UFS 2.0 NAND Device Controller with SSD-Like Higher Read Performance

Konosuke Watanabe

Toshiba

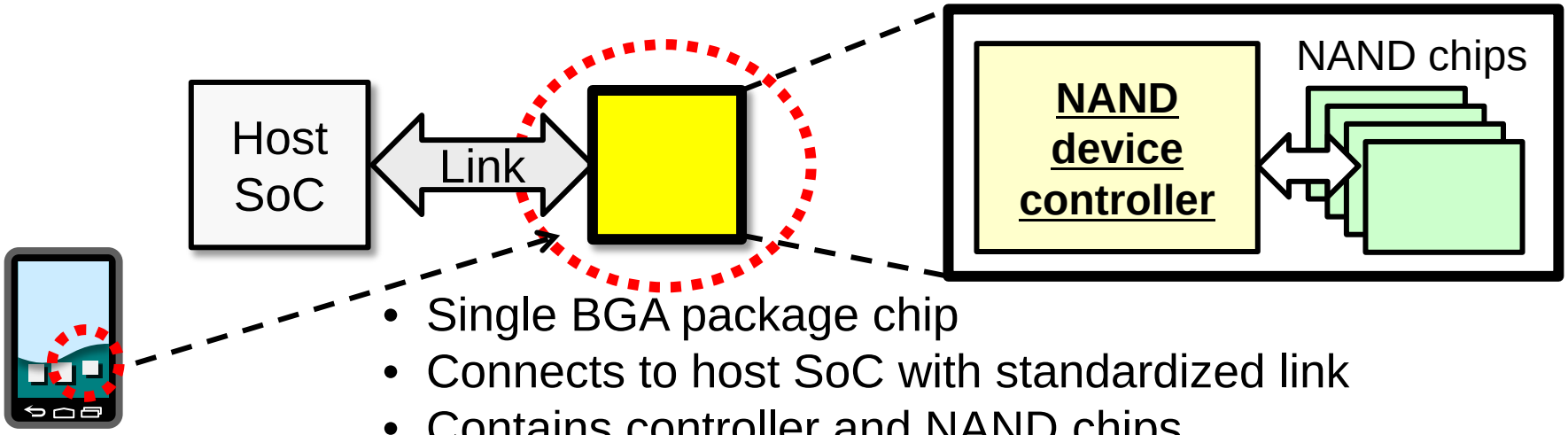
TOSHIBA
Leading Innovation

How did our embedded NAND storage device get SSD-like higher read performance?

Outline

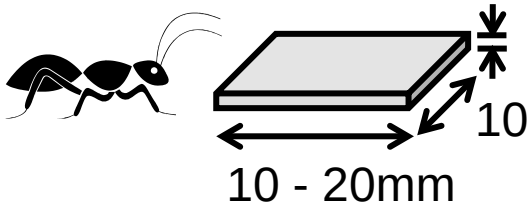
- Background
- Approaches
- Example results

Embedded NAND Storage Device



- Single BGA package chip
- Connects to host SoC with standardized link
- Contains controller and NAND chips

