PENNSTATE

1855

# ***NANDFlashSim*** : Intrinsic Latency Variation Aware NAND Flash Memory System Modeling and Simulation at Microarchitecture Level

**Myoungsoo Jung (MJ)**,
Ellis H. Wilson III, David Donofrio, John Shalf, Mahmut T. Kandemir

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

# Agenda

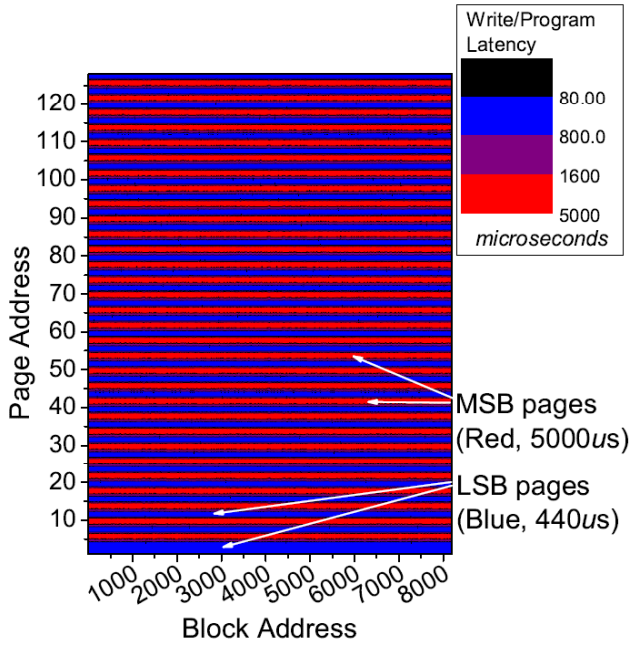
- Revisiting NAND flash technology
- Advance NAND flash operations
- NANDFlashSim
- Evaluation

# Intrinsic Latency Variation

- Fowler-Nordheim Tunneling
  - Making an electron channel
  - Voltage is applied over a certain threshold
- Incremental step pulse programming (ISPP)

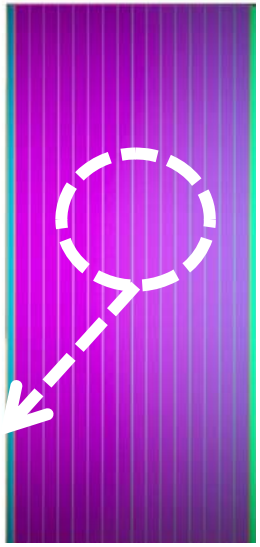
# Intrinsic Latency Variation

- Each step of ISPP needs different programming duration (latency)
- Latencies of the NAND flash memory fluctuate depending on the address of the pages in a block



# NAND Flash Architecture

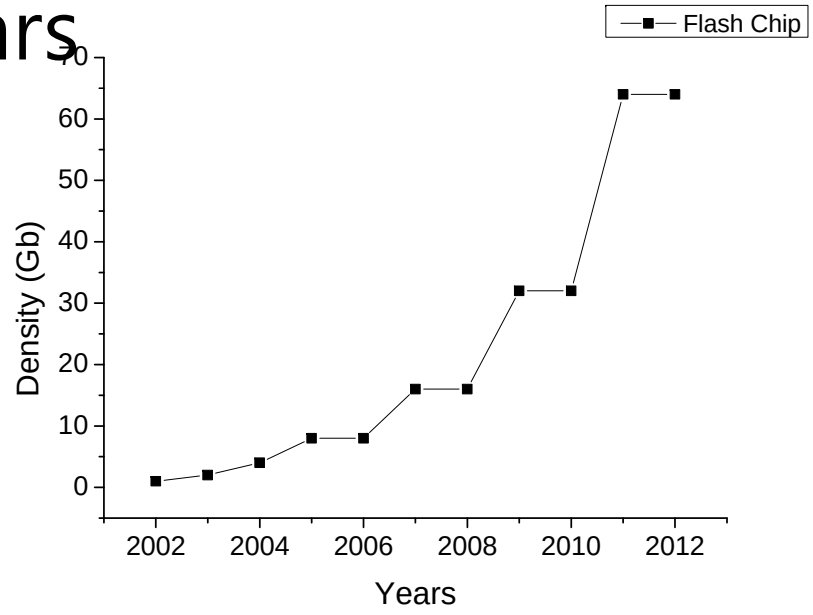
- Employing cache and data registers
- Multiple planes (memory array)
- Multiple dies



*Memory Array (Plane)*

# Density Trend

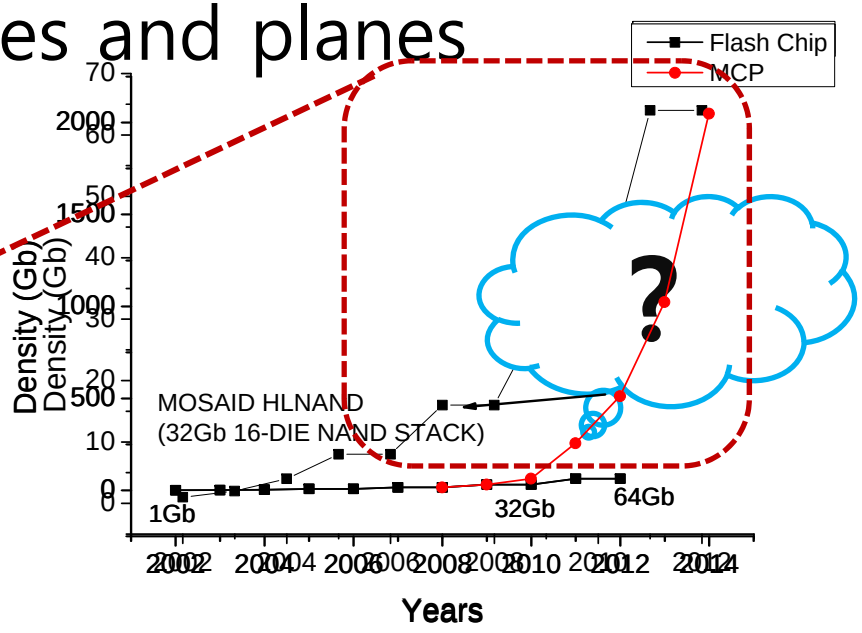
- Flash technology
  - Each cell is capable to store multiple bits
  - manufacturing feature size is scaling down
- So far, density is increasing by two to four times every 2 years



# Density Trend

- Shrinking manufacturing feature size might be limited around 12 nanometer
- Multi-die stack technology
  - Flash packages continue to scale up by employing multiple dies and planes

