# SSD Firmware Implementation Project
# - Lab. #1 -

Sang-Phil Lim (lsfeel0204@gmail.com)
SKKU VLDB Lab.

2011-03-24

# Contents

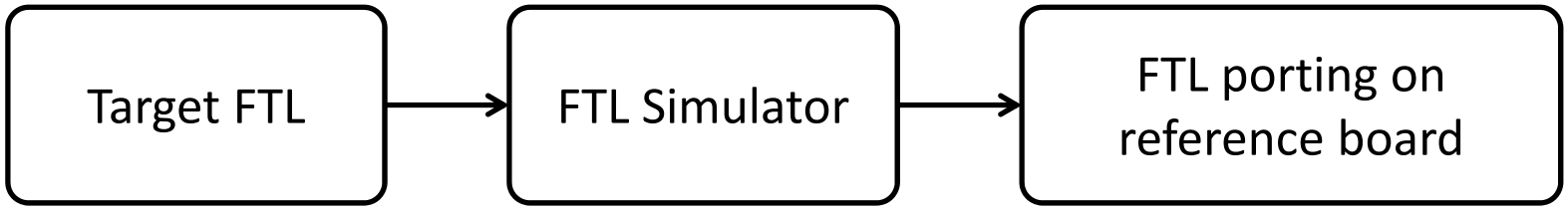
- Project Overview
- Lab. Time Schedule
- Project #1 Guide – FTL Simulator Development

# Project Overview

- Project #1: FTL Simulator Development
  - Implement a popular FTL scheme on the simple FTL simulator
  - Perform FTL simulations
- Project #2: SSD Firmware Implementation
  - Porting own FTL code on actual SSD platform
  - Evaluate SSD performance with benchmarking tool

# Project Overview

- SSD Firmware Implementation Project
- Goal: “Achieve in-depth knowledge of embedded software design and practical experience”

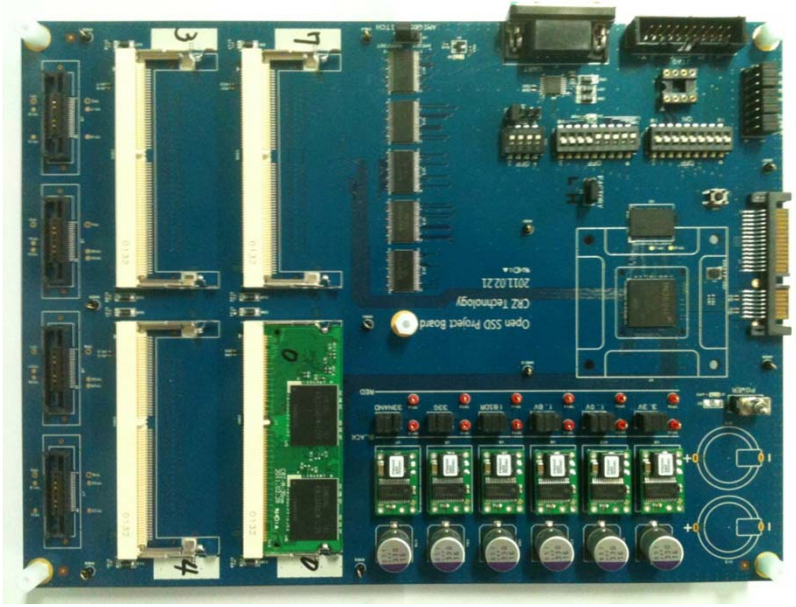


# Lab. Time Schedule

| Lab. | Title |
|---|---|
| #1 | FTL Simulator Development Guide |
| #2 | FTL Simulation Guide |
| #3 | Project 1 Presentation |
| #4 | Jasmine OpenSSD platform tutorial #1 |
| #5 | Jasmine OpenSSD platform tutorial #2 |
| #6 | FTL Porting Guide |
| #7 | Firmware Debugging Guide |
| #8 | SSD Performance Evaluation Guide |
| #9 | Project 2 Presentation |

# Jasmine OpenSSD Platform

- Based on Indilinx Barefoot™ SSD controller
  - 96KB SRAM, 64MB DRAM, SATA 2.0 host interface
  - maximum 256GB capacity



# Project #1 – FTL Simulator Development

# Project Guide Line

- First, each team investigate a popular FTL schemes such as BAST, FAST, LAST, DAC, etc.
- Next, implement  the FTL scheme on the simple FTL simulator
- (Generate I/O workloads for simulation)
- Simulate FTL algorithm and evaluate the performance results

