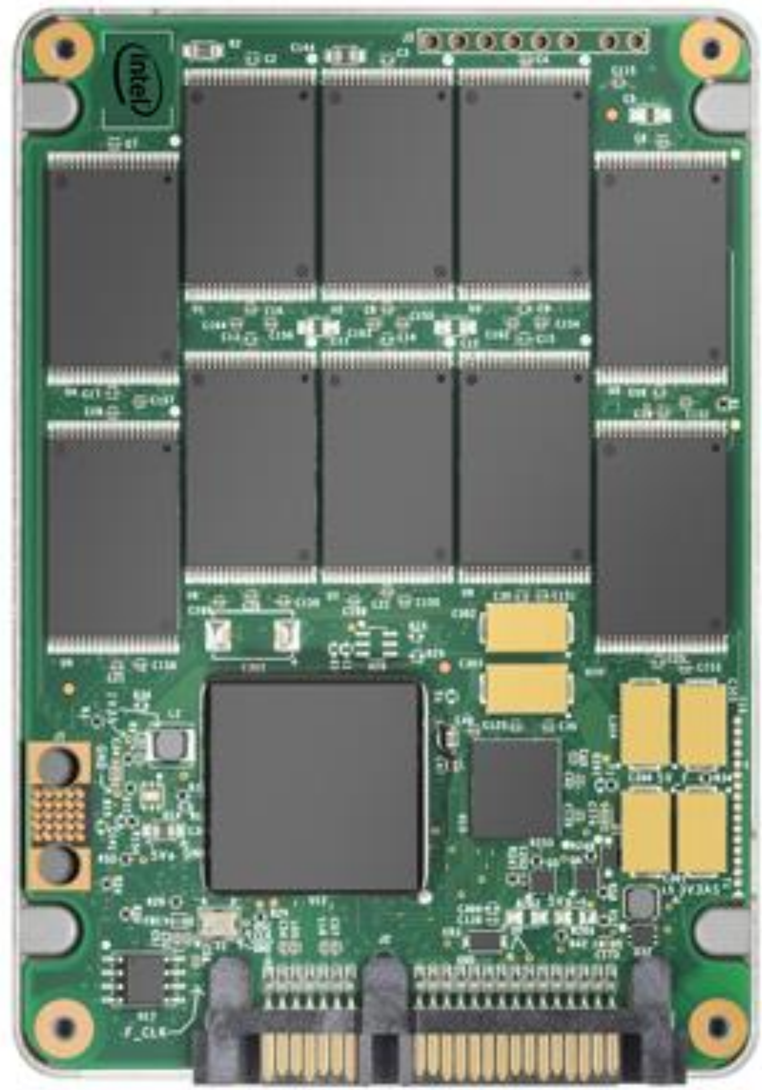
- More and more SSDs are in use
- A digital forensics examiner might have to:
  - Explain why there is a hash mismatch during imaging but that the data is still reliable
  - Why deleted files can't be found in unallocated space
  - Why a judge's order to delete certain files and spaces on a drive cannot be carried out
  - Determine the limitation of current forensic tools

What Is An SSD?

- SSDs are storage devices based on NAND flash memory
  - Thumb drives
  - Hard drive replacements

- Instead of spinning platters, SSDs have some number of NAND memory chips on a board
- SSDs are packaged as HDD replacements
  - E.g., standard 2½" and 3½" drives with a SATA interface

HDD and SSD



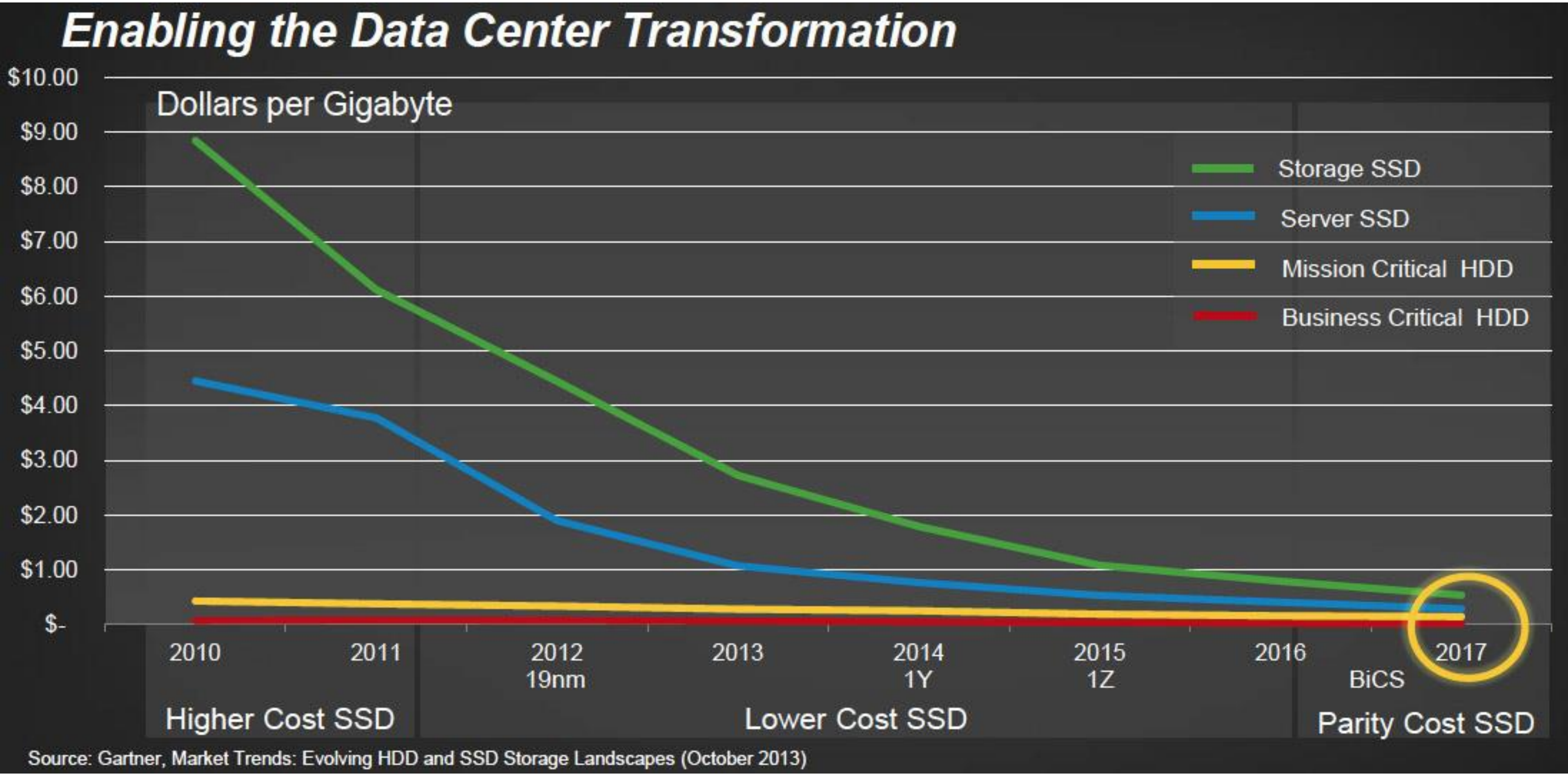
Why Use SSDs?

- Pros
  - They're cool
    - I.e., the latest and greatest
  - Low power
  - Fast I/O
- Cons
  - Expensive
    - But prices are dropping
  - Lower capacity than HDDs
    - But catching up...