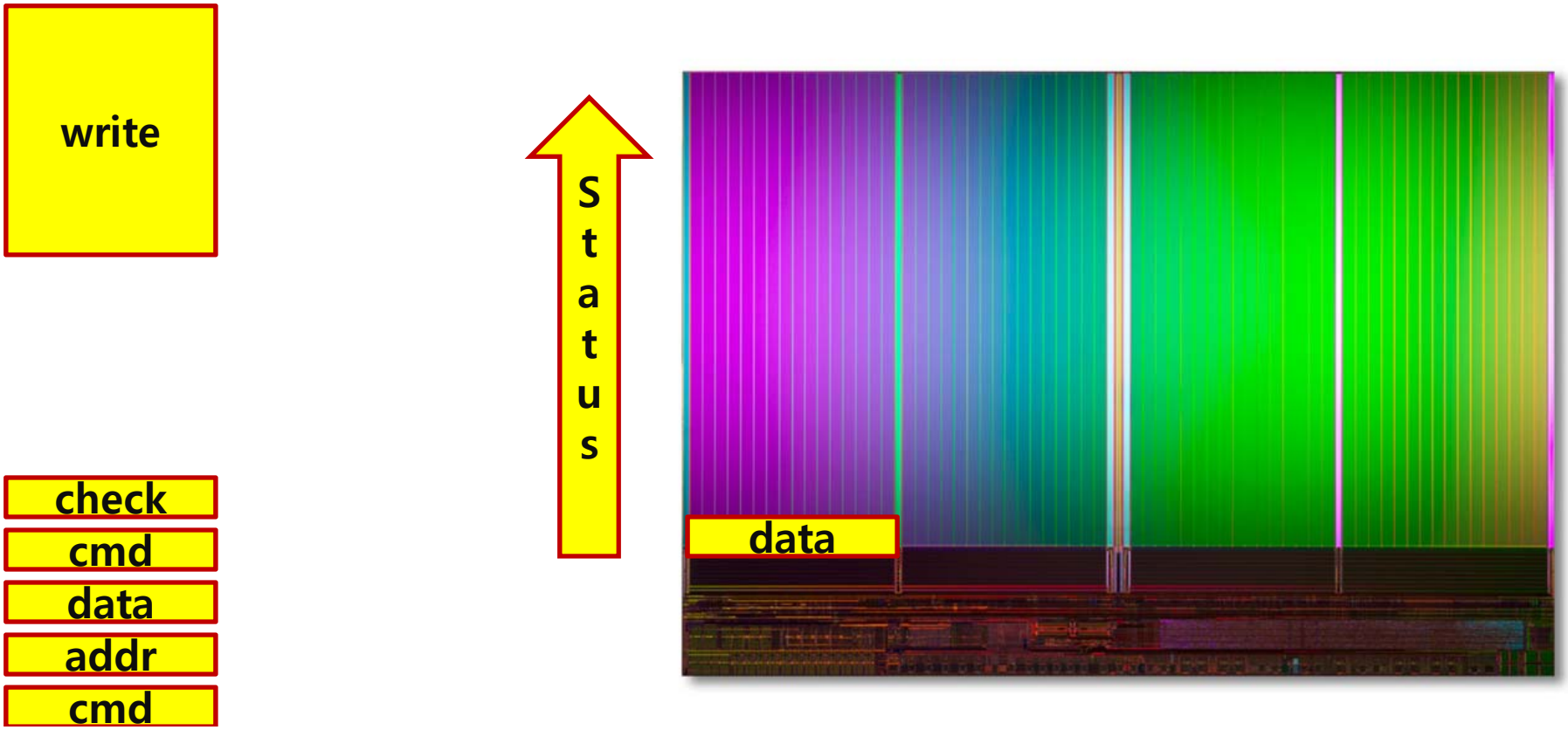
***How does performance behavior change?***



# Advance NAND Flash Operation

# Legacy Operation

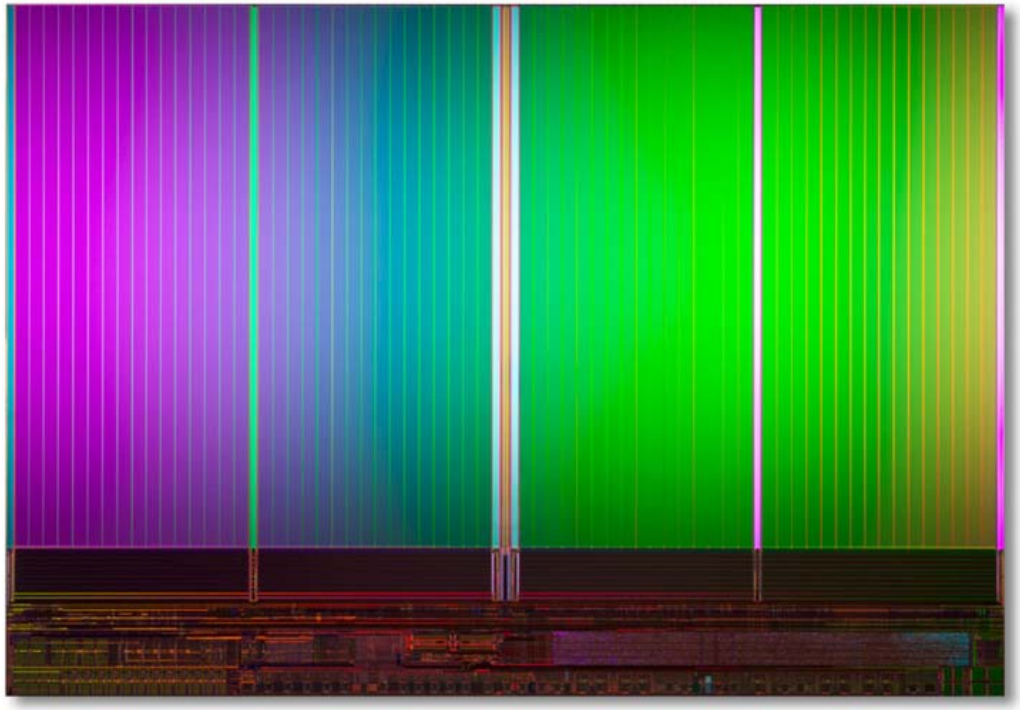
- An I/O operation splits into several operation stages
- Each stage should be appropriately handled by device drivers



# Cache Operation

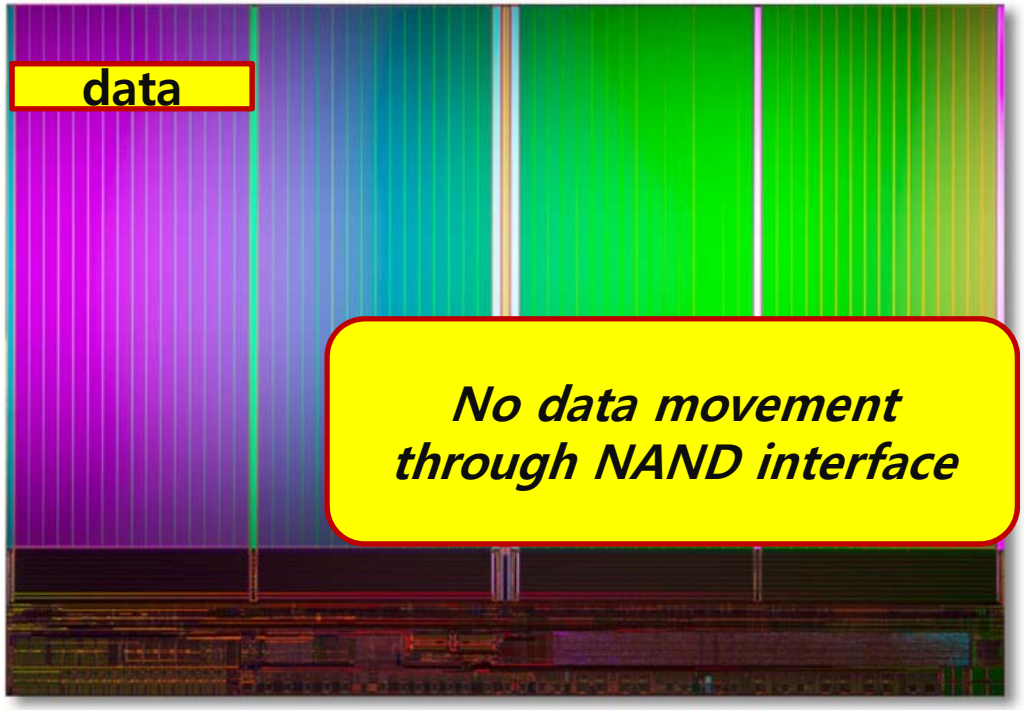
- Cache mode operations use internal registers in an attempt to hide performance overhead from data movements