It must be small and lower power



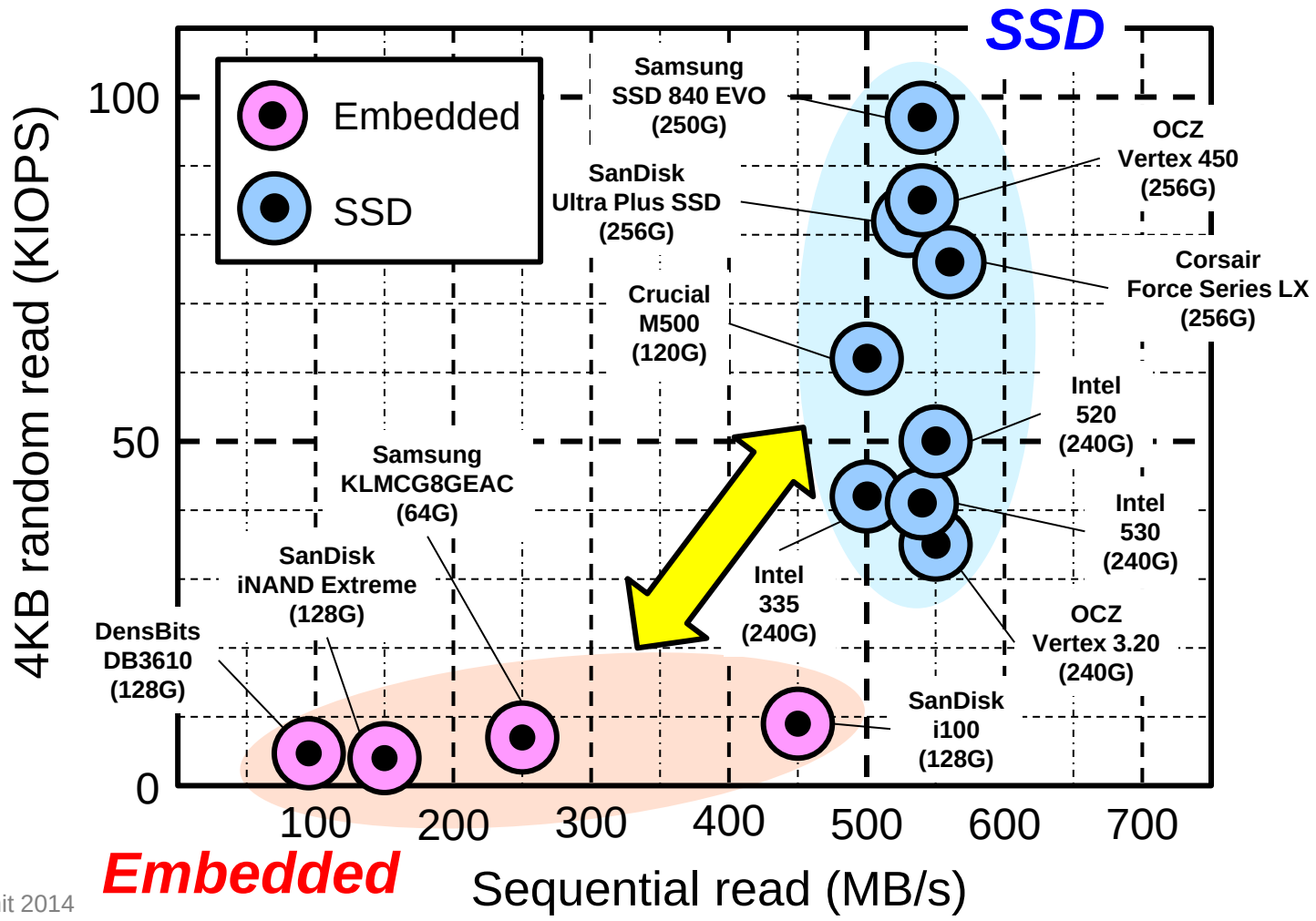
1 - 2 W (active)

...but now higher performance is also expected

UFS 2.0: 1160MB/s (5.8Gbps x 2-lane)