# Development Environment

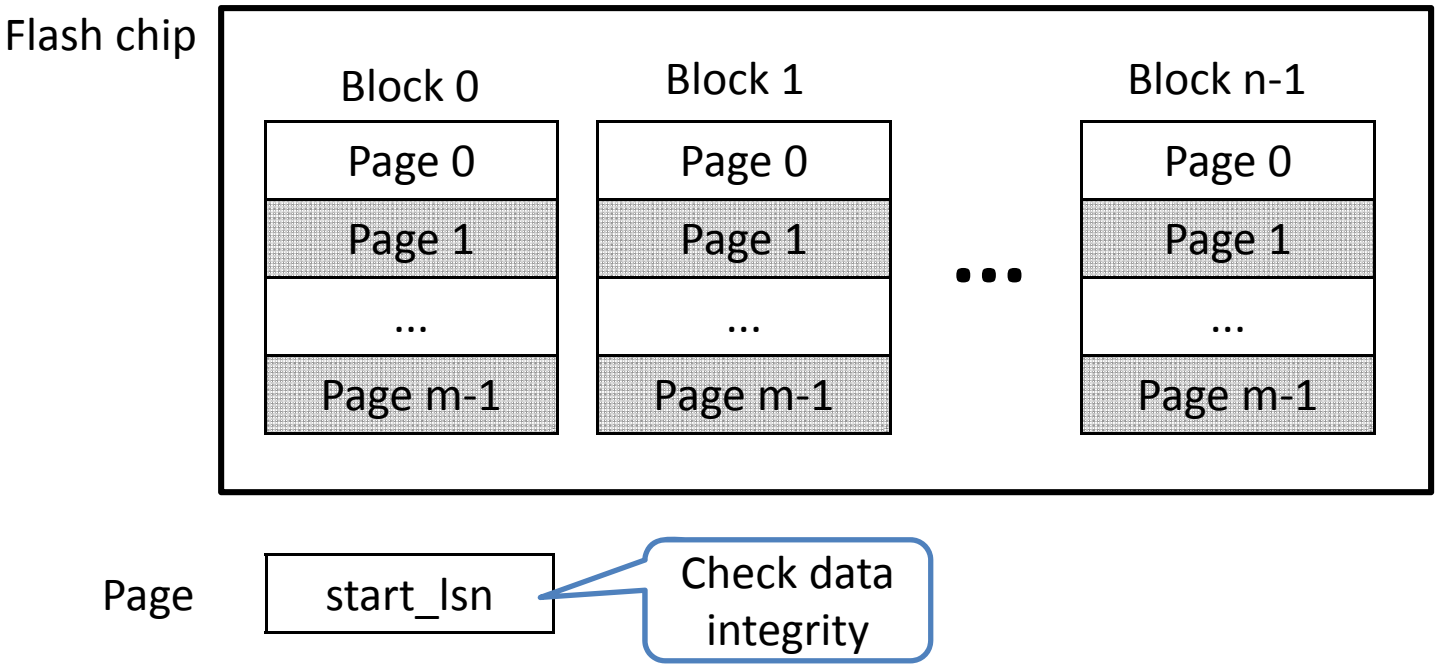
- OS: Windows
- Build tool: Microsoft Visual Studio 2010 Express edition (free)
  - http://www.microsoft.com/express/Downloads/#2010-Visual-CPP

