Layer-by-layer Adaptively Optimized ECC for NAND Flash SSD Storing CNN Weights

Keita Mizushina, Toshiki Nakamura
and Ken Takeuchi

Chuo University, Tokyo, Japan

Outline

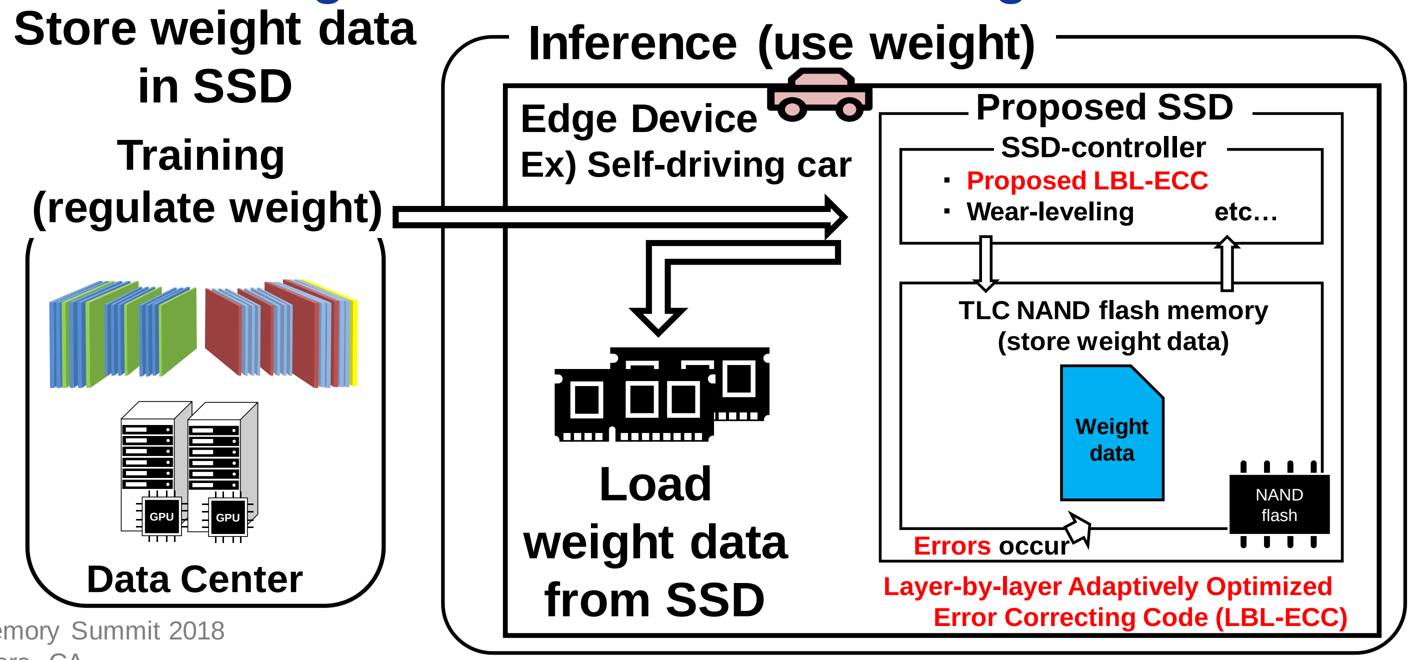
- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