cf. SATA 3.0: 600MB/s

Performance of Current Embedded NAND Storage Device Products is Quite Low



Read performance of embedded NAND storage device and SSD products

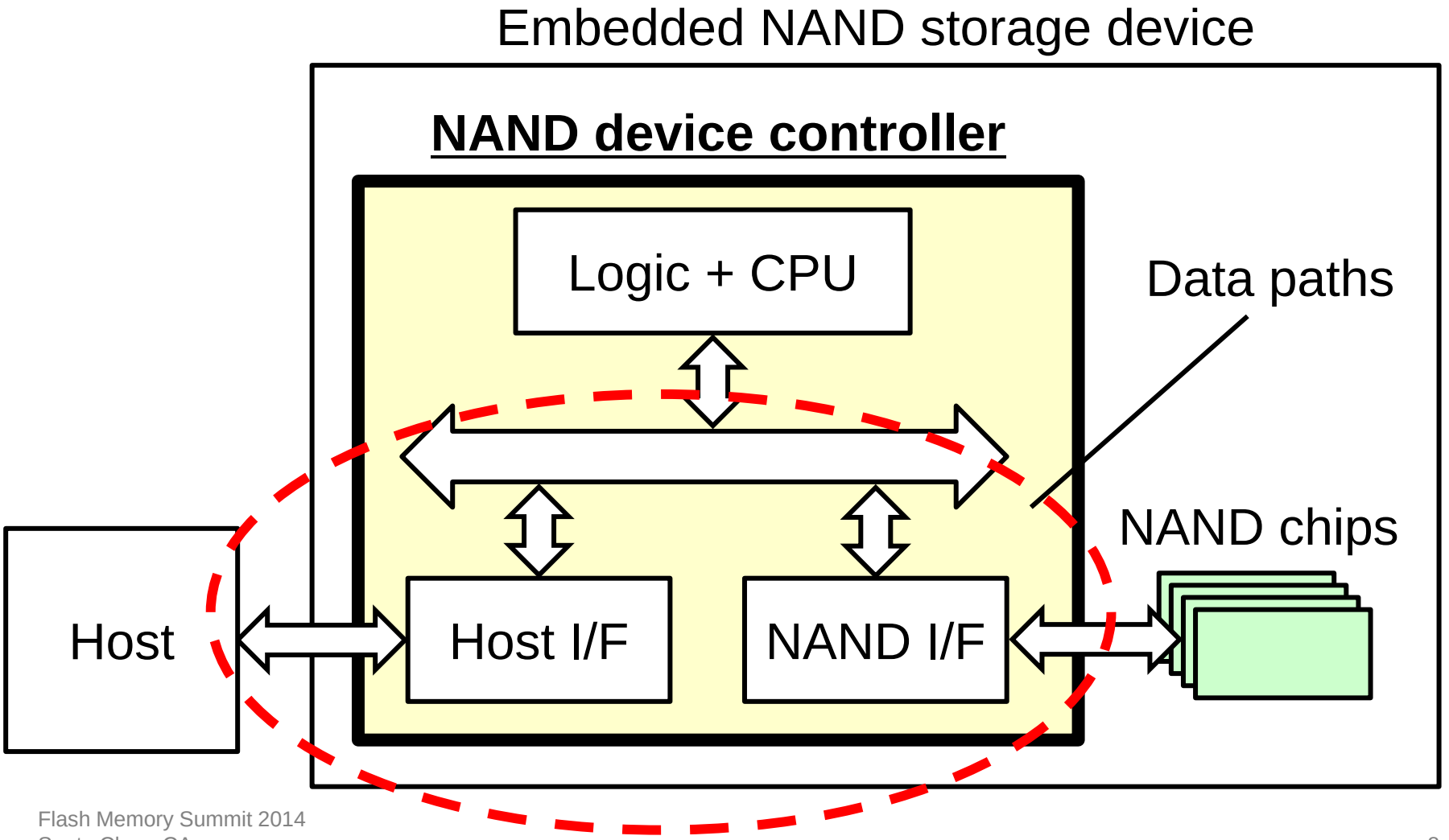
Reason for Their Poor Performance and Our Challenge

- Embedded NAND storage device has strict limitation
  - It must be **small** and **lower power consumption**
  - Cannot take SSD's "rich man's" approach
    - Improve performance by utilizing rich resources
      - Massive NAND chips / channels
      - Powerful CPU
      - Large capacity RAM
- Goal: Achieve SSD-like higher performance without rich resources
  - Focused on read performance of UFS 2.0 device

