NVDIMM-N Cookbook: A Soup-to-Nuts Primer on Using NVDIMM-Ns to Improve Your Storage Performance

Arthur Sainio
Director Marketing, SMART Modular
Mario Martinez
Director Marketing, Netlist

SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Abstract

- Non-Volatile DIMMs, or NVDIMMs, have emerged as a go-to technology for boosting performance for next generation storage platforms. The standardization efforts around NVDIMMs have paved the way to simple, plug-n-play adoption. If you're a storage developer who hasn't yet realized the benefits of NVDIMMs in your products, then this tutorial is for you! We will walk you through a soup-to-nuts description of integrating NVDIMMs into your system, from hardware to BIOS to application software. We'll highlight some of the "knobs" to turn to optimize use in your application as well as some of the "gotchas" encountered along the way.

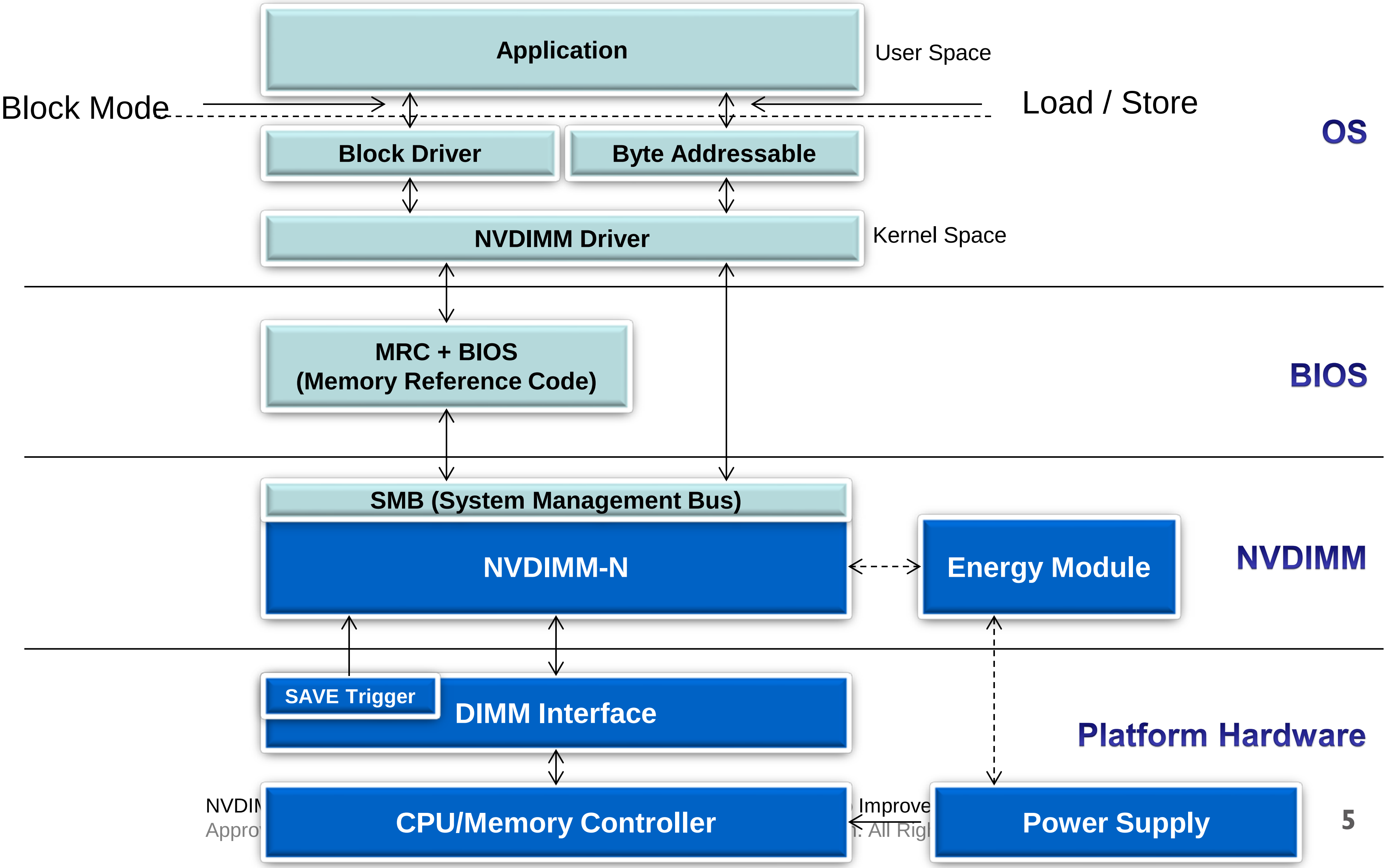
- **Learning Objectives**
  - Understand what an NVDIMM is
  - Understand why an NVDIMM can improve your system performance
  - Understand how to integrate an NVDIMM into your system

- Part I
  - NVDIMM
- Part II
  - BIOS
- Part III
  - OS (Linux)
- Part IV
  - System Implementations & Use Cases

The “Ingredients”

Software

Hardware



Part 1
NVDIMM

NVDIMMs - JEDEC Taxonomy

NVDIMM-N

Standardized

- Memory mapped DRAM. Flash is not system mapped
- Access Methods -> byte- or block-oriented access to DRAM
- Capacity = DRAM DIMM (1’s -10’s GB)
- Latency = DRAM (10’s of nanoseconds)
- Energy source for backup
- DIMM interface (HW & SW) defined by JEDEC

NVDIMM-F

Vendor Specific

- Memory mapped Flash. DRAM is not system mapped.
- Access Method -> block-oriented access to NAND through a shared command buffer (i.e. a mounted drive)
- Capacity = NAND (100’s GB-1’s TB)
- Latency = NAND (10’s of microseconds)

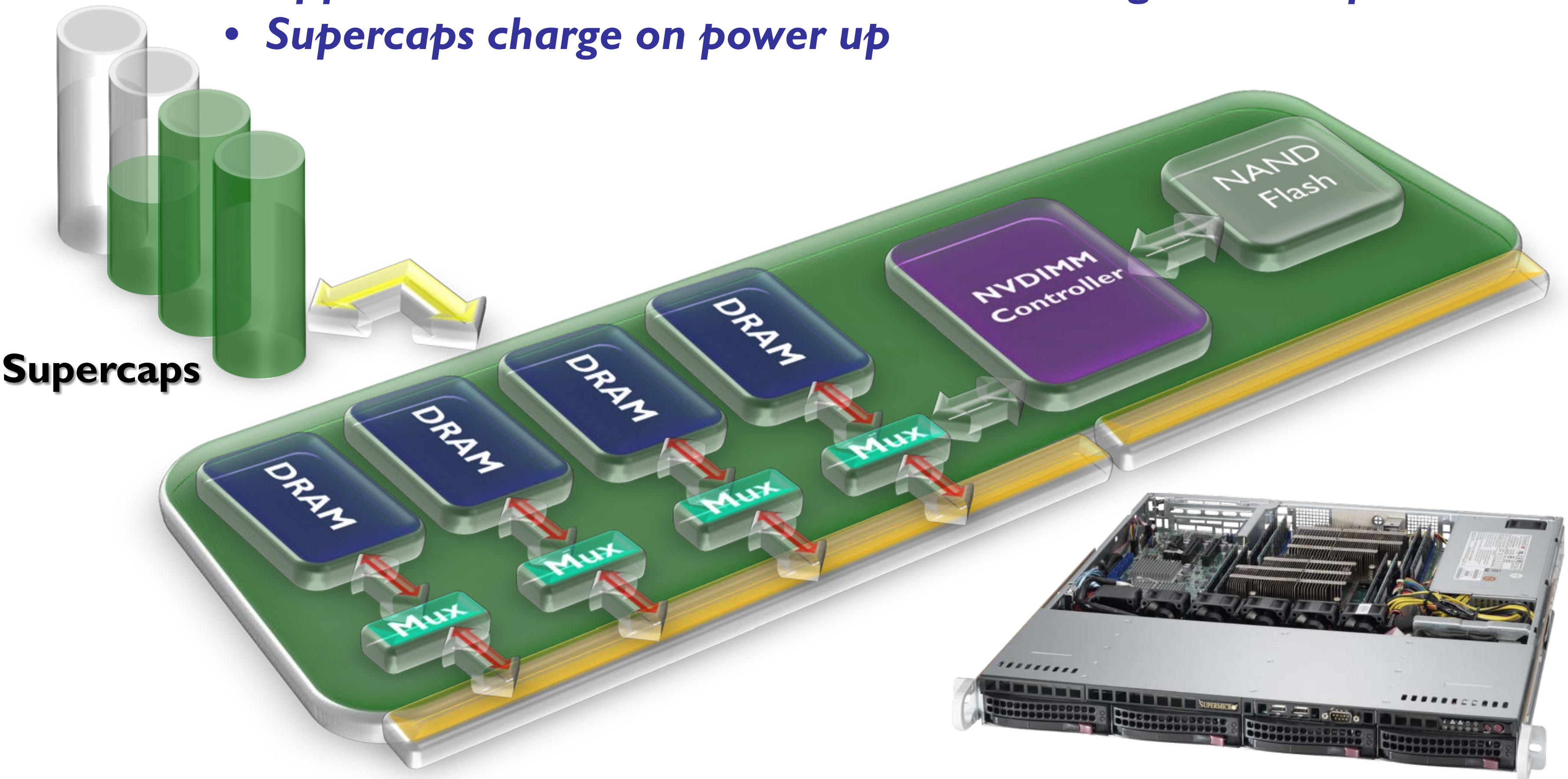
NVDIMM-P

Proposals in progress

- Memory-mapped Flash and memory-mapped DRAM
- Two access mechanisms: persistent DRAM (–N) and block-oriented drive access (–F)
- Capacity = NVM (100’s GB-1’s TB)
- Latency = NVM (100’s of nanoseconds)

NVDIMM-N How It Works

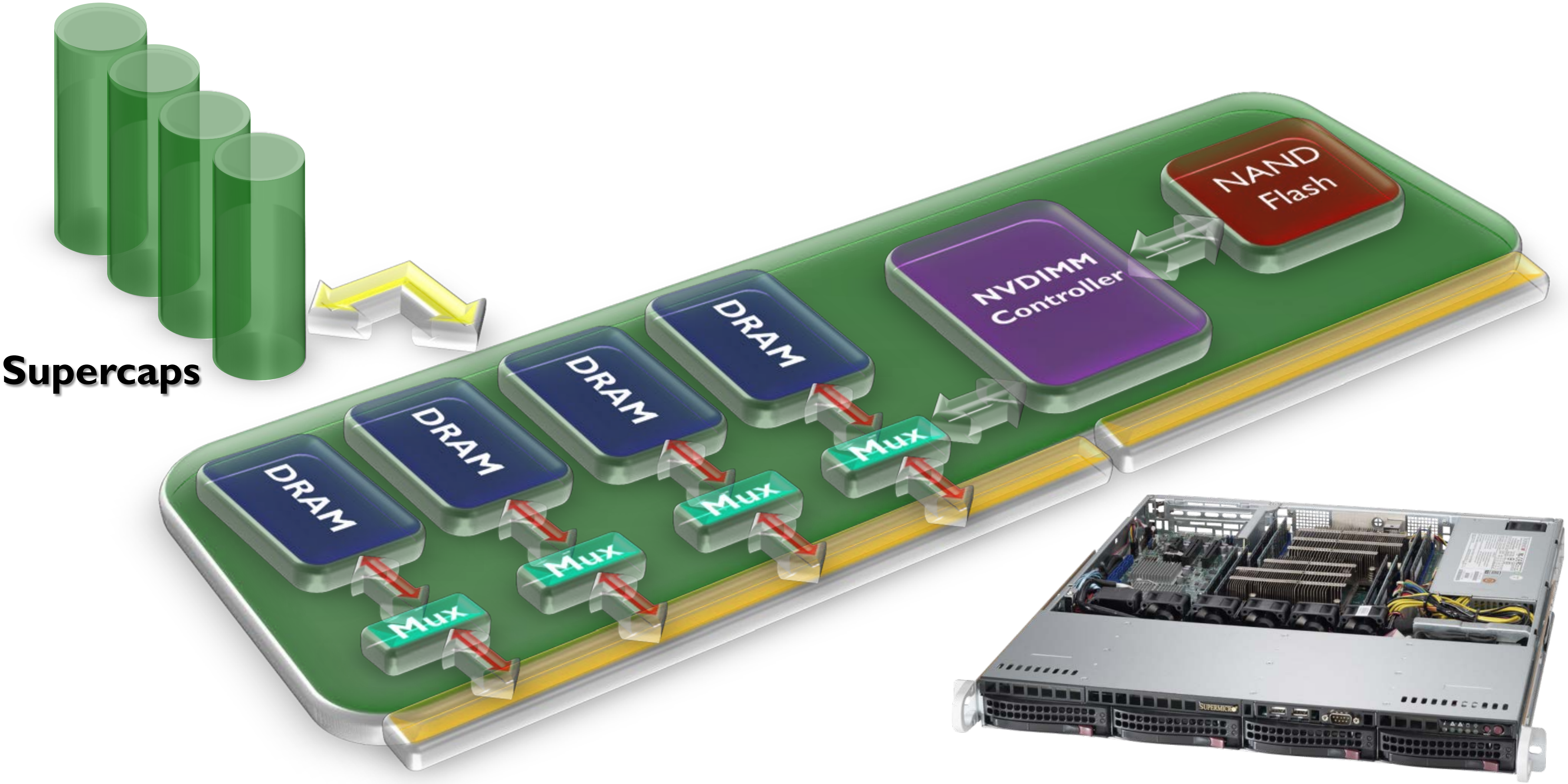
- *Plugs into JEDEC Standard DIMM Socket*
- *Appears as standard RDIMM to host during normal operation*
- *Supercaps charge on power up*