Outline

- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

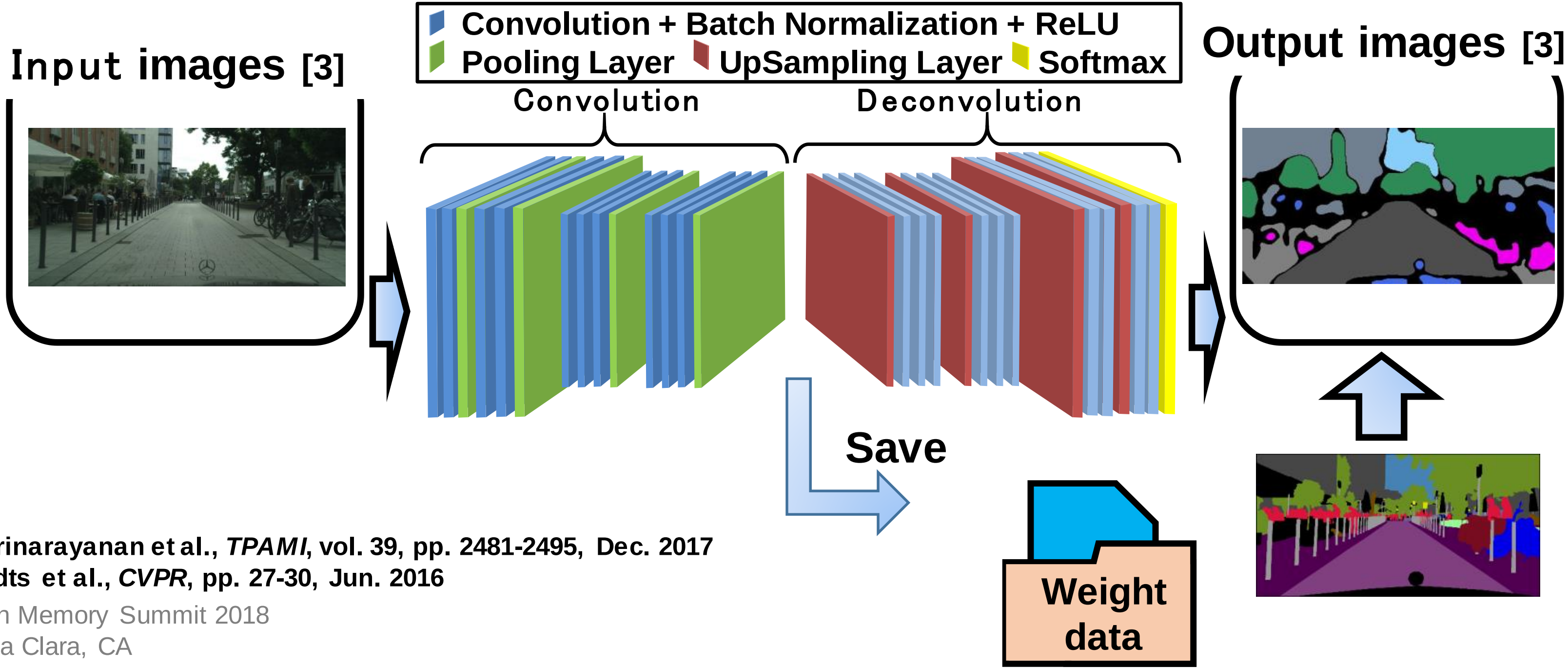
Concept of This Work

- Train network in data center with GPUs
- Store weight data and infer on Edge devices



SegNet Architecture

- SegNet [2] is deep convolutional encoder-decoder architecture for semantic pixel wise labelling