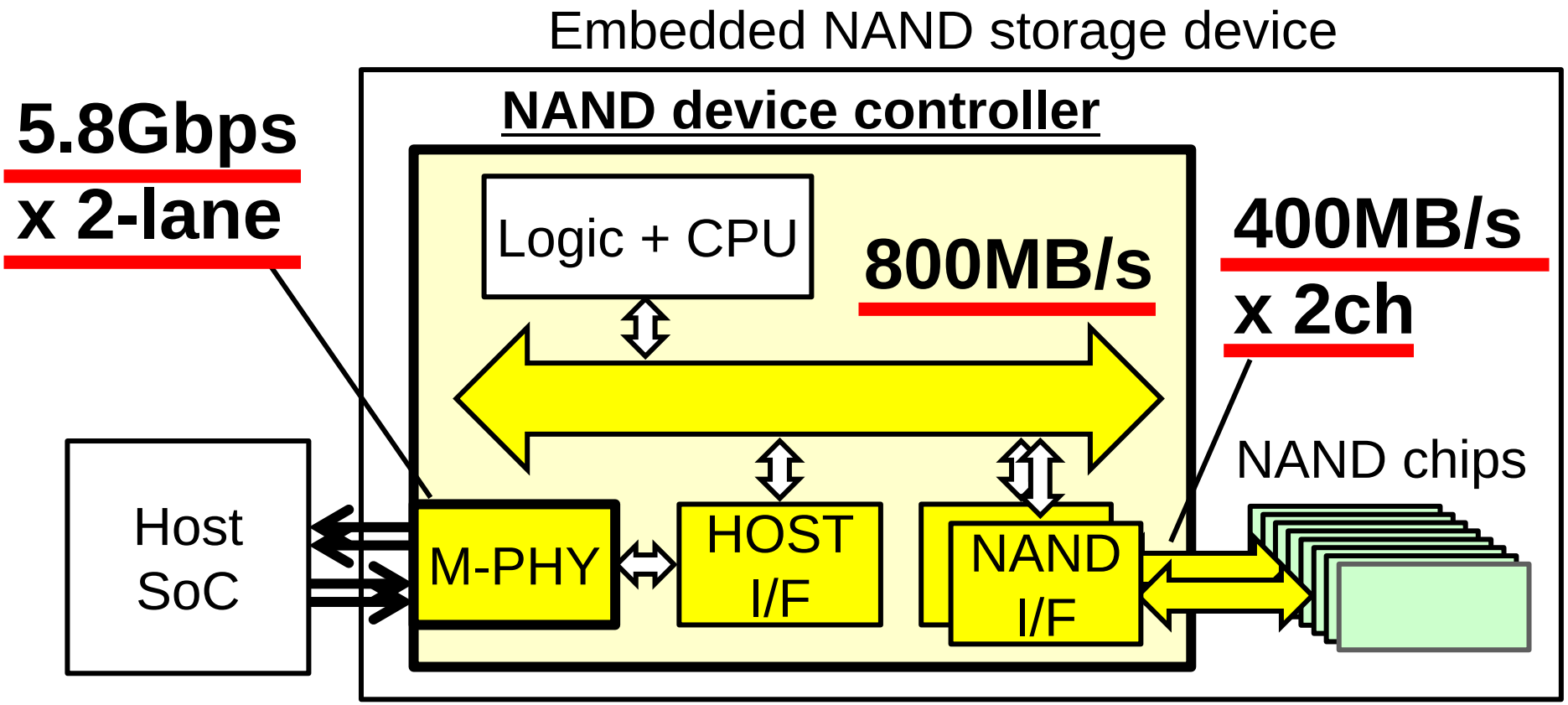
Outline

- Background
- Approaches
- Example results

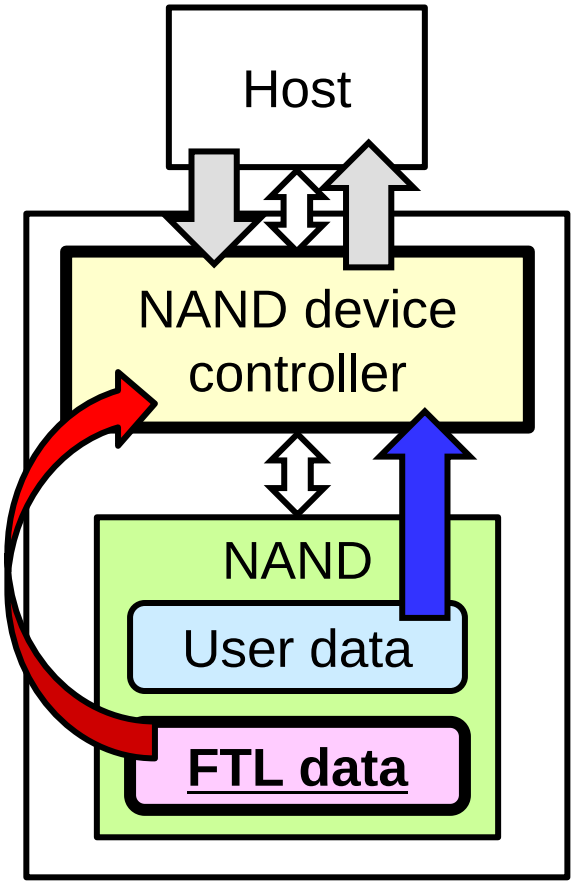
Data Paths Strengthen (1)
Conventional NAND Device Controller