Cost and Performance

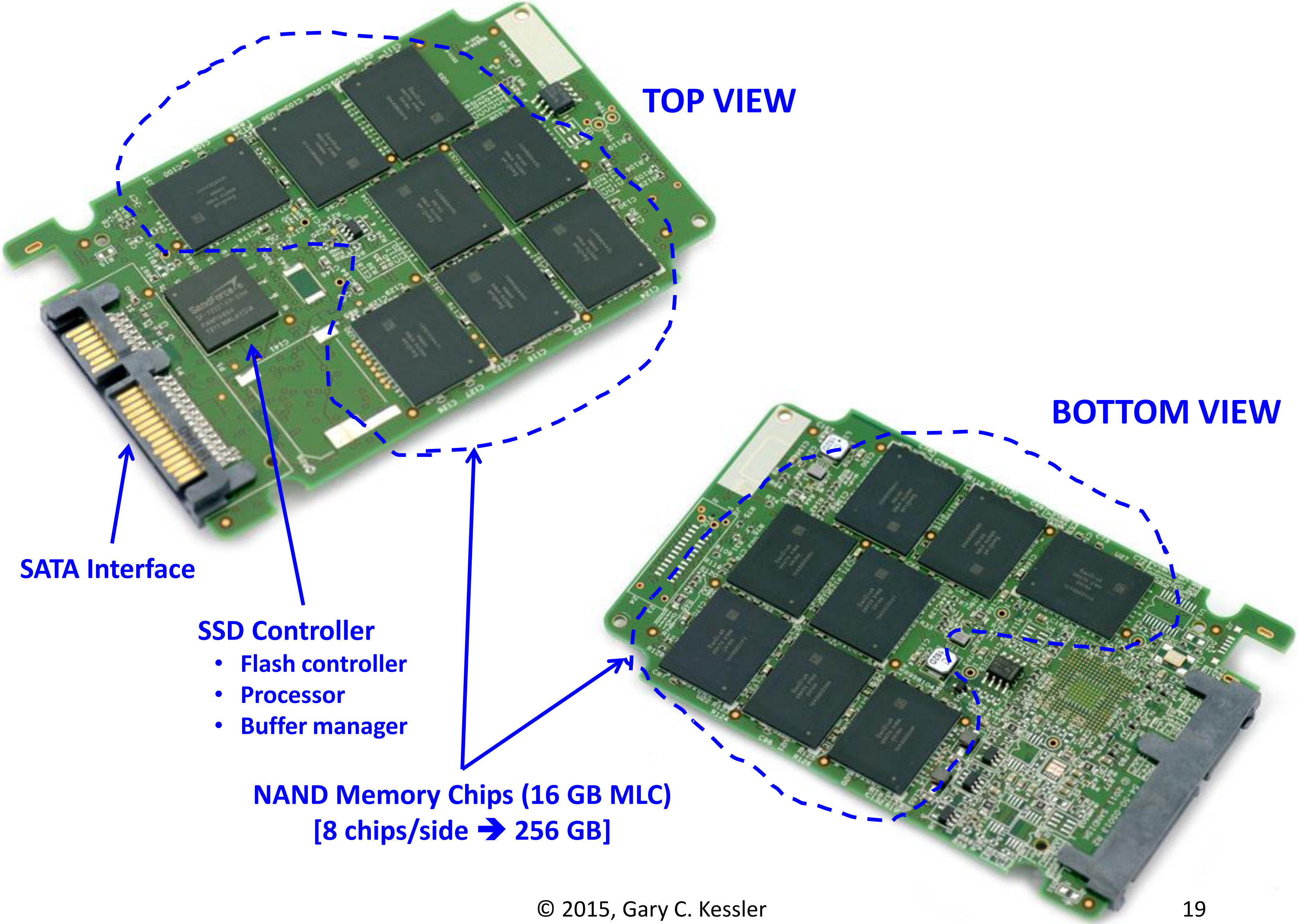
- Cost
  - SSD <40¢/GB and dropping
  - HDD 10¢/GB (2½"), 5¢/GB (3½"), and stable
- Performance
  - SSD read/write latency is lower and data transfer rates higher than HDDs
    - Typical SSD: 10 µs read latency, 100 µs write latency, 275 MBps transfer rate
    - Typical HDD: 3-10 ms read/write latency, 150 MBps transfer rate

Cost per Gigabyte

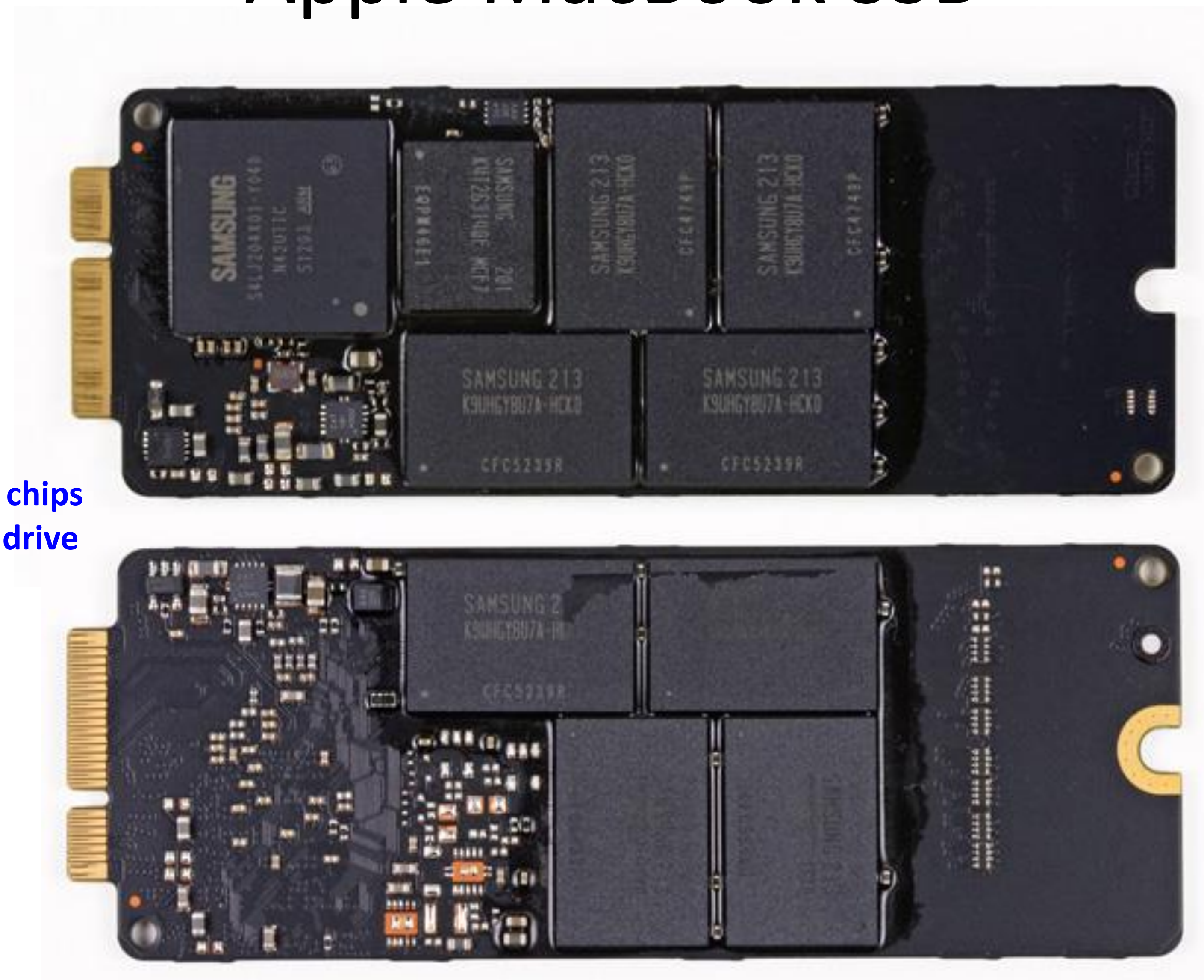


Example: SanDisk Ultra SSD 240 GB

NOTE: No CHS geometry information on the case...


TOP VIEW
BOTTOM VIEW
SATA Interface
SSD Controller
• Flash controller
• Processor
• Buffer manager
NAND Memory Chips (16 GB MLC)
[8 chips/side → 256 GB]

Apple MacBook SSD



64 GB MLC chips
→ 512 GB drive

Techie Side Note

- Two types of memory chip: NAND and NOR
  - Non-volatile
  - Electrically erasable programmable read-only memory (EEPROM, aka E^2PROM)
  - Must be erased before writing

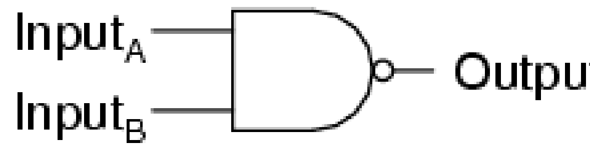
- NOR (Not OR)
  - Random access
  - Long erase times
  - ~100,000 erase/write cycles

- NAND (Not AND)
  - Sequential read/write
  - Short erase time
  - ~1,000,000 erase/write cycles

NAND Techie Side Note

- NAND named for the *not AND* Boolean logic function
  - Output is false (0; 0 V) only if all inputs are true (1; 5 V)
  - *Functionally complete* because any Boolean function can be implemented using combinations of NAND gates