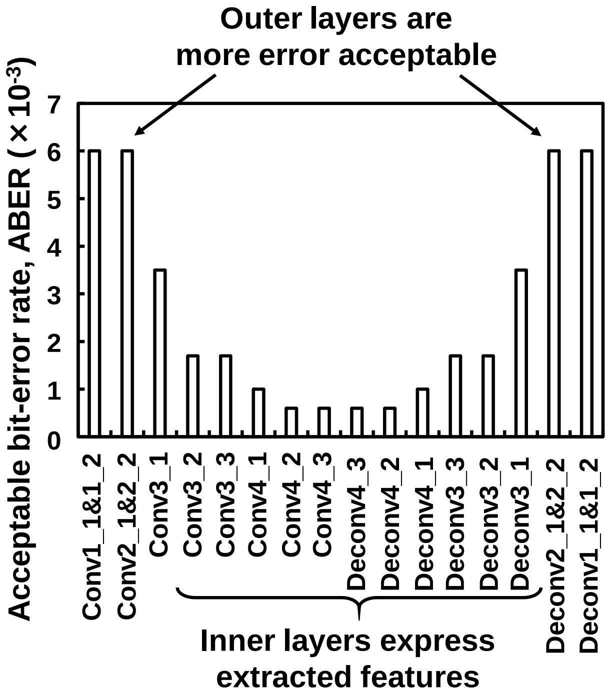


[2] V. Badrinarayanan et al., *TPAMI*, vol. 39, pp. 2481-2495, Dec. 2017

[3] M. Cordts et al., *CVPR*, pp. 27-30, Jun. 2016

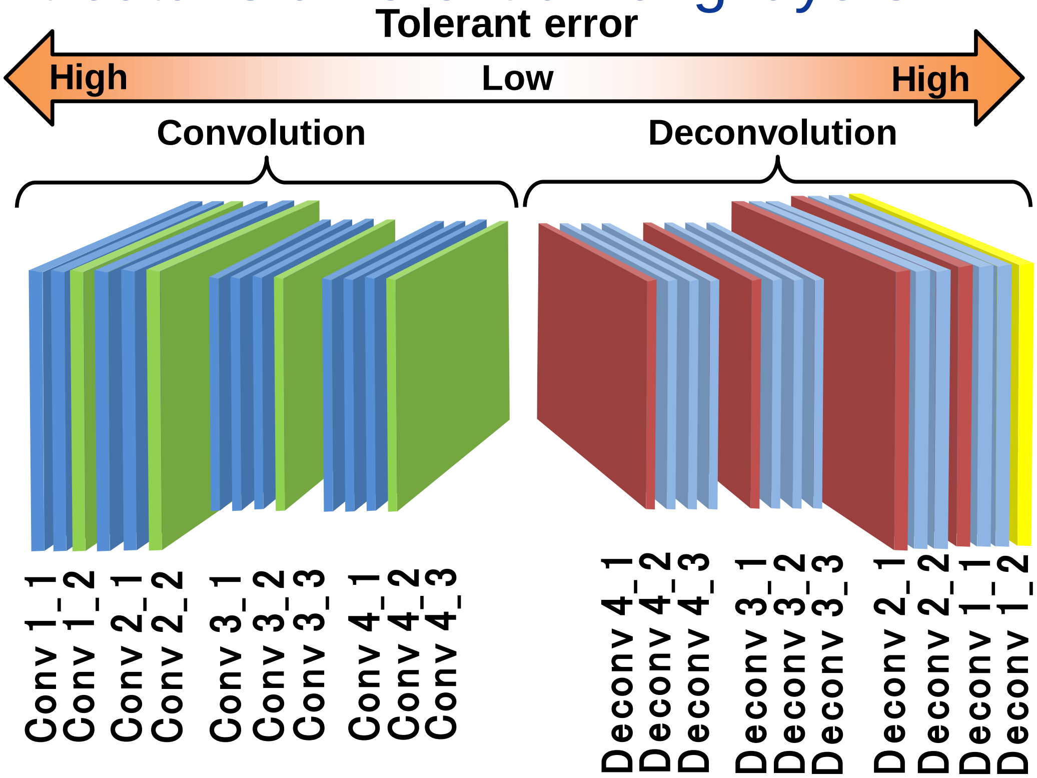
Importance of Weight Data

- Importance of weight data is different among layers



Outer layers are more error acceptable

Inner layers express extracted features



Tolerant error

High Low High

Convolution

Deconvolution

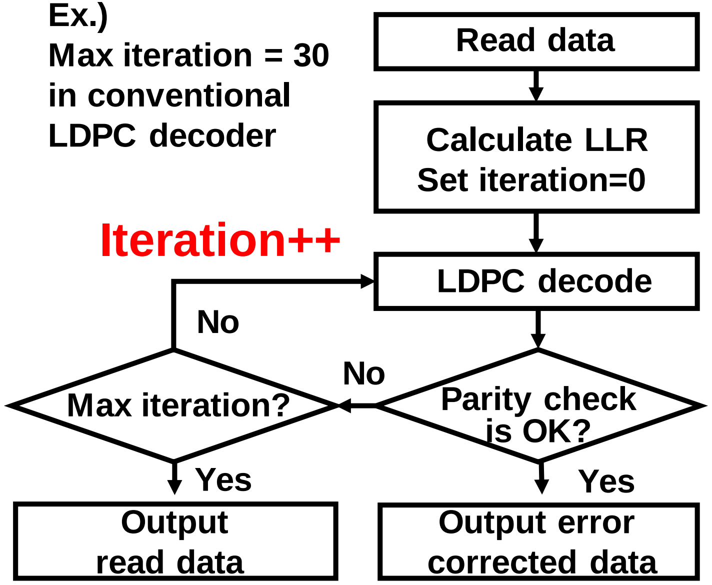
Outline

- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

Low-Density Parity-Check (LDPC) ECC

- LDPC ECC corrects errors gradually by repeating LDPC ECC decoding [4]

Ex.)
Max iteration = 30
in conventional
LDPC decoder



Read data

Calculate LLR
Set iteration=0

Iteration++

LDPC decode

Parity check is OK?

No

Max iteration?

