Understanding SSDs with the OpenSSD Platform

Sang-Won Lee & Jin-Soo Kim

Sungkyunkwan University
{swlee, jinsookim}@skku.edu
http://www.openssd-project.org

Outline

- Introduction to flash memory & SSD
- The OpenSSD project
- Jasmine Hardware
- Jasmine Firmware

Introduction to Flash Memory and SSD

Storage Device Metrics

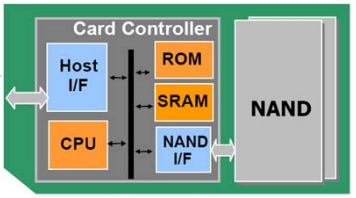
- Capacity ($/GB) : Harddisk >> Flash SSD
- Bandwidth (MB/sec): Harddisk < Flash SSD
- Latency (IOPS): Harddisk << Flash SSD
- Weight/energy/shock resistance/heat & cooling ….
  - Harddisk << Flash SSD

- e.g. Harddisk
  - 7.2K HDD: 50$ / 1TB /100MB/s /100 IOPS
  - 15K HDD: 250$/ 72GB /200MB/s /500 IOPS
  - The HDD price is said to be proportional to IOPS, not capacity.

NAND Applications

- USB

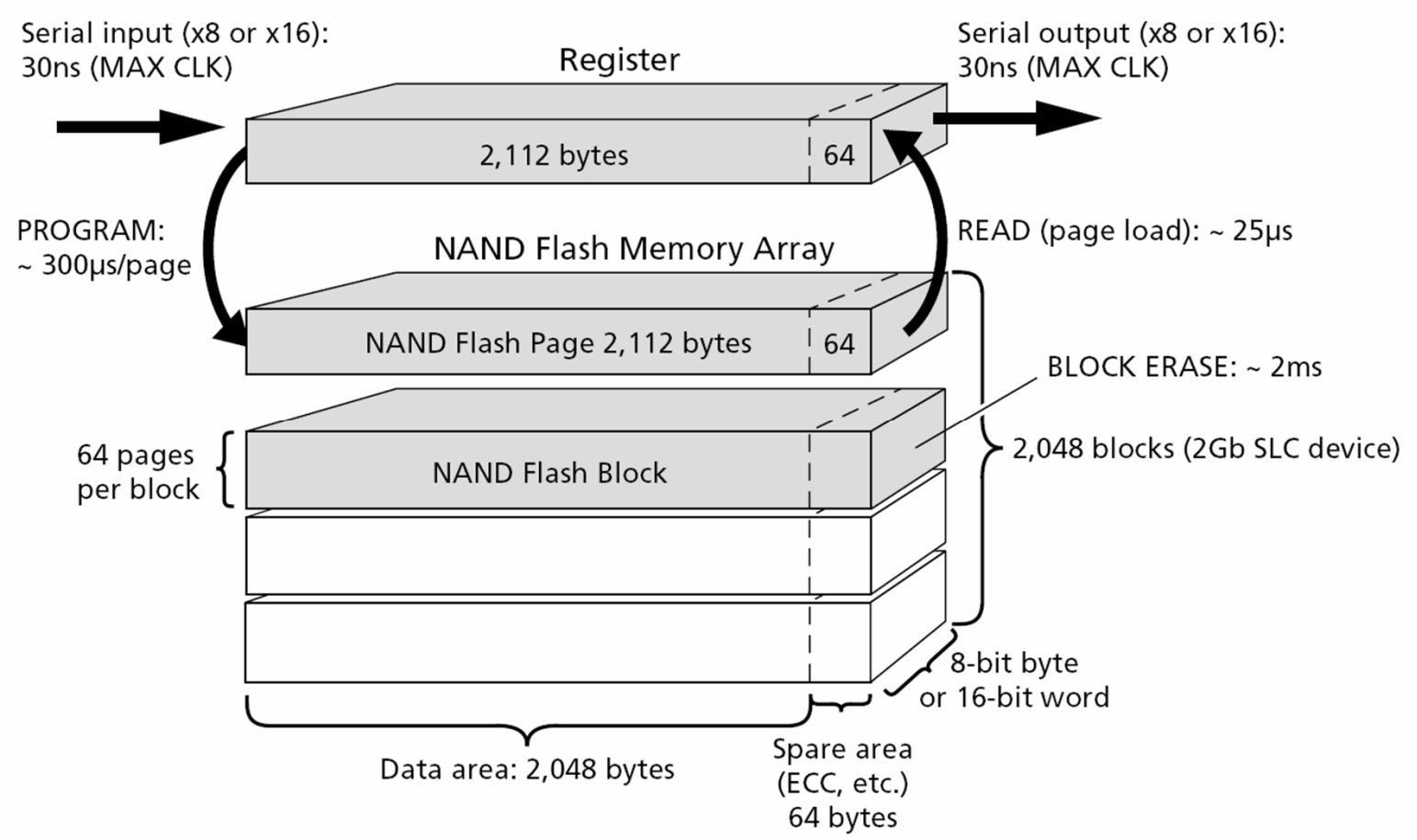
- Flash Cards



  - CompactFlash, MMC, SD/miniSD, xD, …

- Ubiquitous CE
  - MP3, Smartphone, Navigator, DTV, Set-Up
- Hybrid HDD, Intel Turbo Memory, e-MMC
- Flash SSDs for PC/Laptop, Enterprise

NAND Flash Device Organization

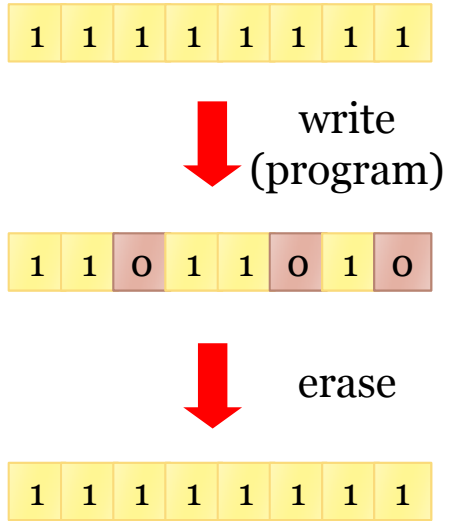


Source: Micron Technology, Inc.

HDD vs. Flash Memory Chip

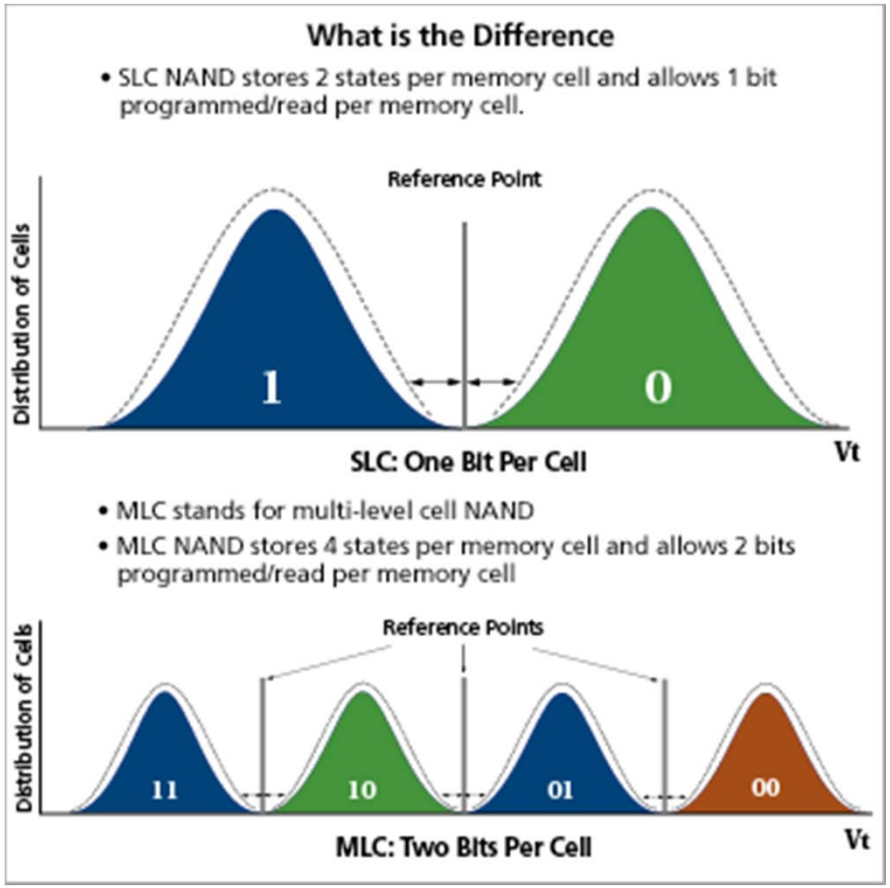
Media	Access time		
	Read	Write	Erase
Magnetic† Disk	12.7 ms (2 KB)	13.7 ms (2 KB)	N/A
NAND Flash‡	80 μs (2 KB)	200 μs (2 KB)	1.5 ms (128 KB)

- Erase-before-overwrite
- No mechanical latency
- Asymmetric read/write speed



NAND Flash Types (1)

- SLC NAND flash
  - Small block (≤ 1Gb)
  - Large block (≥ 1Gb)
- MLC NAND flash
  - 2 bits/cell
- TLC NAND flash
  - 3 bits/cell



Source: Micron Technology, Inc.