# FTL Simulator Design Principles

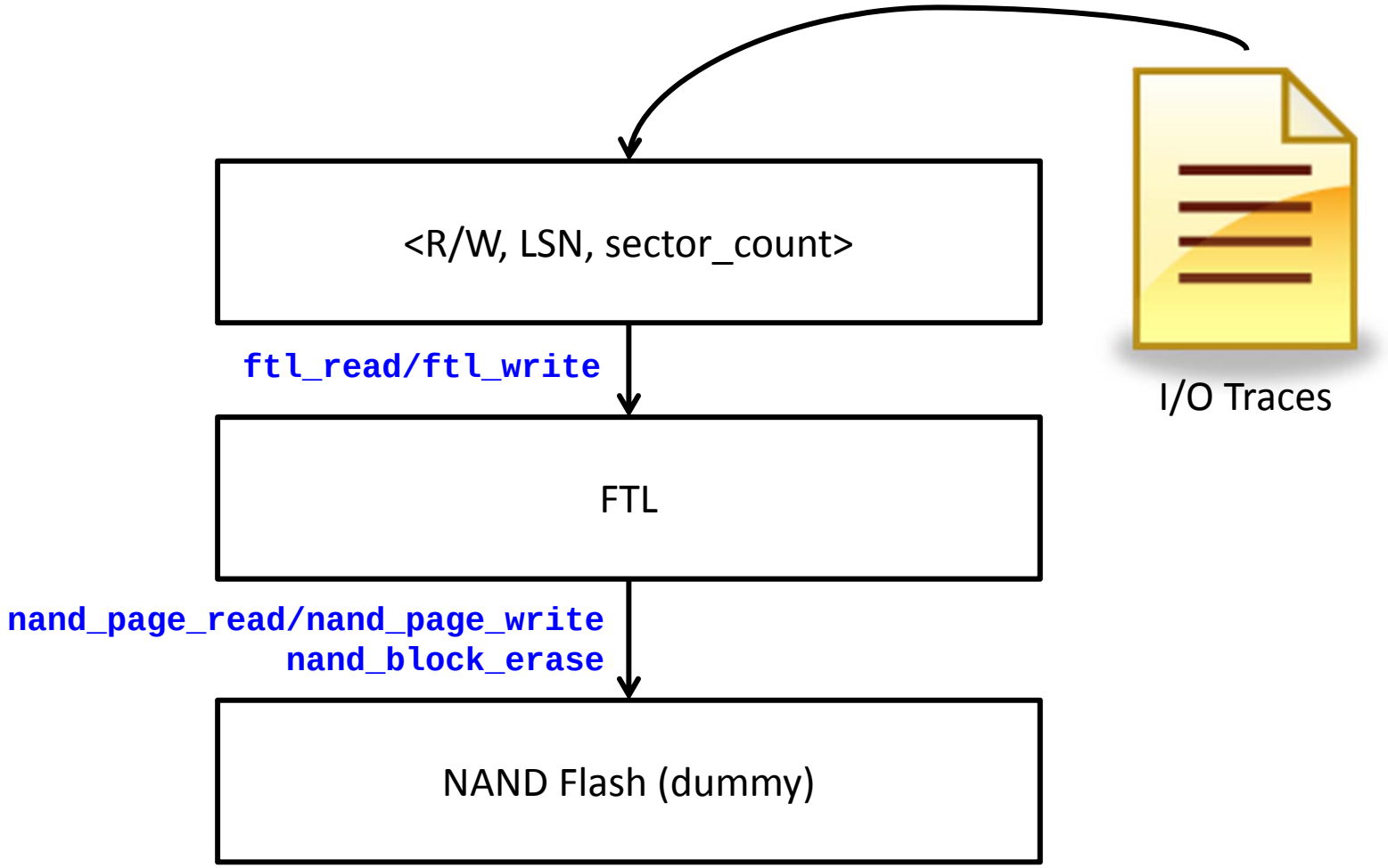
- Basic architecture
  - Single chip, Synch IO (Not support I/O parallelism)
- DRAM
  - Sufficient DRAM
  - All metadata cached in DRAM
- NAND flash
  - NAND Flash NOP(**N**umber **O**f **P**rogramming) = 1
  - Only measuring NAND flash's chip-level overhead
  - Only count NAND flash's primitive operations
    - Page read/write
    - Block erase

