Cosmos OpenSSD: A PCIe-based Open Source SSD Platform

Yong Ho Song\textsuperscript{1}, Sanghyuk Jung\textsuperscript{1}, Sang-Won Lee\textsuperscript{2}, Jin-Soo Kim\textsuperscript{2}

Hanyang University\textsuperscript{1}, Sungkyunkwan University\textsuperscript{2}, Korea
OpenSSD Introduction
What’s the OpenSSD Project

- Open-source SSD platform for research and education on the SSD technology since 2011
- New “OpenSSD platform” for developing SSD firmware, controller hardware, and host software

Contribution
- Indilinx (Merged to OCZ in 2012)
- HYU (Hanyang University), Korea
- SKKU (Sungkyunkwan University), Korea
Why OpenSSD?

- Solve your problem in a real system
- Share your solution with people in society
- Design your own SSD controller, if possible
- Contribute to “open” community
- Use it as a PC disk
- Play for fun
OpenSSD Project History

- **Jasmine OpenSSD (2011)**
  - SSD controller: Indilinx Barefoot (SoC w/SATA-2)
  - Firmware: SKKU VLDB Lab
  - Users from 10+ countries
  - 10+ papers published
OpenSSD Project History

- Cosmos OpenSSD (2014)
  - SSD controller: HYU Tiger3 (FPGA w/PCIe Gen2)
  - Firmware: HYU ENC Lab, SKKU VLDB Lab
  - Users from ?? countries (at least one)
  - ?? papers to be published (at least three)
### Comparison among the Platforms

<table>
<thead>
<tr>
<th></th>
<th>Jasmine OpenSSD</th>
<th>Cosmos Prototype (Tiger2)</th>
<th>Cosmos OpenSSD (Tiger3)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SSD Controller</strong></td>
<td>Indilinx Barefoot (SoC)</td>
<td>HYU Tiger2 (FPGA)</td>
<td>HYU Tiger3 (FPGA)</td>
</tr>
<tr>
<td><strong>Year</strong></td>
<td>2011</td>
<td>2012</td>
<td>2014</td>
</tr>
<tr>
<td><strong>Host Interface</strong></td>
<td>SATA-2</td>
<td>PCIe Gen1.1 (AHCI Subset)</td>
<td>PCIe Gen2 (AHCI Subset)</td>
</tr>
<tr>
<td><strong>Storage Capacity</strong></td>
<td>128 GB</td>
<td>512 GB</td>
<td>512 GB</td>
</tr>
<tr>
<td><strong>NAND Data Interface</strong></td>
<td>Asynchronous</td>
<td>Asynchronous</td>
<td>Synchronous</td>
</tr>
<tr>
<td><strong>DRAM Capacity</strong></td>
<td>64 MB</td>
<td>512 MB</td>
<td>1 GB</td>
</tr>
</tbody>
</table>
The OpenSSD Project

The OpenSSD Project is an initiative to promote research and education on the recent SSD (Solid State Drive) technology by providing easy access to OpenSSD platforms on which open source SSD firmware can be developed. Currently, we offer an OpenSSD platform based on the commercially successful Barefoot™ controller from Infinx Co., Ltd. This site is also intended to be a forum to share various simulators, tools, and workload generators and traces related to SSDs, among researchers in academia and industry.

OpenSSD Platforms

Note: A new OpenSSD platform is coming this year...

We are preparing the second OpenSSD platform called Cosmos. The Cosmos OpenSSD platform is based on the PCIe interface and will debut in the Flash Memory Summit in August, 2014. So, stay tuned!

Cosmos OpenSSD Platform

Coming soon.

Jasmine OpenSSD Platform

http://www.openssd-project.org/wiki/The_OpenSSD_Project
Design Objective

- A simple NAND storage platform to be used for
  - Firmware development and evaluation
  - Controller architecture exploration
  - New controller design (e.g. Intelligent SSD)

→ An open-source SSD platform with AP-class CPUs and other hardware resources
# Cosmos SSD Hardware Features