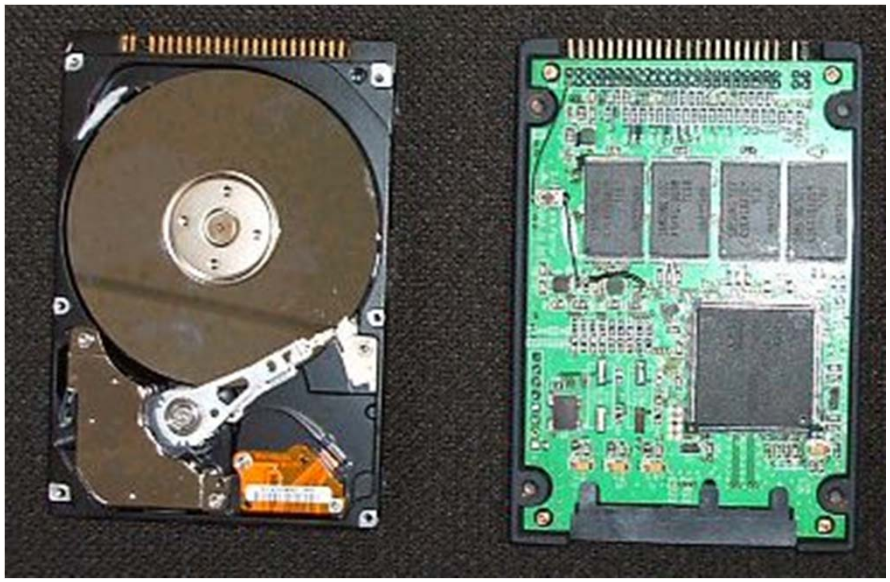
NAND Flash Types (2)

	SLC NAND[1] (small block)	SLC NAND[2] (large block)	MLC NAND[3]
Page size (Bytes)	512+16	2,048+64	4,096+128
Pages / Block	32	64	128
Block size	16KB	128KB	512KB
t_R (read)	15 µs (max)	20 µs (max)	50 µs (max)
t_{PROG} (program)	200 µs (typ) 500 µs (max)	200 µs (typ) 700 µs (max)	600 µs (typ) 1,200 µs (max)
t_{BERS} (erase)	2 ms (typ) 3 ms (max)	1.5 ms (typ) 2 ms (max)	3 ms (typ)
NOP	1 (main), 2 (spare)	4	1
Endurance Cycles	100K	100K	10K
ECC (per 512Bytes)	1 bit ECC 2 bits EDC	1 bit ECC 2 bits EDC	4 bits ECC 5 bits EDC

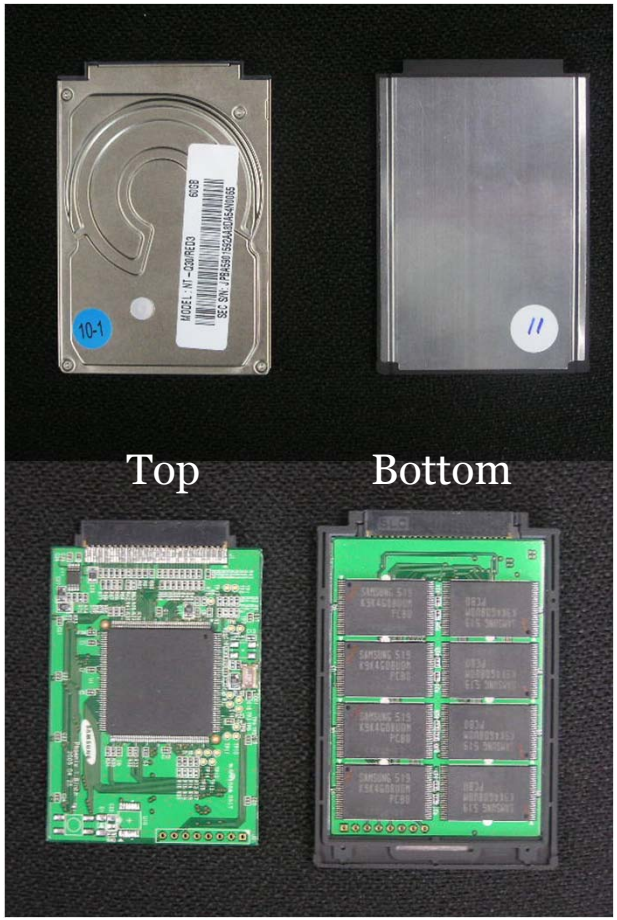
[1] Samsung K9F1208X0C (512Mb) [2] Samsung K9K8G08U0A (8Gb) [3] Micron Technology Inc.

HDDs vs. SSDs (1)

2.5” HDD Flash SSD
(101x70x9.3mm)



1.8” HDD Flash SSD
(78.5x54x4.15mm)

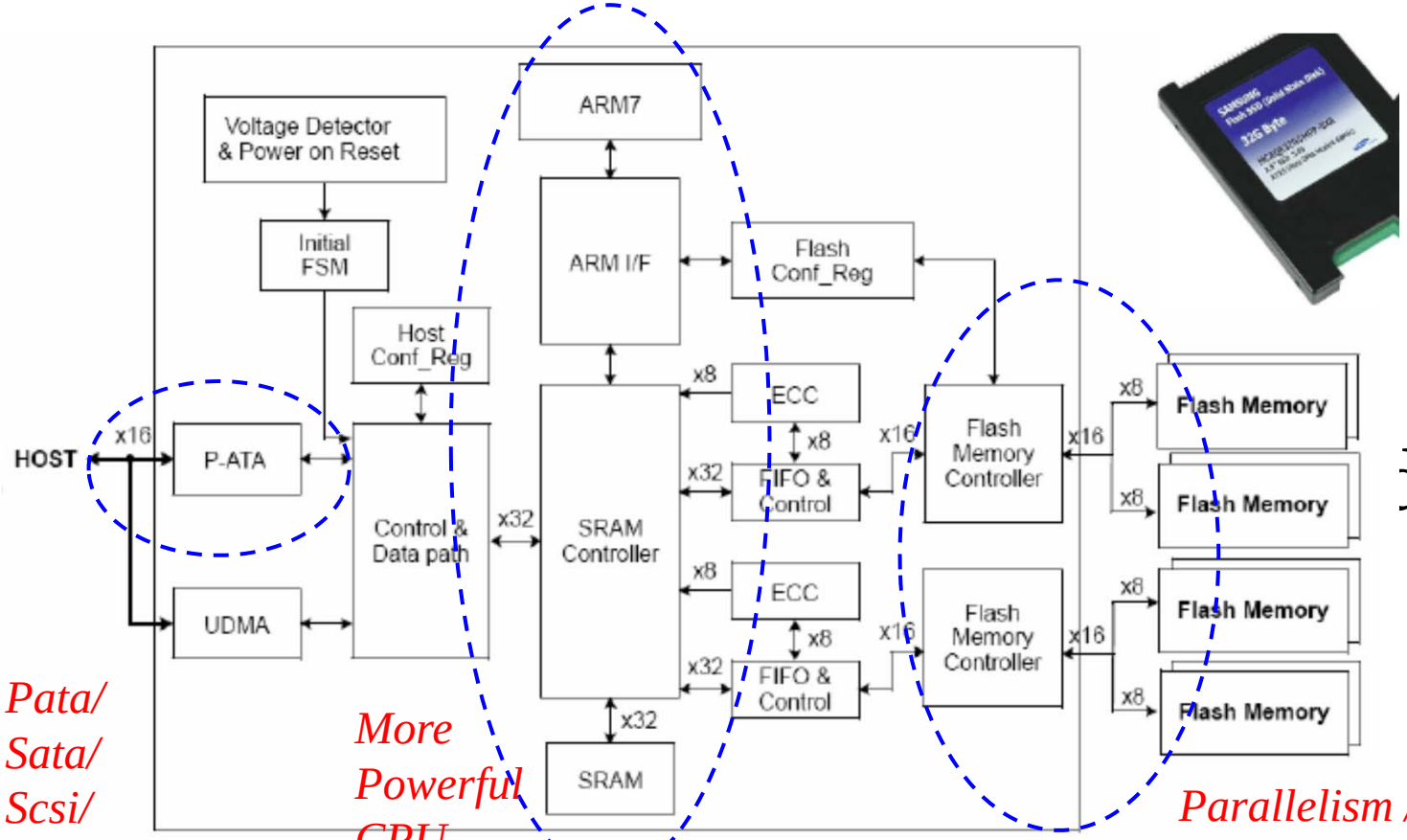


HDDs vs. SSDs (2)

Feature	SSD (Samsung)	HDD (Seagate)
Model	MMDOE56G5MXP (PM800)	ST9500420AS (Momentus 7200.4)
Capacity	256GB (16Gb MLC x 128, 8 channels)	500GB (2 Discs, 4 Heads, 7200RPM)
Form factor	2.5” Weight: 84g	2.5” Weight: 110g
Host interface	Serial ATA-2 (3.0 Gbps) Host transfer rate: 300MB	Serial ATA-2 (3.0 Gbps) Host transfer rate: 300MB
Power consumption	Active: 0.26W Idle/Standby/Sleep: 0.15W	Active: 2.1W (Read), 2.2W (Write) Idle: 0.69W, Standby/Sleep: 0.2W
Performance	Sequential read: Up to 220 MB/s Sequential write: Up to 185 MB/s	Power-on to ready: 4.5 sec Average latency: 4.17 msec
Measured performance[1] (On MacBook Pro, 256KB for sequential, 4KB for random)	Sequential read: 176.73 MB/s Sequential write: 159.98 MB/s Random read: 10.56 MB/s Random write: 2.93 MB/s	Sequential read: 86.07 MB/s Sequential write: 84.64 MB/s Random read: 0.61 MB/s Random write: 1.28 MB/s
Price[2]	539,190 won	80,400 won

[1] *Source: http://forums.macrumors.com/showthread.php?t=658571* [2] *Source: http://www.danawa.com (As of Mar. 17, 2011)*

Flash SSD Block Diagram

- 

del

Storage Abstraction

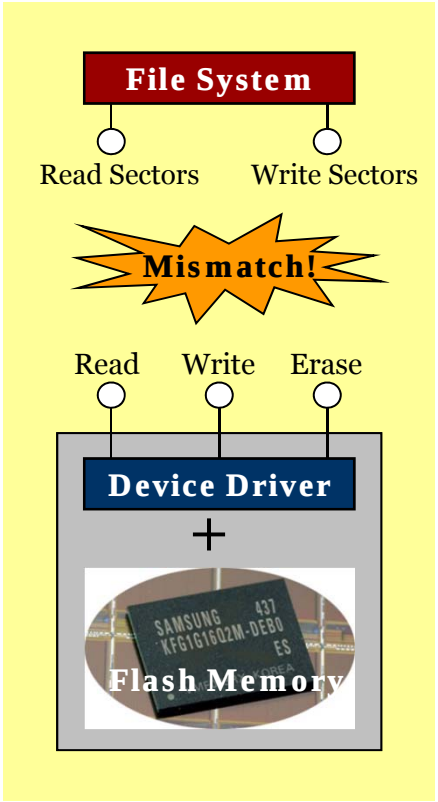
- Abstraction given by block device drivers:

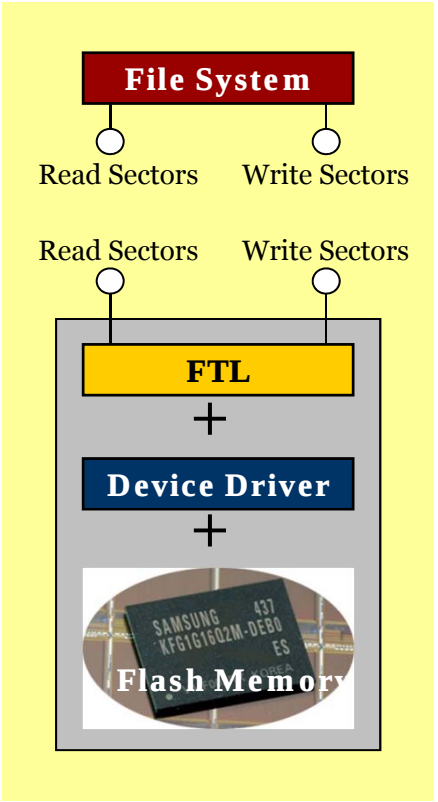
- Operations
  - Identify(): returns N
  - Read (start sector #, # of sectors)
  - Write (start sector #, # of sectors, data)

Source: Sang Lyul Min (Seoul National Univ.)