No

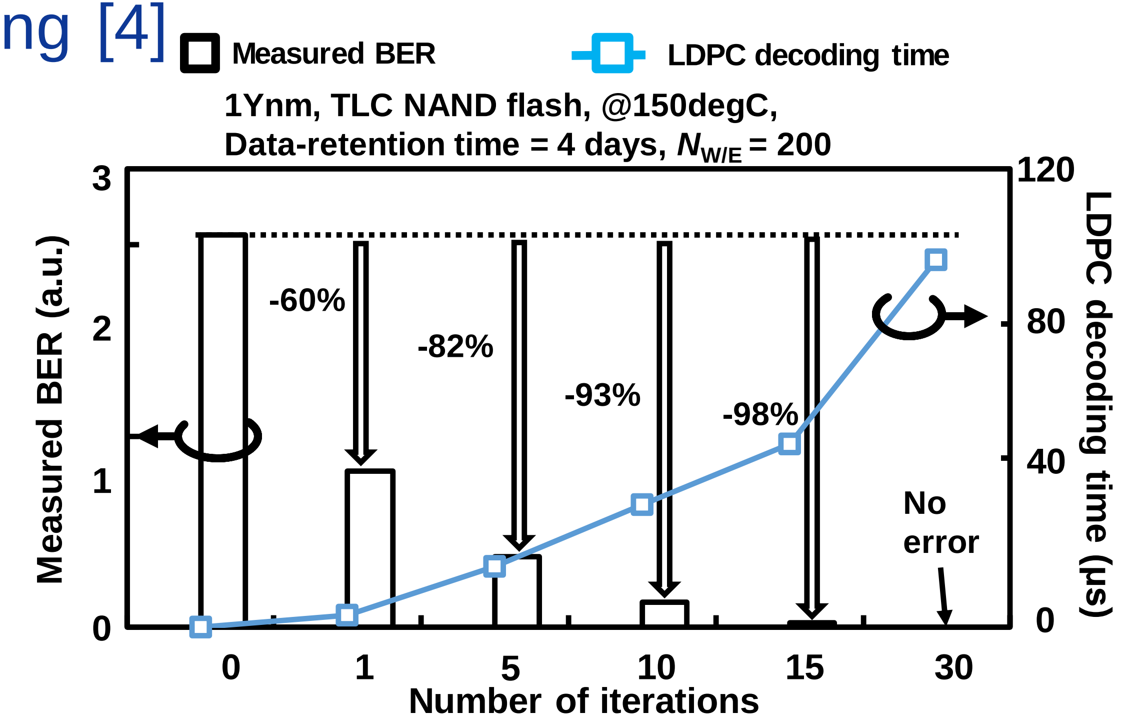
Yes

Output read data

Yes

Output error corrected data

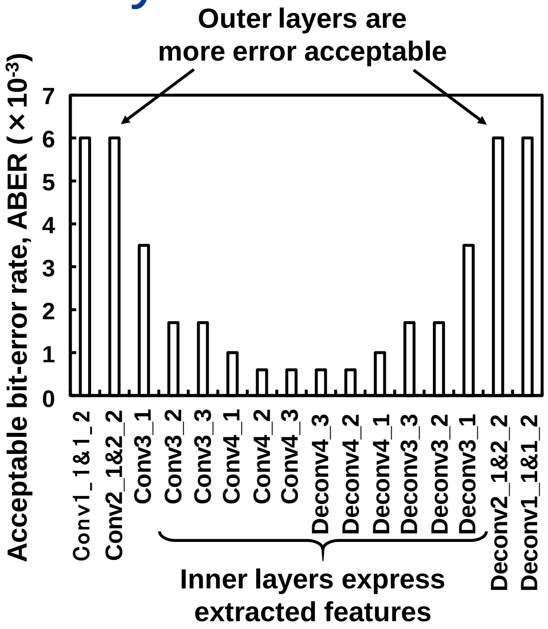
LLR : Log Likelihood Ratio

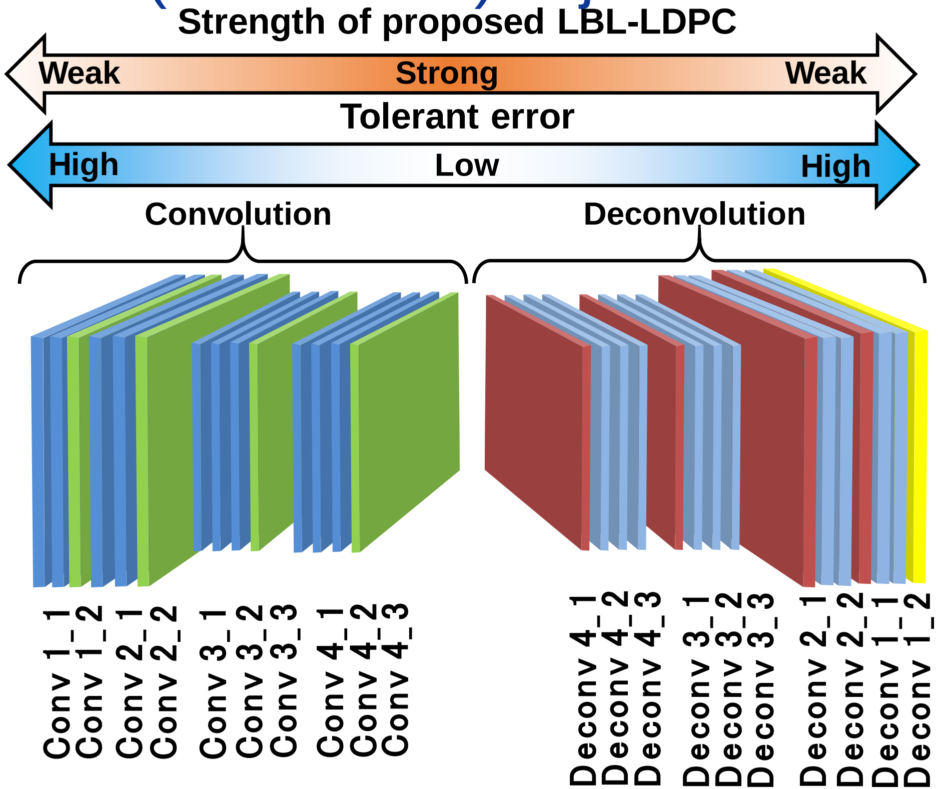