Data Paths Strengthen (2)
Strengthened Data Paths Minimally

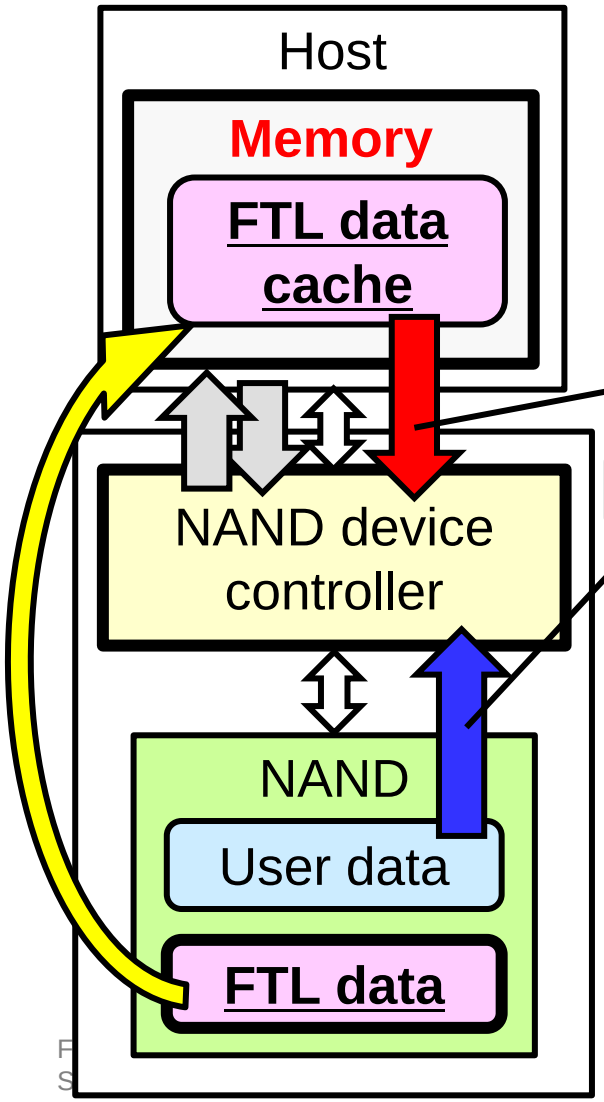


Random Read Latency Reduction (1) Latency of Random Read



- Random read is accompanied by a couple of NAND reads
  - Target data (user data) read
  - A couple of **FTL data reads**
    - Special information for address translation
- NAND read latency is several 10s µsec
- Random read latency is larger than 100µsec
- Can be reduced by caching FTL data on large capacity RAM
  - With increasing of package size and power consumption

Random Read Latency Reduction (2) Using Unified Memory Architecture (UMA)



1. Made NAND device controller available to access host memory
   - Extended UFS standard
2. Cached FTL data on host memory