What is FTL?

- A software layer to make NAND flash fully emulate traditional block devices (or disks)





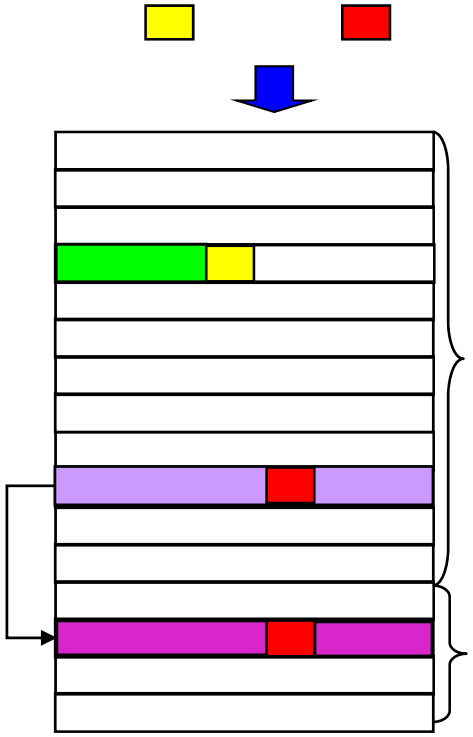
Source: Zeen Info. Tech.

Roles of FTL

- For performance
  - Indirect mapping (address translation)
  - Garbage collection
  - Over-provisioning
  - Hot/cold separation
  - Interleaving over multiple channels/flash chips/planes
  - Request scheduling
  - Buffer management
  - …
- For Reliability
  - Bad block management
  - Wear-leveling
  - Power-off recovery
  - Error correction code (ECC)
  - …
- Other Features
  - Encryption
  - Compression
  - Deduplication
  - …

Overwrites in Flash Memory

- Naïve approach



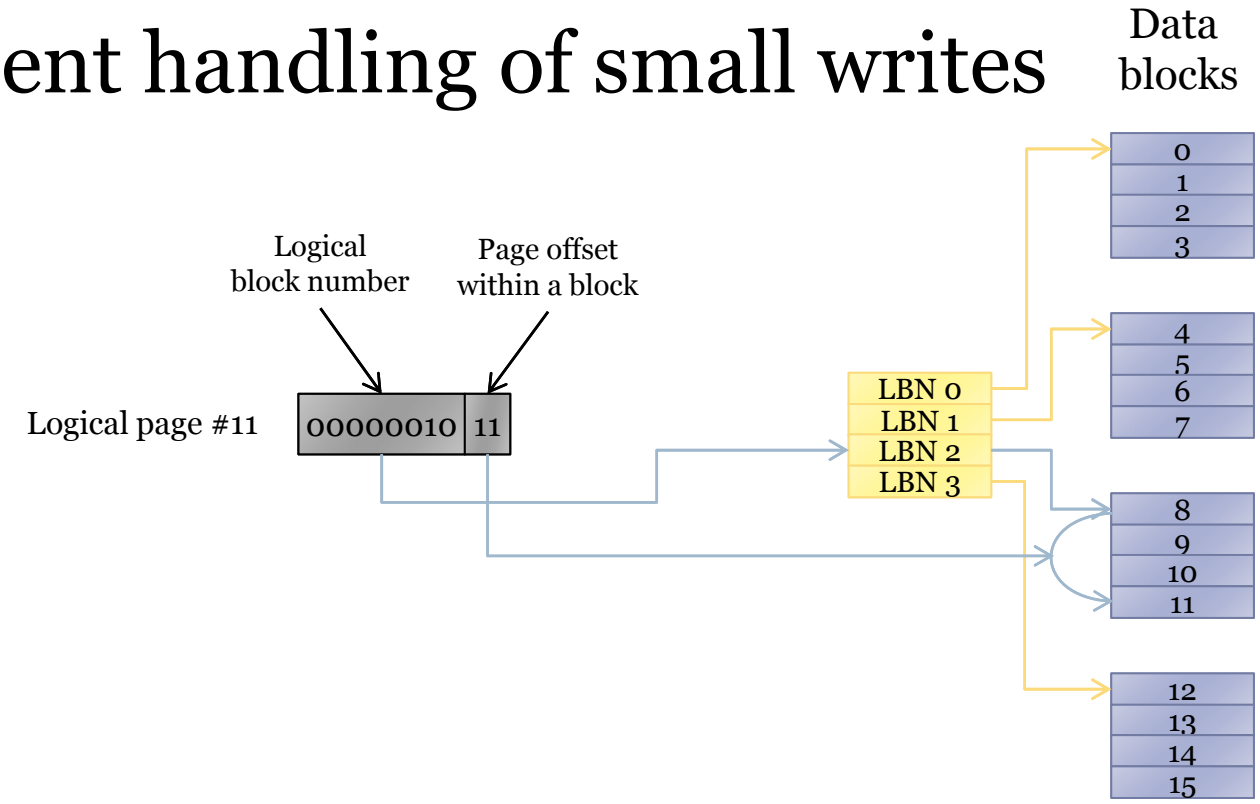
- New write (2K): 0.2ms
- Overwrite (2K)
  - 63 2K-reads = 6.3ms
  - 63 2K-writes = 12.6ms
  - 1 2K-write = 0.2ms
  - 1 erase = 1.5ms
  - Total = 20.6ms

Various Mapping Techniques

- Block mapping
  - Logical block vs. physical block
- Page mapping
  - Logical page vs. physical block
- Hybrid mapping
  - Block mapping + page mapping

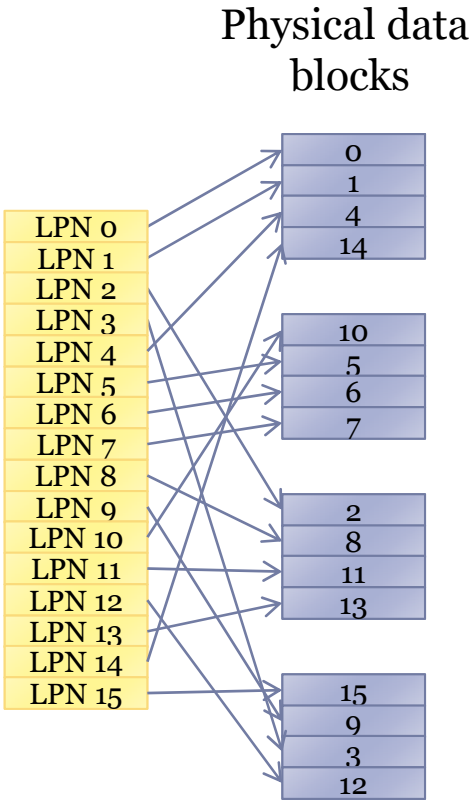
Block Mapping

- Each table entry maps one block
- Small RAM usage
- Inefficient handling of small writes

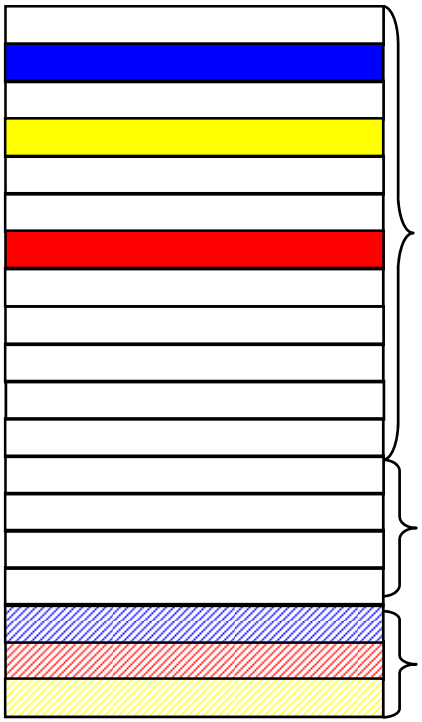


Page Mapping

- Most flexible
- Efficient handling of small writes
- Large memory footprint
  - 32MB for 32GB MLC (4KB page)
- Sensitive to the amount of reserved blocks
- Performance affected as the system ages



Hybrid Mapping: An Example



- Every 2K writes can be buffered in 0.2 ms, not in 20.6 ms
- When a log block is full, it should be merged
  - *"Merge"* operation
- BAST: Block Associative Sector Translation
- Log block thrashing
- Other hybrid scheme: FAST, Superblock, LAST, ...

- Based on "Locality" and "Log Structured File System" idea

Roles of SSDs (in Enterprise)

- Special purpose disk
  - Swap device, redo log device, temporary tablespace
- Complementing disk
  - Extended {buffer / disk} cache
  - A new component in memory hierarchy
  - E.g. SSD as faster disk, separate layer between ram and disk
- Replacing disk
  - IOPS Booster
  - “Flash is disk, disk is tape, and tape is dead” (Gray)

Current SSD Usage Trends

Oracle for TPC-C (2010 Dec.)

- Oracle Exadata
- Total cost: 49M
  - Storage: 23M
  - Sun Flash Array: 22M
  - 720 * 2TB 7.2K HDD: 0.7M

- IBM SSD Buffer (VLDB 10)
- MS SQL Server (SIGMOD 11)

ORACLE

SPARC SuperCluster with T3-4 Servers

TPC-C 5.11.0
TPC-Pricing 1.5.0
Report Date
December 2, 2010

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$30,528,863USD	30,249,688 tpmC	$1.01USD/tpmC	June 1, 2011

Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
SPARC T3 1.65GHz 108 / 1,728 / 13,824	Oracle Database 11g Release 2 Enterprise Ed. With Oracle Real Application Clusters and Partitioning	Oracle Solaris 10 09/10	Tuxedo CFS-R Tier 1 Oracle iPlanet Web Server	24,300,000

Clients

81 Sun Fire X4170M2
2.93GHz Intel Xeon X5670 HC
48GB Memory
2 146GB SAS disk

Database Nodes

27 Sun SPARC T3-4 Servers
4 1.65GHz SPARC T3
512GB Memory
3 300GB 10K RPM SAS
4 8Gb/s FC HBA, 2 port
10GbE SFP+
5RU High

Storage

67 X4270M2 DATA COMSTAR
6 2TB 7.2K RPM SAS
2 Sun F5100 Flash Arrays

2 X4270M2 DATA COMSTAR
5 2TB 7.2K RPM SAS
2 Sun F5100 Flash Arrays

28 X4270M2 REDO COMSTAR
11 2TB 7.2K RPM SAS

System Component	Each Server Node		Each Client	
Processors/Cores/Threads and cache	4/64/512	SPARC T3 1.65GHz 6 MB L2 Cache	2/12/24	Intel Xeon X5670 12MB Smart Cache
Memory		512GB (13.5TB Total)		48GB
Disk Controllers	4	8Gb/s FC HBA 2 Port	1	8 port Internal SAS
OS Disks (each system)	3	300GB 10K RPM SAS	2	146GB 10K RPM SAS
External Storage (Equally visible to all T3-4 Server nodes)	11,040 720	24GB SSD Flash Modules 2TB 7.2K RPM SAS		
Total Storage		1.76PB		

Some Future Trends

- FlashSSD based In-Storage Processing (ISP)
  - Promising in data-intensive applications: OLAP, search, map/reduce, scientific data
    - E.g. Scan/filtering, hashing, sorting

- vs. Disk based ISP History
  - Database machine (Boral and DeWitt)
  - Active disk (Eric Riedel et al.)
    - Processing power in storage
    - CMU, {HP, Sun} + Oracle: Oracle Exadata

The OpenSSD Project

What's the OpenSSD Project?