data2
data1



# Internal Data Move Mode

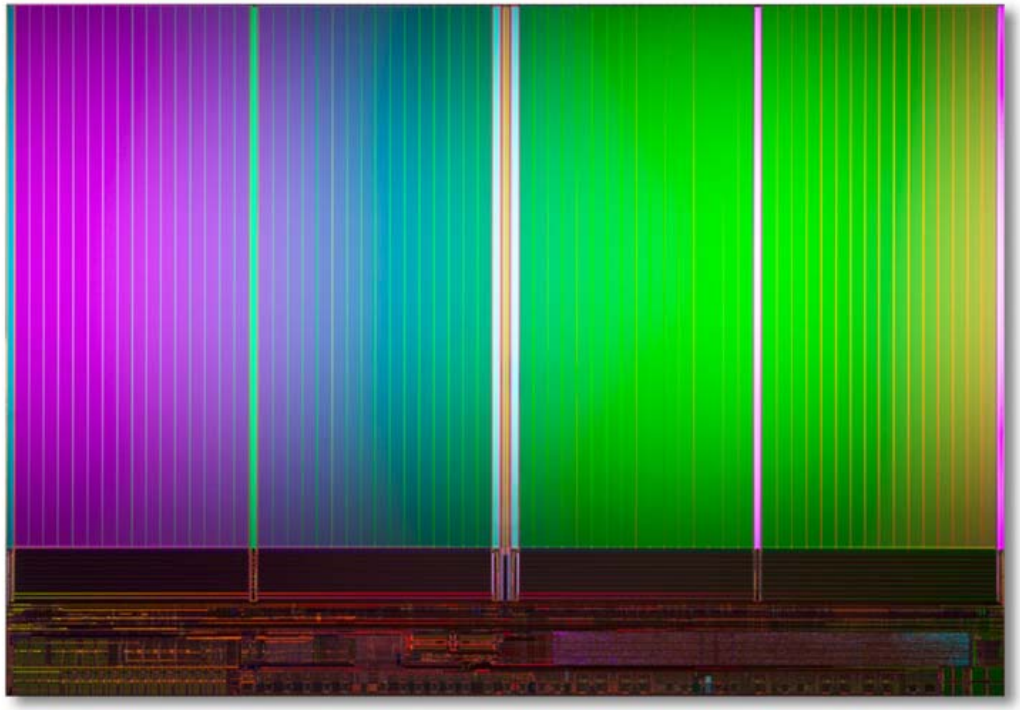
- Saving space and cycles to copy data
- Source and destination page address should be located in the same die




cmd
dst addr
src addr
cmd

# Multi-plane Mode Operation

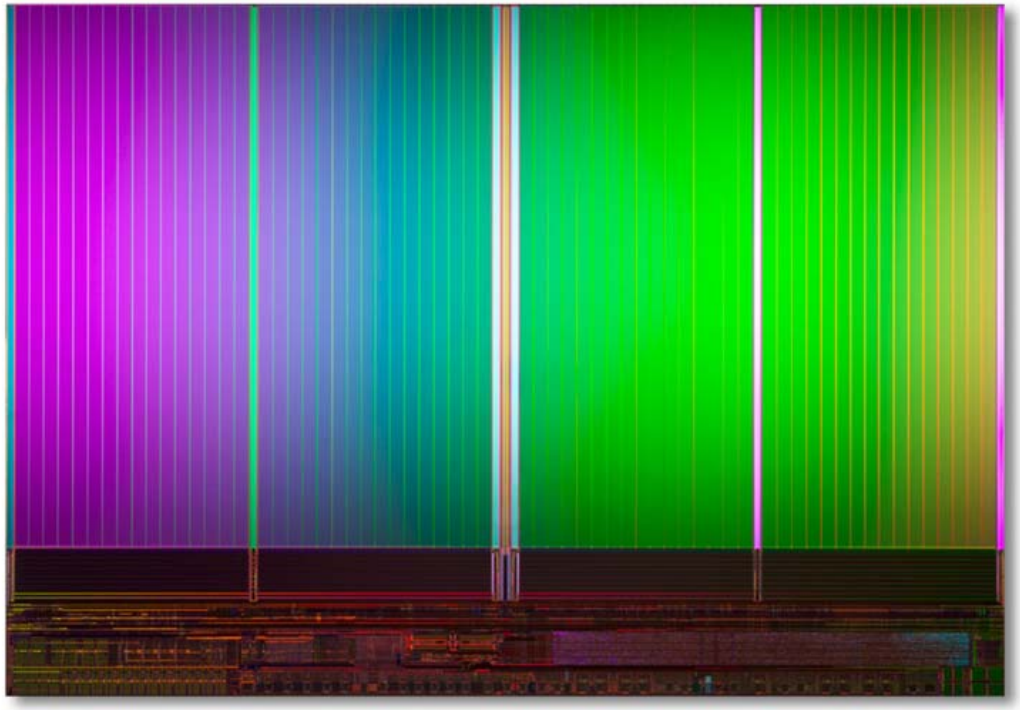
- Two different pages can be served in parallel
- Addresses should indicate same page offset in a block, same die address and should have different plane addresses (*plane addressing rule*)



**check**
**cmd**
**data2**
**data1**
**cmd**
**addr2**
**addr1**
**cmd**

# Interleaved-Die Mode Operation

- providing a way, taking advantage of internal parallelism by interleaving NAND transactions
- Scheduling NAND transactions and bus arbitrations are critical dominant of memory system performance



cmd
data1
addr1
cmd

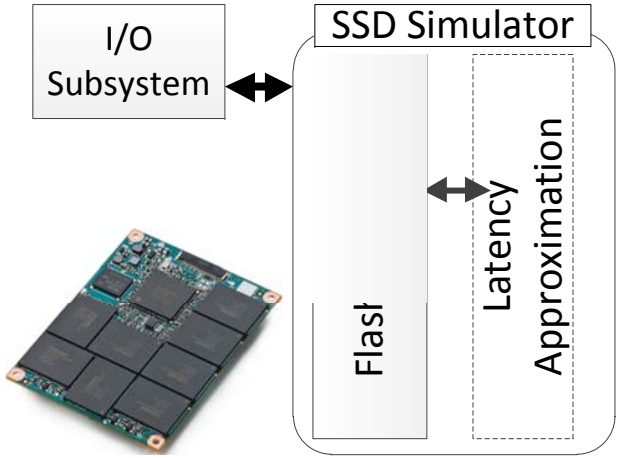
cmd
data2
addr2
cmd

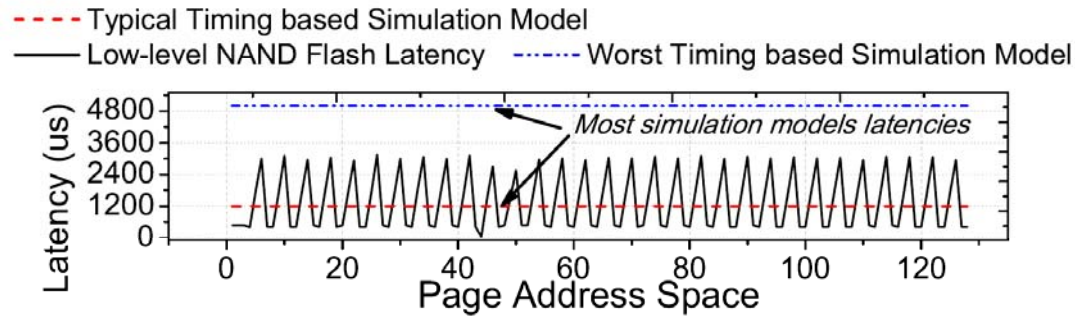
# Challenges

- Performances are varied based on:
  - intrinsic latency variation characteristic
  - internal parallelism
  - advanced flash operations types
- Performances are affected by
  - how to deal with diverse advance flash operations
  - how to effectively schedule NAND transactions