Adapted from SNIA presentations by AgigA Tech

NVDIMM-N How It Works

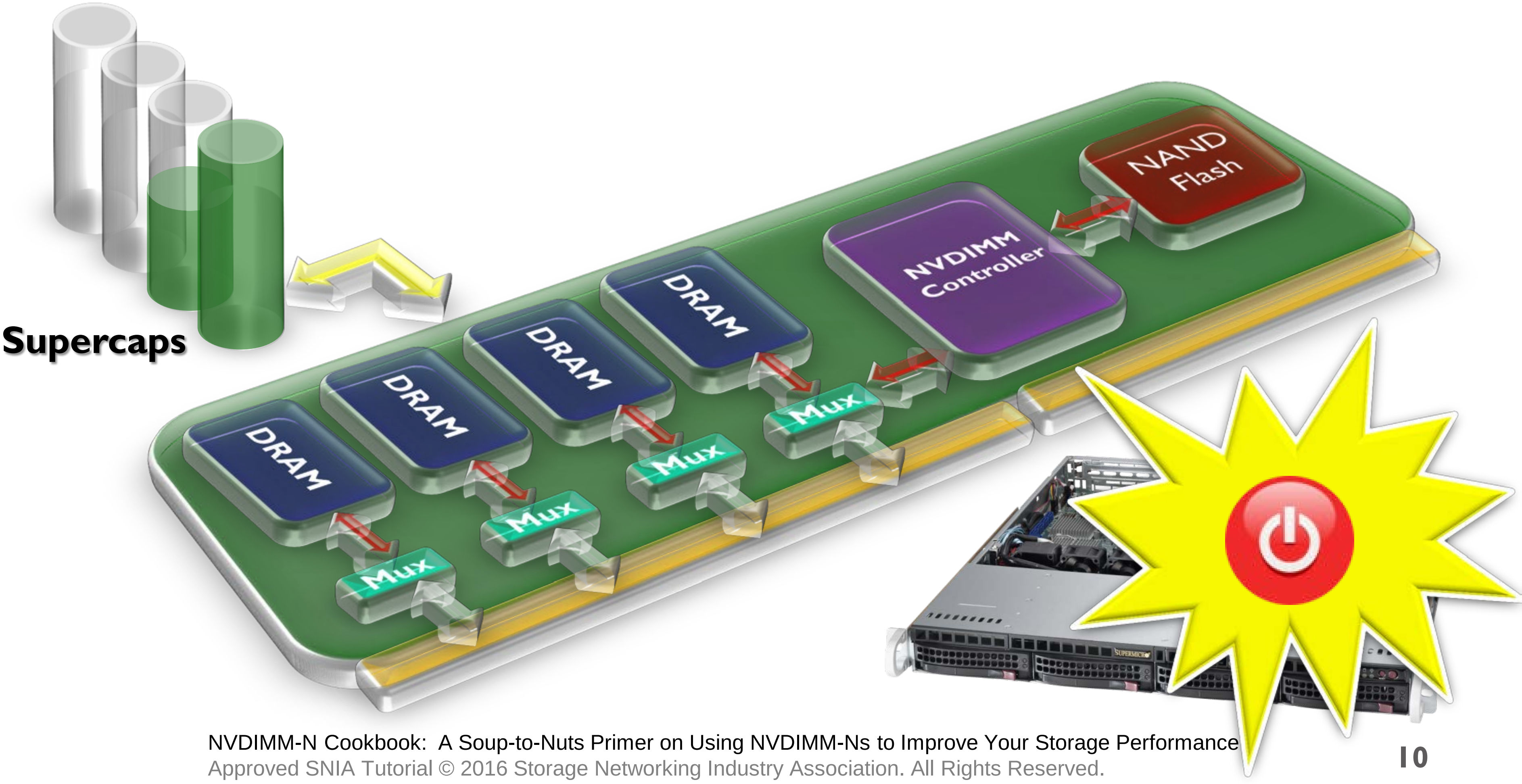
- ***When health checks clear, NVDIMM can be armed for backup***
- ***NVDIMM can be used as persistent memory space by the host***



Adapted from SNIA presentations by AgigA Tech

NVDIMM-N How It Works

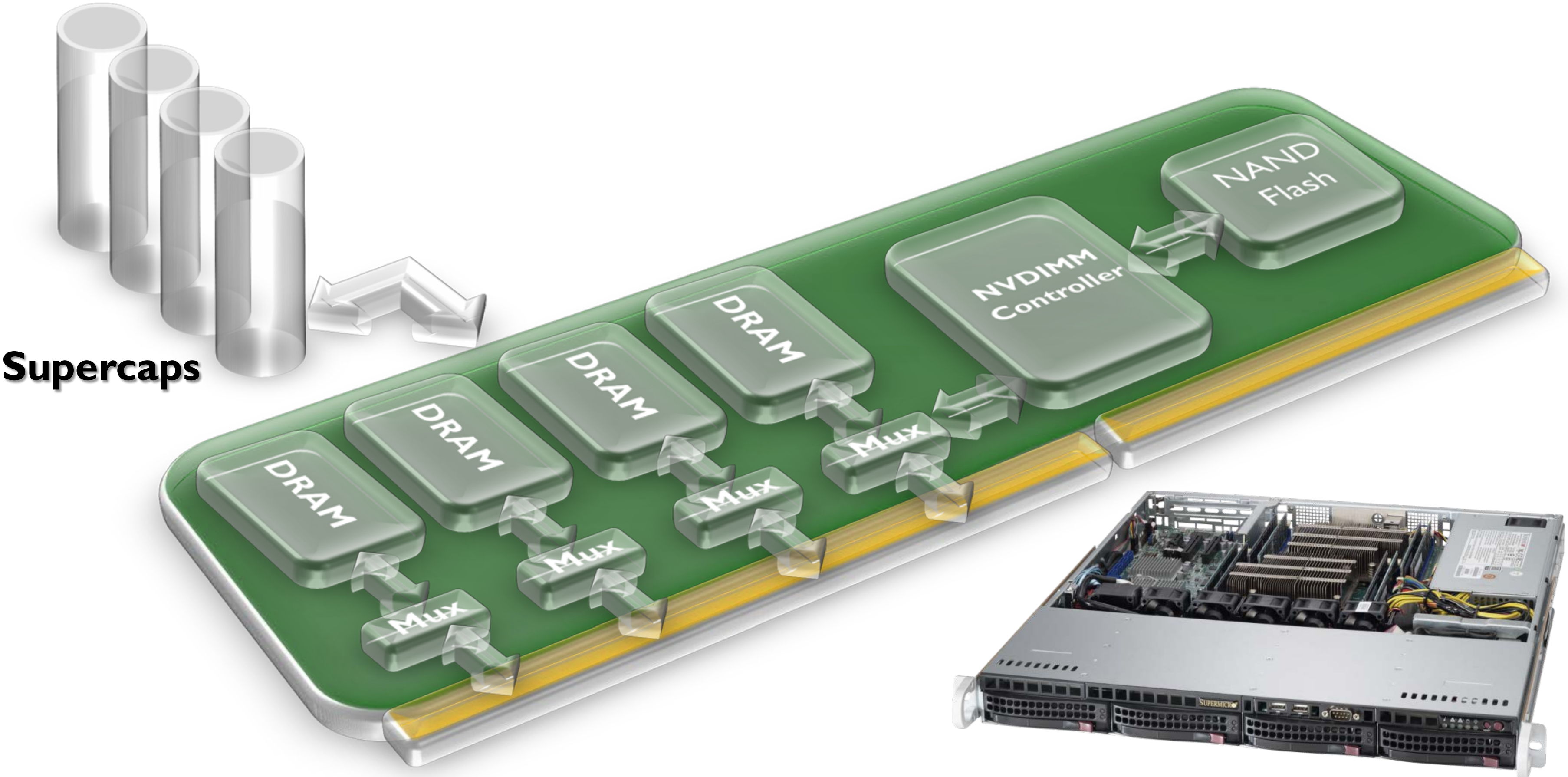
- *During unexpected power loss event, DRAM contents are moved to NAND Flash using Supercaps for backup power*



Adapted from SNIA presentations by AgigA Tech

NVDIMM-N How It Works

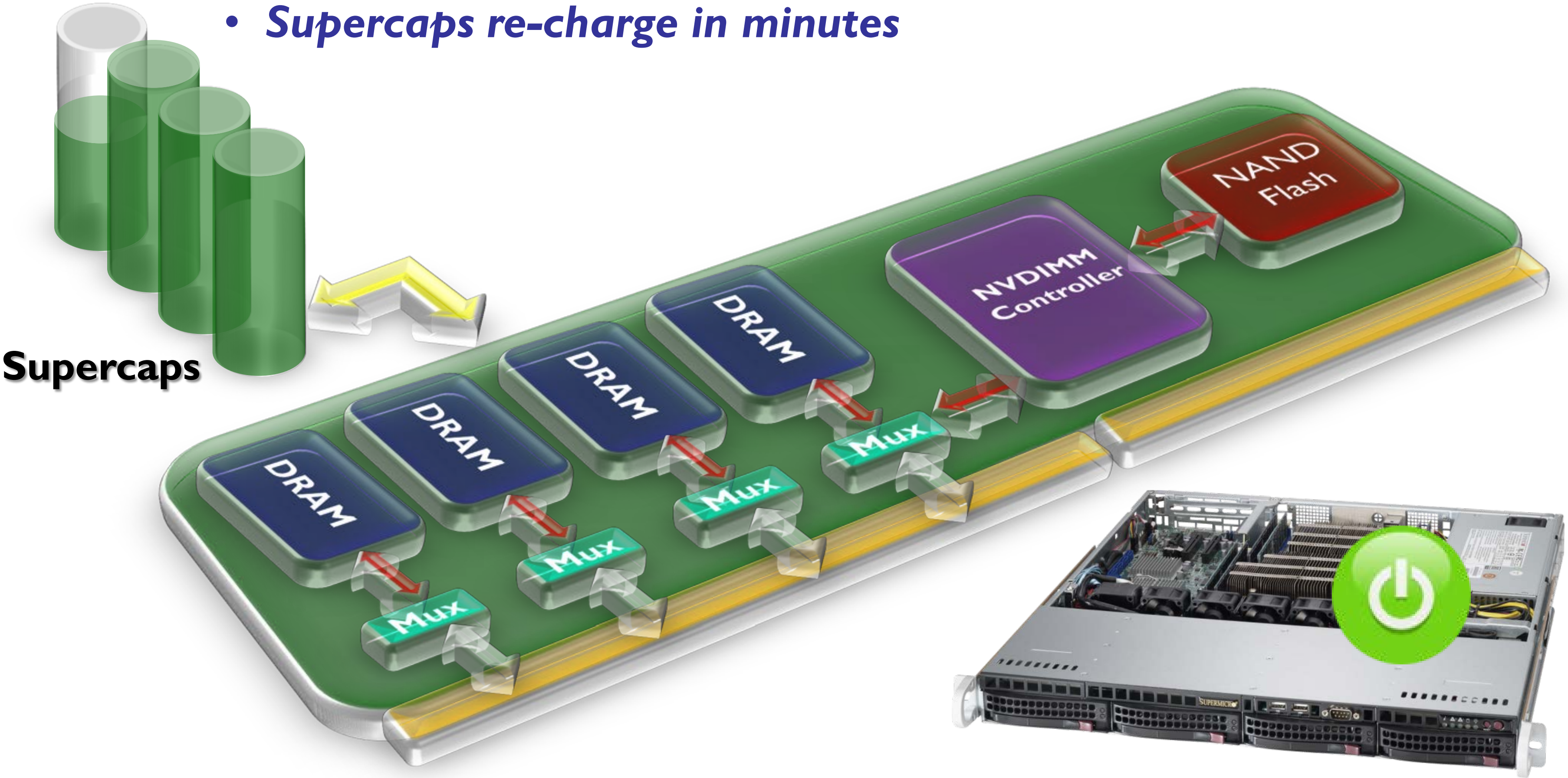
- ***When backup is complete, NVDIMM goes to zero power state***
- ***Data retention = NAND Flash spec (typically years)***