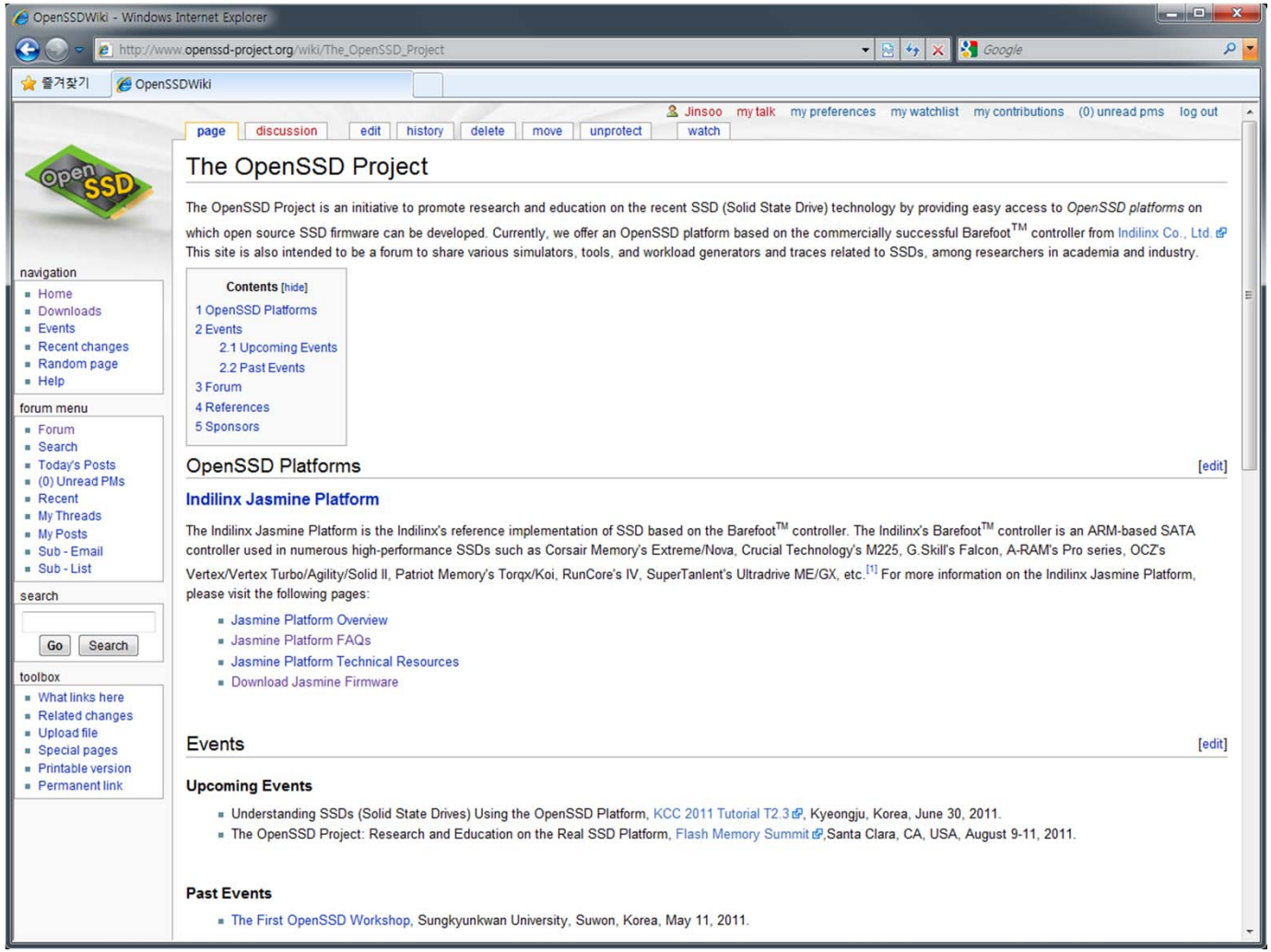
- An initiative to promote research and education on the SSD technology
- Provides an "OpenSSD platform" for developing open-source SSD firmware

- Started as a collaborative work between SKKU and Indilinx

Why OpenSSD?

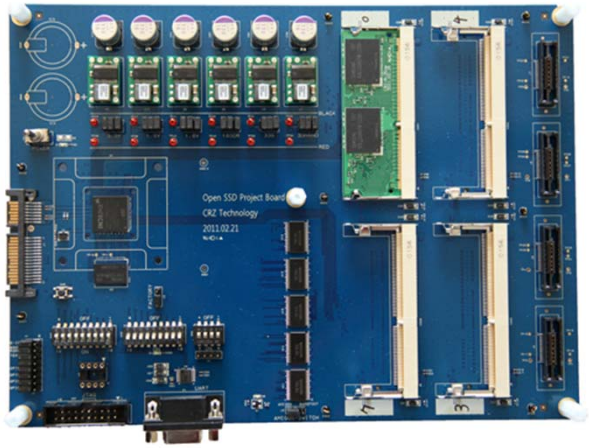
- No more simulations
- Broaden research horizon
- Educate people with a real system
- Share expertise
- Just for fun!
- …

http://www.openssd-project.org



Jasmine OpenSSD Platform

- A reference implementation of SSD based on the Indilinx Barefoot controller




- Sample FTL source codes
- Technical documents

Jasmine Users

- 31 sets shipped to 10 institutions (16 labs)
  - Sungkyunkwan U., Hanyang U., Ajou U., Hongik U., Korea U., Kwangwoon U., POSTECH, Soongsil U., U of Seoul, Inha U.
- 1 set shipped abroad
  - RecoverMyFlashDrive.com
- Inquiries from US, China, Netherlands, …
- Currently preparing more sets

Jasmine Firmware

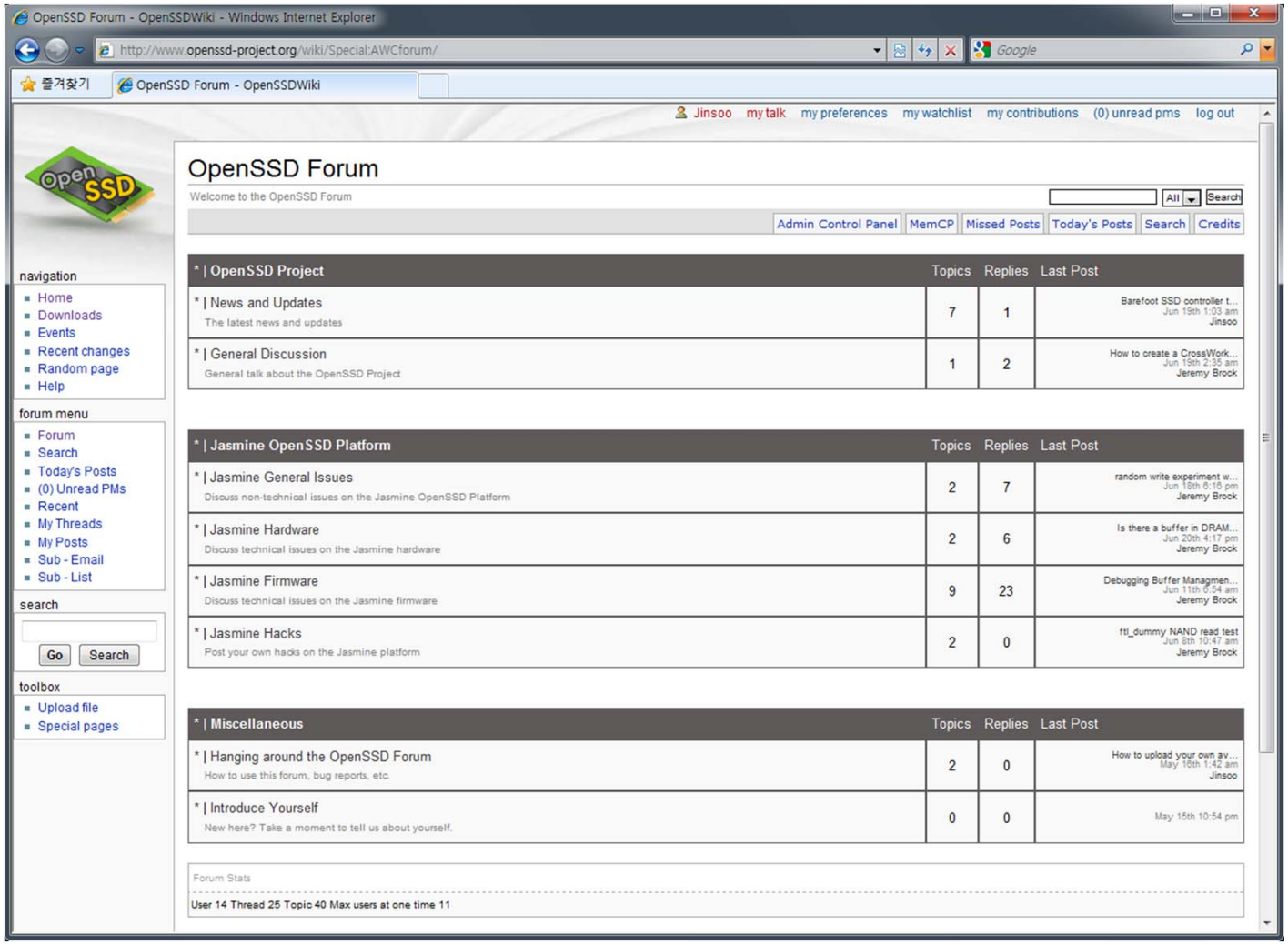
- The firmware source code v1.0.0 released on April 7, 2011 by Indilinx under the GPL
- The latest version: v1.0.6 (June 25, 2011)
- Available at http://www.openssd-project.org/wiki/Downloads