# Prior Simulation Works

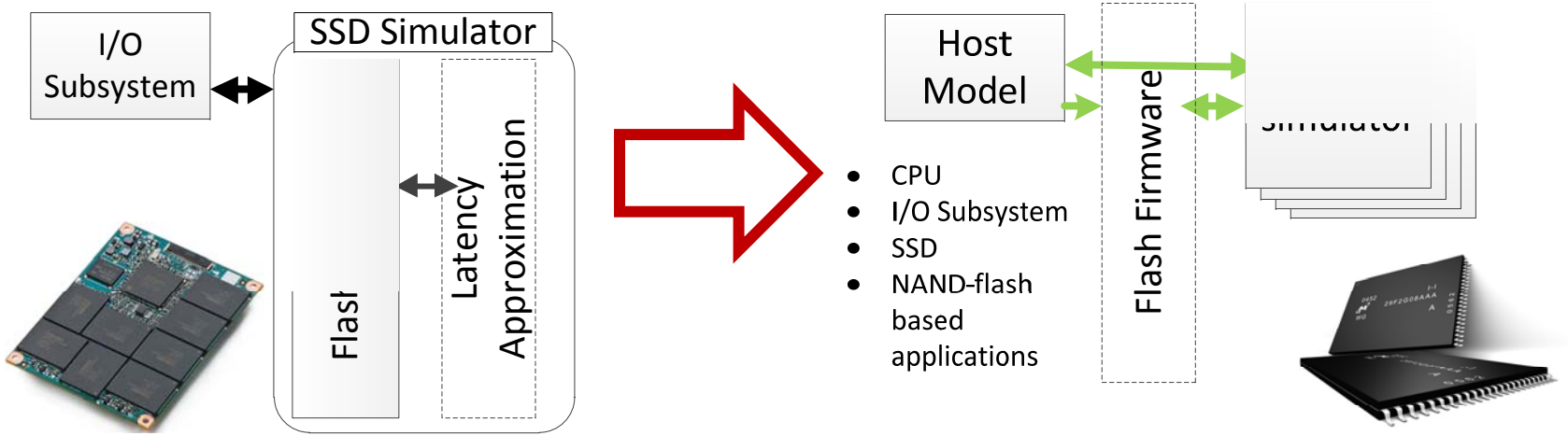
- Flash-based Solid State Disks Simulation
  - Tightly coupled to specific flash firmware
- Unaware of latency variation of NAND flash
  - Latency approximation model with *constants*
- Course-grain NAND command handling
  - In-order execution



# NANDFlashSim

- Simulating and Modeling NAND flash rather than flash firmware or SSDs
  - NANDFlashSim can be applied to diverse application like off-chip caches of a multi-core system and I/O subsystems of mobile systems
  - Multiple instances can be used for building SATA, PCI-e based SSDs

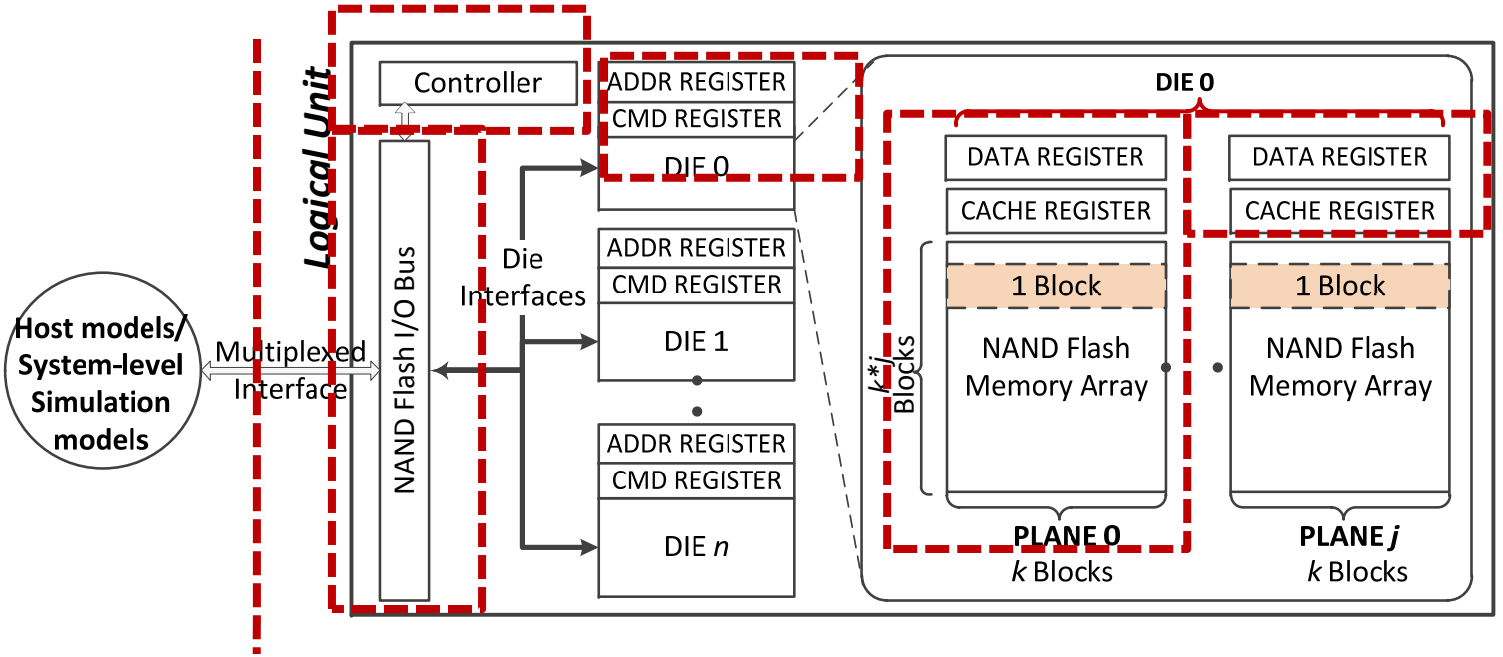


# NANDFlashSim

- Detailed Timing Model
- Awareness of intrinsic latency variation
  - designed to be performance variation-aware and employs different page offsets in a physical block
- Reconfigurable Microarchitecture
  - Supports highly reconfigurable architectures in terms of multiple dies and planes
- Fine-grain NAND flash command handling
  - 16 combinations of advance flash operation
  - Supporting out-of-order execution

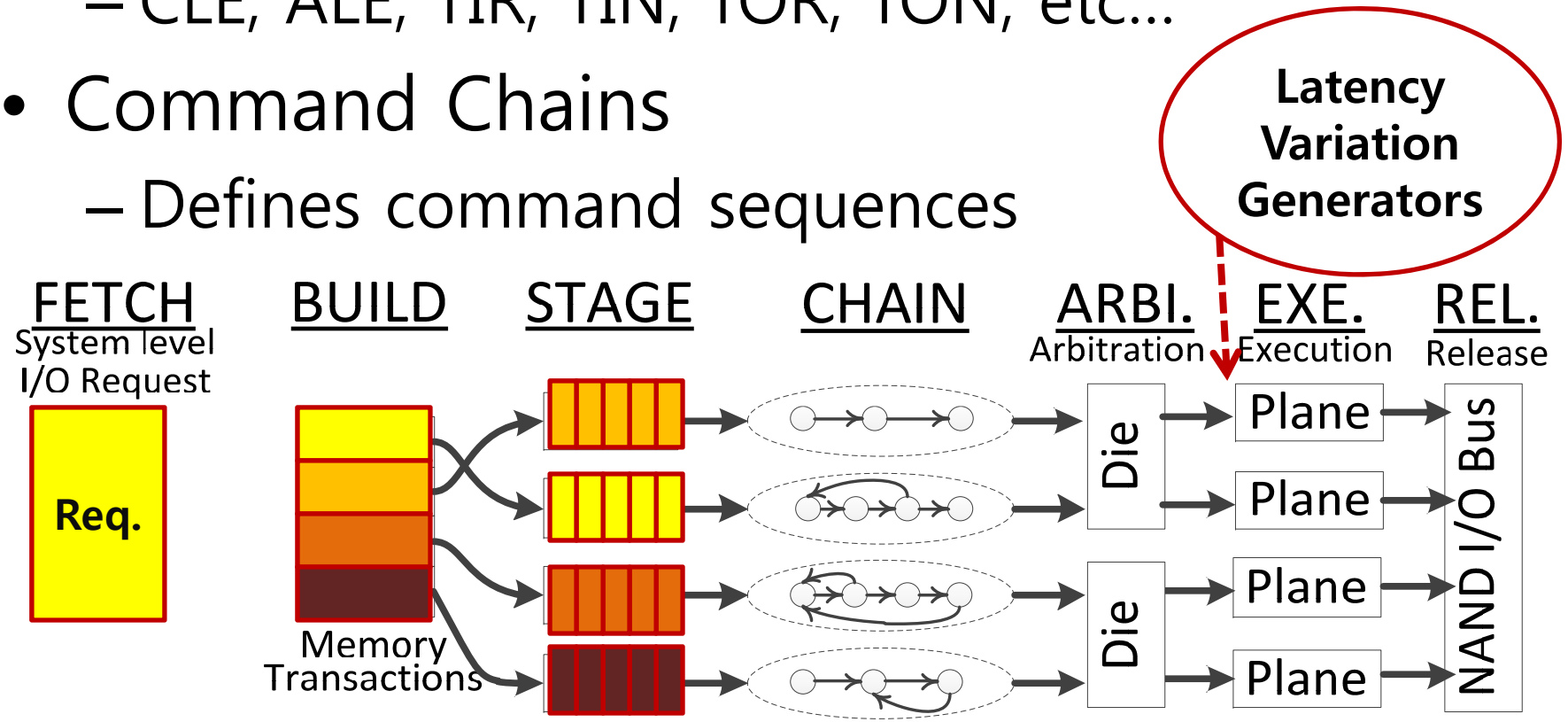
# High-level View

- Command set architecture and individual state machine associated with it
- Host and NAND flash clock domain are separate.
- All entries (controller, register, die, …) are updated at every cycles