# Logical View of NAND Flash

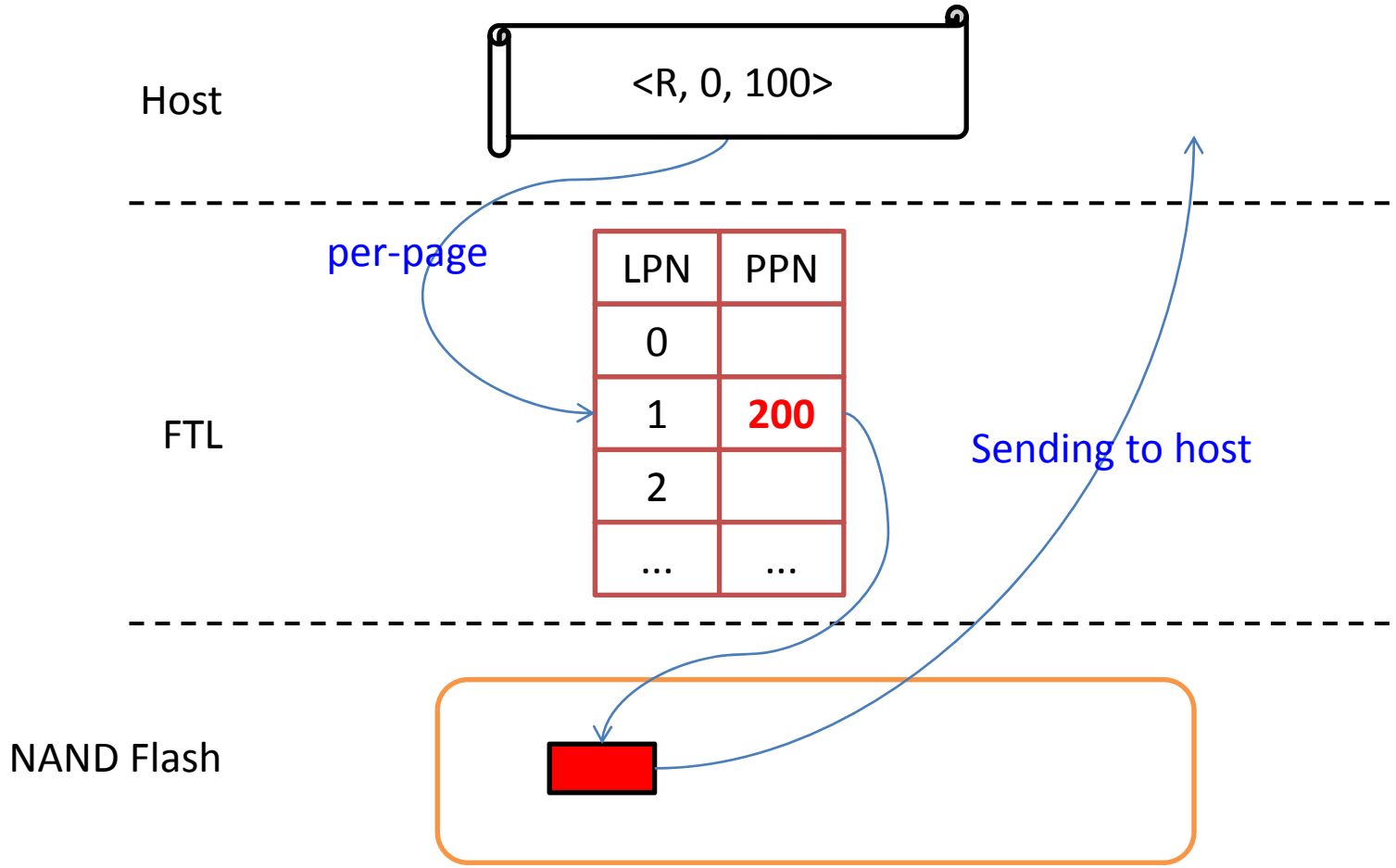
- Single chip basis
- Not contain actual user data