1. Reads host memory to get FTL data
2. Reads NAND to get user data

Host memory read latency << NAND read latency

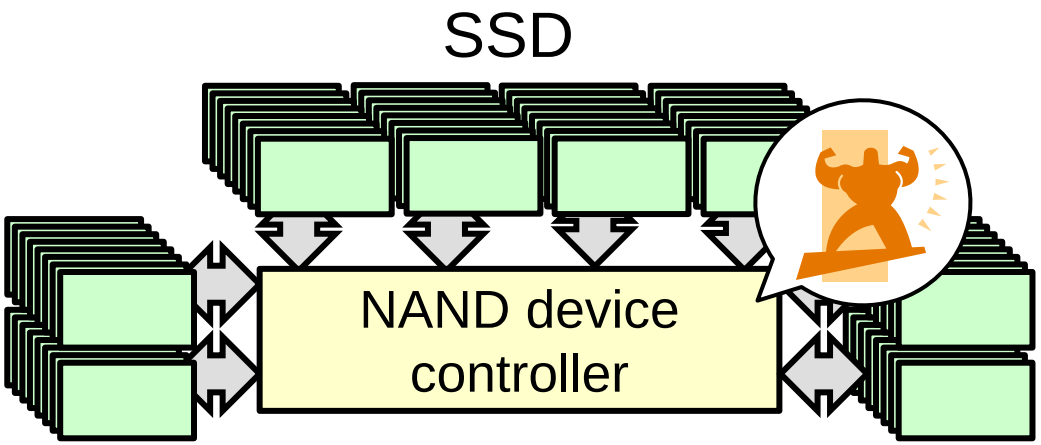
Reduces random read latency

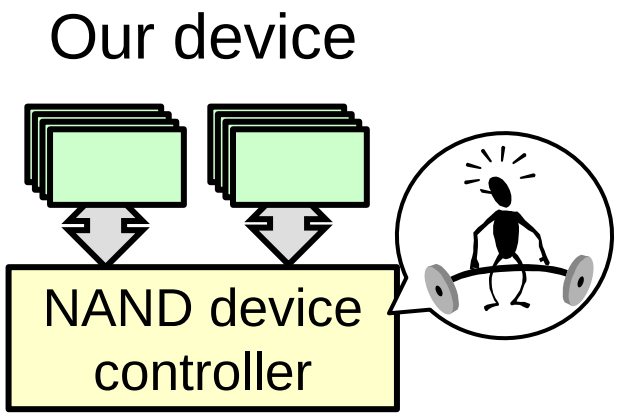
by 50% in our evaluation environment

With no increase in package size and chip power consumption

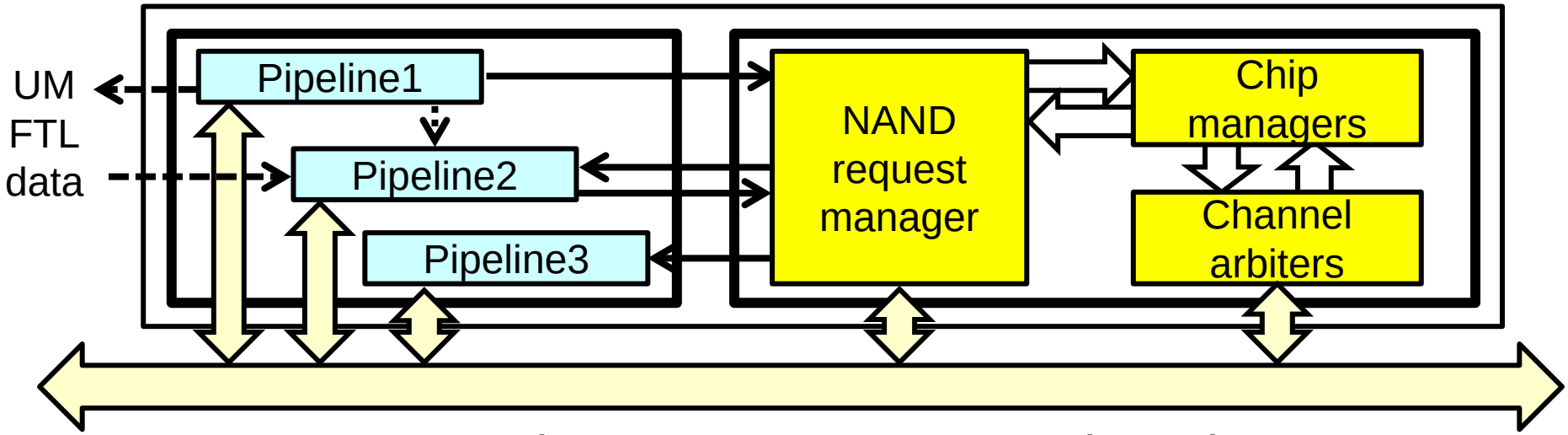
Random Read Throughput Boost (1)
Parallel NAND Reading

- NAND chips can be read in parallel
  - Multiple NAND chips (and channels)
  - Multiple outstanding NAND read requests
- Parallelism is determined by number of NAND chips, channels and read requests





Random Read Throughput Boost (2)
Random Read Command Processor (RRCP)