- Total firmware downloads: 337 times (As of June 28, 2011)

Jasmine Resources

- FAQs
- Forums
- Board schematics
- Technical Reference Manual
- FTL Developer's Guide
- Barefoot Controller Technical Reference
- Contributions from community
  - Developer Information (by Jeremy Brock), …

OpenSSD Forum



1st OpenSSD Workshop

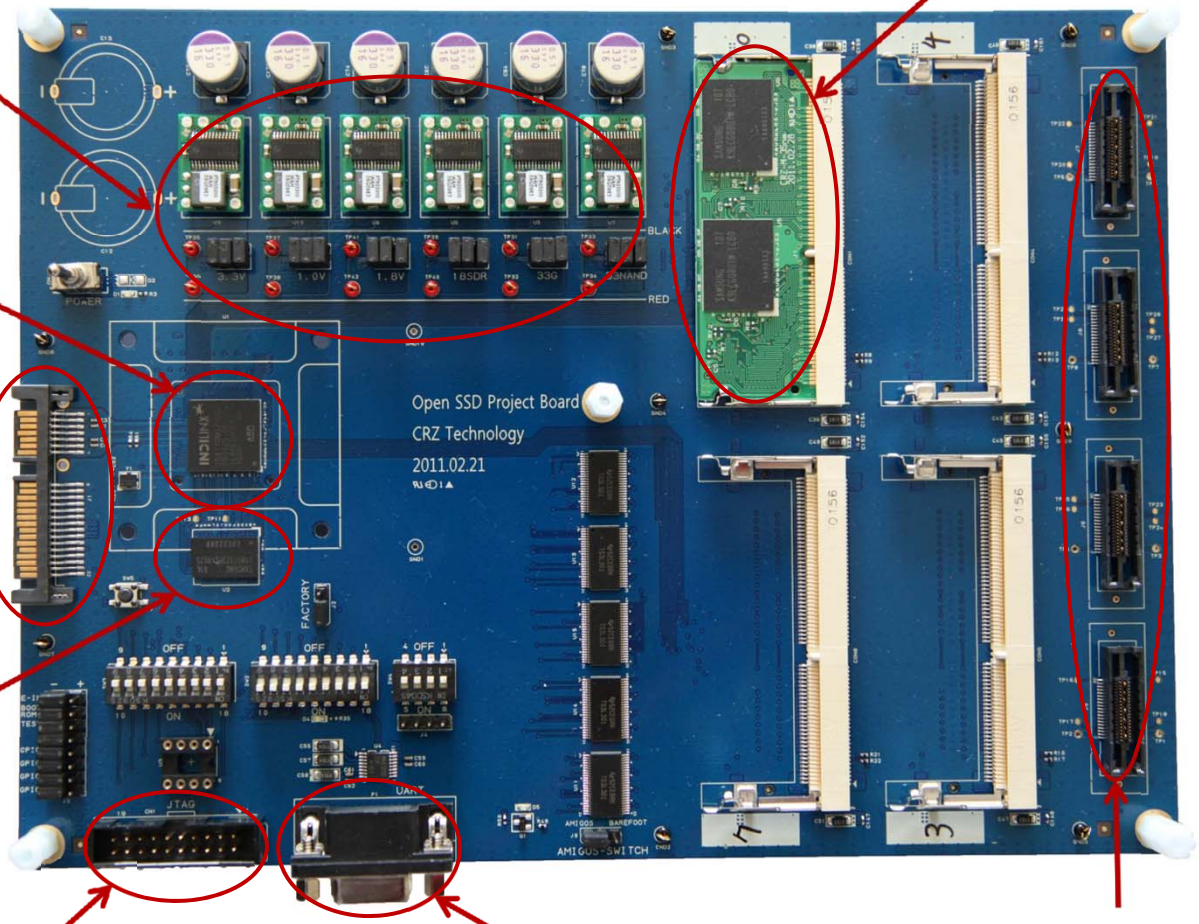
- May 11, 2011 @ Sungkyunkwan Univ.
- 53 participants from 12 institutions

Jasmine Hardware

A Real SSD

Source: benchmarkreviews.com

Jasmine Board



Current Measurement points

NAND flash module (32GB/module)

Indilinx Barefoot SSD controller

SATA 2.0 interface

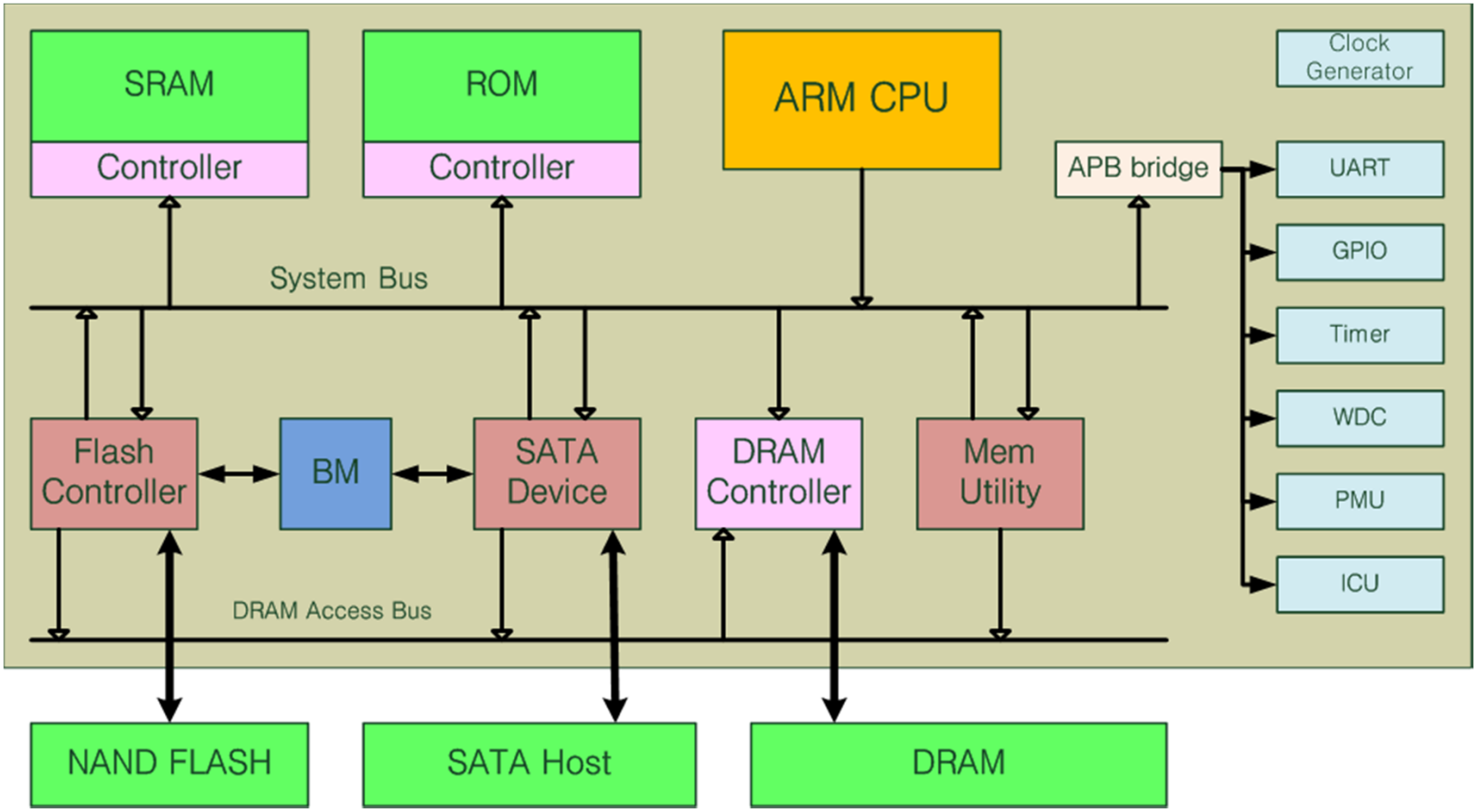
64MB SDRAM

JTAG port

UART port

Mictor connectors for logic analyzers

Indilinx Barefoot Controller



Barefoot Features

- ARM7TDMI-S running up to 87.5MHz
- 96KB SRAM
- SATA 2.0 (3Gbps)
- Mobile SDRAM controller up to 64MB
- NAND flash 8/12/16-bit BCH ECC per sector
- SDRAM 2-byte RS ECC per 128 +4 bytes
- Maximum 64CE's (4 channels, 8 banks/ch)
- System bus running up to 175MHz