Adapted from SNIA presentations by AgigA Tech

NVDIMM-N How It Works

- *When power is returned, DRAM contents are restored from NAND Flash*
- *Supercaps re-charge in minutes*

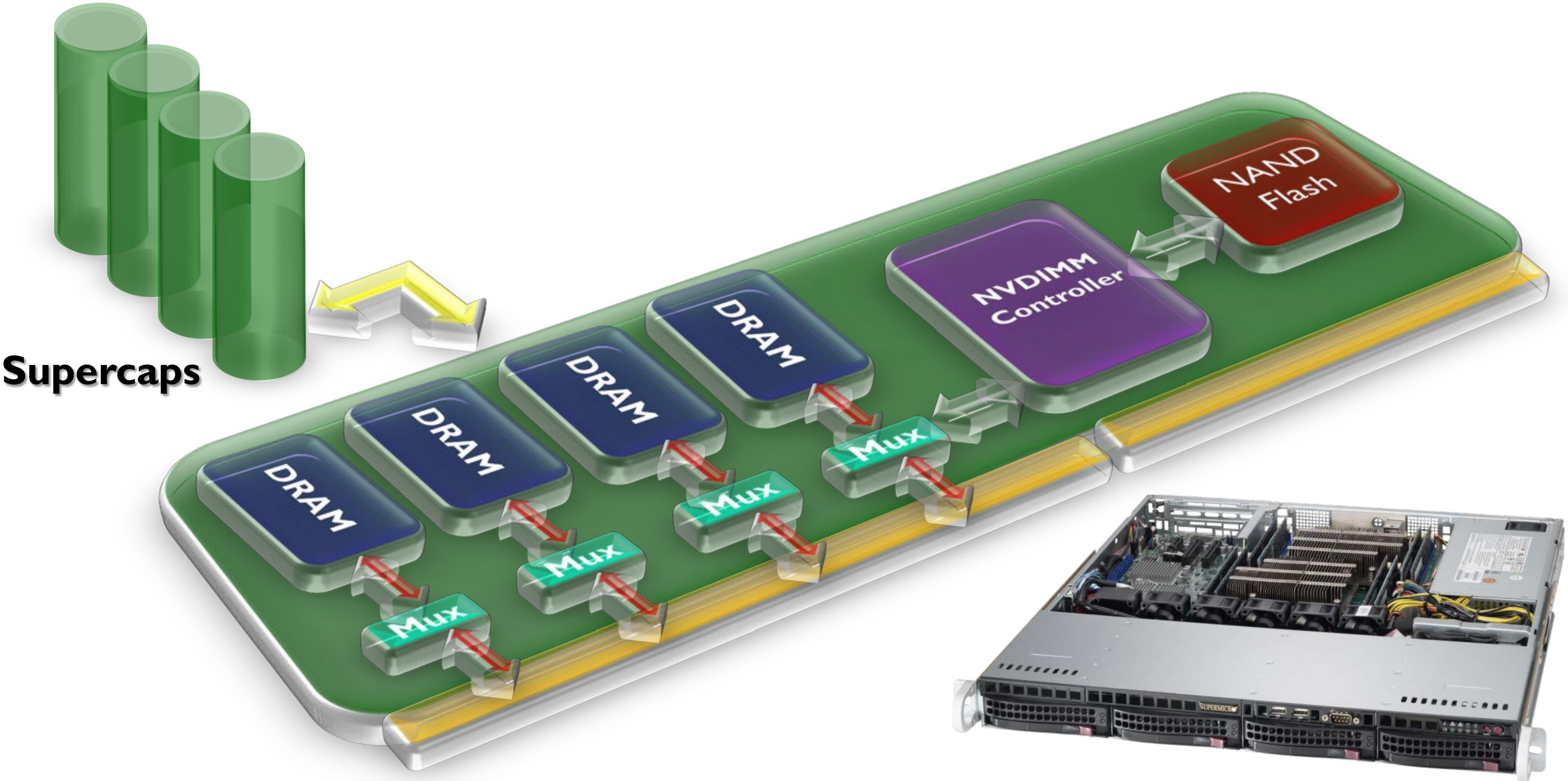


NVDIMM-N Cookbook: A Soup-to-Nuts Primer on Using NVDIMM-Ns to Improve Your Storage Performance

Adapted from SNIA presentations by AgigA Tech

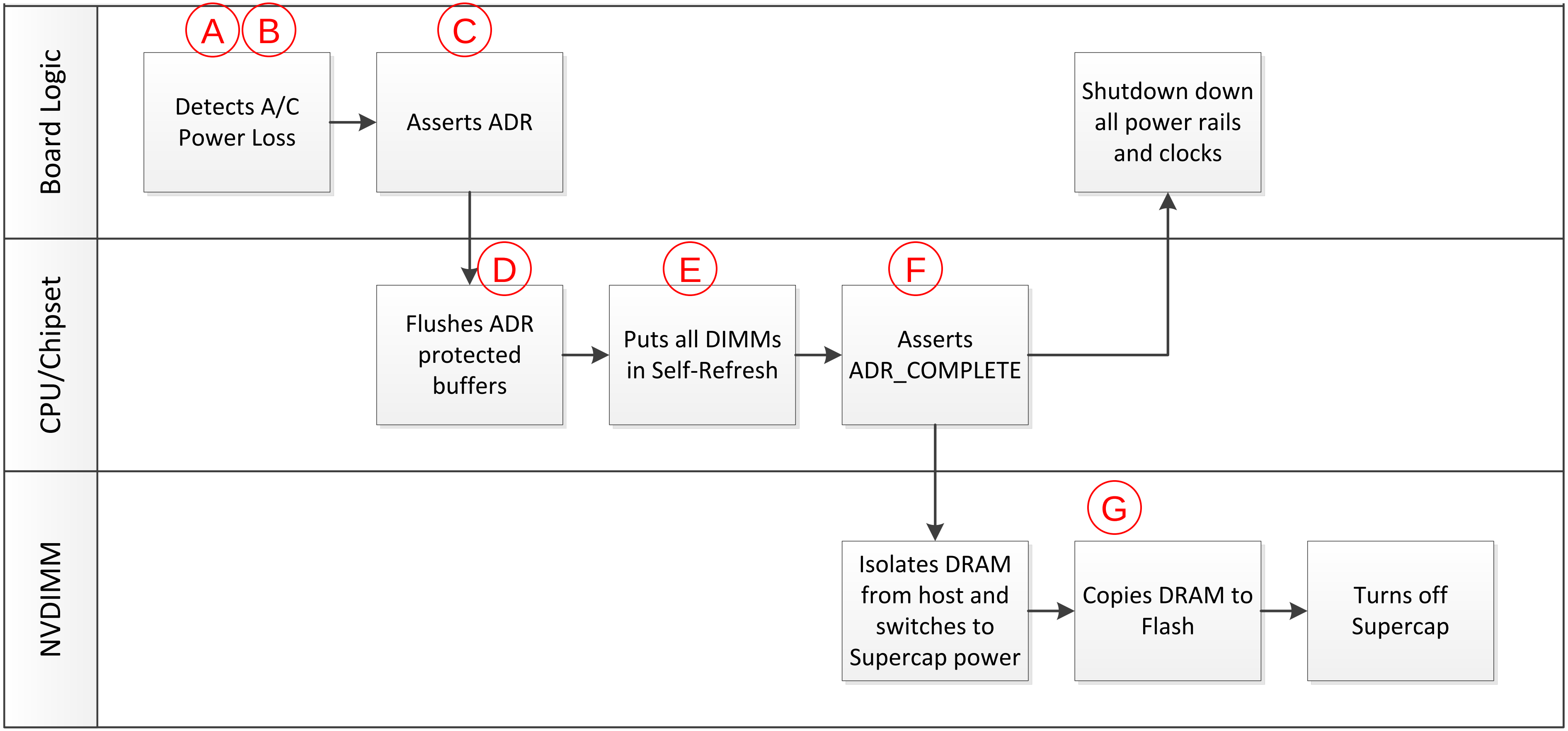
NVDIMM-N How It Works

- ***DRAM handed back to host in restored state prior to power loss***



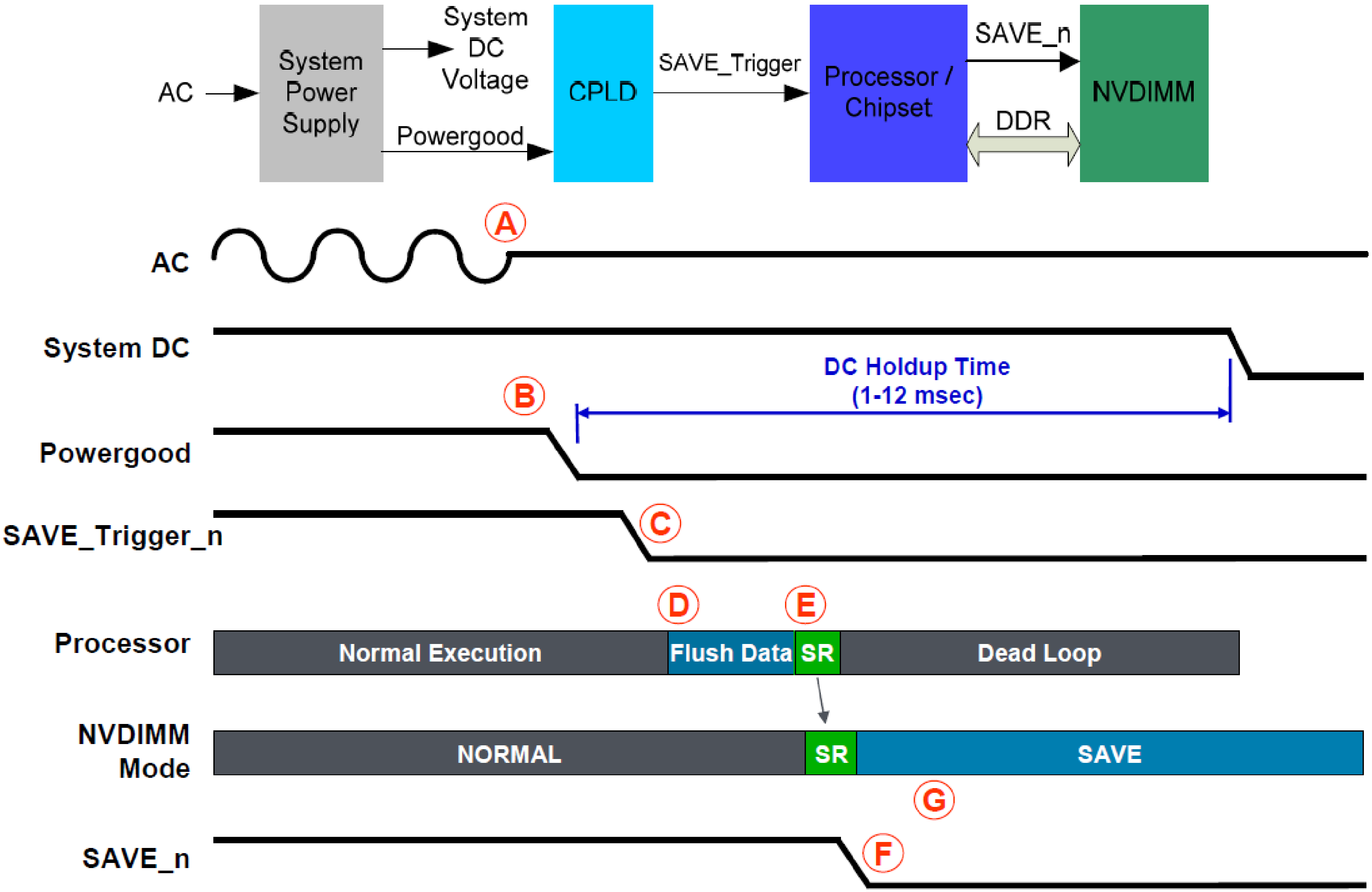
Adapted from SNIA presentations by AgigA Tech

NVDIMM Entry Process using ADR (Asynchronous DRAM Re-fresh)