Front-end

Parallel out-of-order random read command processing

Back-end

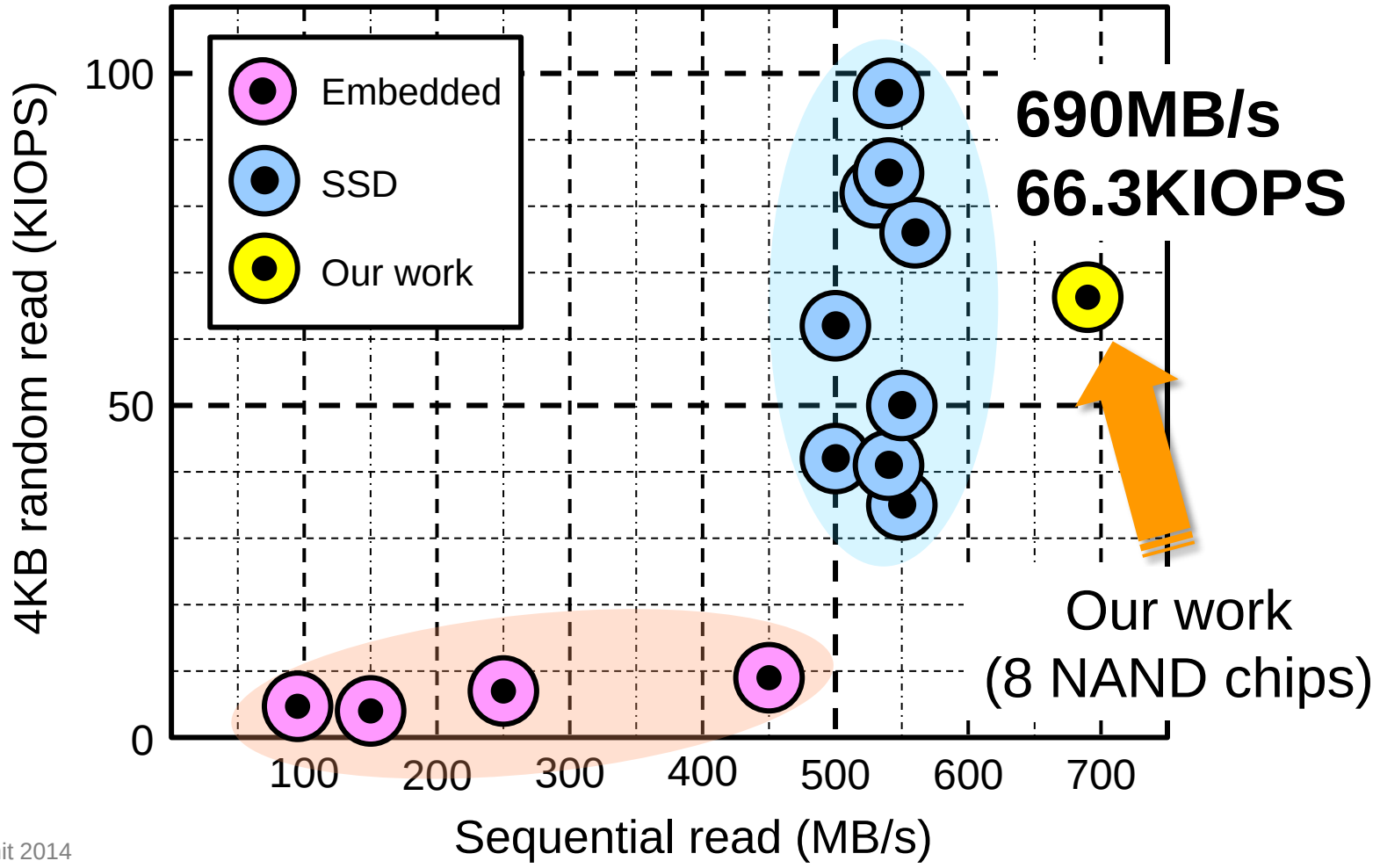
Efficient NAND read access arbitration and management

7.0 NAND chips are activated in parallel on average in our evaluation environment (with 8 NAND chips)

Outline

- Background
- Approaches
- Example results

SSD-like Read Performance is Achieved



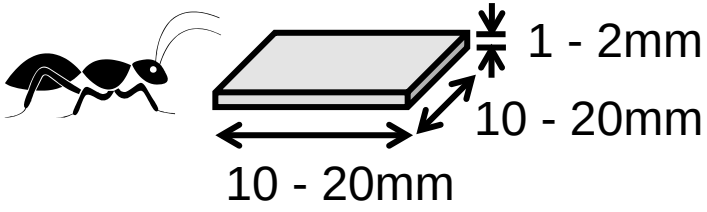
Package Size and Power Consumption can be Maintained

Package size (with NAND chips)		**11.5mm x 13.0mm** x 1.2mm
Package power	Active (Seq. read)	< **1.5 W**‡
	Idle / Sleep	< 1.45 mW‡ / < 0.30 mW

‡ Depends on storage capacity

cf. Typical level

Size



Power consumption

1 - 2 W (active)

Conclusion

- Developed UFS 2.0 embedded NAND storage device
- Improved read performance with three approaches
  - Strengthen data paths minimally
  - Reduce random read latency by using unified memory architecture and FTL data caching
  - Boost random read throughput by maximizing read command parallelism with Random Read Command Processor
- SSD-like read performance is achieved
  - 4KB random read: 66.3KIOPS
  - Sequential read: 690MB/s
- Package size and power consumption can be maintained
  - Package size (with NAND chips) : 11.5mm x 13.0mm x 1.2mm
  - Active (seq. read / write) : < 1.5 W