[4] S. Tanakamaru et al., *IRPS*, pp. 3B.3.1-3B.3.6, Apr. 2015.

Concept of Proposed LBL-LDPC

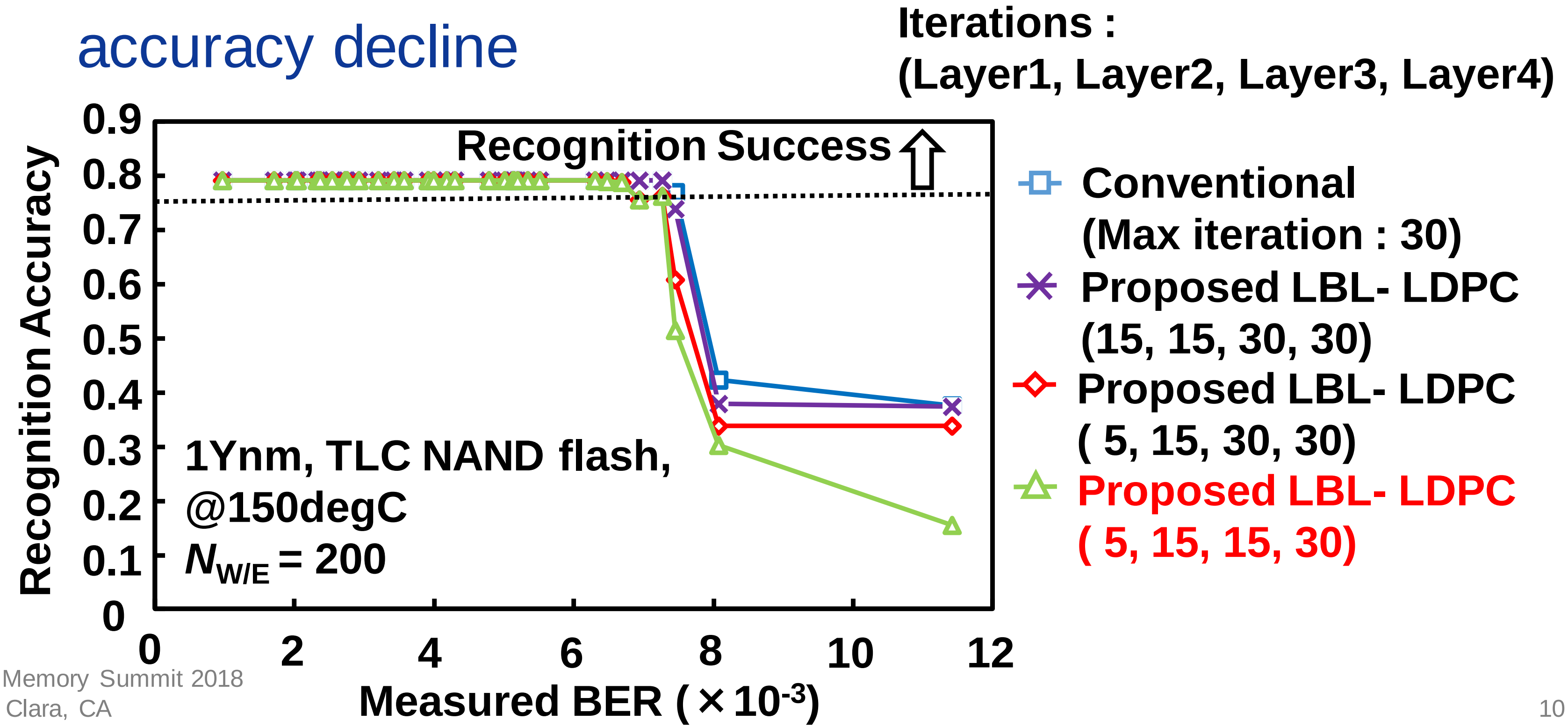
- Layer-by-layer LDPC (LBL-LDPC) adjusts iteration cycle