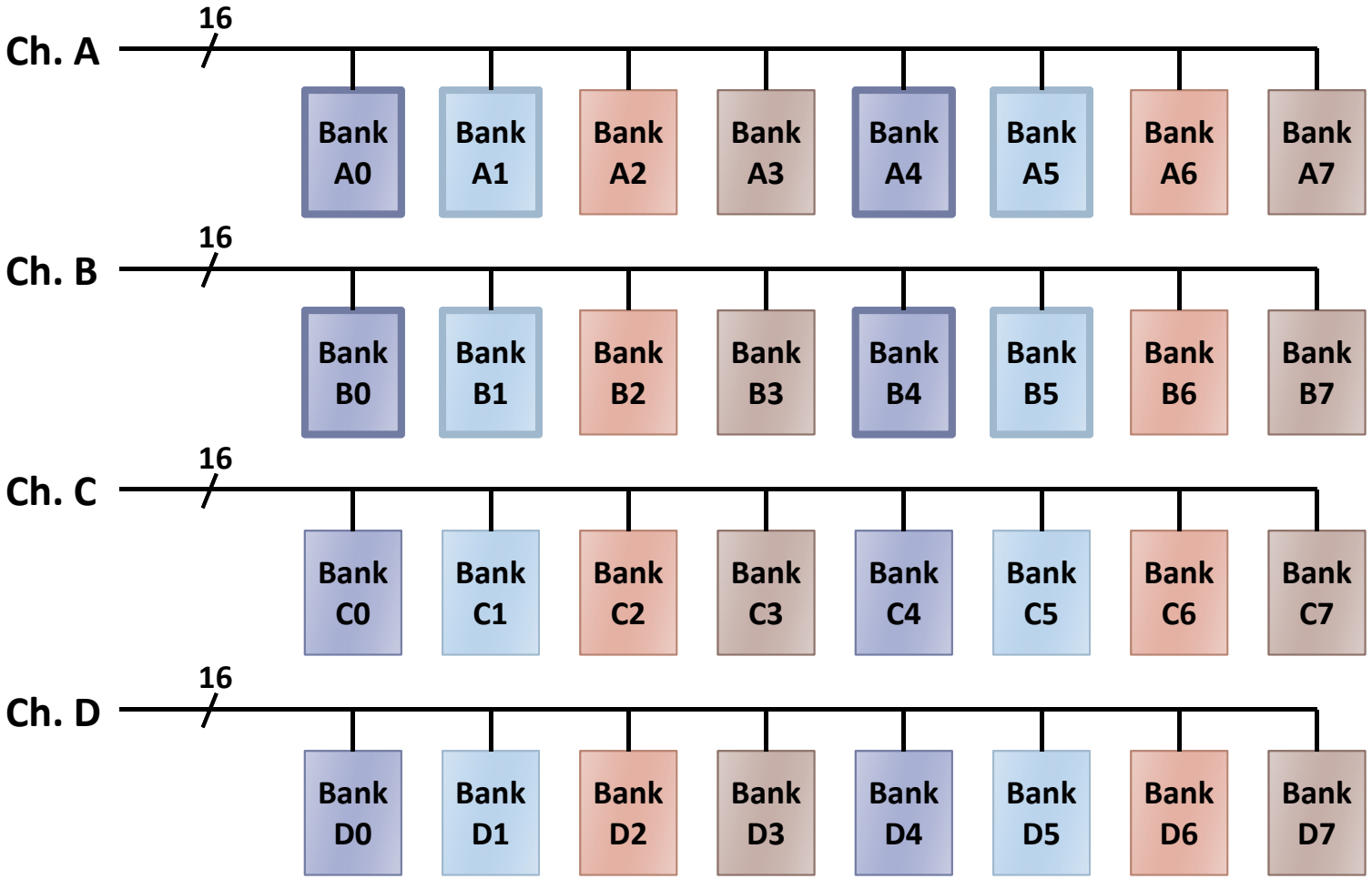
SDRAM & Flash

- Mobile SDRAM
  - Samsung 64MB (subject to change)
- NAND Flash
  - Samsung 64GB (subject to change) in two NAND flash modules
  - Four K9LCG08U1M (8GB) packages / module
  - 32Gb (4GB) per die, 2 CE signals / package (Dual Die Package)

Debugging/Monitoring Aids

- JTAG
- UART
- 1 LED and 6 GPIO pins
- Mictor connectors to NAND flash signals for logic analyzers
- Separate current measurement points for core, IO, SDRAM, and NAND

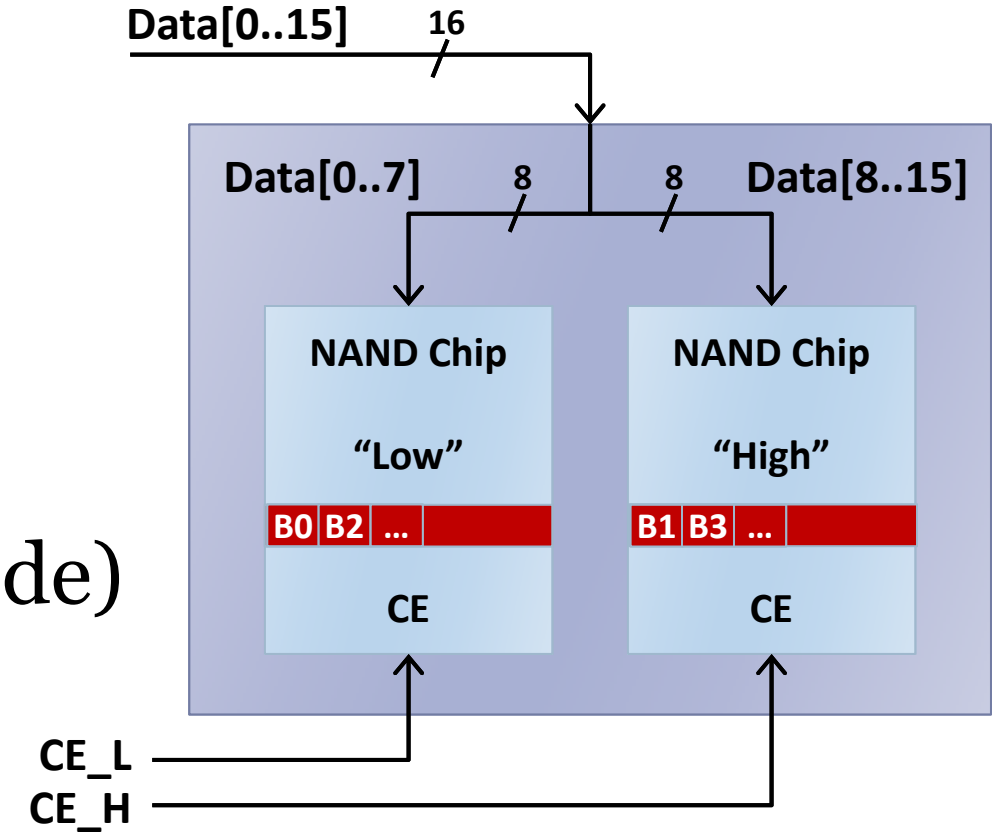
NAND Flash Configuration



Bank Configuration

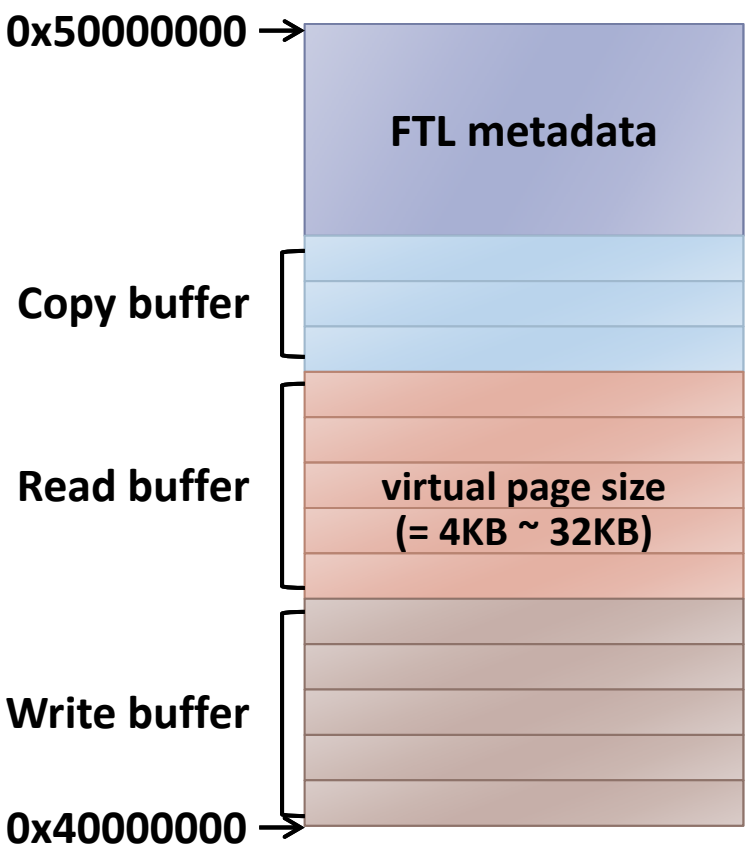
- 16-bit IO/bank

- Virtual page size
  = Page size * 2
- Virtual page size
  = Page size * 4
  (for 2-plane mode)

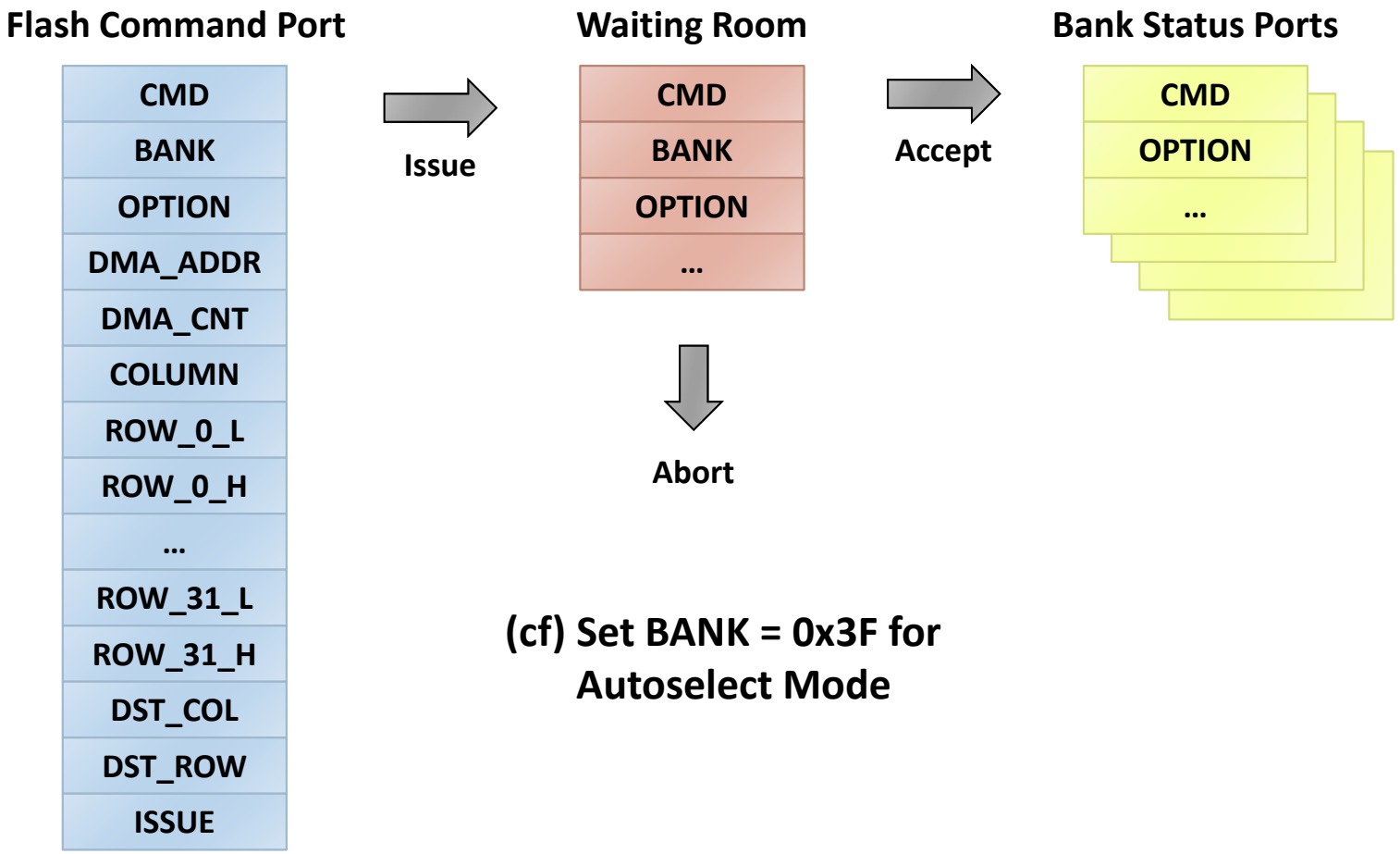


SDRAM Layout

- The location & size of each region fixed at init time
- Buffer is segmented in fixed size (4~32KB)
- Read/write buffers in circular buffer scheme
- 4-byte ECC parity added to every 128B