2-input NAND gate

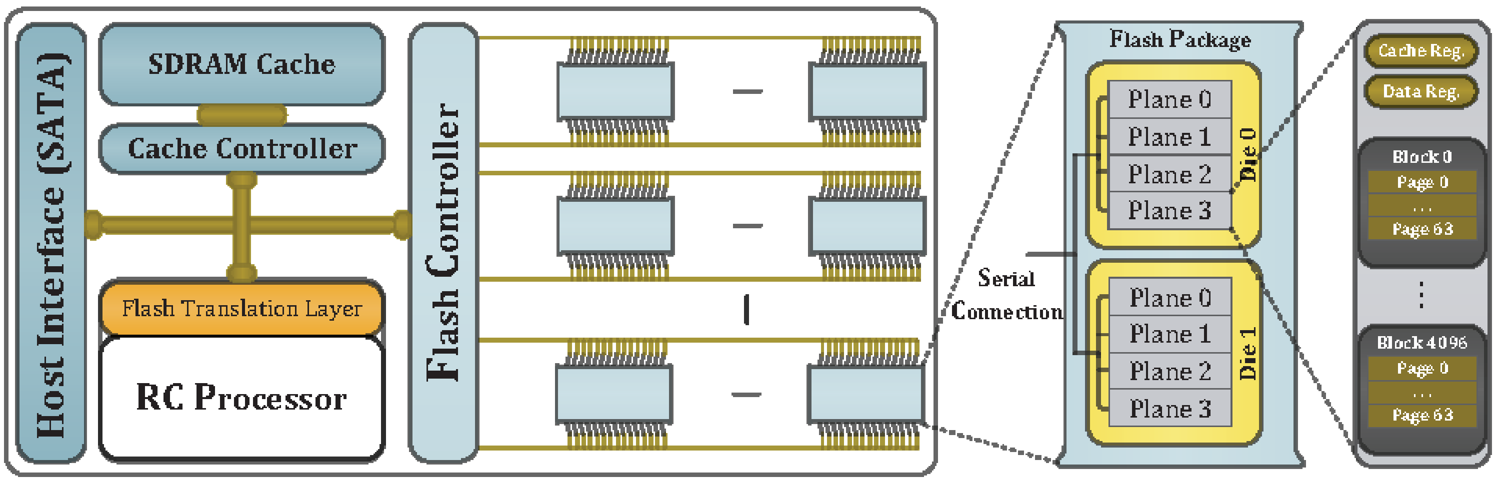


A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

File Systems on SSDs

- Traditional file systems that were designed for traditional HDDs still work on SSDs
  - E.g., ext3/ext4, FAT32, HFS/HFS+, NTFS, and UFS all work on SSDs
- Some file systems have been created specifically for flash memory devices
  - E.g., exFAT, F2FS, JFFS2, YAFFS2

Terms and Concepts



Poll question #3

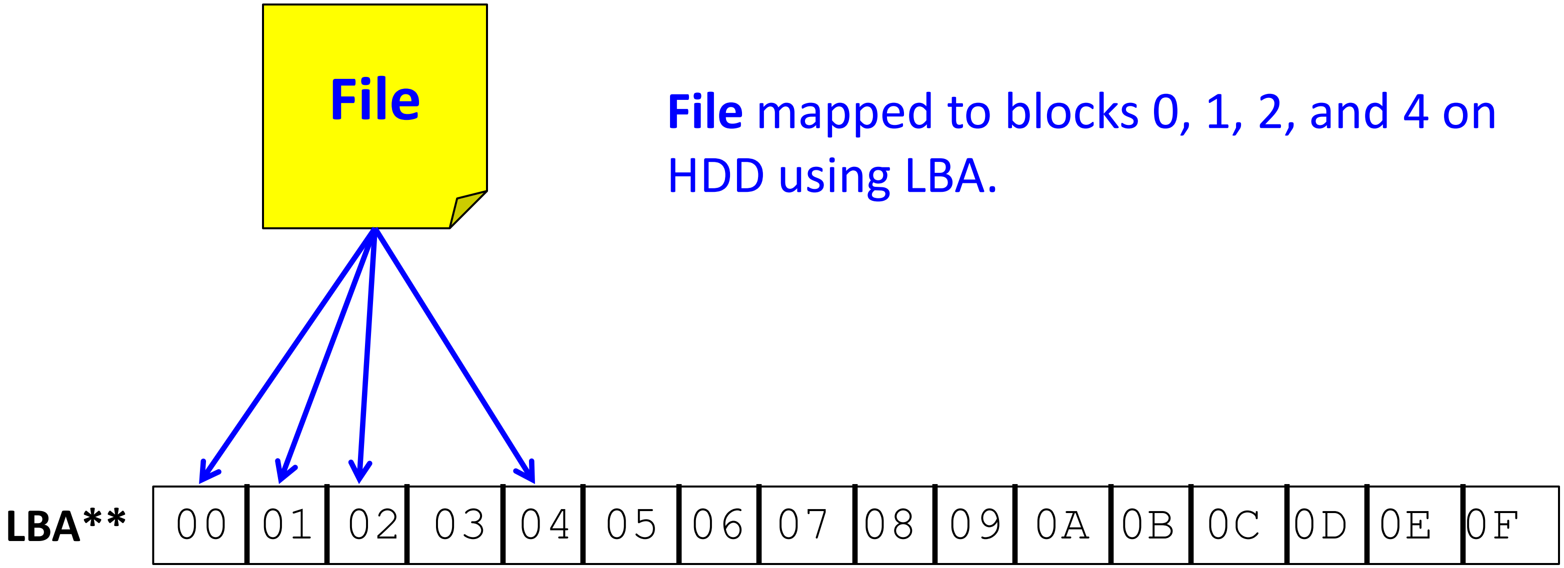
Terms, Terms, and More Terms...

- Addressing, Architecture, and Operation
  - Addressing
  - Page, block, and more!
  - Reading, writing, and erasing
  - Memory cell density
- Improving SSD performance
  - Wear leveling
  - Write amplification
  - Garbage collection
  - TRIM
  - Over provisioning

Addressing in HDD and SSD

- HDD
  - 1-1 mapping between logical and physical address
    - Disk drive controller translates logical block address (LBA) to cylinder/head/sector (CHS) address
  - Data does not move
- SSD
  - Memory cells wear out
  - No 1-1 address mapping
    - SSD's Flash Translation Layer (FTL) translates LBA to physical address on device
  - Data moves due to wear leveling, garbage collection, and over-provisioning

HDD LBA



File mapped to blocks 0, 1, 2, and 4 on HDD using LBA.

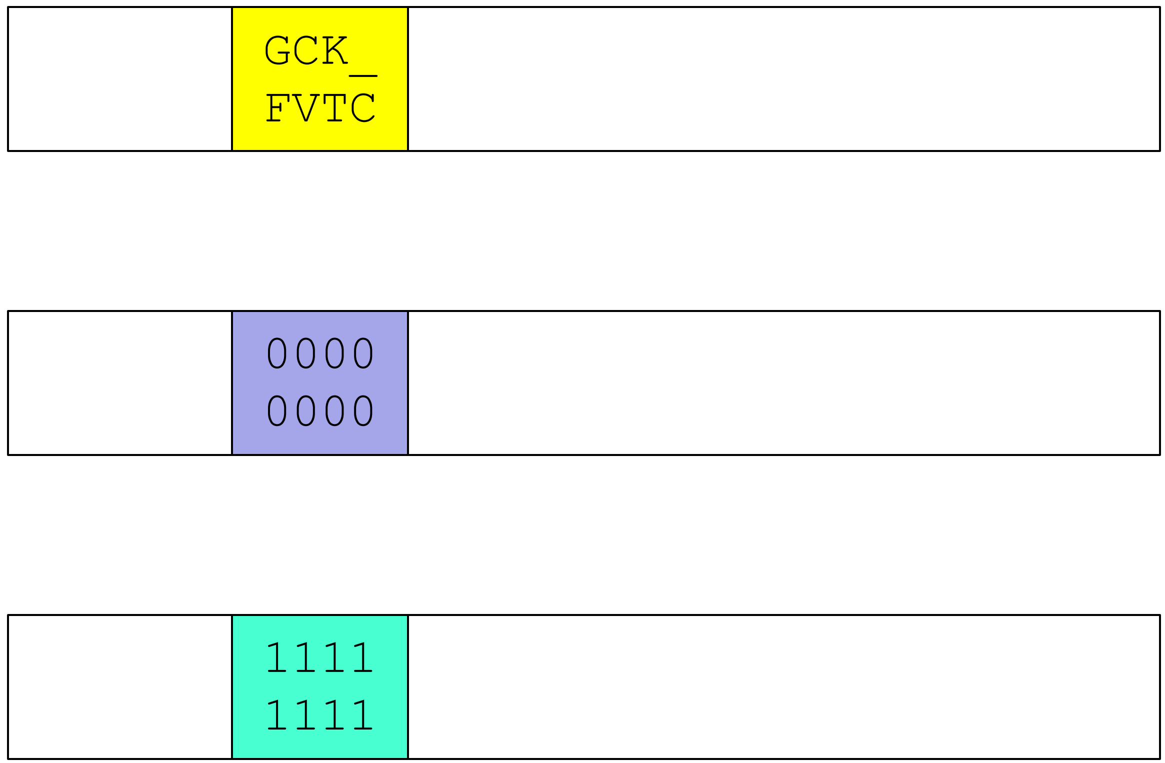
** LBA is translated to physical CHS addressing by the disk controller.