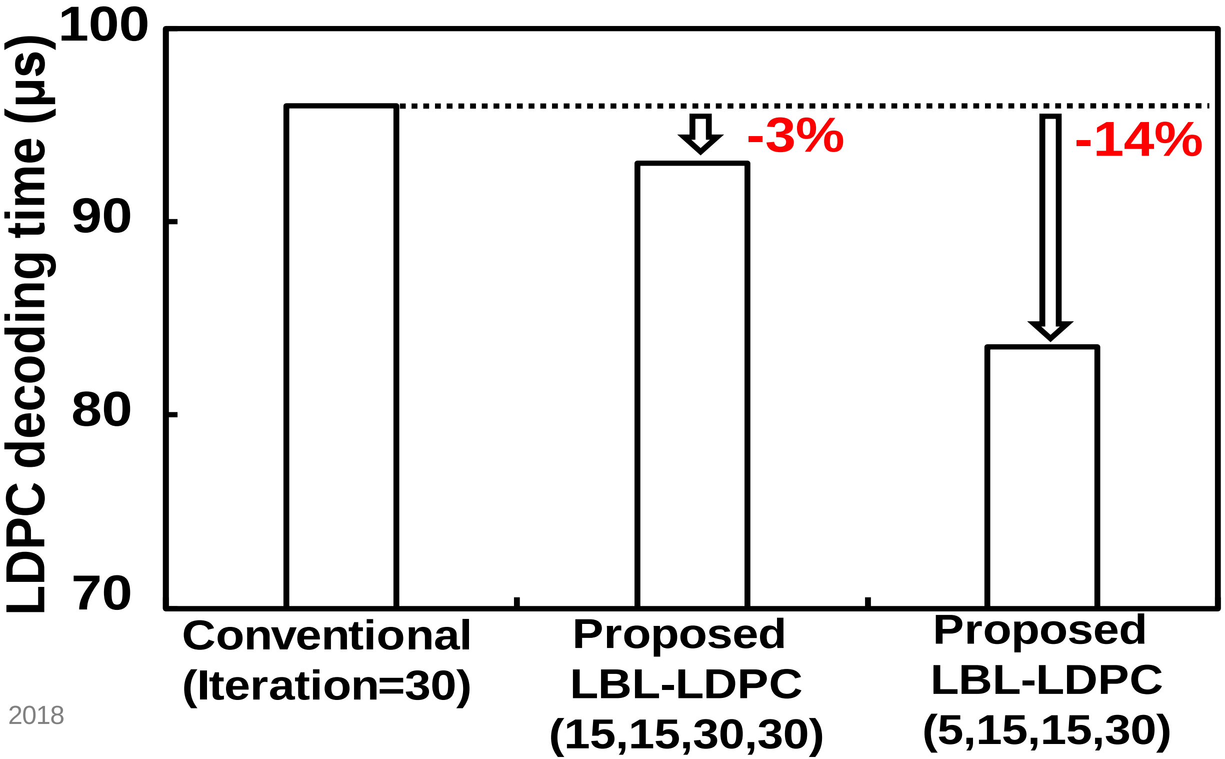
Performance of Proposed LBL-LDPC

- LBL-LDPC decreases iteration count without accuracy decline



Performance of Proposed LBL-LDPC

- LBL-LDPC reduces decoding time by 14%



Result of Proposed LBL-LDPC

- Proposed LBL-LDPC decreases decoding time by 14% compared with conventional LDPC ECC

	Conventional LDPC	Proposed LBL-LDPC (5, 15, 15, 30)
LDPC ECC decoding time	96 µs	83 µs