# Command Set Architecture

- Multi-stage Operation
  - Stage are defined by common operations
  - CLE, ALE, TIR, TIN, TOR, TON, etc…
- Command Chains
  - Defines command sequences



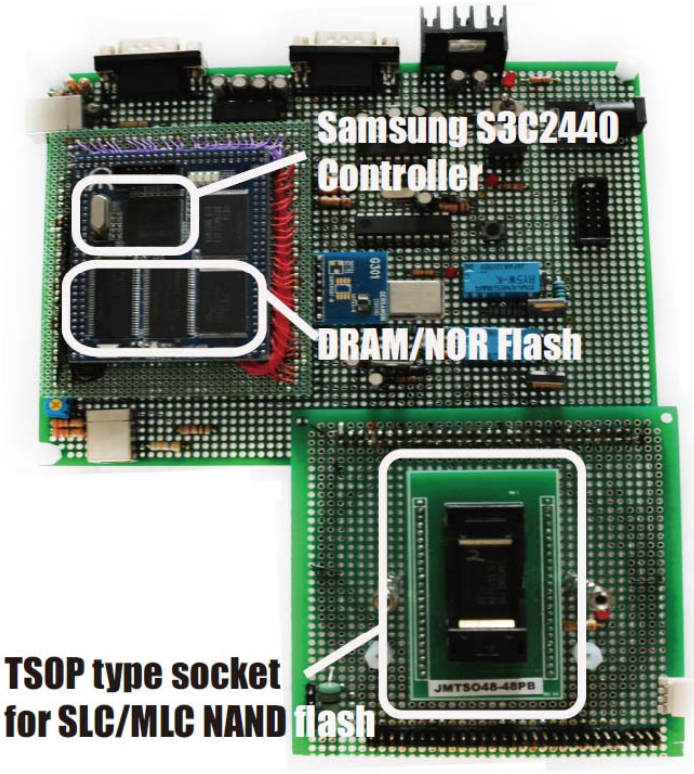
# Evaluation Methodology

| Device Type | Feature | Value |
|---|---|---|
| Single Level Cell | Page Size(Byte) | 2048 |
| | # of Page Per Block | 64 |
| | # of Block | 4096 |
| | Write Latency(us) | 250 |
| | Read Latency(us) | 25 |
| | Erase Latency(us) | 1500 |
| Multi Leve Cell 1 (MLC1) | Page Size(Byte) | 2048 |
| | # of Page Per Block | 128 |
| | # of Block | 8196 |
| | Write Latency(us) | 250~2200 |
| | Read Latency(us) | 50 |
| | Erase Latency(us) | 2500 |
| Multi Leve Cell 2 (MLC2) | Page Size(Byte) | 8192 |
| | # of Page Per Block | 256 |
| | # of Block | 8196 |
| | Write Latency(us) | 440~5000 |
| | Read Latency(us) | 200 |
| | Erase Latency(us) | 2500 |