Data Doesn't Move On An HDD

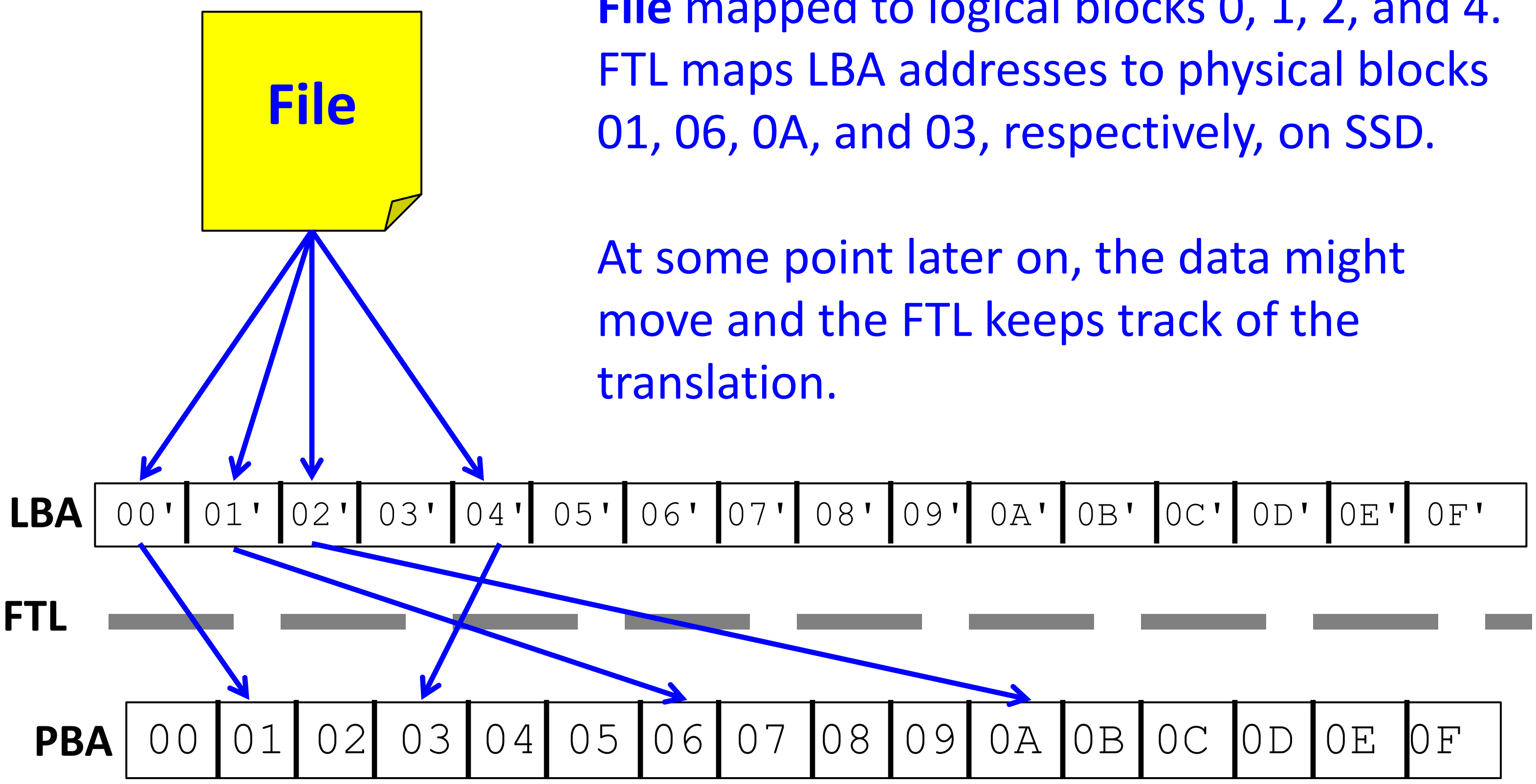
1) Write "GCK_FVTC" to LBA 0x4000

2) Write 0s to LBA 0x4000

3) Write 1s to LBA 0x4000



SSD LBA-PBA Translation



File mapped to logical blocks 0, 1, 2, and 4. FTL maps LBA addresses to physical blocks 01, 06, 0A, and 03, respectively, on SSD.

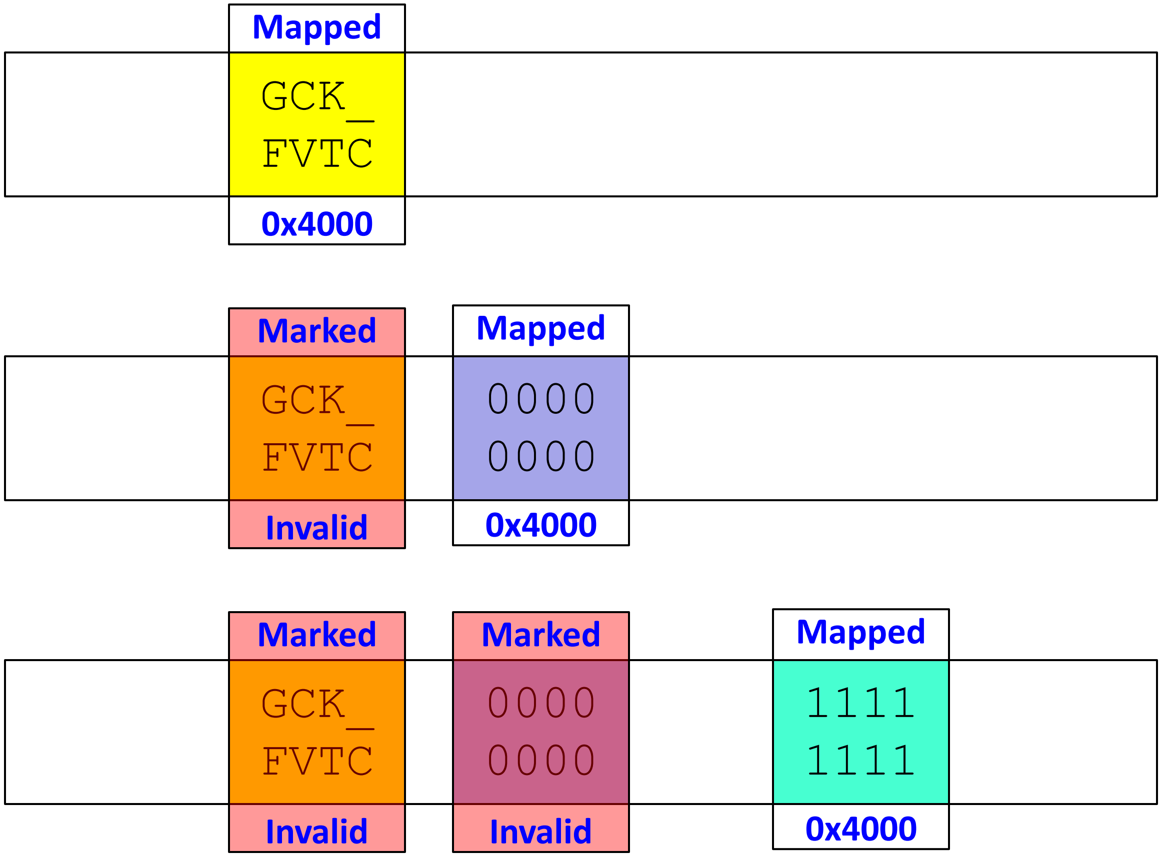
At some point later on, the data might move and the FTL keeps track of the translation.