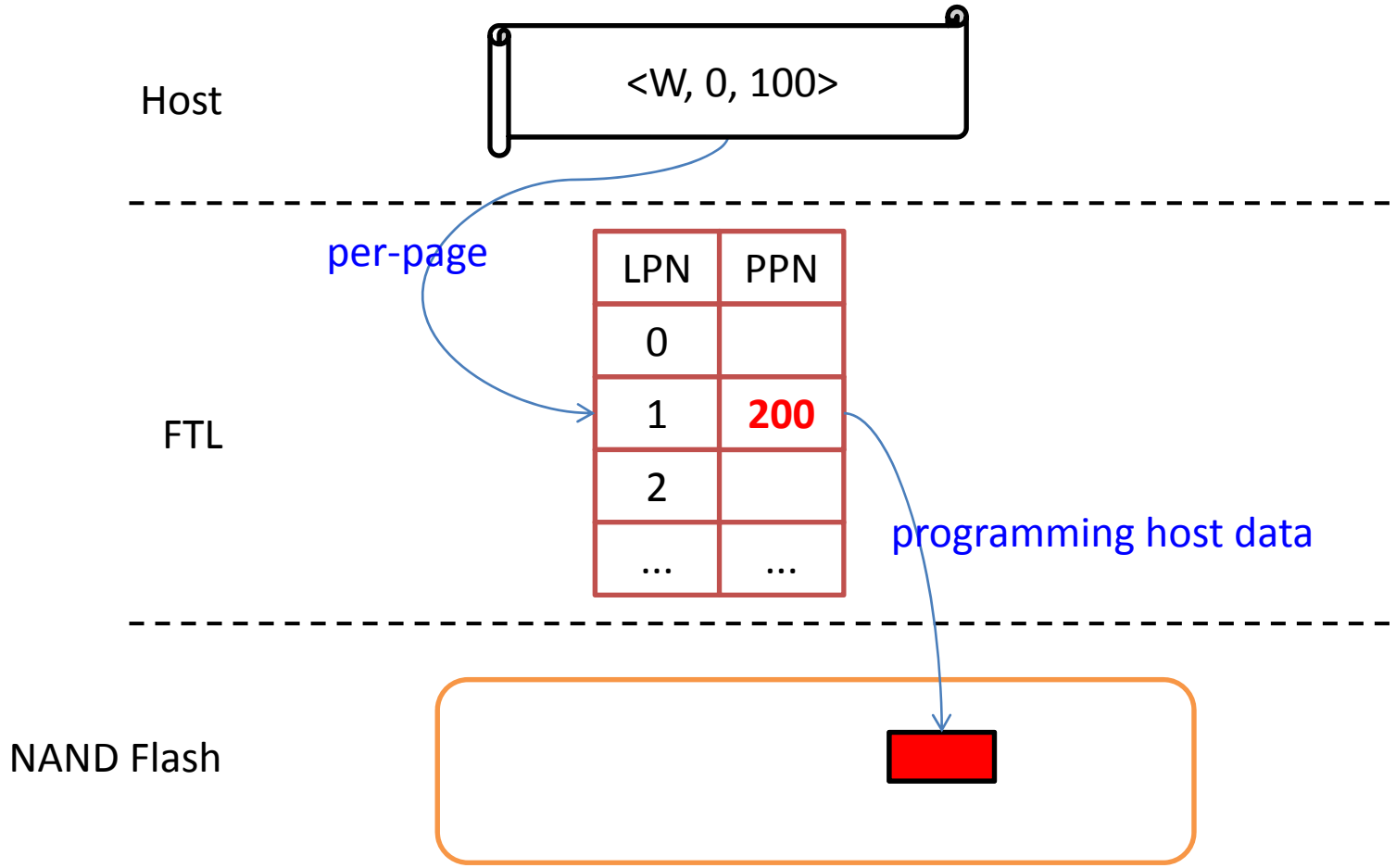
# FTL Simulator Overview



# Basic Read Operation



# Basic Write Operation



# Notice for Lab #2

- Each team should investigate an FTL scheme
  - Fully understand the target FTL include mapping algorithm and NAND usage
  - ! Reading list related to FTLs is downloadable @ http://csl.skku.edu/ICE3028S11/Resources
- Next Lab time, you should summarize key idea of target FTL and present within 5 slides

# Paper Reading List


1. M.-L. Chiang, P. C. H. Lee, and R.-C. Chang, "Using Data Clustering to Improve Cleaning Performance for Flash Memory," Software - Practice and Experience, Vol. 29, No. 3, 1999.
2. J. Kim, J. M. Kim, S. Noh, S. L. Min, and Y. Cho, "A Space-efficient Flash Translation Layer for CompactFlash Systems," IEEE Transactions on Consumer Electronics, Vol. 48, No. 2, pp.366-375, 2002.
3. S.-W. Lee, D.-J. Park, T.-S. Chung, D.-H. Park, .-J. Song, "A Log-buffer based Flash Translation Layer using Fully-Associative Sector Translation," ACM Transactions on Embedded Computing Systems, Vol. 6, No. 3, 2007.
4. D. Jung, J.-U. Kang, H. Jo, J.-S. Kim, and J. Lee, "Superblock FTL: A Superblock-based Flash Translation Layer with a Hybrid Address Translation Scheme," ACM Transactions on Embedded Computing Systems, Vol. 9, No. 4, 2010.
5. S. Lee, D. Shin, Y.-J. Kim, and J. Kim, "LAST: Locality-aware Sector Translation for NAND Flash Memory-based Storage Systems," Proc. of the 1st International Workshop on Storage and I/O Virtualization, Performance, Energy, Evaluation and Dependability (SPEED08), pp.36-42, 2008.
6. Y.-G. Lee, D. Jung, D. Kang, and J.-S. Kim, "μ-FTL: A Memory-Efficient Flash Translation Layer Supporting Multiple Mapping Granularities," Proceedings of the 8th Annual ACM Conference on Embedded Software, 2008.
7. A. Gupta, Y. Kim, and B. Urgaonkar, "DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mappings," Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2009.
8. H. Kwon, E. Kim, J. Choi, D. Lee, and Sam H. Noh, "Janus-FTL: Finding the Optimal Point on the Spectrum between Page and Block Mapping Schemes," Proceedings of the 10th ACM Conference on Embedded Software, 2010.
9. F. Chen, T. Luo, and X. Zhang, "CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives," Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST), 2011.

# Contact with TA

- Office: VLDB Lab. (#26314A), 2nd Engr. Bldg.
- By e-mail: lsfeel0204@gmail.com
- By phone: 031-290-7988
- By twitter: @ice3028skku

# Any Questions?