Board Logic
CPU/Chipset
NVDIMM
A
B
Detects A/C Power Loss
C
Asserts ADR
Shutdown down all power rails and clocks
D
Flushes ADR protected buffers
E
Puts all DIMMs in Self-Refresh
F
Asserts ADR_COMPLETE
Isolates DRAM from host and switches to Supercap power
G
Copies DRAM to Flash
Turns off Supercap

- Letters correspond to the timing diagram on the next page

SAVE Operation



Source: Viking NVDIMM tutorial for SNIA

NVDIMM-N DDR4 Platform HW Support/JEDEC Standardization

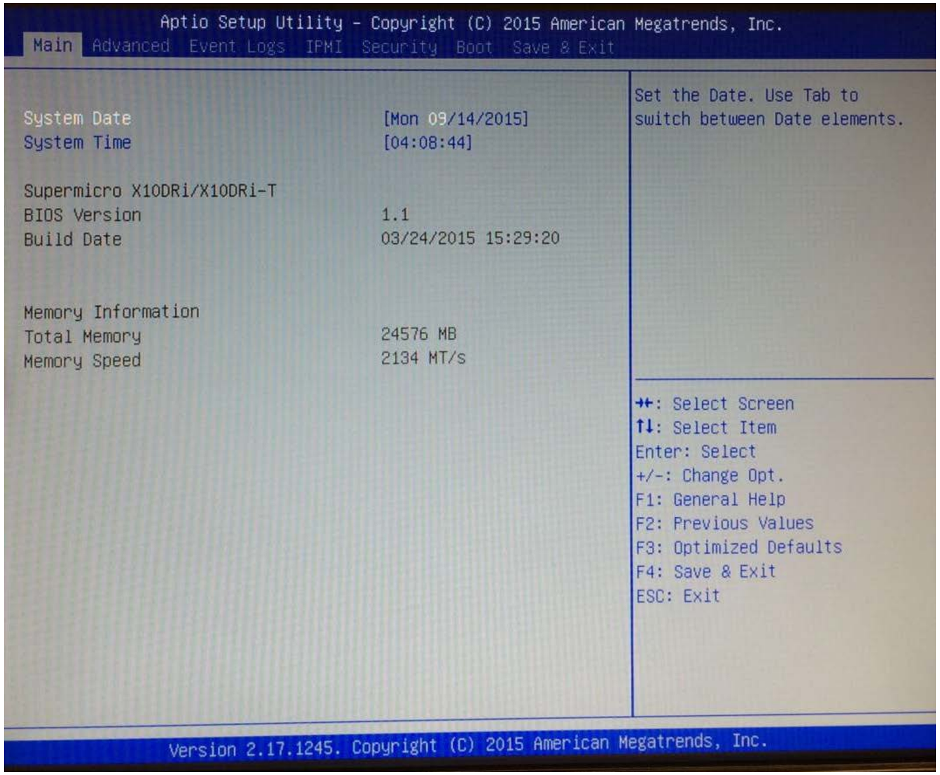
- o 12V: pin 1, 145 provides power for backup energy source
- o SAVE_n: pin 230 sets a efficient interface to signal a backup and SAVE completion
- EVENT_n asynchronous event notification
- Byte Addressable I2C interface (JESD245)
- 12V in DDR4 simplifies NVDIMM power circuitry and cable routing
  - One cable needed between NVDIMM and BPM (Backup Power Module)
  - No cable needed if Host provides 12V backup power via DDR4 12V

Part 2
BIOS

NVDIMM-N BIOS Support Functions

NVDIMMs rely on the BIOS/MRC (Memory Reference Code)

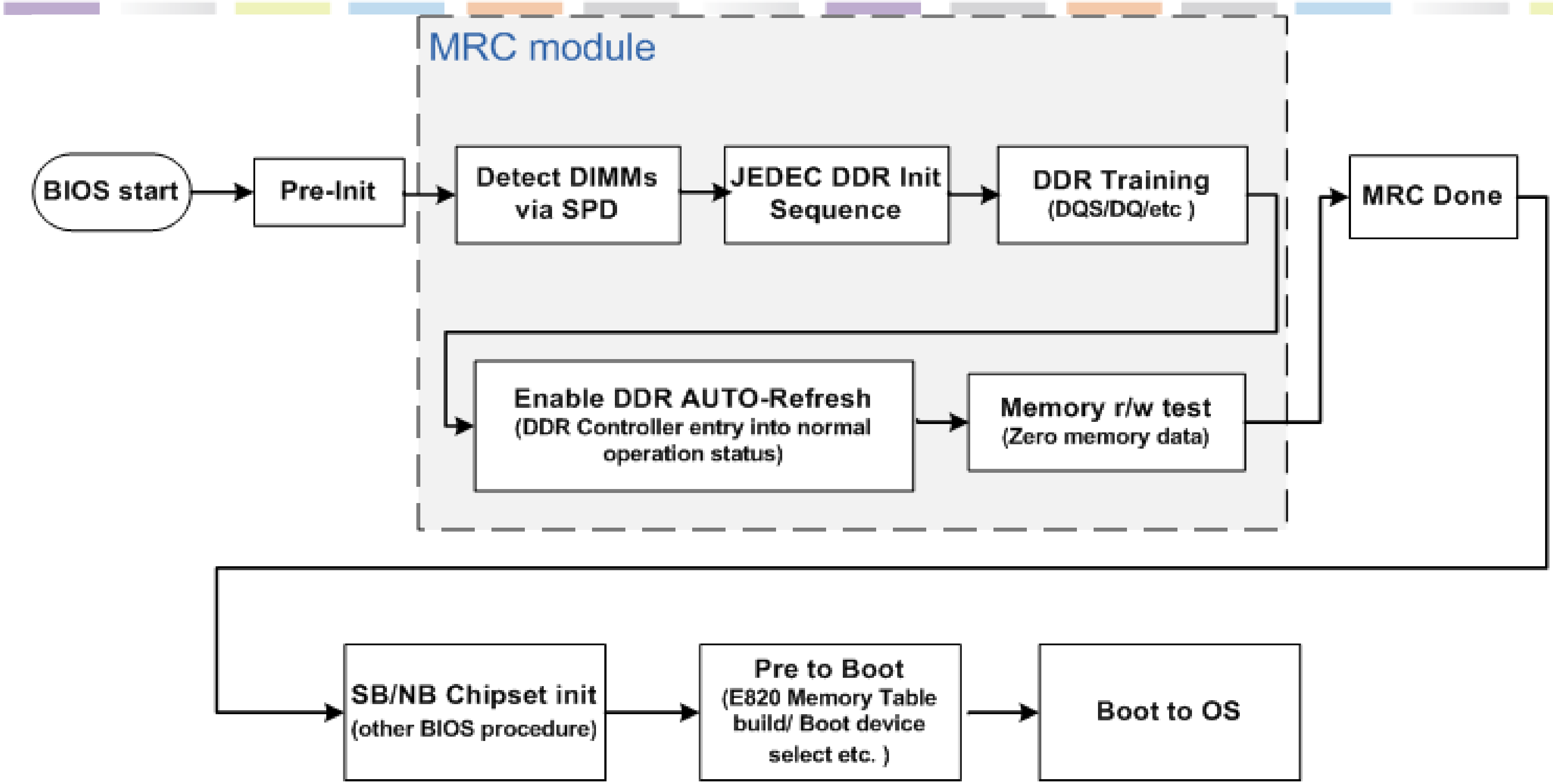
1. Detect NVDIMMs
2. Setup Memory Map
3. ARM for Backup
4. Detect AC Power Loss
5. Flush Write Buffers
6. RESTORE Data On Boot
7. Enable I2C R/W Access



```
                 Aptio Setup Utility - Copyright (C) 2015 American Megatrends, Inc.
 Main  Advanced  Event Logs  IPMI  Security  Boot  Save & Exit

                                                            Set the Date. Use Tab to
 System Date                        [Mon 09/14/2015]        switch between Date elements.
 System Time                        [04:08:44]

 Supermicro X10DRi/X10DRi-T
 BIOS Version                       1.1
 Build Date                         03/24/2015 15:29:20


 Memory Information
 Total Memory                       24576 MB
 Memory Speed                       2134 MT/s

                                                            ↔: Select Screen
                                                            ↑↓: Select Item
                                                            Enter: Select
                                                            +/-: Change Opt.
                                                            F1: General Help
                                                            F2: Previous Values
                                                            F3: Optimized Defaults
                                                            F4: Save & Exit
                                                            ESC: Exit

              Version 2.17.1245. Copyright (C) 2015 American Megatrends, Inc.
```

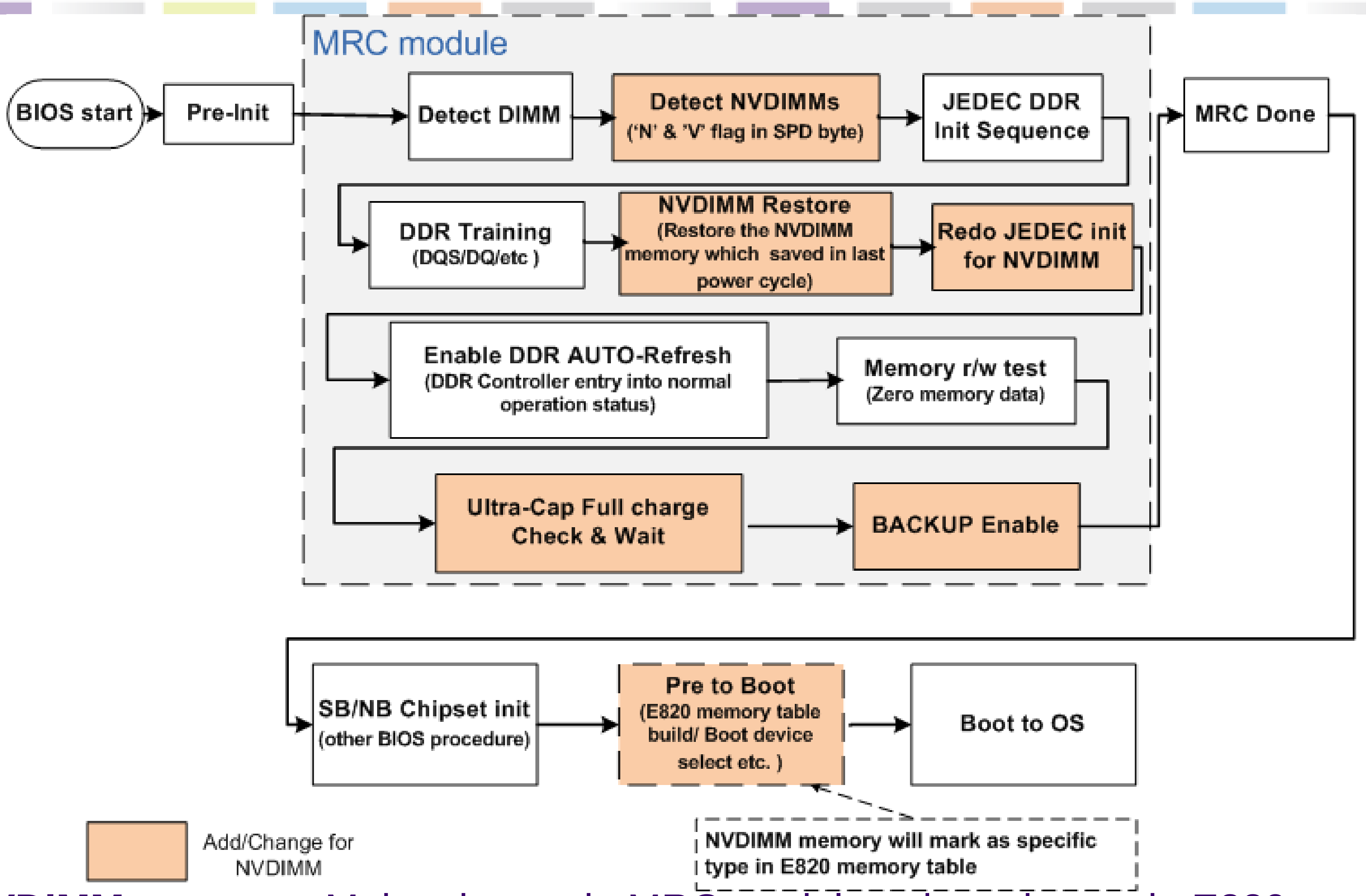
Source; SNIA NVDIMM SIG

Standard BIOS Flow


MRC module
BIOS start
Pre-Init
Detect DIMMs via SPD
JEDEC DDR Init Sequence
DDR Training (DQS/DQ/etc)
MRC Done
Enable DDR AUTO-Refresh (DDR Controller entry into normal operation status)
Memory r/w test (Zero memory data)
SB/NB Chipset init (other BIOS procedure)
Pre to Boot (E820 Memory Table build/ Boot device select etc.)
Boot to OS