Data Moves On An SSD

1) Write "GCK_FVTC" to LBA 0x4000

2) Write 0s to LBA 0x4000

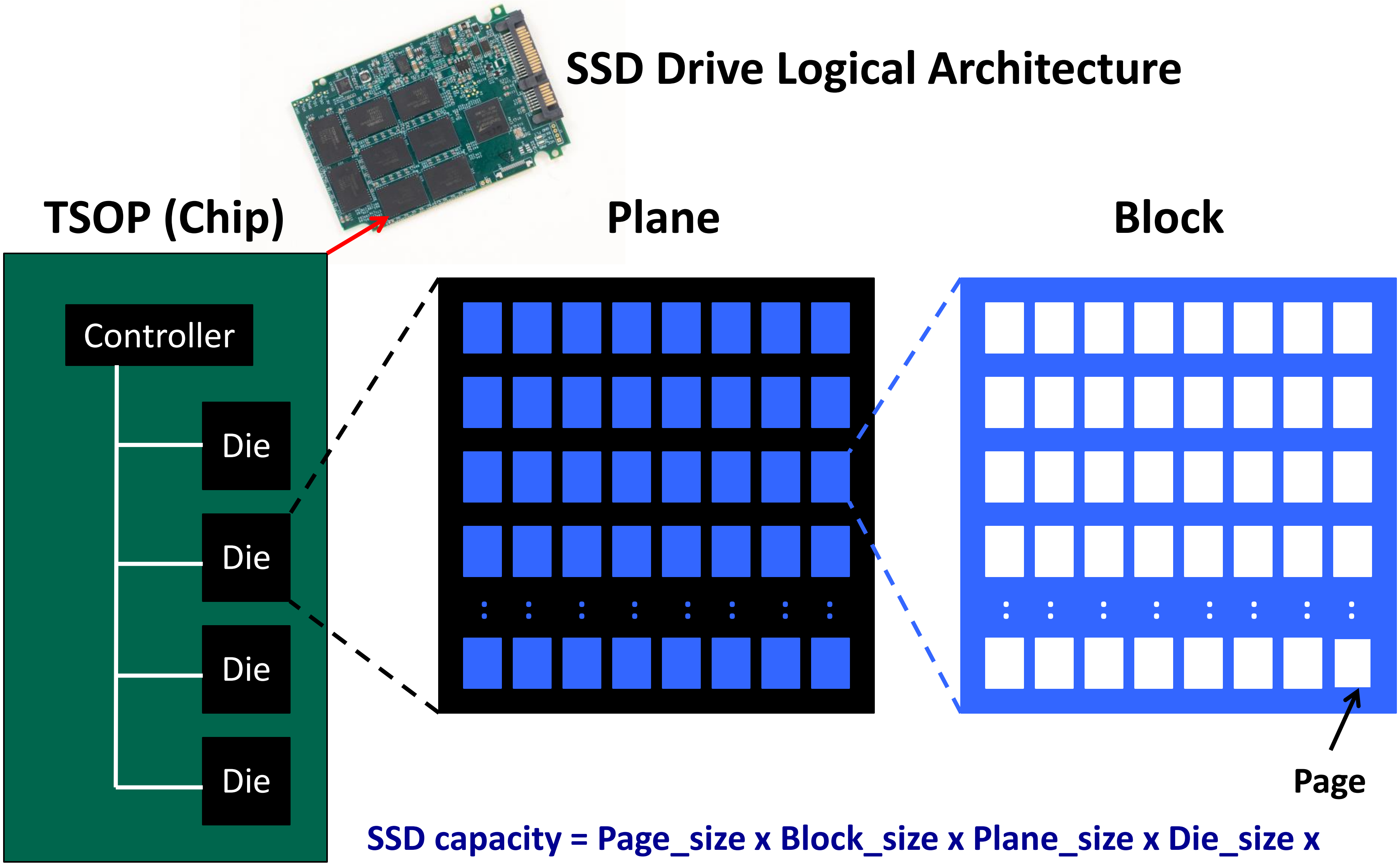
3) Write 1s to LBA 0x4000



Pages, Blocks, and Planes... Oh my!

- *Page* -- smallest unit of a read/write operation
  - 2 KB, 4 KB, 8 KB, 16 KB
- *Block* -- smallest unit of an erase operation
  - Composed of 64, 128, 256, or 512 pages
- *Plane*
  - Composed of 1,024 or 2,048 blocks
- *Die* -- independently functioning block
  - Composed of 1, 2, or 4 planes
- *Thin Small Outline Package (TSOP)* -- NAND chip
  - Composed of 1, 2, or 4 die
- *SSD device* composed of some number of TSOPs

SSD Drive Logical Architecture



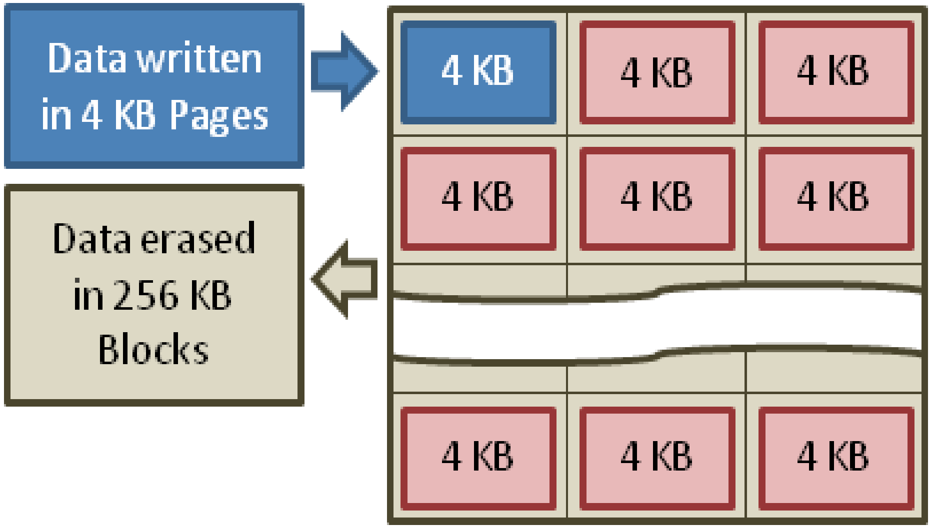
SSD capacity = Page_size x Block_size x Plane_size x Die_size x TSOP_size x #TSOPs/drive

E.g., 8 KB page, 256 pages/block, 1024 blocks/plane, 2 planes/die, 4 die/TSOP, 16 TSOPs

SSD size = 8 KB * 256 * 1024 * 2 * 4 * 16 = 256 GB

Reading, Writing, and Erasing

- Reading is performed at the page level
- Writing is performed at page level
  - A page must be erased before it can be overwritten *(see next page)*
- Erasing is performed at the block level
  - This process is slow, up to 10 ms

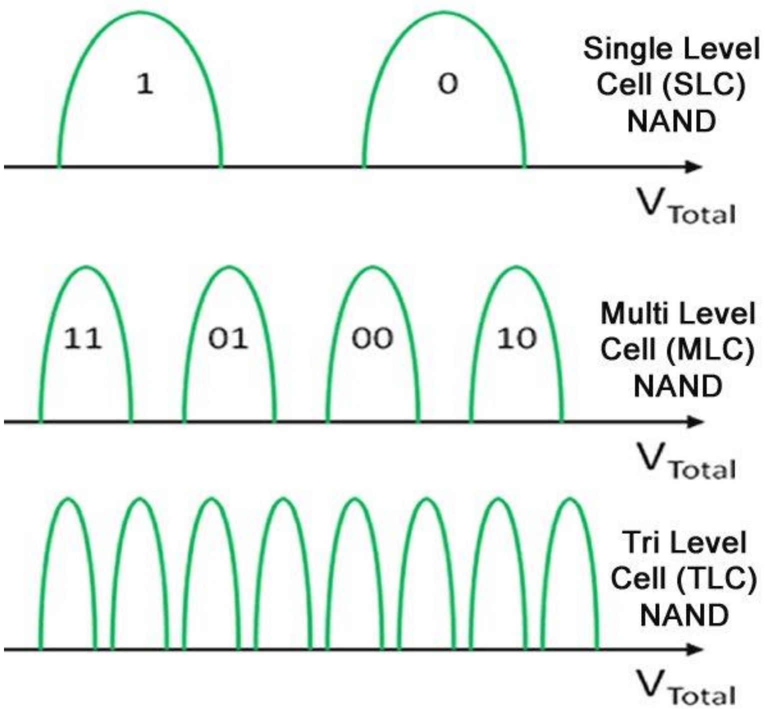


Program and Erase Operations

- NAND flash has only two operations
  - *Erase:* Set bits to 1
  - *Program (*aka *write):* Change bit from 1 to 0
  - Cannot change a bit from 0 to 1
- To write a page within a block, all of the page's bits need to be set to 1 (erase) and then other bits set to 0 (program), as appropriate
- NAND memory becomes unreliable after a finite number of program/erase (P/E) cycles
  - Dense memory cells have reduced P/E cycles