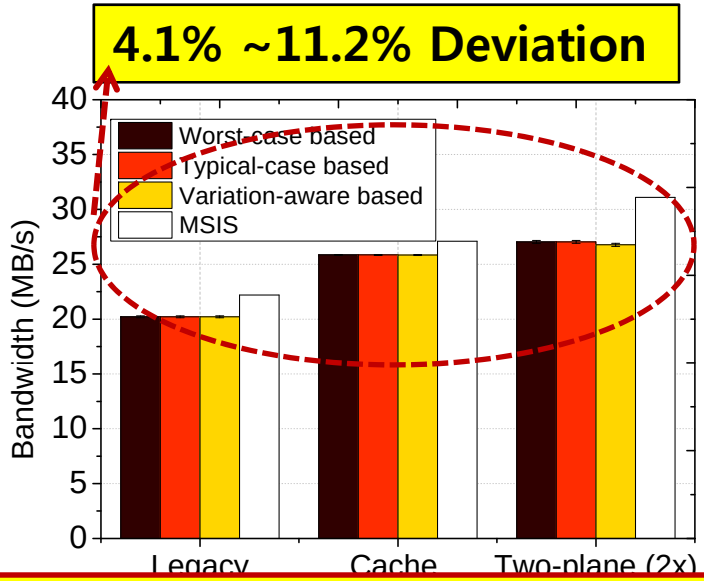
SAMSUNG

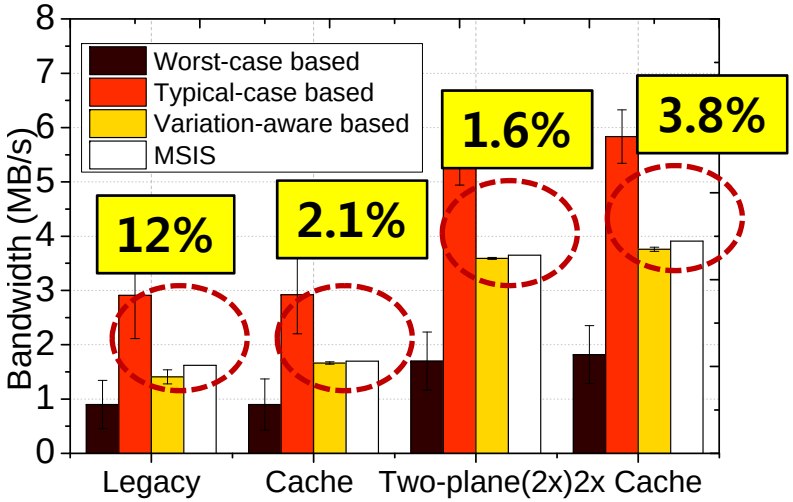
MICRON

SK HYNIX

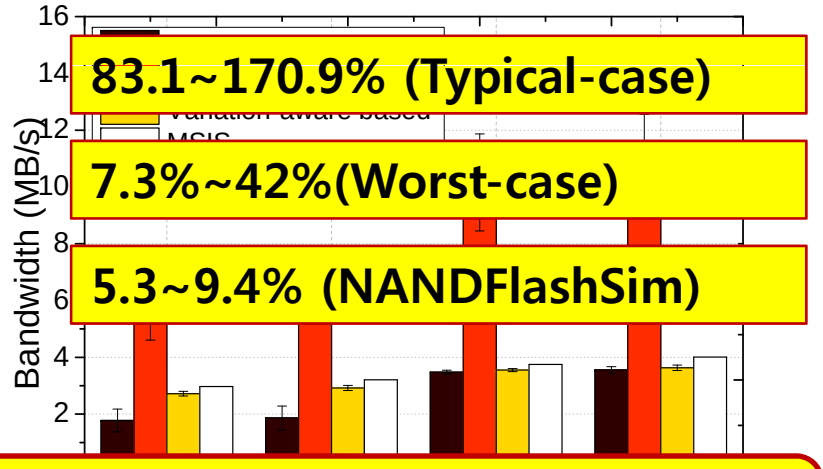


# Validation (Throughput)



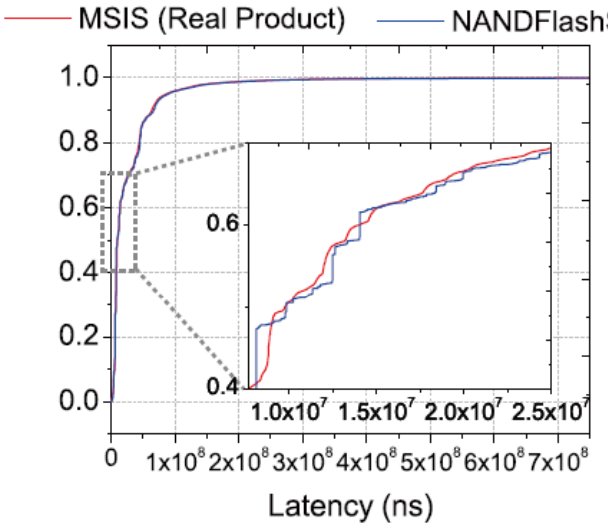



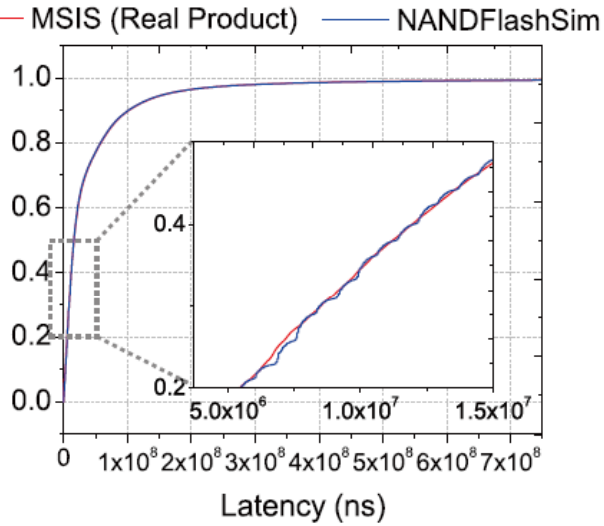

**Write Performance (Single Die)**



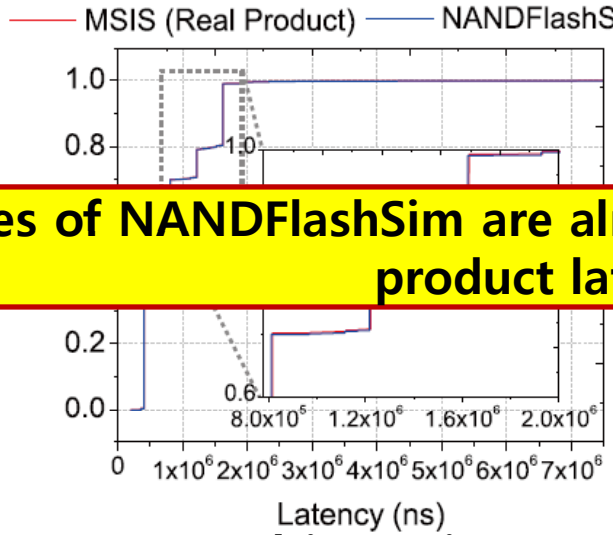

***Improvement*** from deviation
**48.7%~79.6%** in typical-case model, **44.4 ~ 53.5 %** in worst-case model

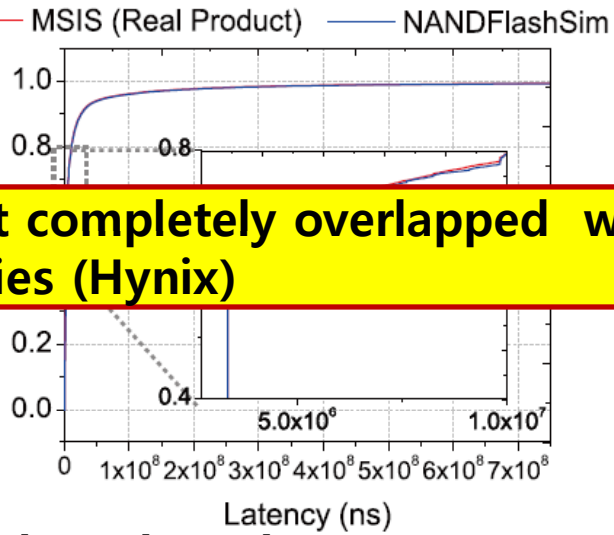
# Validation (Latency)




**Write-intensive Workload (MSN Server, Financial OLTP)**



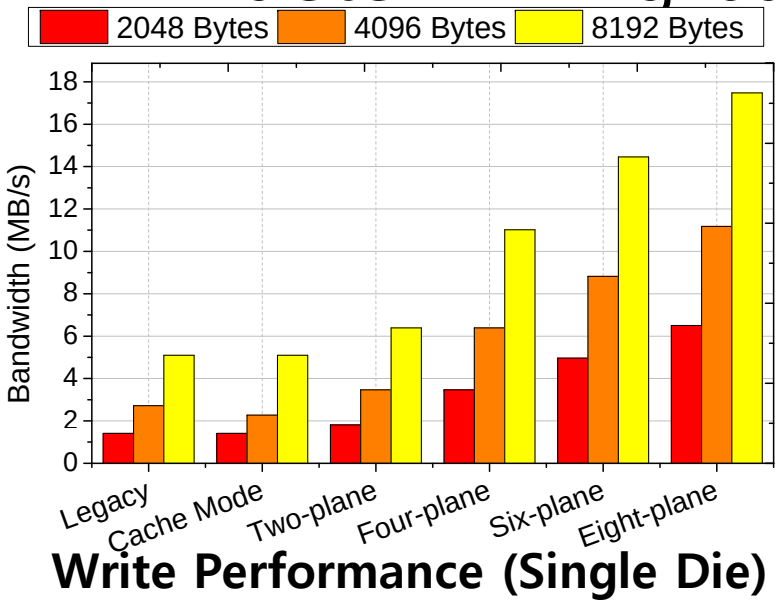




**Latencies of NANDFlashSim are almost completely overlapped with real product latencies (Hynix)**

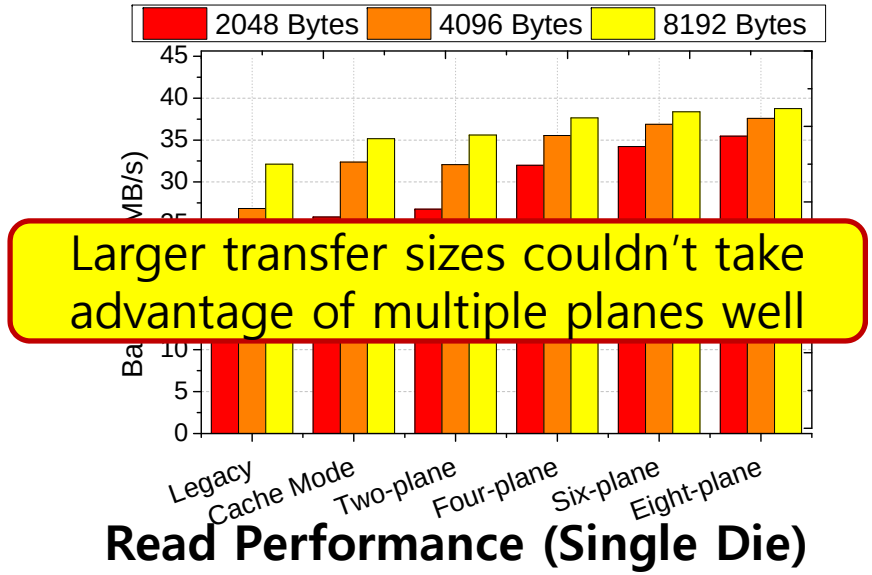
**Read-intensive Workload (Webserch, User)**

# Performance of Multiple Planes

- Performance of write are significantly enhanced as the number of plane increases
  - Cell activities (TIN) can be executed in parallel
- Data movement (TOR) is a dominant factor in determining bandwidth