-14%

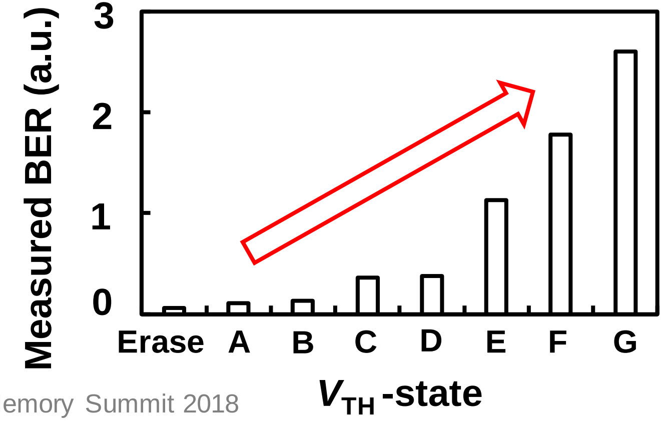
Outline

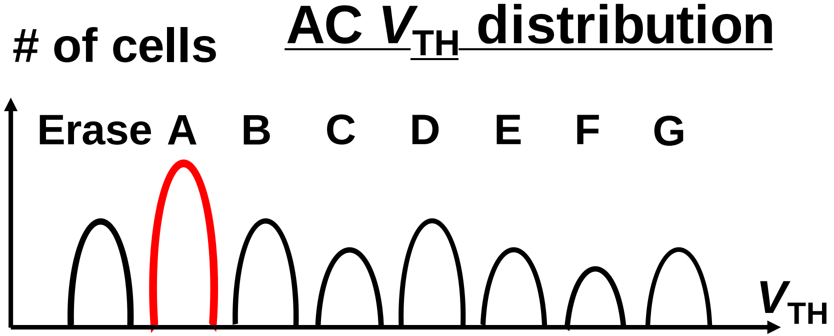
- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

Conventional Asymmetric Coding(AC)

- At high V_{TH} states, many errors occur
- Asymmetric Coding (AC) [5] increases A-state to minimize total errors

1Ynm, TLC NAND flash, @150degC
Data-retention time = 4 days,
Write/Erase cycles ($N_{W/E}$)=200





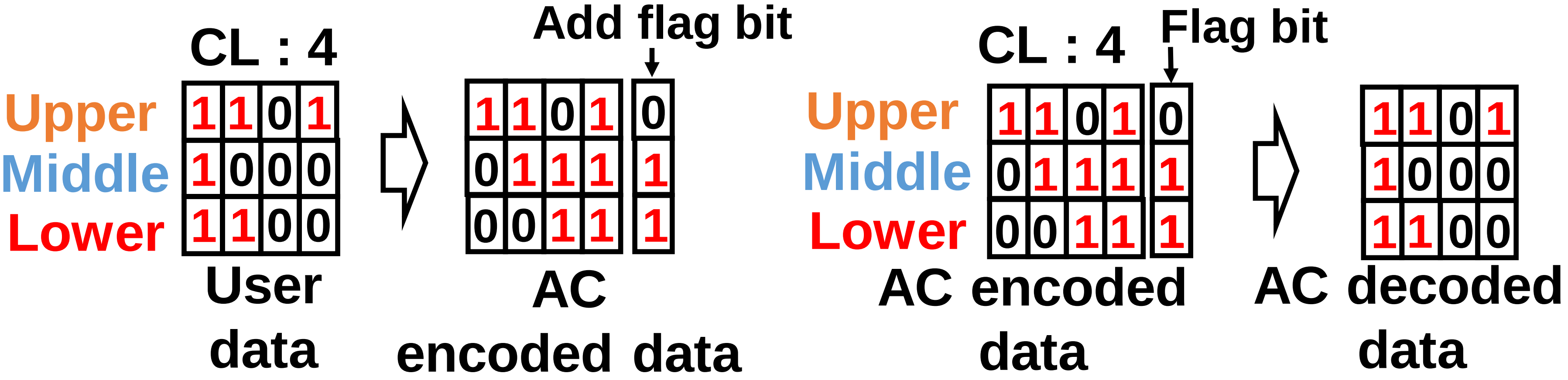
Upper	1	1	1 ↔	0	0	0	0 ↔	1
Middle	1	1 ↔	0	0 ↔	1	1 ↔	0	0
Lower	0 ↔	1	1	1	1 ↔	0	0	0

Increase “A”-state (AC)

[5] S. Tanakamaru et al., *JSSC*, vol. 47, no. 1, pp. 85-96, Jan. 2012.