Memory Cell Density

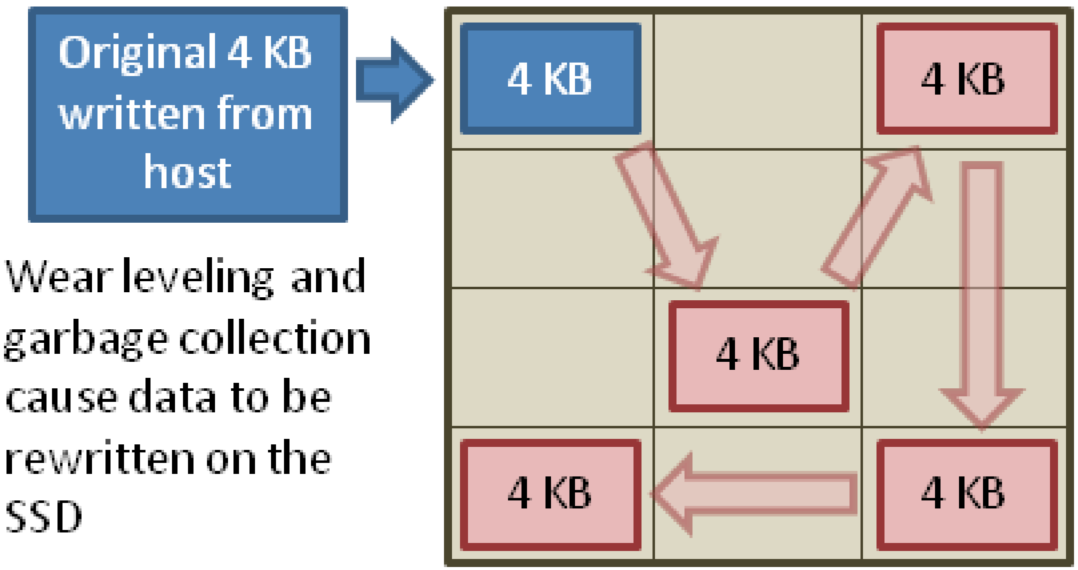
- *SLC* NAND stores 1 bit per memory cell (i.e., 2 states)
  - Long lifetime (i.e., most P/E cycles)
- *MLC* NAND stores 2 bits per cell (i.e., 4 states)
  - Higher latency and reduced lifetime compared to SLC
- *TLC* NAND stores 3 bits per cell (i.e., 8 states)
  - Even higher latency, reduced lifetime
- *16 Level Cell (16LC)* stores 4 bits per cell (i.e., 16 states)
  - Not (yet) in use



Page Architecture

- *Pages* are measured in bytes
- Pages are composed of memory cells, each of which holds some number of bits
  - SLC = 1 b/cell, MLC = 2 b/cell, TLC = 3 b/cell, 16LC = 4b/cell
- #_cells/page = page_size (B) / #bits_per_cell
  - E.g., a 4 KB page requires 4096 SLC cells, 2048 MLC cells, 1366 TLC cells, and 1024 16LC cells

Wear Leveling & Garbage Collection



Wear leveling and garbage collection cause data to be rewritten on the SSD

Solid-state drive Flash memory

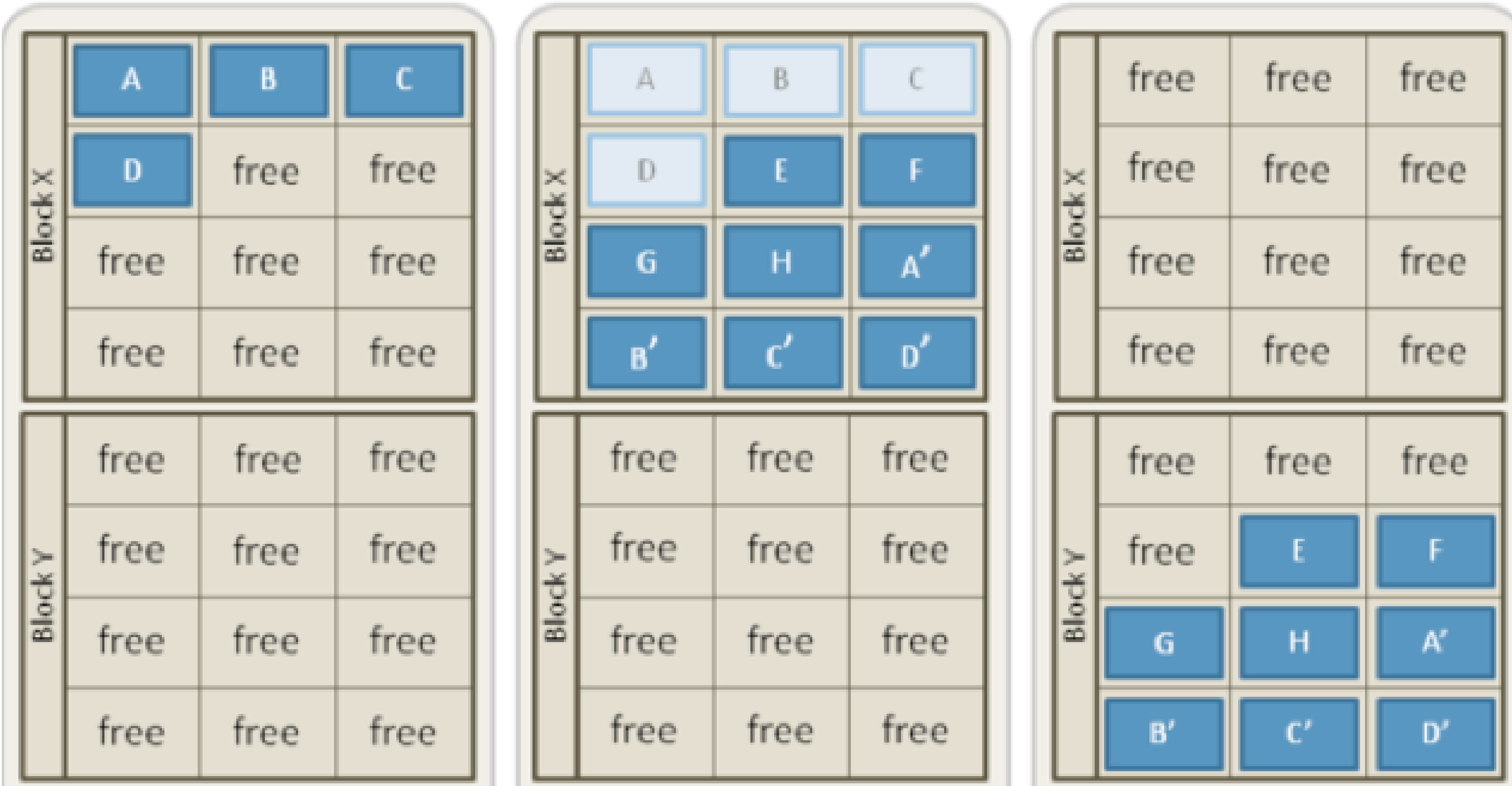
Wear Leveling

- *Wear leveling* is a form of statistical "write balancing"
  - The FTL ensures that writing is spread out to less-written blocks to prevent premature failure of cells
  - Pages can be individually written (programmed), but an entire block has to be erased at one time
    - If a block is erased as part of wear leveling, the valid contents of the block are re-written to other pages
    - This results in *write amplification*, where even a non-changing page might be written to memory multiple times due to the effects of wear leveling

Garbage Collection

- A worst-case scenario of wear leveling would be the need to erase an entire block due to having to write a single byte to a single page
  - Large hit to performance since erase_time >> program_time
- *Garbage collection* (aka *self-healing*) identifies blocks that appear to be prone to such behavior and erase them when they are not otherwise in use

Write Amplification and Garbage Collection



Issues With Garbage Collection

- A problem with GC is that "not otherwise in use" varies by file system and the fact that the SSD controller has no way of knowing what file system is being used
  - E.g., Samsung developed a proprietary algorithm optimized for NTFS by examining the $Bitmap structure and employs NTFS exclusively on their SSDs
- GC algorithms are proprietary and vary by SSD make and model