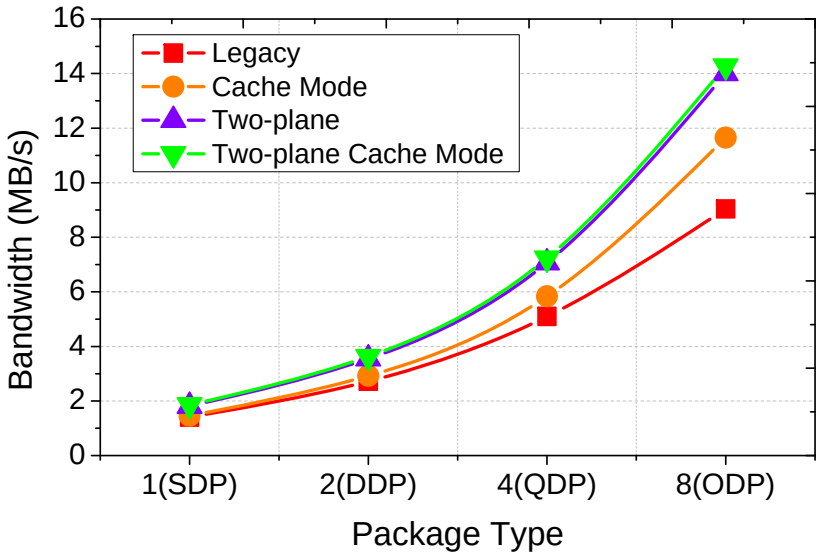
**Write Performance (Single Die)**

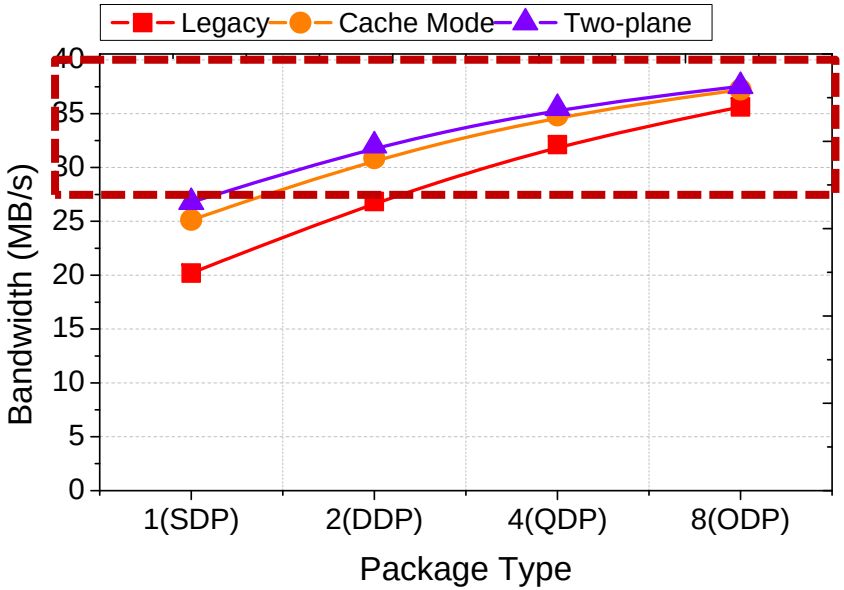


**Read Performance (Single Die)**

# Performance of Multiple Dies

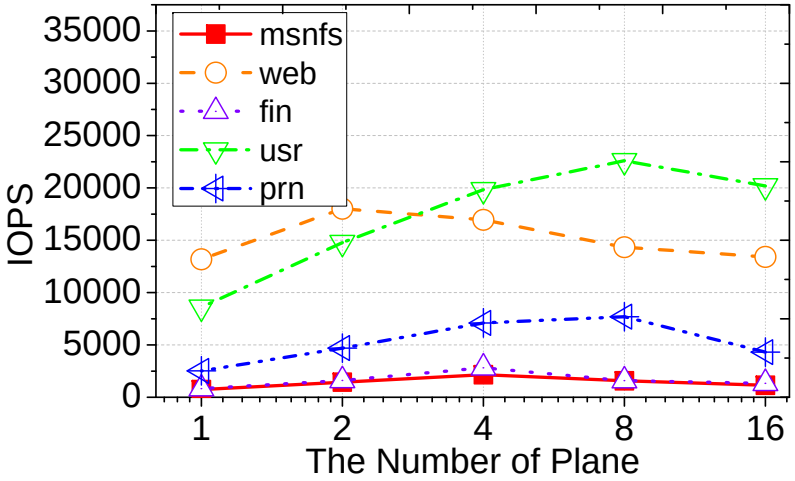
- Similar to multi-plane, write performance are improved by increasing the number of dies
- Multiple dies architecture provides a little worse performance than multi-plane

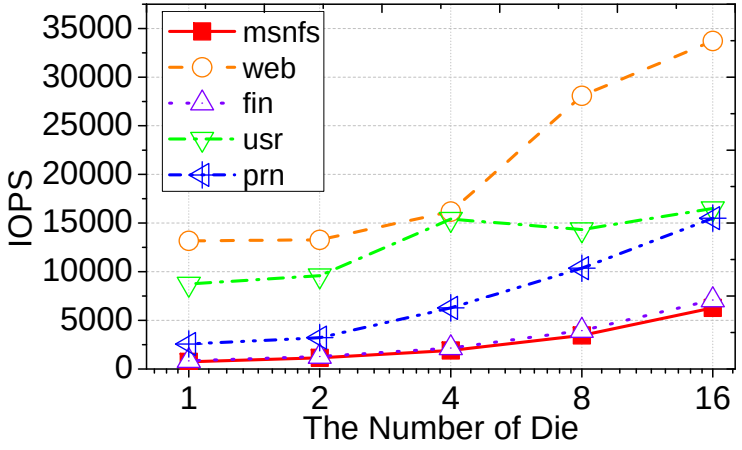


# Multi-plane VS Multi-die

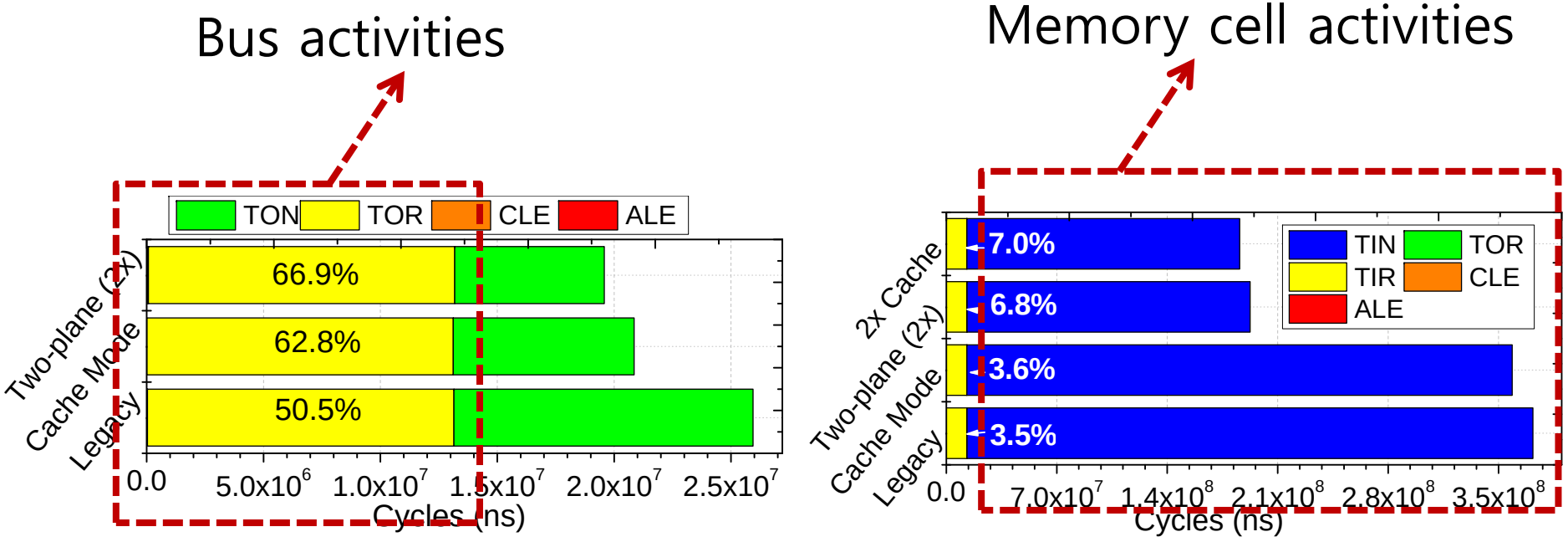
- Under disk-friendly workload
  - The performance of interleaved-die operation is 54.5% better than multi-plane operation on average
  - Interleaved-die operations have less restrictions for addressing