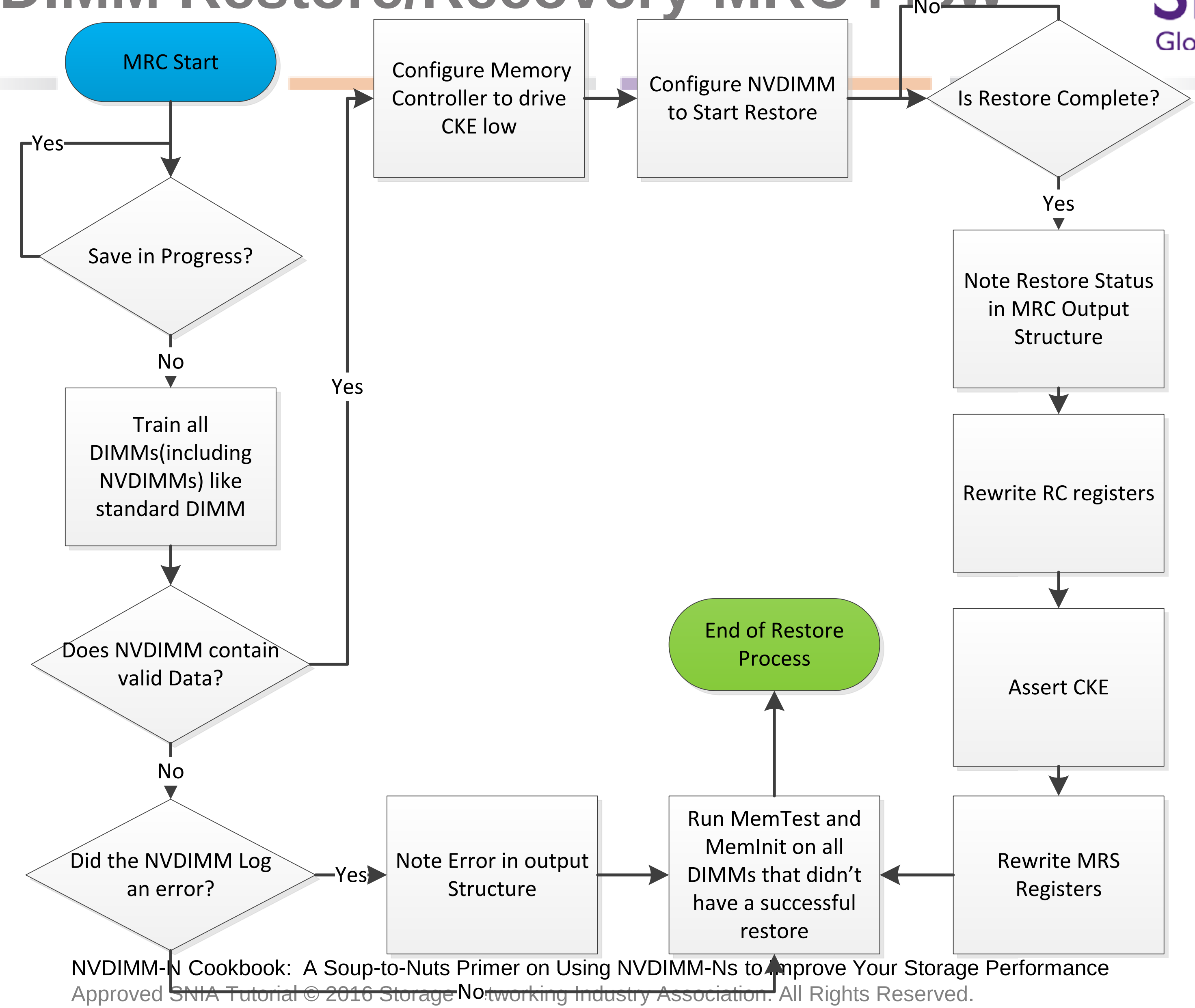
Memory **R**eference **C**ode (MRC) module provides the memory initialization procedure. This module is maintained by Intel (for Intel-based platforms of course) and released to all BIOS vendors.

NVDIMM Supported BIOS Flow


MRC module
BIOS start
Pre-Init
Detect DIMM
Detect NVDIMMs
('N' & 'V' flag in SPD byte)
JEDEC DDR
Init Sequence
MRC Done
DDR Training
(DQS/DQ/etc)
NVDIMM Restore
(Restore the NVDIMM memory which saved in last power cycle)
Redo JEDEC init for NVDIMM
Enable DDR AUTO-Refresh
(DDR Controller entry into normal operation status)
Memory r/w test
(Zero memory data)
Ultra-Cap Full charge Check & Wait
BACKUP Enable
SB/NB Chipset init
(other BIOS procedure)
Pre to Boot
(E820 memory table build/ Boot device select etc.)
Boot to OS
Add/Change for NVDIMM
NVDIMM memory will mark as specific type in E820 memory table

NVDIMM support : Major change in MRC module, minor change in E820 module

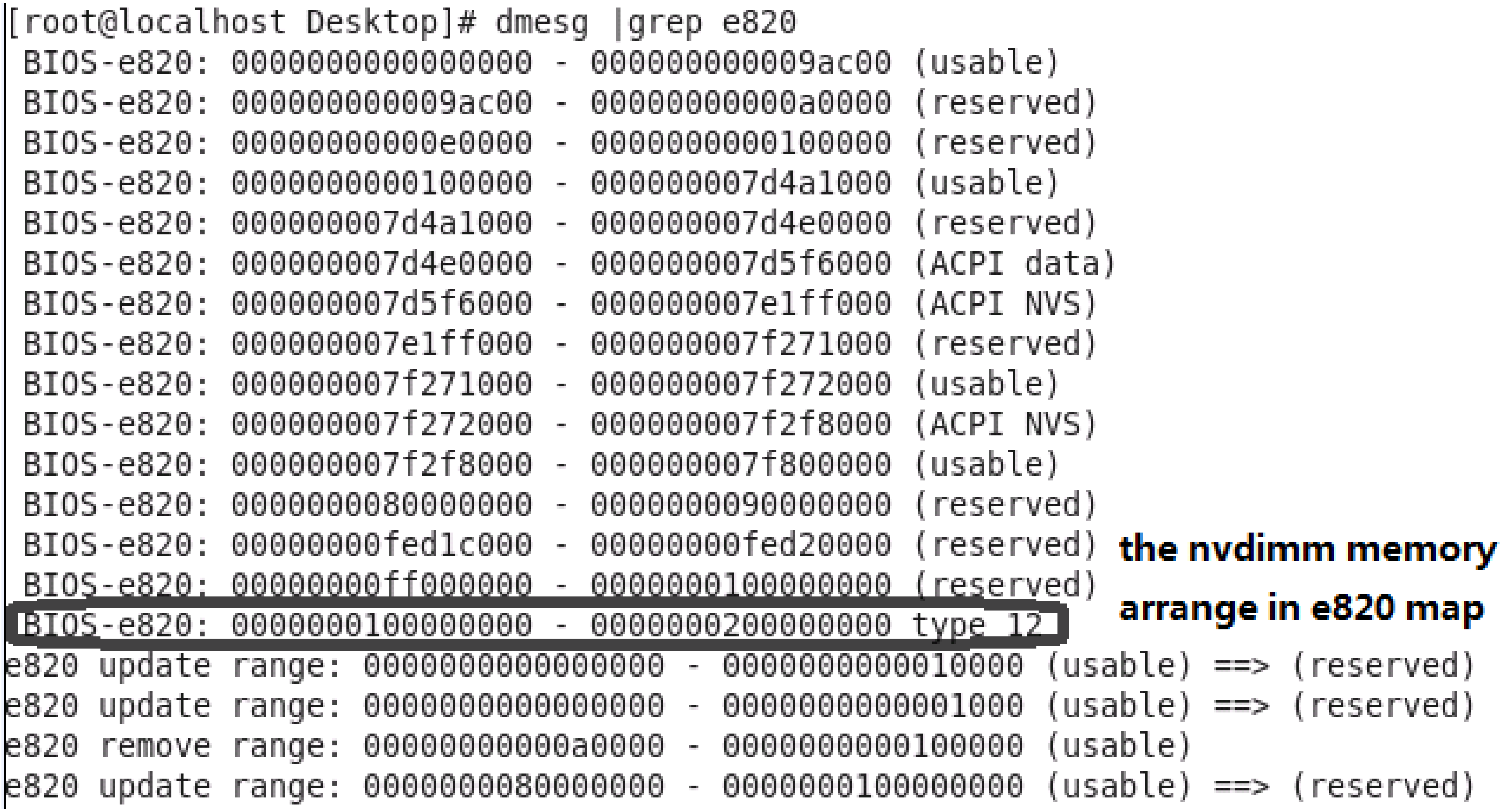
NVDIMM Restore/Recovery MRC Flow


MRC Start
Yes
Save in Progress?
No
Train all DIMMs(including NVDIMMs) like standard DIMM
Does NVDIMM contain valid Data?
Yes
No
Did the NVDIMM Log an error?
Yes
No
Configure Memory Controller to drive CKE low
Configure NVDIMM to Start Restore
Is Restore Complete?
No
Yes
Note Restore Status in MRC Output Structure
Rewrite RC registers
Assert CKE
Rewrite MRS Registers
Run MemTest and MemInit on all DIMMs that didn't have a successful restore
Note Error in output Structure
End of Restore Process

NVDIMM training slides provided by Intel

E820 Table Example

- E820 is shorthand to refer to the facility by which the BIOS of x86-based computer systems reports the memory map to the operating system or boot loader.



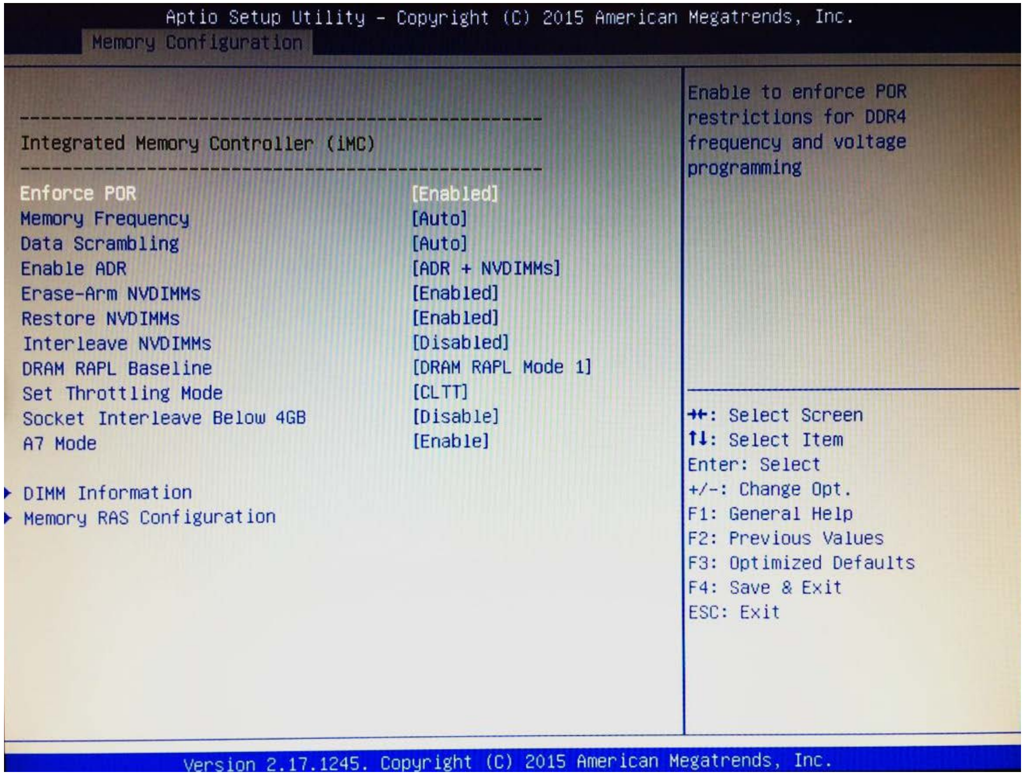
```
[root@localhost Desktop]# dmesg |grep e820
 BIOS-e820: 0000000000000000 - 000000000009ac00 (usable)
 BIOS-e820: 000000000009ac00 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000e0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000007d4a1000 (usable)
 BIOS-e820: 000000007d4a1000 - 000000007d4e0000 (reserved)
 BIOS-e820: 000000007d4e0000 - 000000007d5f6000 (ACPI data)
 BIOS-e820: 000000007d5f6000 - 000000007e1ff000 (ACPI NVS)
 BIOS-e820: 000000007e1ff000 - 000000007f271000 (reserved)
 BIOS-e820: 000000007f271000 - 000000007f272000 (usable)
 BIOS-e820: 000000007f272000 - 000000007f2f8000 (ACPI NVS)
 BIOS-e820: 000000007f2f8000 - 000000007f800000 (usable)
 BIOS-e820: 0000000080000000 - 0000000090000000 (reserved)
 BIOS-e820: 00000000fed1c000 - 00000000fed20000 (reserved)
 BIOS-e820: 00000000ff000000 - 0000000100000000 (reserved)
 BIOS-e820: 0000000100000000 - 0000000200000000 type 12
e820 update range: 0000000000000000 - 0000000000010000 (usable) ==> (reserved)
e820 update range: 0000000000000000 - 0000000000001000 (usable) ==> (reserved)
e820 remove range: 00000000000a0000 - 0000000000100000 (usable)
e820 update range: 0000000080000000 - 0000000100000000 (usable) ==> (reserved)
```