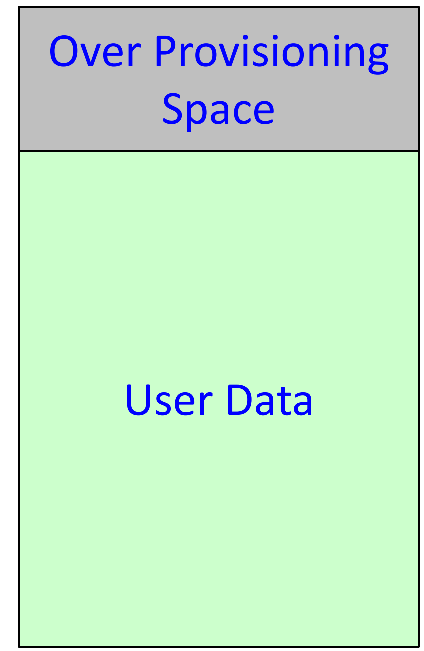
TRIM

- A command that allows an SSD-aware operating system to explicitly tell the SSD that some blocks are no longer in use
  - A mechanism to optimize garbage collection and improve device efficiency
- Available starting in Android 4.3, Mac OS X 10.6.8 (Snow Leopard), Linux 2.6.28, Windows 7, and various Unix versions
  - Does not work on RAID systems or on encrypted disks

```
gck@Moriarty:~# sudo hdparm -I /dev/sda | grep -i trim
            *    Data Set Management TRIM supported (limit 8 blocks)
            *    Deterministic read ZEROs after TRIM
gck@Moriarty:~# echo Gary > tyui.txt ; sync
gck@Moriarty:~# hdparm --fibmap tyui.txt

tyui.txt:
 filesystem blocksize 4096, begins at LBA 41283534, assuming 512 byte sectors.
 byte_offset  begin_LBA    end_LBA    sectors
           0   68356222    68356229          8
gck@Moriarty:~# hdparm --read-sector 68356222 /dev/sda | head -n 4

/dev/sda:
reading sector 68356222: succeeded
6147 7972 000a 0000 0000 0000 0000 0000
gck@Moriarty:~# rm -f tyui.txt
gck@Moriarty:~# fstrim -v . ; sync
.: 2678761472 bytes was trimmed
gck@Moriarty:~# hdparm --read-sector 68356222 /dev/sda | head -n 4

/dev/sda:
reading sector 68356222: succeeded
0000 0000 0000 0000 0000 0000 0000 0000
gck@Moriarty:~#
```

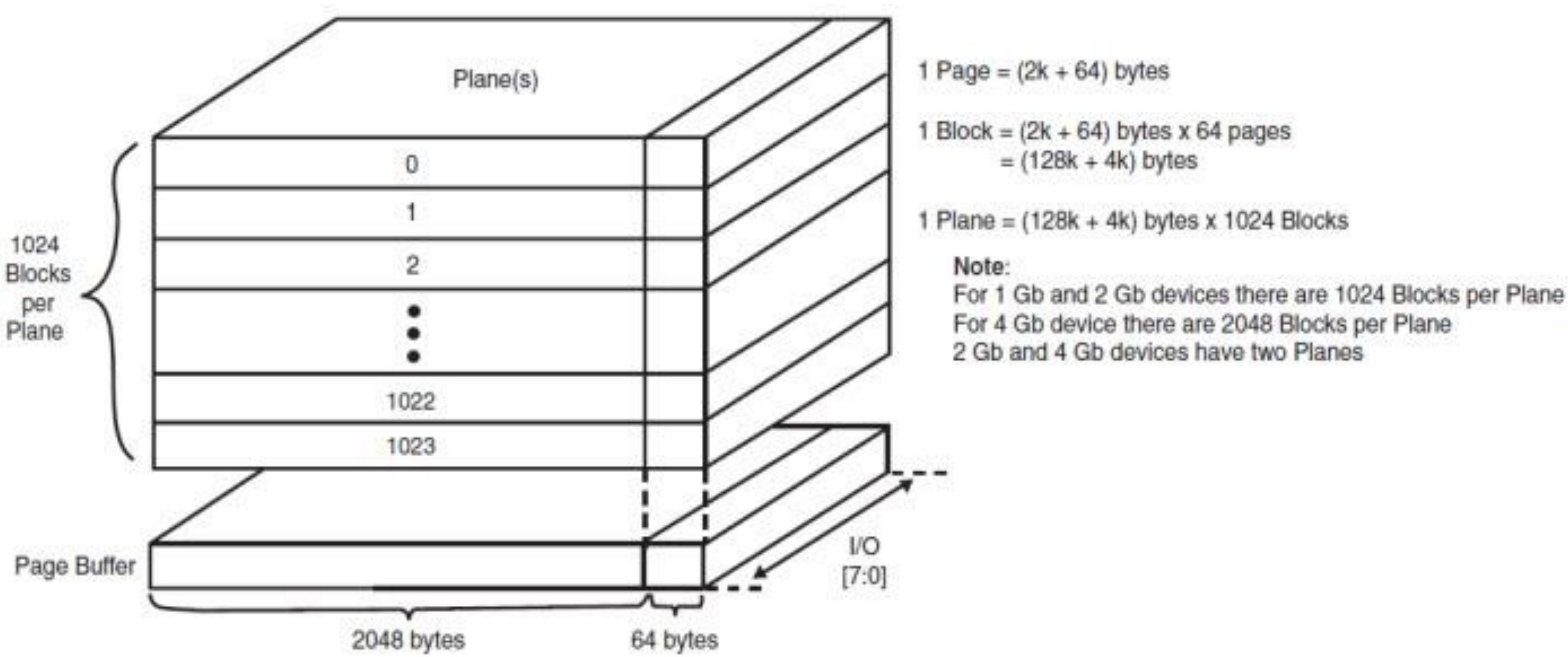
Over Provisioning

- A portion of the total SSD capacity -- generally about 7% -- is reserved for garbage collection and other functions
  - A higher amount of over provisioning (OP) yields higher write performance, lower write amplification, and longer SSD useful life
  - Over provisioned space is not available to standard forensic tools
- Do **not** confuse this with OP on a HDD which provides a permanent, static replacement for bad sectors

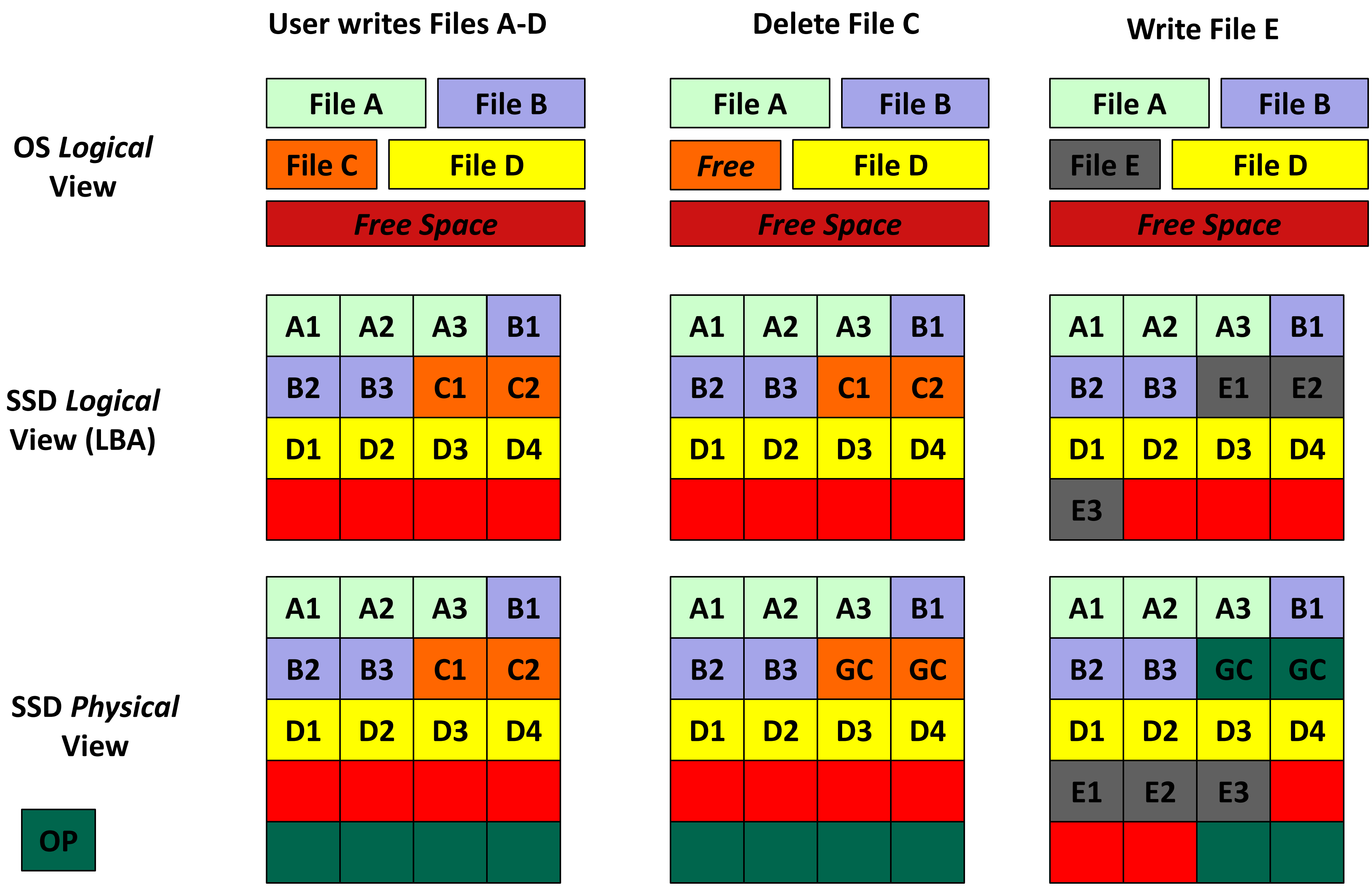


Flash-Based SSD

Over Provisioning



Garbage Collection and Over Provisioning



So, By Way of Review...