<table>
<thead>
<tr>
<th><strong>FPGA</strong></th>
<th><strong>Xilinx Zynq-7000 series</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MCU</strong></td>
<td><strong>Cortex™-A9</strong></td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td><strong>Clock Frequency</strong></td>
</tr>
<tr>
<td><strong>MCU</strong></td>
<td><strong>667 MHz</strong></td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td><strong>Total Capacity</strong></td>
</tr>
<tr>
<td><strong>NAND</strong></td>
<td><strong>512 GB</strong></td>
</tr>
<tr>
<td><strong>Organization</strong></td>
<td><strong>4-Channel / 8-Way</strong></td>
</tr>
<tr>
<td><strong>DRAM</strong></td>
<td><strong>Device Interface</strong></td>
</tr>
<tr>
<td><strong>Device Interface</strong></td>
<td><strong>DDR3 (533 MHz)</strong></td>
</tr>
<tr>
<td><strong>Total Capacity</strong></td>
<td><strong>1 GB</strong></td>
</tr>
<tr>
<td><strong>Bus</strong></td>
<td><strong>System</strong></td>
</tr>
<tr>
<td><strong>System</strong></td>
<td><strong>AXI-Lite (Bus width: 32 bits)</strong></td>
</tr>
<tr>
<td><strong>Storage Data</strong></td>
<td><strong>AXI (Bus width: 64 bits, Burst Length: 256)</strong></td>
</tr>
<tr>
<td><strong>SRAM</strong></td>
<td><strong>256 KB (FPGA Internal)</strong></td>
</tr>
<tr>
<td><strong>Error Correction Code</strong></td>
<td><strong>BCH 32 bits/2 KB</strong></td>
</tr>
<tr>
<td><strong>Host Interface</strong></td>
<td><strong>PCI-Express Gen2 4-Lane (2 GB/s)</strong></td>
</tr>
<tr>
<td><strong>Power Measurement</strong></td>
<td><strong>NAND Flash and Board Power Measurement (External ADC Module or NI DAC)</strong></td>
</tr>
</tbody>
</table>
## NAND Flash Memory Specification

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor / Model</td>
<td>SK Hynix / H27QDG8VEBIR-BCB</td>
</tr>
<tr>
<td>Device Interface</td>
<td>Synchronous (ONFI 2.2)</td>
</tr>
<tr>
<td>Cell Technology</td>
<td>MLC (2 Bits/cell)</td>
</tr>
<tr>
<td>Capacity</td>
<td>16 GBs/package</td>
</tr>
<tr>
<td>Data Area</td>
<td>8192 Bytes/page</td>
</tr>
<tr>
<td>Spare Area</td>
<td>640 Bytes/page</td>
</tr>
<tr>
<td>Dies per Package</td>
<td>4 dies (QDP)</td>
</tr>
<tr>
<td>Data Cycle</td>
<td>20 ns (DDR: 10 ns)</td>
</tr>
<tr>
<td>Page Read</td>
<td>70 us (typical)</td>
</tr>
<tr>
<td>Page Program</td>
<td>1400 us (typical)</td>
</tr>
<tr>
<td>Block Erase</td>
<td>3.5 ms (typical)</td>
</tr>
<tr>
<td>Speed</td>
<td>NV-DDR100</td>
</tr>
</tbody>
</table>
## DRAM Memory Specification

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vendor</strong></td>
<td>Samsung (K4B4G1646B-HCK0)</td>
</tr>
<tr>
<td><strong>Device Interface</strong></td>
<td>DDR3-1066F</td>
</tr>
<tr>
<td><strong>DRAM Device Bus Width</strong></td>
<td>16 Bits</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>1 GB (512 MBs x2)</td>
</tr>
<tr>
<td><strong>CAS Latency</strong></td>
<td>7 Cycles</td>
</tr>
<tr>
<td><strong>CAS Write Latency</strong></td>
<td>6 Cycles</td>
</tr>
<tr>
<td><strong>RAS to CAS Delay</strong></td>
<td>7 Cycles</td>
</tr>
<tr>
<td><strong>Precharge Time</strong></td>
<td>7 Cycles</td>
</tr>
<tr>
<td><strong>Throughput</strong></td>
<td>8.5 GB/s</td>
</tr>
</tbody>
</table>
Cosmos SSD Platform Board

<table>
<thead>
<tr>
<th>No.</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Xilinx Zynq-7000 (HYU Tiger3 Controller)</td>
</tr>
<tr>
<td>2</td>
<td>DDR3 (512 MB x2)</td>
</tr>
<tr>
<td>3</td>
<td>NAND SODIMM x2</td>
</tr>
<tr>
<td>4</td>
<td>External PCIe x2</td>
</tr>
<tr>
<td>5</td>
<td>USB JTAG</td>
</tr>
<tr>
<td>6</td>
<td>UART</td>
</tr>
<tr>
<td>7</td>
<td>USB 2.0 OTG</td>
</tr>
<tr>
<td>8</td>
<td>SD Card</td>
</tr>
<tr>
<td>9</td>
<td>Power Measurement Connector</td>
</tr>
</tbody>
</table>
Statistics on the performance and utilization of hardware resources
Use of Performance Monitor

Cosmos OpenSSD Platform

- CPU
- I/O Ports
- I2C / SPI
- UART
- Memory Controller
- Host I/F Controller (PCIe)
- Performance Monitor
- BCH Engine
- Channel Controller
- Storage Controller
- DRAM
- NAND Array

Application (C/C++)
Software
Software module performance

Host System
Storage performance

RTL (Verilog/VHDL)
Hardware
Hardware module performance
Example: Operation Time Breakdown