the nvdimm memory address arrange in e820 map

Note: ACPI 6.0 defines Type 7 for Persistent Memory and NFIT

Additional BIOS Considerations

- BIOS also presents various menu options to setup NVDIMM operation
- Examples:
  - Enable ADR
  - Enable ARM in BIOS
  - Enable RESTORE
  - Write Cache options



```
          Aptio Setup Utility - Copyright (C) 2015 American Megatrends, Inc.
   Memory Configuration

                                                       | Enable to enforce POR
 ------------------------------------------------      | restrictions for DDR4
 Integrated Memory Controller (iMC)                    | frequency and voltage
 ------------------------------------------------      | programming
 Enforce POR                    [Enabled]              |
 Memory Frequency               [Auto]                 |
 Data Scrambling                [Auto]                 |
 Enable ADR                     [ADR + NVDIMMs]        |
 Erase-Arm NVDIMMs              [Enabled]              |
 Restore NVDIMMs                [Enabled]              |
 Interleave NVDIMMs             [Disabled]             |
 DRAM RAPL Baseline             [DRAM RAPL Mode 1]     |
 Set Throttling Mode            [CLTT]                 |
 Socket Interleave Below 4GB    [Disable]              | →←: Select Screen
 A7 Mode                        [Enable]               | ↑↓: Select Item
                                                       | Enter: Select
► DIMM Information                                     | +/-: Change Opt.
► Memory RAS Configuration                             | F1: General Help
                                                       | F2: Previous Values
                                                       | F3: Optimized Defaults
                                                       | F4: Save & Exit
                                                       | ESC: Exit

          Version 2.17.1245. Copyright (C) 2015 American Megatrends, Inc.
```

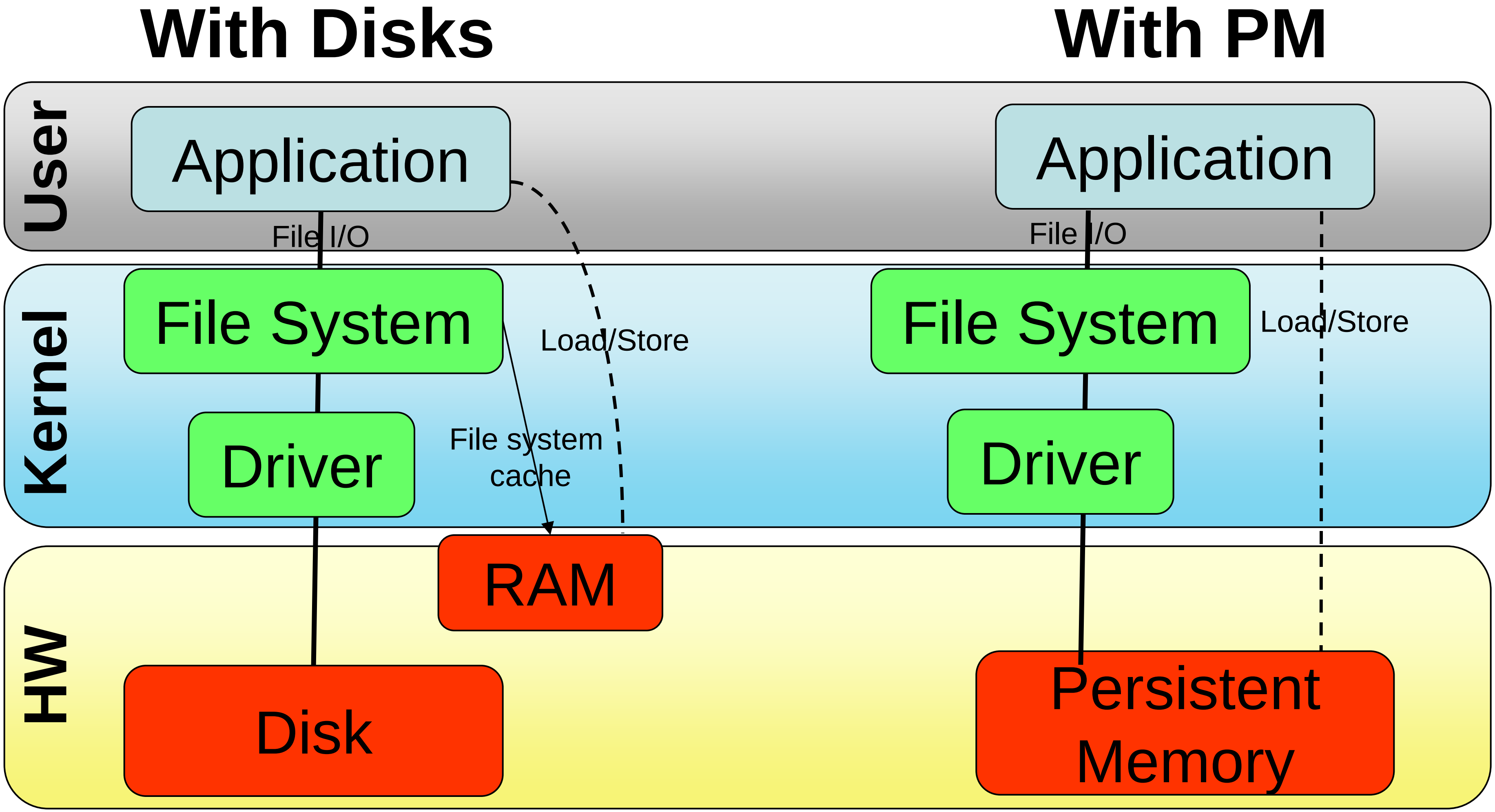
Legacy vs JEDEC I2C Register Implementation

- BIOS implementations for DDR3 platforms and prior were specific to an NVDIMM vendor’s command set (although high level commands were common)
- Early DDR4 platforms follow this same basic method. BIOS with MRC 1.10 to 1.14 all have Vendor Specific I2C support
- MRC with JEDEC I2C Register Support include BIOS support for ACPI 6.0, NFIT (NVDIMM Firmware Interface Table), and DSM (Driver Specific Method), cf. http://pmem.io
- Systems starting to launch now that use the JEDEC I2C command set

SNIA™
Global Education

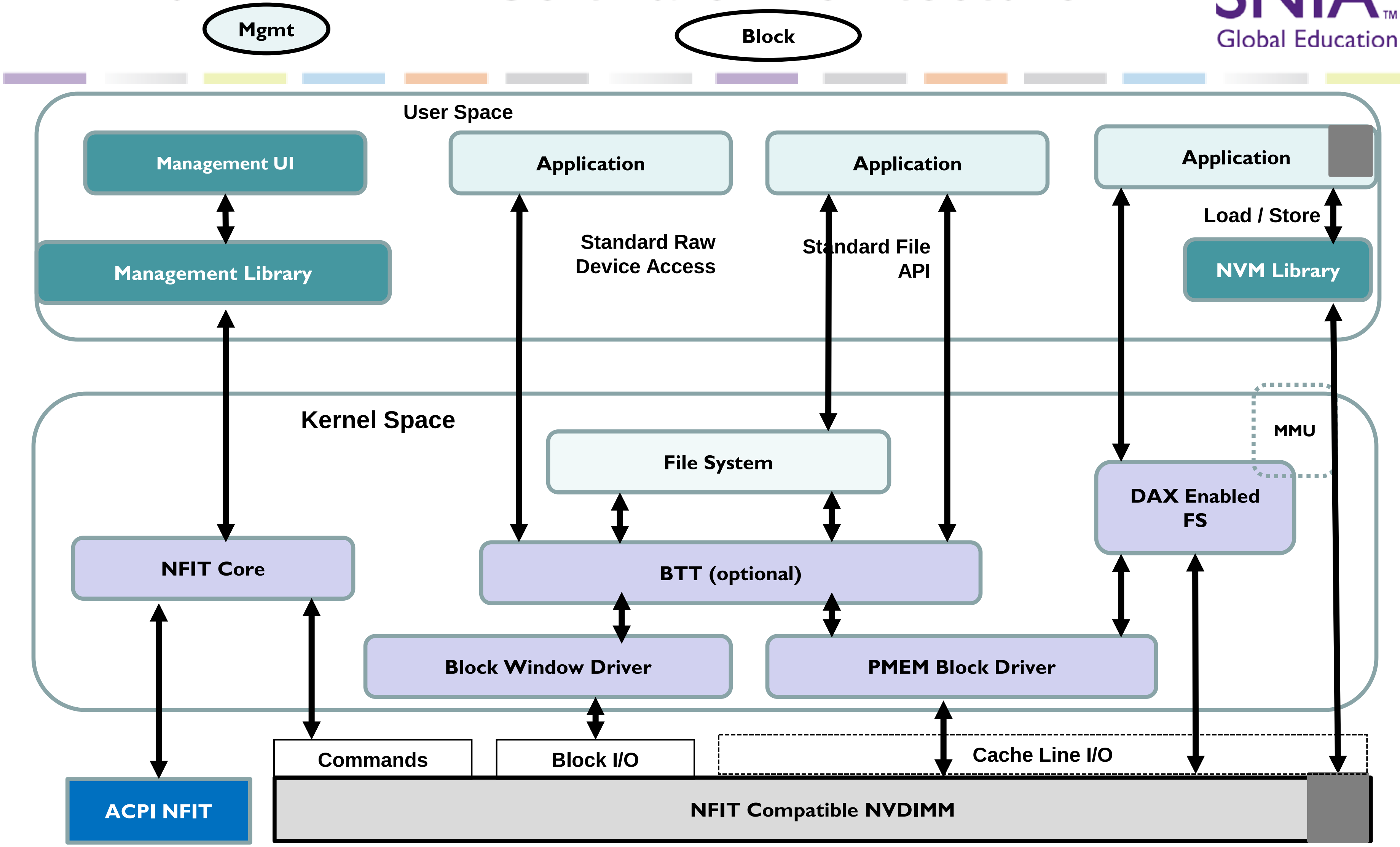
Part 3
OS
(Linux & Microsoft)

Memory Mapped File Programming Model



Source; SNIA NVM Programming TWG

Linux NVDIMM Software Architecture



Mgmt

Block

User Space

Management UI

Management Library

Application

Standard Raw Device Access

Application

Standard File API

Application

Load / Store

NVM Library

Kernel Space

MMU

File System

DAX Enabled FS

NFIT Core

BTT (optional)