NAND Flash Controller



(cf) Set BANK = 0x3F for Autoselect Mode

Example: Reading a VPage

```
SETREG (FCP_CMD, FC_COL_ROW_READ_OUT);
SETREG (FCP_DMA_CNT, SECTORS_PER_PAGE * BYTES_PER_SECTOR);
SETREG (FCP_COL, 0);
SETREG (FCP_DMA_ADDR, RD_BUF_PTR(g_ftl_read_buf_id));
SETREG (FCP_OPTION, FO_P | FO_E | FO_B_SATA_R);
SETREG (FCP_ROW_L(bank), row);
SETREG (FCP_ROW_H(bank), row);
SETREG (FCP_BANK, bank);
while ((GETREG(WR_STAT) & 0x00000001) != 0);
SETREG (FCP_ISSUE, NULL);
```

Flash Operation Parallelism

BSP_FSM of Bank A0

IDLE	BUSY	IDLE

RB signal of Bank A0

READY	BUSY (cell operation)	READY

BSP_FSM of Bank A1

IDLE	BUSY	IDLE

RB signal of Bank A1

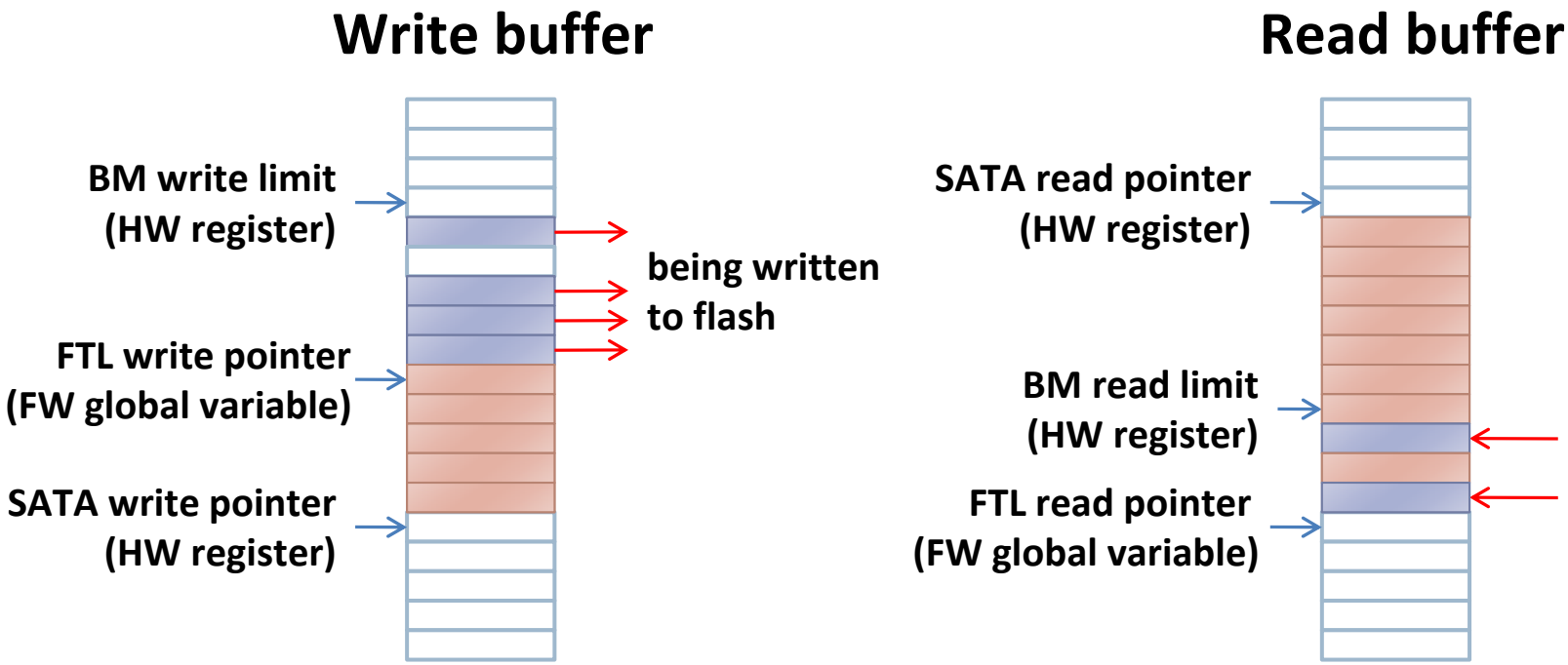
READY	BUSY (cell operation)	READY

Owner of Channel A (= who is doing an interface operation)

FREE	A0	A1	FREE	A1	FREE	A0	FREE

Buffer Management

- Buffer management and flow control in hardware

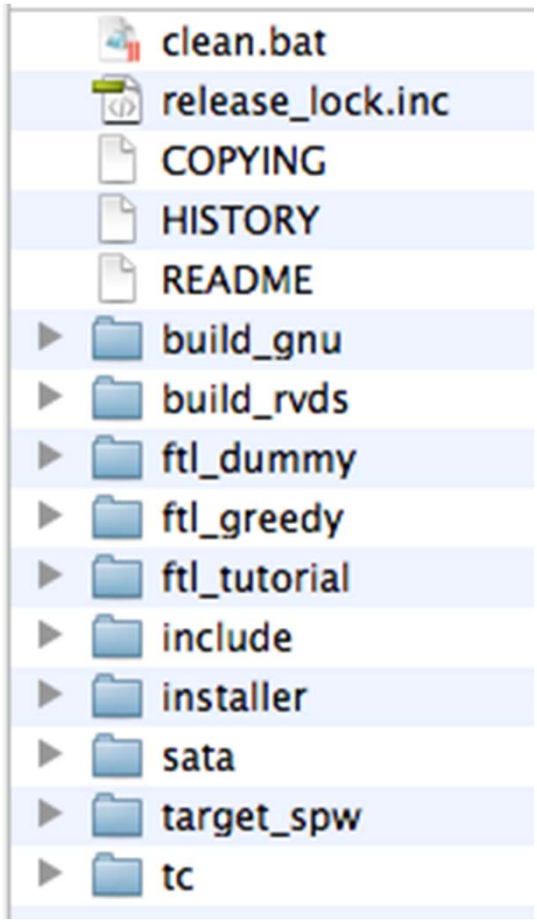


Memory Utility

- Hardware accelerator for memory operations
  - Initializing a memory region with a given value
  - Copying a memory between SRAM & DRAM
  - Finding a bit pattern
  - Finding a given value
  - Finding a min/max value
  - Reading/writing DRAM with ECC handling
  - …

Jasmine Firmware

Firmware Source Tree

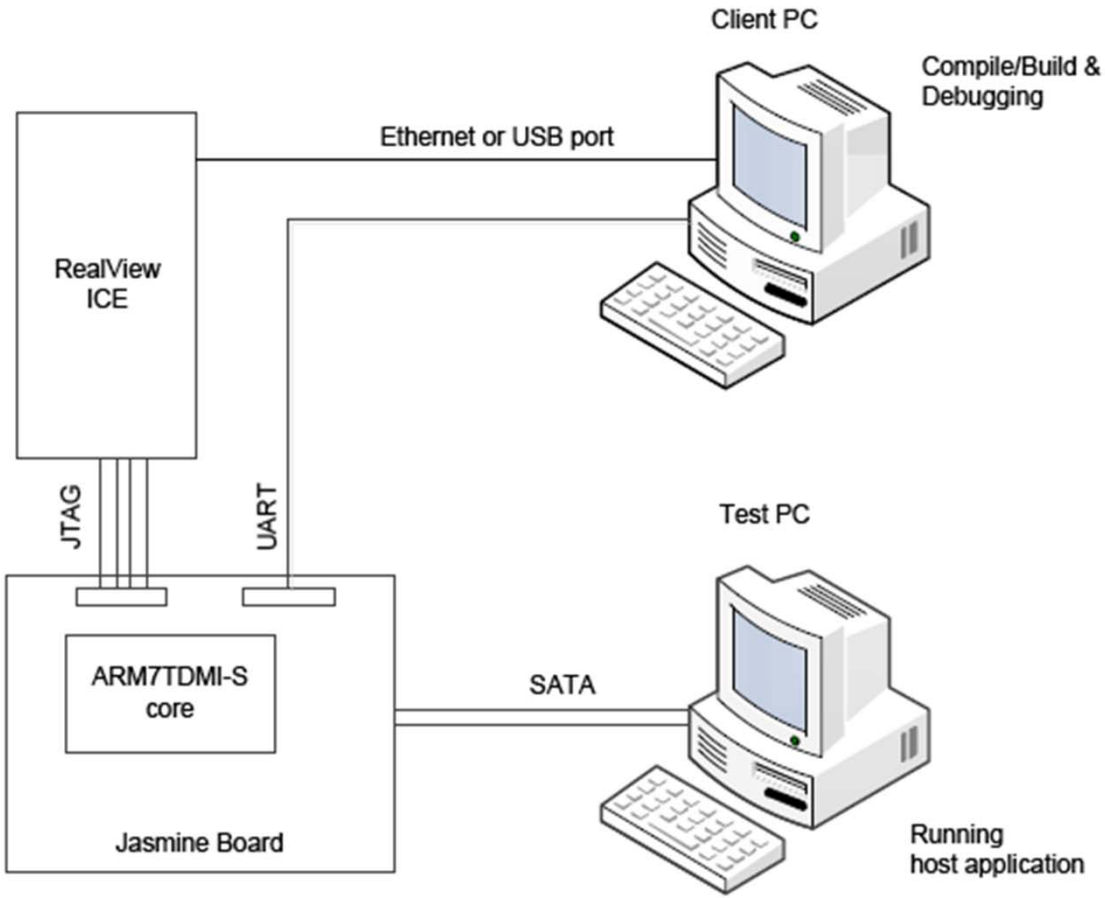


; Build directory for Sourcery G++ toolchain
; Build directory for ARM RVDS toolchain
; Sample FTL: DummyFTL
; Sample FTL: GreedyFTL
; Sample FTL: TutorialFTL
; Header files
; Firmware installer utility (installer.exe)
; SATA protocol handling
; Initialization, flash control, etc.
; Testing code

Toolchain

- For building firmware
  - ARM RealView Development Suite (RVDS) 3.0 or higher ($$$$)
  - CodeSourcery G++ Lite Edition for ARM EABI
  - CrossWorks for ARM from Rowley Associates ($$$)
- For building installer utility (install.exe)
  - Microsoft Visual Studio 2005 or Microsoft Visual C++ 2010 Express Edition