Host → Buffer (DRAM) → NAND

Response Time

Page

Page

Page

Data Register

tPROG

buf2nand

host2buf

cflru

AVG

LSB

MSB (FAST)

MSB (SLOW)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

tPROG

LSB

MSB (FAST)

MSB (SLOW)

Typical

Flash Memory Summit 2014
Santa Clara, CA
Available FTLs (under GPL)

- **TutorialFTL** (by HYU ENC Lab)
  - Page-mapping FTL w/o GC
  - Device initialization and interface definition

- **GreedyFTL** (by HYU ENC Lab)
  - Page-mapping FTL w/ Greedy GC

- **DramDisk** (by HYU ENC Lab)
  - DRAM Disk for measuring maximum read/write performance (PCIe + DRAM)
Common Operation Flow

Device Initialization

Firmware Start ()

CMD READY?

Yes

No

CMD END?

Yes

No

IDLE OP?

Not Available

Available

Background OP

Call buf_read()

Call NAND_read()

Call buf_write()

Call NAND_write()
Host Software

- **AS-IS**
  - Makes the storage work as a drive
  - Handles read/write requests from kernel
  - Needs to install storage-specific device driver
    - Windows7 64-bit device driver (by HYU ENC Lab)
    - Linux 3.2.X kernel device driver (by HYU ENC Lab)

- **TO-BE**
  - Compatible with PCIe system drivers
    - AHCI support
    - NVMe support
Some Use Cases
When compared to the simulator, the performance of cosmos prototype is 16% lower and 22.3% higher in read and program operations, respectively.

Note: Cosmos prototype is used in this experiment

*Ref.*) FlashSim: A Simulator for NAND Flash-Based Solid-State Drives
X-FTL

- Transactional FTL for SQLite (SIGMOD 2013)
  - Atomic write of data pages
    - Offload the semantic of all-or-nothing propagation of data pages carried out in host system down to FTL layer
    - Avoid “redundant writes”
  - Originally on top of Jasmine OpenSSD; now in porting to Cosmos

![Execution time vs. # of updated pages per transaction graph]
X-FTL (Continued)

- **Architecture**

File System Interface

- **Read(Pi), Write(Pi), fsync, ioctl(abort)**

Storage Interface

- **Read(Ti, Pi), Write (Ti, Pi), Commit(Ti), Abort(Ti)**

Traditional FTL with Garbage Collection

- **Page Mapping Table (L2P)**

<table>
<thead>
<tr>
<th>LPN</th>
<th>PPN</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>P1</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Pn</td>
<td>:</td>
</tr>
</tbody>
</table>

Propagation at commit

- **X-FTL**

<table>
<thead>
<tr>
<th>TID</th>
<th>LPN</th>
<th>PPNnew</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>Ti</td>
<td>P1</td>
<td>Active</td>
<td>:</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
<td>Active</td>
</tr>
<tr>
<td>Ti</td>
<td>Pn</td>
<td>Active</td>
<td>:</td>
</tr>
</tbody>
</table>

Write/Read

Commit/Abort

Recovery

Old copy of P1, ..., Pn

New copy of P1, ..., Pn
ActiveSort (HotStorage 2014)
- Accelerates external sorting
- Performs on-the-fly merge inside the SSD when the results are requested
- Eliminates extra data transfer
- Increases the lifetime of SSDs

*Ref.)* Accelerating External Sorting via On-the-fly Data Merge in Active SSDs
Stay Tuned

- Sources will be available soon at the OpenSSD webpage
- Sources could be updated by other users except us (welcome all the time!)
- And more activities will be posted to the webpage as well
Call For Participation

- Welcome any contributions from
  - SSD manufacturers
  - NAND flash vendors
  - Research groups
  - Individual developers
  - …
Contributors

Prof. Yong Ho Song Ph.D.
Hanyang University

Jaehyeong Jeong Ph.D.
System Architecture & SSD HW Controller

Youngnam Kim
SSD HW Architecture

Ilyong Jung
Controller Architecture & ECC Engine

Gi-Hwan Oh
xFTL Design

Prof. Sang-Won Lee Ph.D.
Sungkyunkwan Univ

Sanghyuk Jung Ph.D.
System Architecture & SSD SW Firmware

Taeyeong Huh
SSD HW Controller

Gyeongyong Lee
SSD SW Firmware

Young-Sik Lee
Active SSD

Prof. Jin-Soo Kim Ph.D.
Sungkyunkwan Univ

Youngin Jo
SSD Architecture & PCIe Interface

Sangjin Lee
Controller Architecture & PCIe Interface

Luis Cavazos
Active SSD

Prof. Jin-Soo Kim Ph.D.
Sungkyunkwan Univ

Sung-Rae Kim Ph.D.
ECC Algorithm
Thank you

For further information, visit http://www.openssd-project.org