Block Window Driver

PMEM Block Driver

Commands

Block I/O

Cache Line I/O

ACPI NFIT

NFIT Compatible NVDIMM

4.2 Kernel

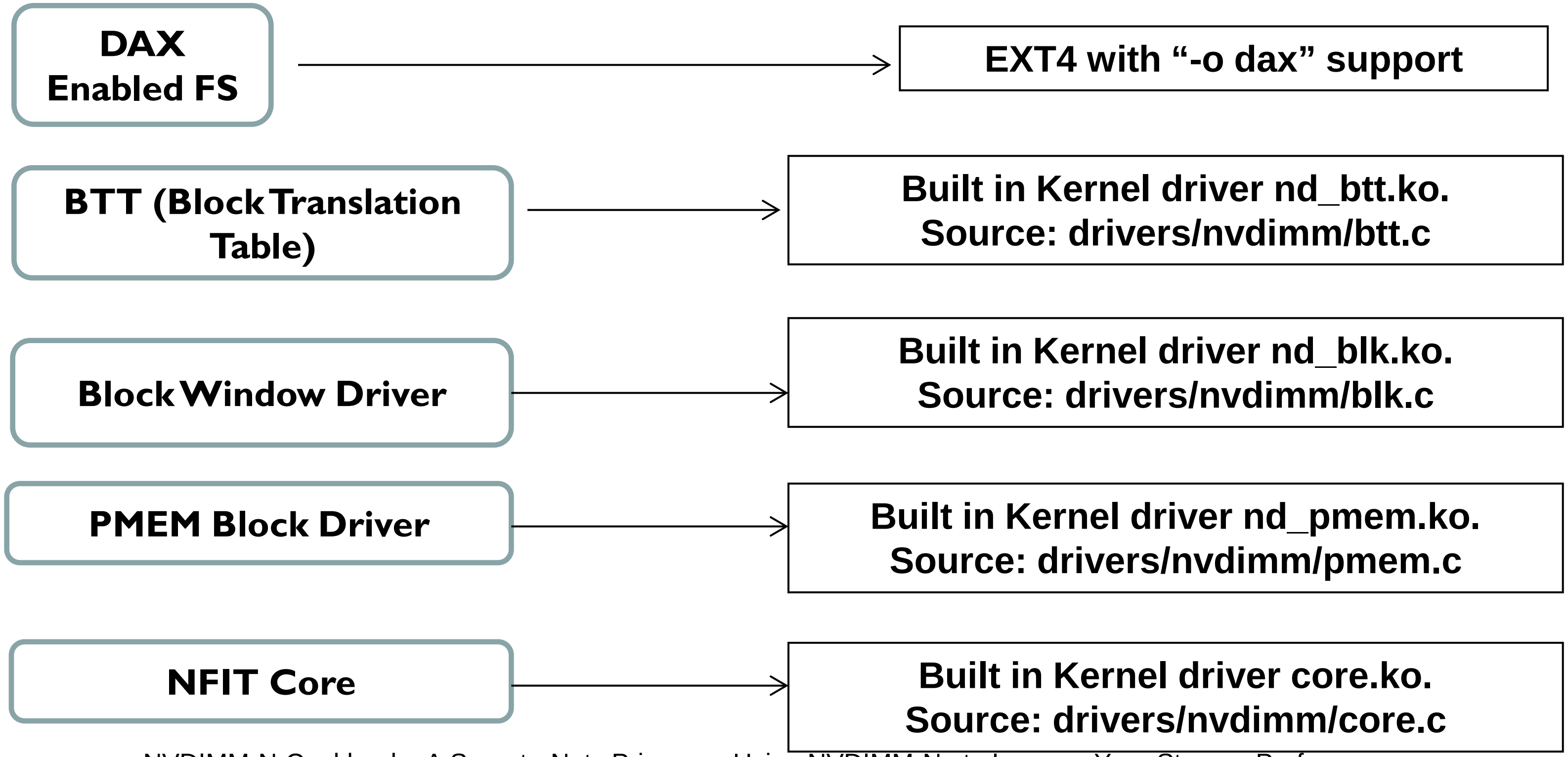
Intel® GIT Hub

ACPI 6.0 Compatible

Existing File Systems

What’s available in Linux 4.4 Kernel?

- Linux 4.4 subsystems added and modified in support of NVDIMMs
- Core Kernel support for ACPI 6.0 with NFIT BIOS, Device Drivers, Architectural Code, and File System with DAX support (ext4)
- Distributions (Open Source Initiatives)
  - Ubuntu 16.04 LTS (4.4 Kernel)
  - Fedora 23 (4.2.0 Kernel)



DAX Enabled FS → **EXT4 with “-o dax” support**

BTT (Block Translation Table) → **Built in Kernel driver nd_btt.ko. Source: drivers/nvdimm/btt.c**

Block Window Driver → **Built in Kernel driver nd_blk.ko. Source: drivers/nvdimm/blk.c**

PMEM Block Driver → **Built in Kernel driver nd_pmem.ko. Source: drivers/nvdimm/pmem.c**

NFIT Core → **Built in Kernel driver core.ko. Source: drivers/nvdimm/core.c**

Microsoft - NVDIMM-N OS Support

Microsoft

- At this year's //Build conference MS made public that Windows Server 2016 supports JEDEC-compliant DDR4 NVDIMM-N
  - https://channel9.msdn.com/Events/Speakers/tobias-klima
- Technical Preview 5 of Windows Server 2016, has NVDIMM-N support
  - https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-technical-preview)

Part 4
System Implementations & Use Cases

Examples of NVDIMM Systems
Intel DDR4

	2015	2016	2017
Processor	Haswell Grantley	Broadwell Grantley	Skylake Purley
Memory Speeds	DDR4-2133	DDR4-2133 DDR4-2400	DDR4-2400 DDR4-2666
Single Socket, 8 DIMMs Half-width compute module		Adams Pass S7200AP	Buchannan Pass S2600BP
Dual Socket, 24 DIMMs 1U/2U Rack Optimized Server	Wildcat Pass S2600WT	→	Wolf Pass S2600WF
Dual Socket, 8/16 DIMMs Half-width 2U Node	Taylor Pass S2600TP	→	Buchannan Pass S2600BP
Dual Socket, 8/16 DIMMs Half-width 2U Node	Kennedy Pass S2600KP	→	
Dual Socket 16 DIMMs 4U Pedestal Chassis	Cottonwood Pass S2600CW	→	Sawtooth Pass S2600ST

Examples of NVDIMM Systems Supermicro DDR4

Market Segment	X10 Model	Available Configurations
Channel	X10DRC/i-LN4+/T4+, X10DRi(T), X10DRX, X10DRH-C/I(T), X10DRH-C/iLN4	Motherboard, barebones or complete server
Enterprise	X10DRU-i+ (Ultra Series)	Complete server-only
HPC	X10DRT-H/HIBF, X10DRT-P/PT/PIBF, X10DRT-L/LIBQ/LIBF, X10DRT-PS, X10DRFR(N)(T), X10DRFF(-C), X10DRFF(C/TG)	Motherboard or complete server
Data Center	X10DRD-L/I(N)T, X10DRD-LTP/I(N)TP, X10DDW-I(N), X10DRW-I(T), X10DRW-E/N(T)	Motherboard, barebones or complete server
Storage	X10DRS-2U/3U/4U, X10DSC+, X10DSC-TP4S, X10DRH-C/I(T), X10DRH-C/iLN4	Motherboard, barebones or complete server
GPU	X10DRG-Q	Motherboard, barebones or complete server

NVDIMM-N DDR4 Platform Energy Source Options

- JEDEC JC45.6 Byte Addressable Energy Backed Interface

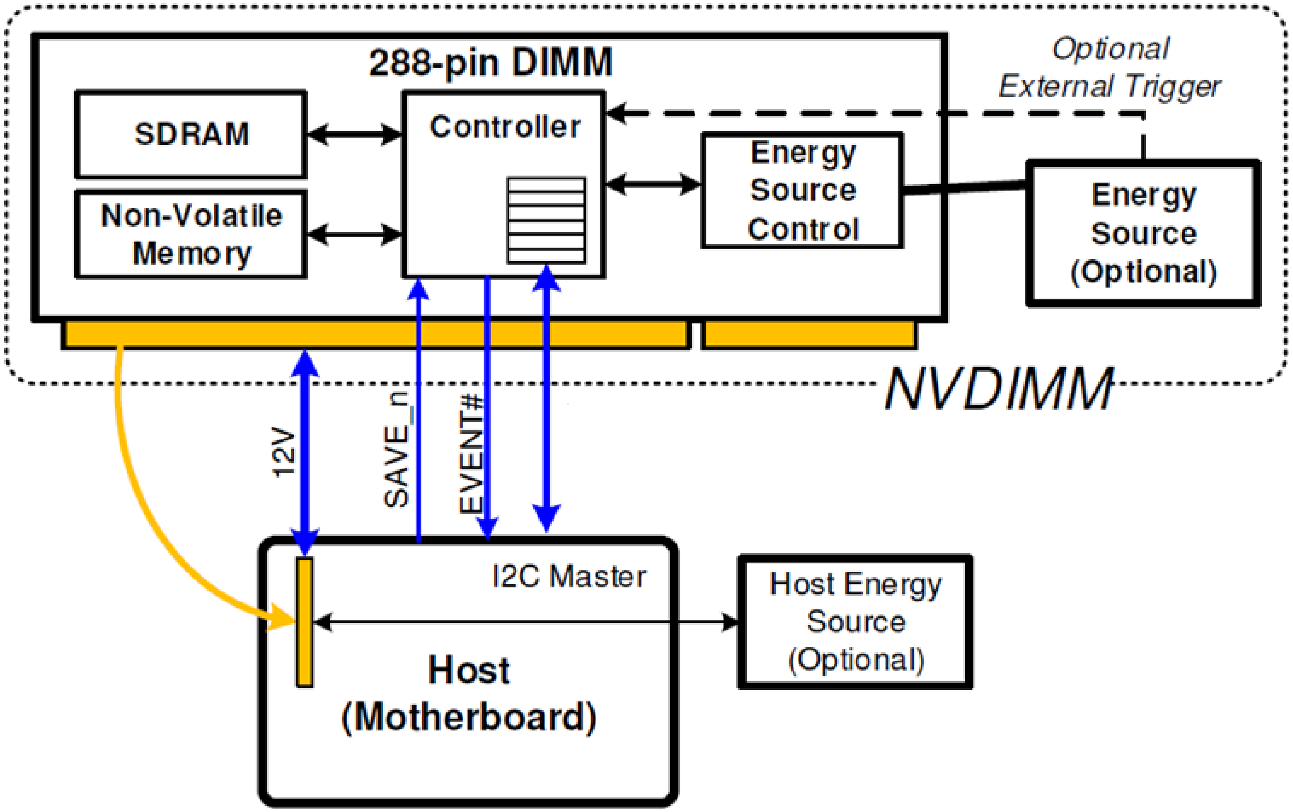


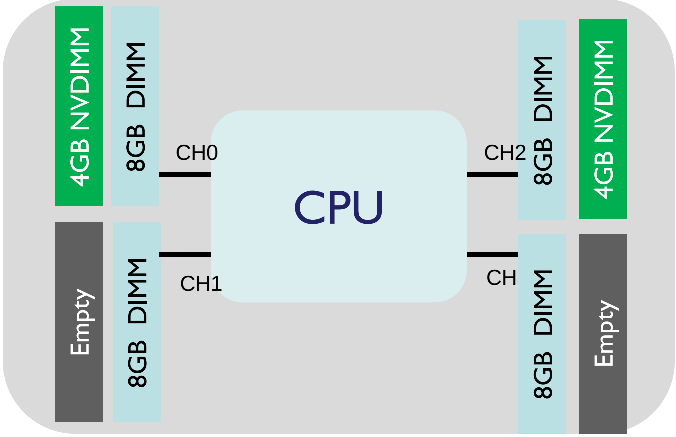
Figure 1: NVDIMM overview

Population Rules

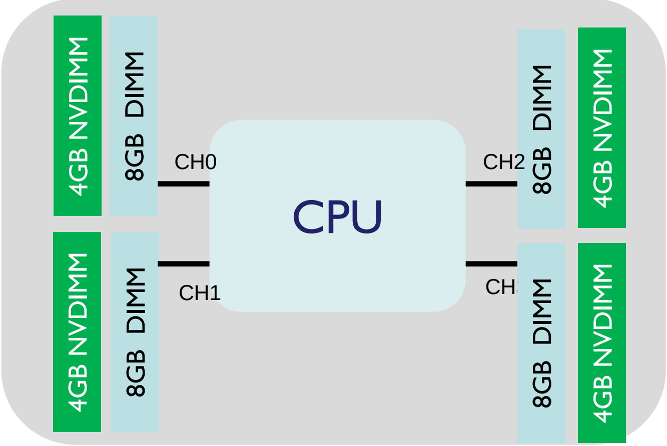
- There are no NVDIMM specific population rules
  - Normal DIMM population rules still apply(ex RDIMMs and LRDIMMs can’t be mixed)
  - NVDIMMs and normal DIMMs may be mixed in the same channel
  - NVDIMMs from different vendors may be mixed in the same system and even the same channel.
- How the DIMMs are installed in a system will affect performance, so thought should be put into how DIMMs are populated
- NVDIMM population tips
  - Interleaving DIMMs within a channel provides a very **small** performance benefit
  - Interleaving DIMMS across a channel provides a very **large** performance benefit
  - Two DIMMs of the same type should not be installed in the same channel unless all other channels in the system have at least one of that type DIMM.