- Write (program) a page; erase entire block
- Wear leveling prevents early demise of SSD by distributing the P/E operations
- Erase operations take a long time; garbage collection minimizes the performance hit
- TRIM commands allow the OS to tell the SSD about unused pages, further optimizing garbage collection
- Over provisioned space allows even better efficiency of SSD operation
  - But OP pages are inaccessible to standard software

SSD Forensics

Poll question #4

Imaging an SSD

- Imaging an SSD will follow the same process as imaging a spinning HDD
  - Employ a write-blocker and standard imaging software
- "Unallocated space" will not be as plentiful as on an HDD
- The very act of connecting an SSD into a write-blocker will supply power to the SSD and possibly start garbage collection
  - *Corollary:* Image the SSD as soon as possible after seizure

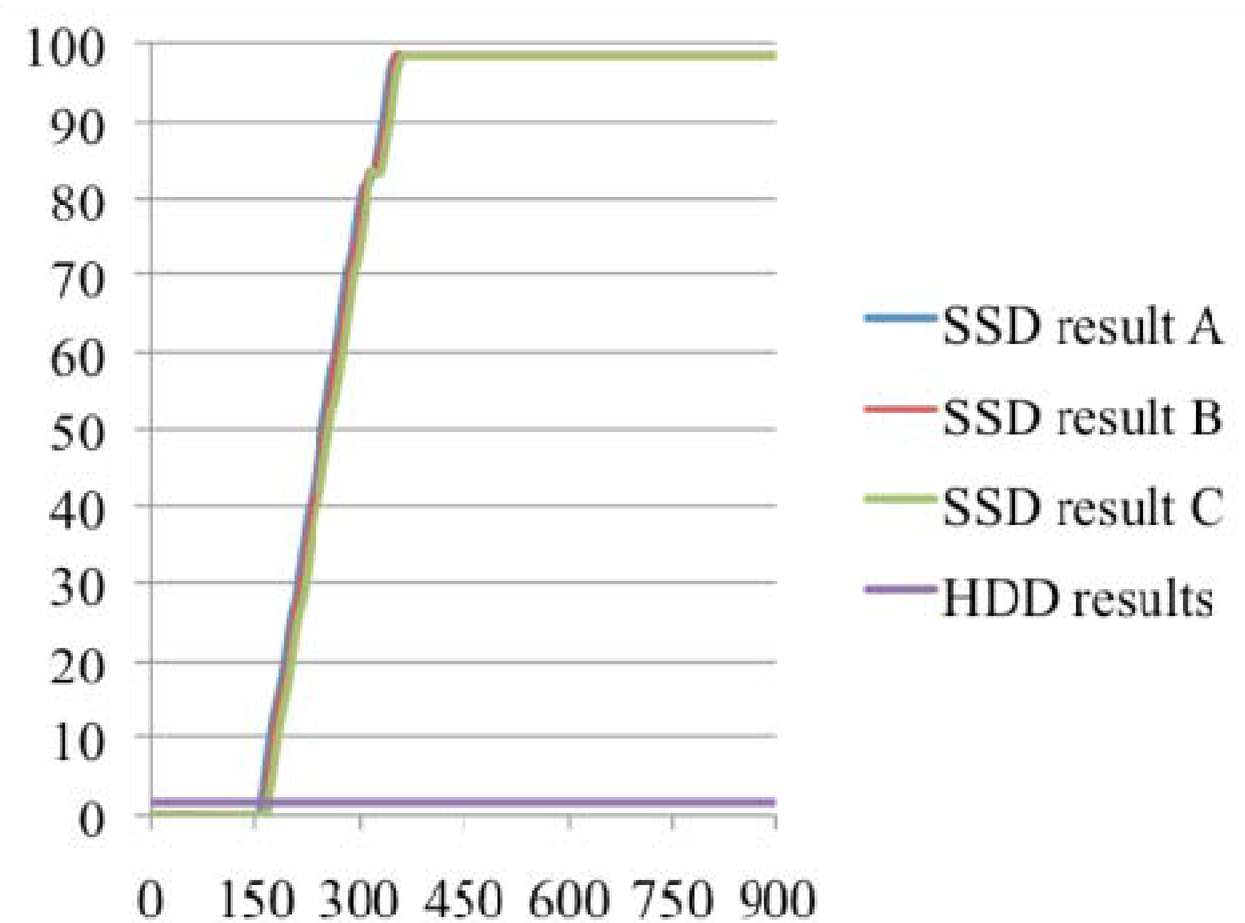
Garbage Collection During Imaging

- A GC operation that occurs while imaging an SSD will likely cause *evidence corrosion*
- GC can also cause problems with image validation
  - If the image hash doesn't match the SSD hash, it is most likely that the *original* device has lost data compared to the copy
  - We tend to interpret hash mismatches as an error with the copy
- Without knowing the GC algorithm and schedule, it is impossible to know when GC last occurred, when GC will next occur, and what is impacted by GC

Formatting an SSD

- Bell & Boddington (2010) compared how long an HDD and SSD held on to "deleted" data after a Quick Format
  - HDD holds the "deleted" space forever
  - SSD starts GC within a few minutes and completely wipes the device in ~5 minutes
- On an SSD, a Quick Format and Full Format are essentially equivalent

The Minutes After A *Quick Format*



Percentage of drive that is zeroed out

Seconds

Bell & Boddington, 2010

Can the Original SSD State be Restored?

- Bell & Boddington (2010) imaged an HDD and SSD behind a write-blocker (disks plugged in for several minutes prior to imaging)
  - Since SSD Controller is *in* the SSD, the SSD internals are transparent to the OS and file system, and there is no control by the computer
  - Write-blocker will block any computer instructions that might trigger GC, but every model SSD has its own GC algorithm and schedule; plugging in the SSD powers it
- Analysis
  - Recovered 395,762 of 395,763 files (>99.999%) on HDD
  - Recovered 1,090 of 316,666 files (0.34%) on SSD

Do Write Blockers Help?

- Bell & Boddington (2010) tried different configurations of imaging, write-blockers, and power
  - Write-blockers can help preserve "deleted" data but cannot completely preserve the target drive
  - Start imaging as soon as possible after connecting

About the TRIM Command...

- After the TRIM command is issued, data might still reside on the SSD *but* be inaccessible via today's read or forensics methods
  - E.g., pages might be in over provisioned space

Final Comments and Observations

Some Predictions...

- GC algorithms will become more aggressive in order to improve performance of larger, faster SSDs
- TRIM-initiated GC -- issued by the OS rather than waiting to be initiated by the drive itself -- will cause even faster evidence corrosion
- These issues might also start to impact newer USB flash drives

More Predictions...

- It will become next-to-impossible to reliably:
  - Recover "deleted" files
  - Conclude any user intent when finding a "wiped" drive
- Chip-off forensics techniques will be developed
  - GC will still destroy a lot of potentially probative information
  - We'll get access to OP space

Conclusions

- Need to educate our "clients"
  - I.e., investigators, prosecutors, and judges
- The up and coming reality
  - Unallocated space will not yield reliable and complete information
  - Inferences cannot be drawn about a wiped drive
  - Data loss can occur on an SSD without a command by the computer, arguably causing loss of probative information
  - A hash mismatch after imaging may not mean what we think it means
  - It is impossible to reliably delete an SSD

Useful References

- AnandTech Web Site
  - *http://www.anandtech.com*
- The SSD Database
  - http://www.thessdreview.com/SSDDatabase/
- Coding for SSDs
  - *http://codecapsule.com/2014/02/12/coding-for-ssds-part-1-introduction-and-table-of-contents/*
- *"Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?" (Bell & Boddington)*
  - *Journal of Digital Forensics, Security and Law, 5*(3)
  - *http://ojs.jdfsl.org/index.php/jdfsl/article/viewFile/21/45*
- "Why SSD Drives Destroy Court Evidence, and What Can Be Done About It" (Gubanov & Afonin)
  - *http://forensic.belkasoft.com/en/why-ssd-destroy-court-evidence*
- "Recovering Evidence from SSD Drives in 2014" (Gubanovis & Afonin)
  - *http://articles.forensicfocus.com/2014/09/23/recovering-evidence-from-ssd-drives-in-2014-understanding-trim-garbage-collection-and-exclusions/*

Summary

- Introduction to solid state devices
- SSD terms, concepts, and operation
- Impact on digital forensics
- Conclusion