Development Setup



Building Firmware (1)

- Set compile options (`./include/jasmine.h`)

```
#define  OPTION_2_PLANE           1  // 1: use 2-plane mode
#define  OPTION_ENABLE_ASSERT     0  // 1: enable ASSERT()
#define  OPTION_FTL_TEST          0  // 1: FTL test w/o SATA
#define  OPTION_UART_DEBUG        0  // 1: enable UART msgs
#define  OPTION_SLOW_SATA         0  // 1: SATA1 (1.5Gbps)
#define  OPTION_SUPPORT_NCQ       1  // 1: NCQ support
#define  OPTION_REDUCED_CAPACITY  0  // 1: for testing
```

Building Firmware (2)

- Specify the target FTL (`./build_gnu/Makefile`)

```
FTL     = [dummy|tutorial|greedy]
...
```

- Compile the firmware source

```
> cd build_gnu
> build.bat
```

  - Firmware binary: `./build_gnu/firmware.bin`

Building Firmware (3)

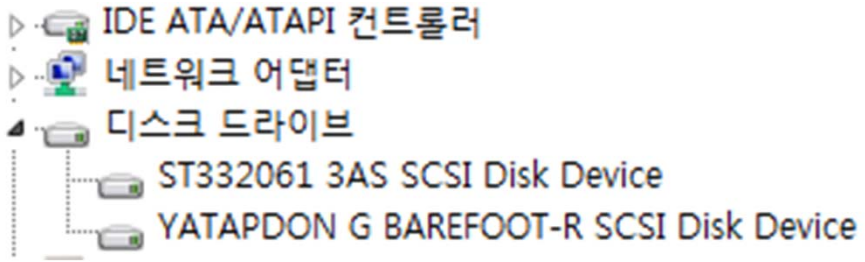
- Compile the installer utility
  - Open `./installer/installer.sln` in MS Visual Studio
  - Build the project
  - Move `./installer/install.exe` to `./build_gnu`

Installing Firmware (1)

- Boot the Jasmine board in “Factory mode”
  - J2 jumper need to be set

Factory Mode

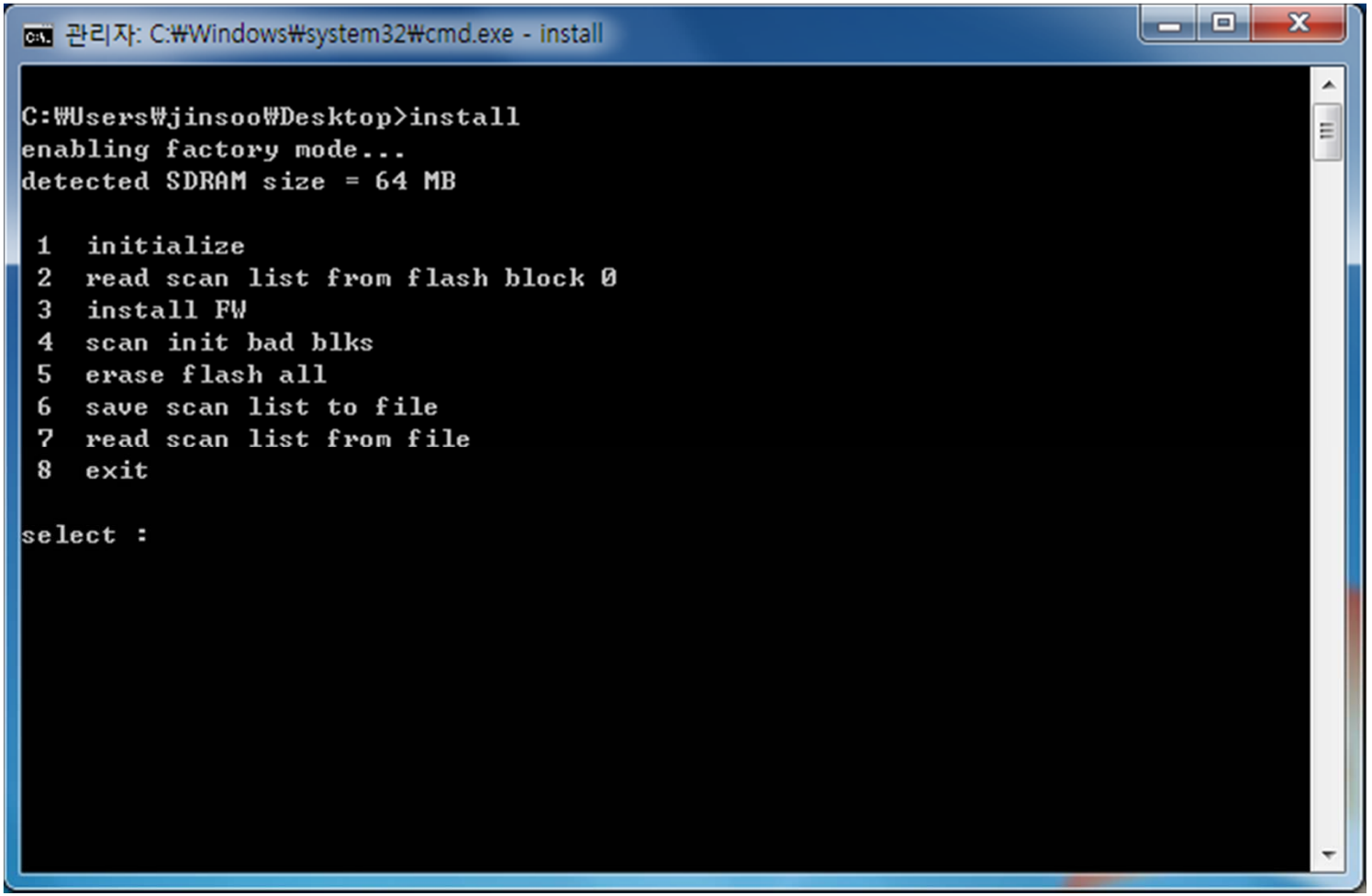
Normal Mode
(Default)



- Install firmware

```
> cd build_gnu
> install.exe
```

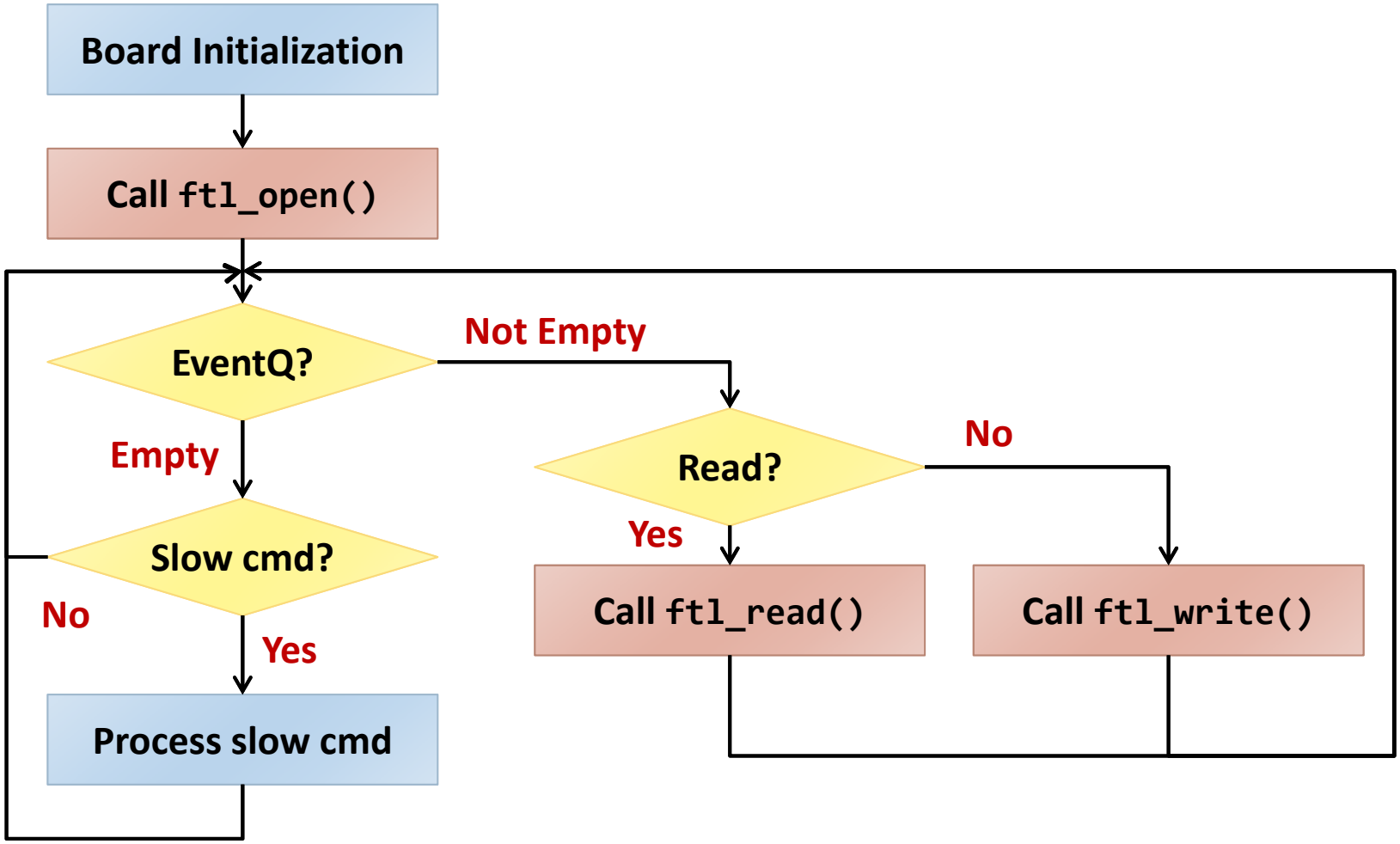
Installing Firmware (2)



Available FTLs

- TutorialFTL (by Indilinx)
  - Page-mapping FTL
  - No garbage collection
  - NAND flash initialized at power-on
- GreedyFTL (by SKKU VLDB Lab.)
  - Page-mapping FTL with garbage collection
  - Data survive a normal power-off
- DummyFTL (by Indilinx)
  - For measuring SATA and DRAM speed

Overall Flow



Debugging Firmware

- On-board LED
- UART
- ARM JTAG ICE + ARM RVDS
- USB ARM JTAG devices + CodeWorks

- Monitoring Signals with Logic Analyzers

Call For Participation

- Welcome any contributions from
  - SSD manufacturers
  - NAND flash vendors
  - Research groups
  - Individual developers
  - …
- You can create and edit most of pages after registration

Thank You!