Source; Intel

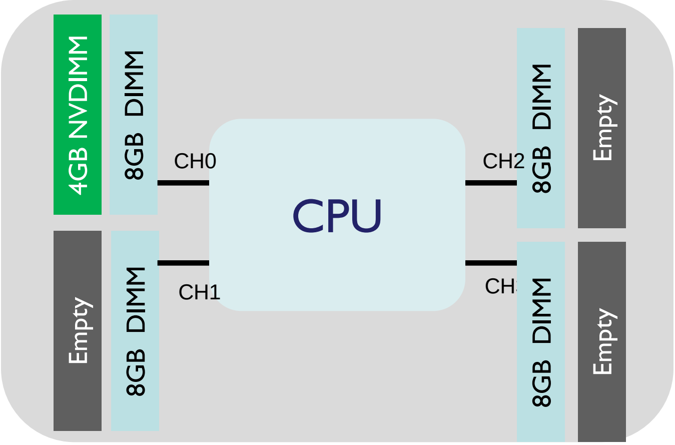
Example Optimal Interleaves



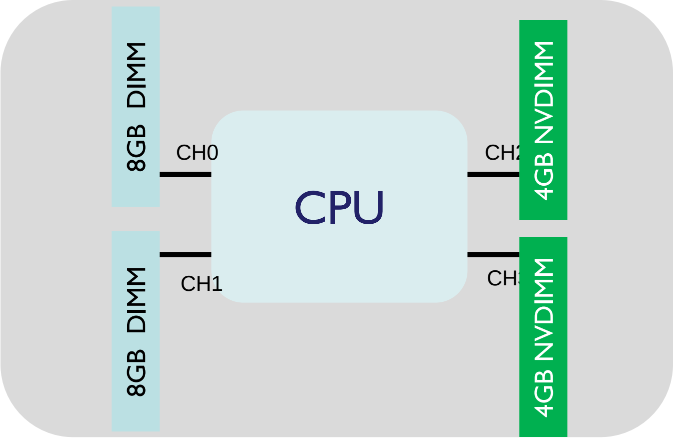
Has a 4-way Interleave between normal DIMMs, and optionally a 2-way interleave between the NVDIMMs



Has a 4-way Interleave between normal DIMMs, and optionally a 4-way interleave between the NVDIMMs



Has a 4-way Interleave between normal DIMMs



Has a 2-way Interleave between normal DIMMs, and optionally a 2-way interleave between the NVDIMMs

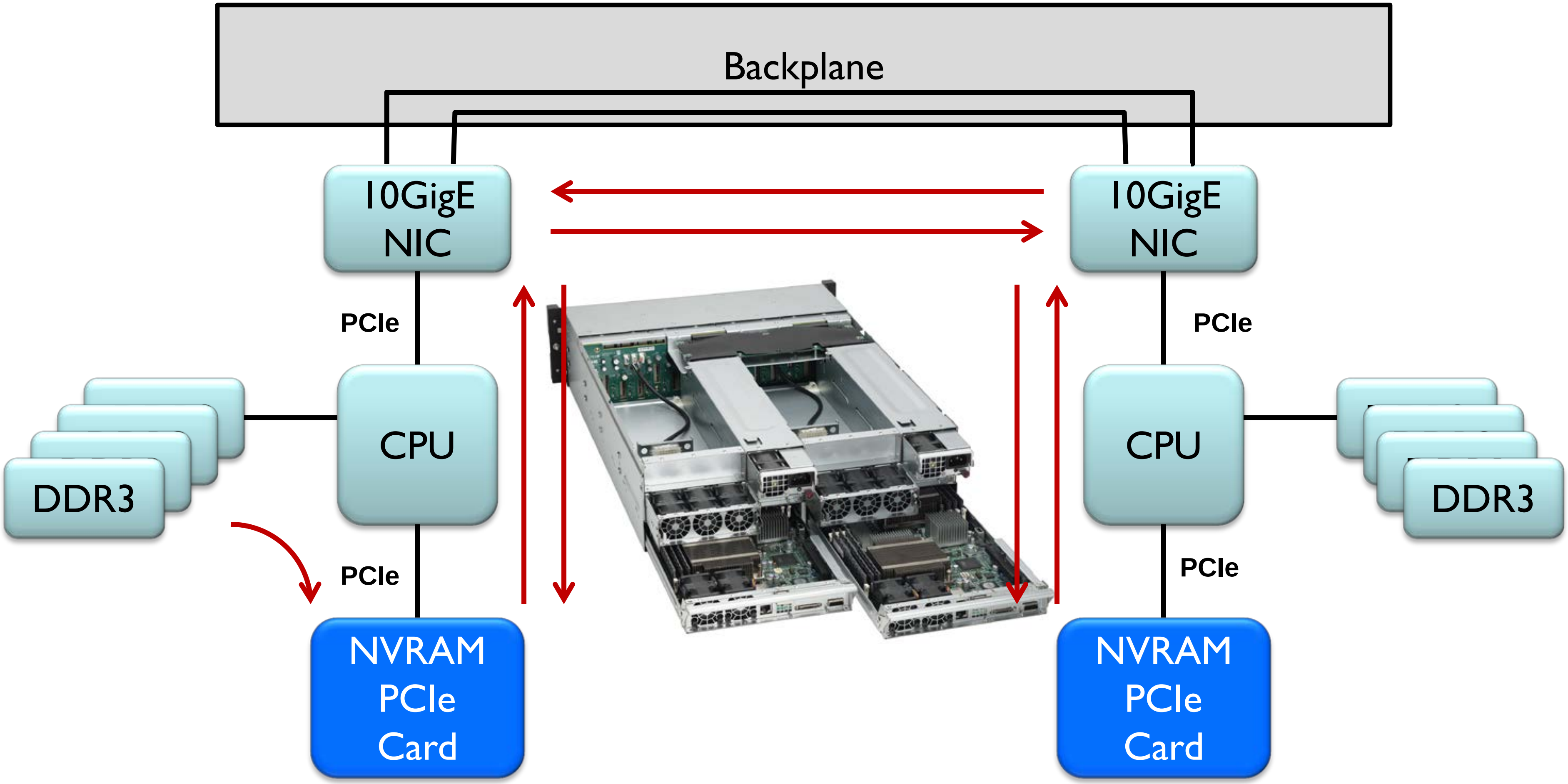
Source; Intel

Use Cases

- ***In Memory Database:*** Journaling, reduced recovery time, Ex-large tables
- ***Traditional Database:*** Log acceleration by write combining and caching
- ***Enterprise Storage:*** Tiering, caching, write buffering and meta data storage without an auxiliary power source
- ***Virtualization:*** Higher VM consolidation with greater memory density
- ***High-Performance Computing:*** Check point acceleration and/or elimination
- ***NVRAM Replacement***: Higher performance enabled by removing the DMA setup/teardown
- ***Other***: Object stores, unstructured data, financial & real-time transactions

Adapted from SNIA presentations by HP

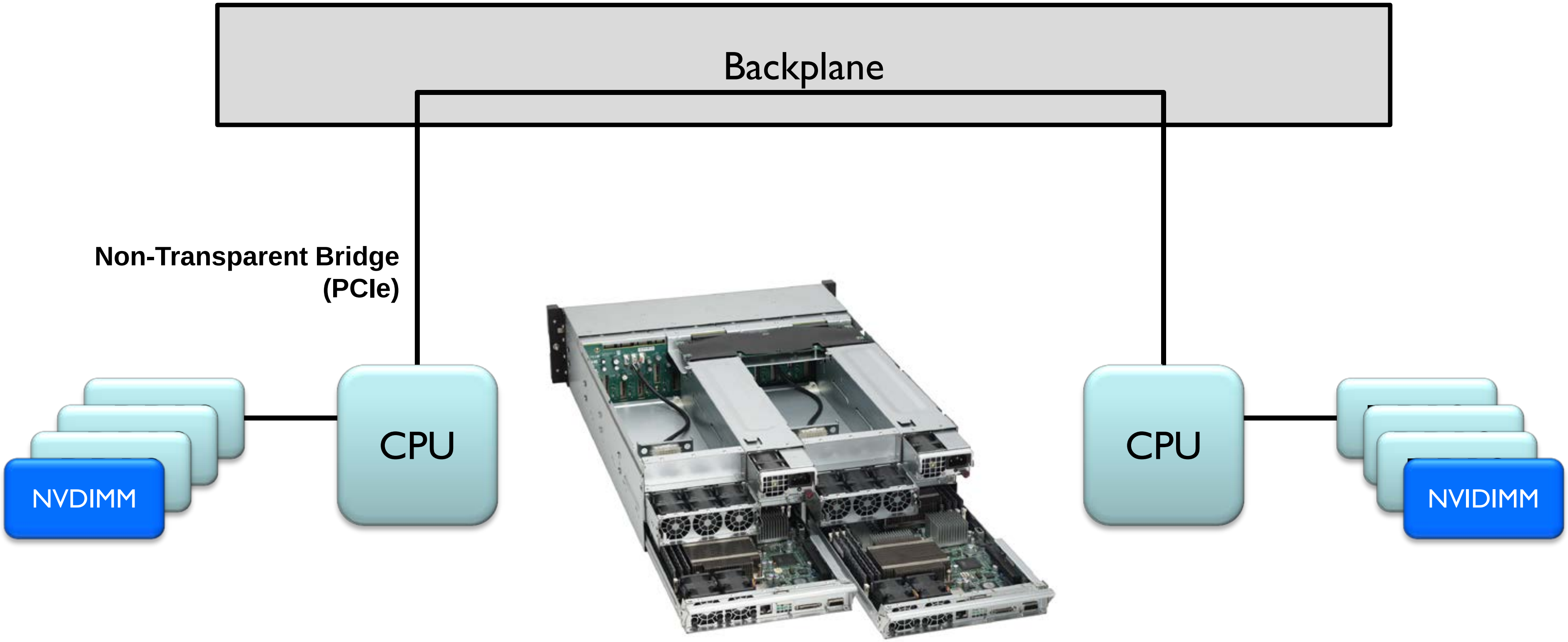
Application Example: Storage Bridge Bay (SBB)



Shadow Writes Required for Failover

Adapted from SNIA presentations by AgigA Tech

SBB: A Simpler/Better/Faster Way



Also a better alternative to Cache-to-Flash implementations:

- Separate failure domain
- No battery maintenance
- System hold-up requirements significantly less severe
- 4x write latency performance improvement

Adapted from SNIA presentations by AgigA Tech

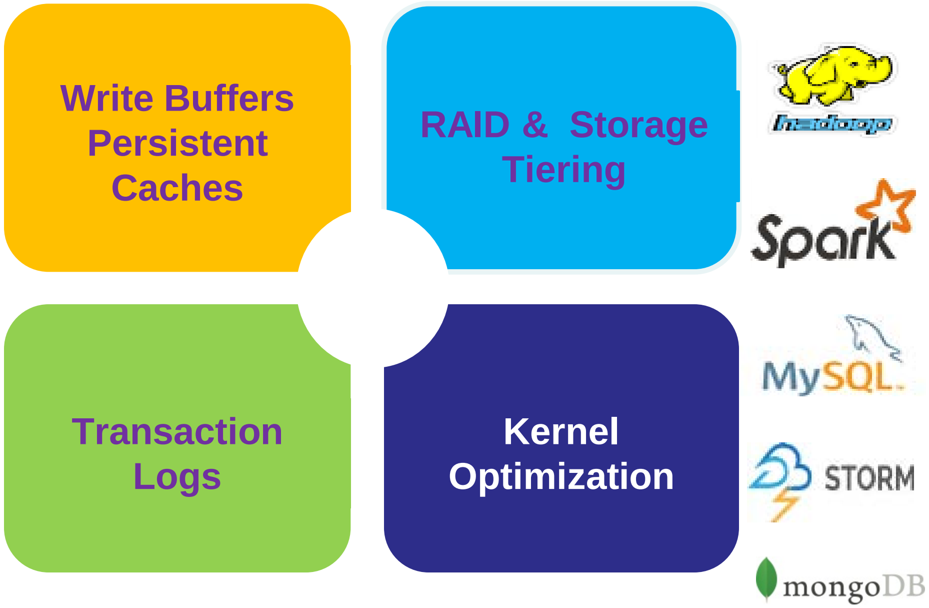
Advantages of NVDIMMs for Applications

Legacy HDD/SSD Solution

- Persistent data stored in HDD or SSD tiers
- Slow & unpredictable software stack

NVDIMM Solution

- Persistent data stored in fast DRAM tier
- Removes software stack from data-path



Accelerates SW-Apps !

- DRAM class latency & thru-put for persistent data
  - 1000X lower latency
  - 10X+ throughput increase

•The value is in application acceleration

Thank You!

Attribution & Feedback

The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History

Jeff Chang/Arthur Sainio - June 2015
Arthur Sainio Revised - August 2016

Additional Contributors

Mario Martinez - July 2015

Please send any questions or comments regarding this SNIA Tutorial to ***tracktutorials@snia.org***