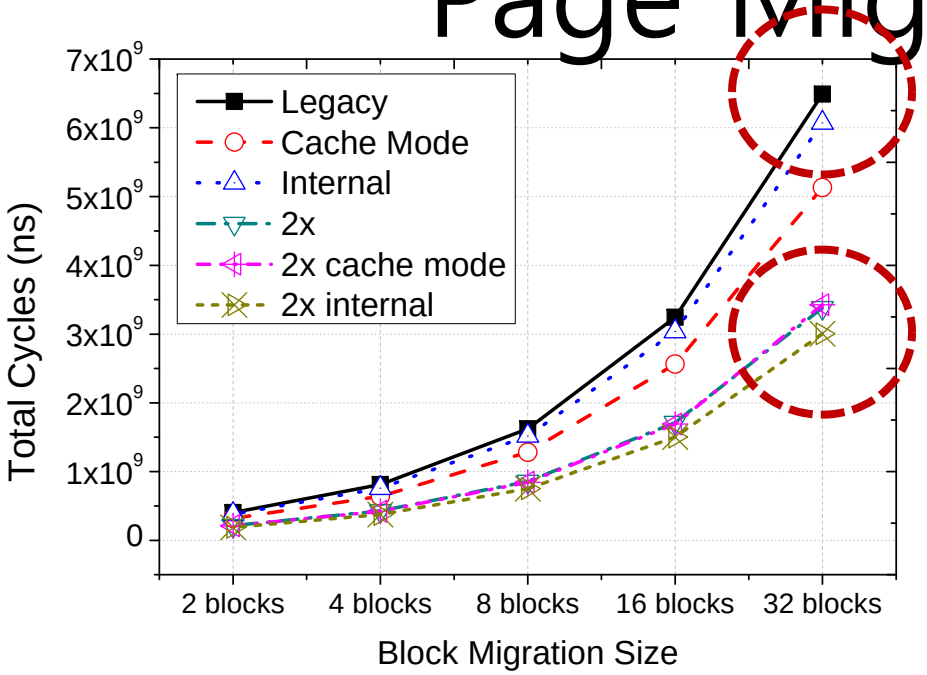



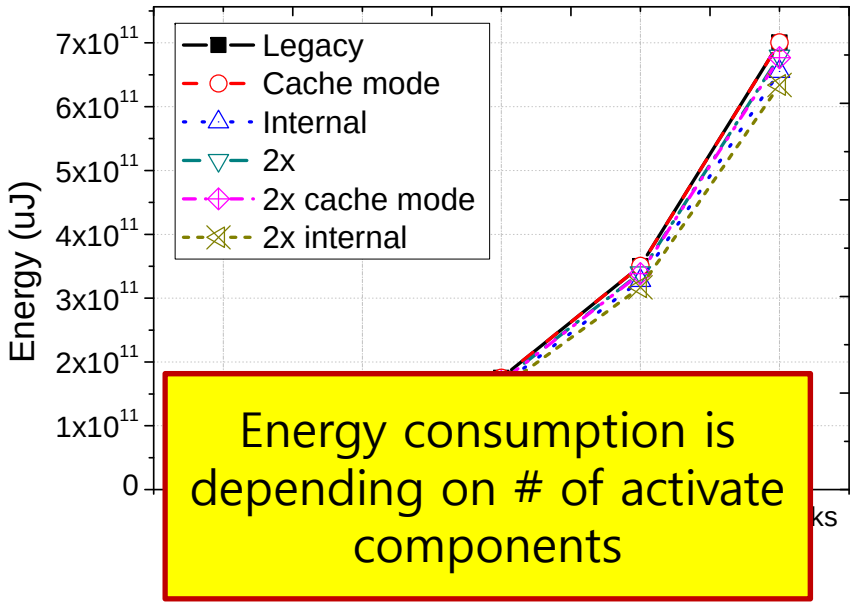
# Breakdown of Cycles

- While writes, most cycles are used for NAND flash itself, reads spend at least 50.5% of the total time doing.

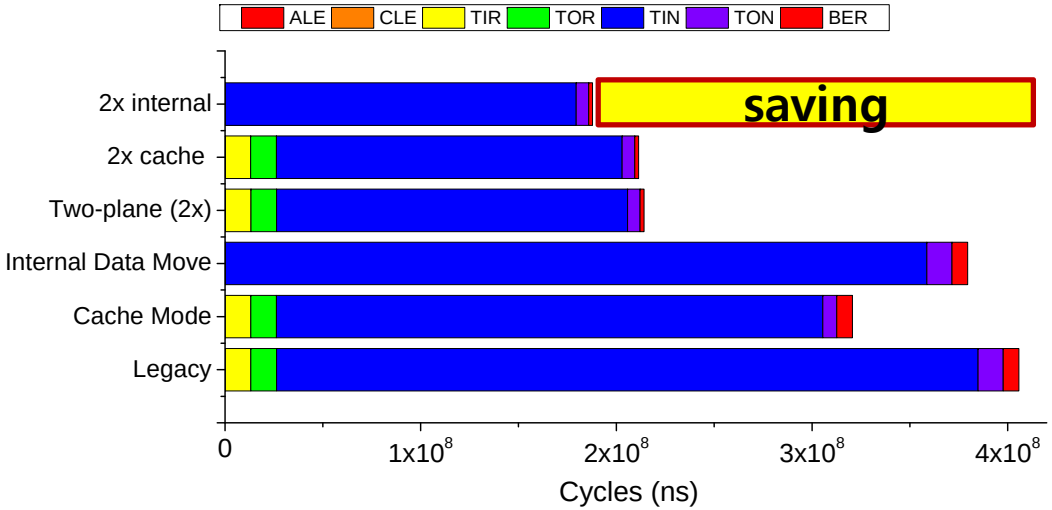


# Page Migration Test



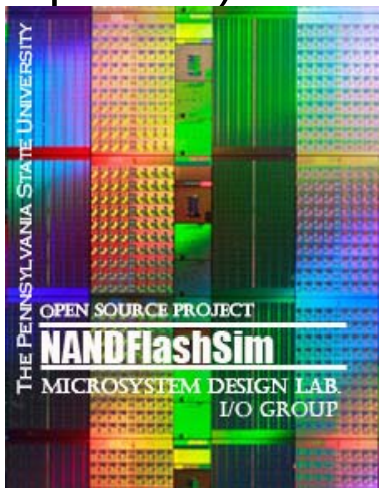




Energy consumption is depending on # of activate components



# Conclusion & Future Works

- A research vehicle for evaluating parallelism and architecture trend
  - Single instance
    - Integrating it into GEM5 and Simics
    - Plan to apply it with Green Flash and Xtensa of CoDEx
  - Multiple instances
    - We successfully built a multi-channel SSD framework with 1024 instances (~16384 dies, ~ 131072 planes)
- Open Source Project
  - Static/shared library
  - Standalone simulation



# Q & A

- Download
  - http://www.cse.psu.edu/~mqj5086/nfs/
- Mailing list
  - nandflashsim@googlegroups.com
- Thanks to
  - Dean Klein, Micron Technology, Inc.
  - Seung-hwan Song, University of Minnesota
  - Michael Kim, Corelinks
  - Kurt Lee, Corelinks
  - Leonard Ko, Corelinks
  - Yulwon Cho, Stanford University