Conventional Asymmetric Coding(AC)

- Flip all bits if “0” is more than “1” and append “1” as flag



Encode AC

CL : 4

	User data				AC encoded data				Add flag bit
Upper	1	1	0	1	1	1	0	1	0
Middle	1	0	0	0	0	1	1	1	1
Lower	1	1	0	0	0	0	1	1	1

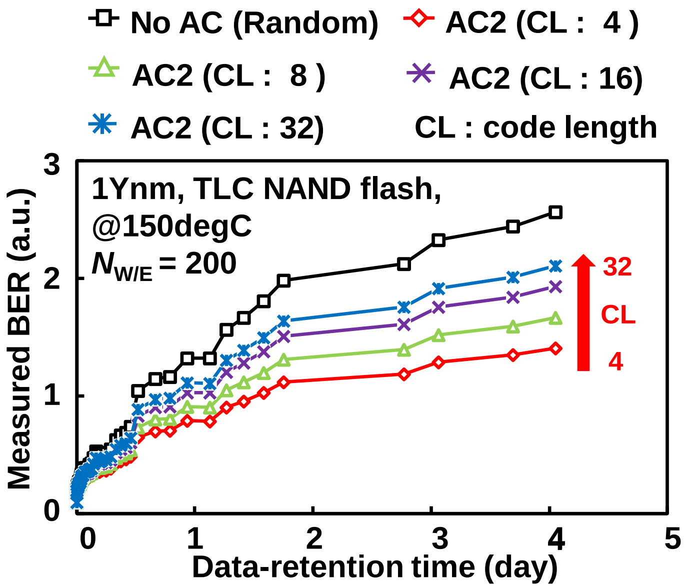
Decode AC

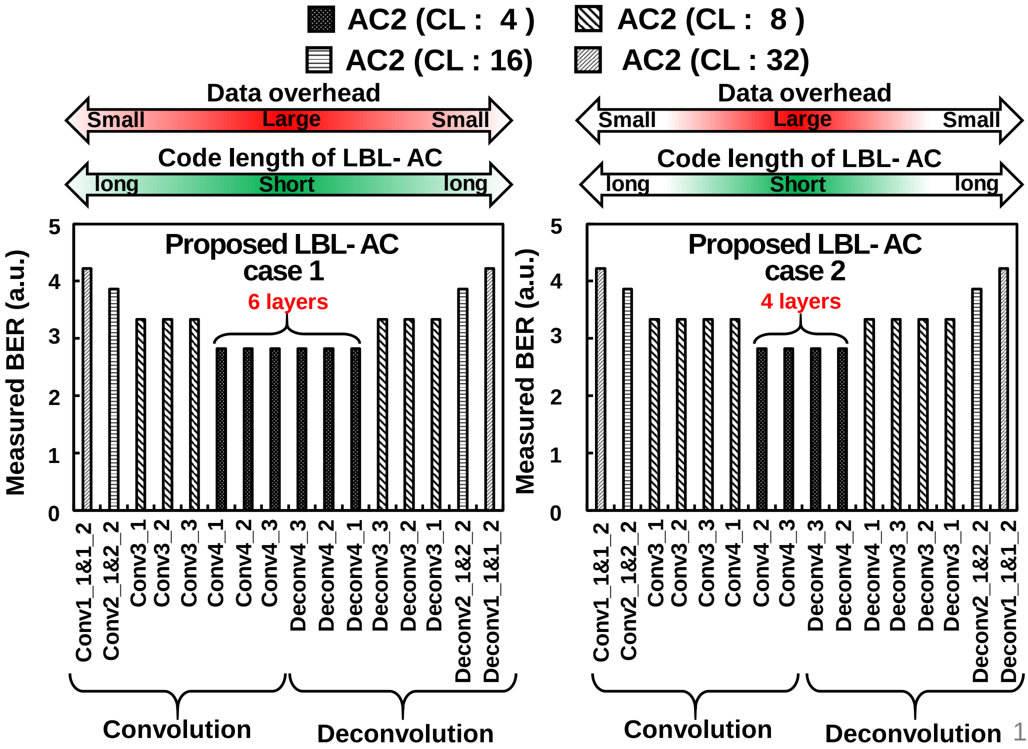
CL : 4

	AC encoded data				Flag bit	AC decoded data			
Upper	1	1	0	1	0	1	1	0	1
Middle	0	1	1	1	1	1	0	0	0
Lower	0	0	1	1	1	1	1	0	0

Concept of Proposed LBL-AC

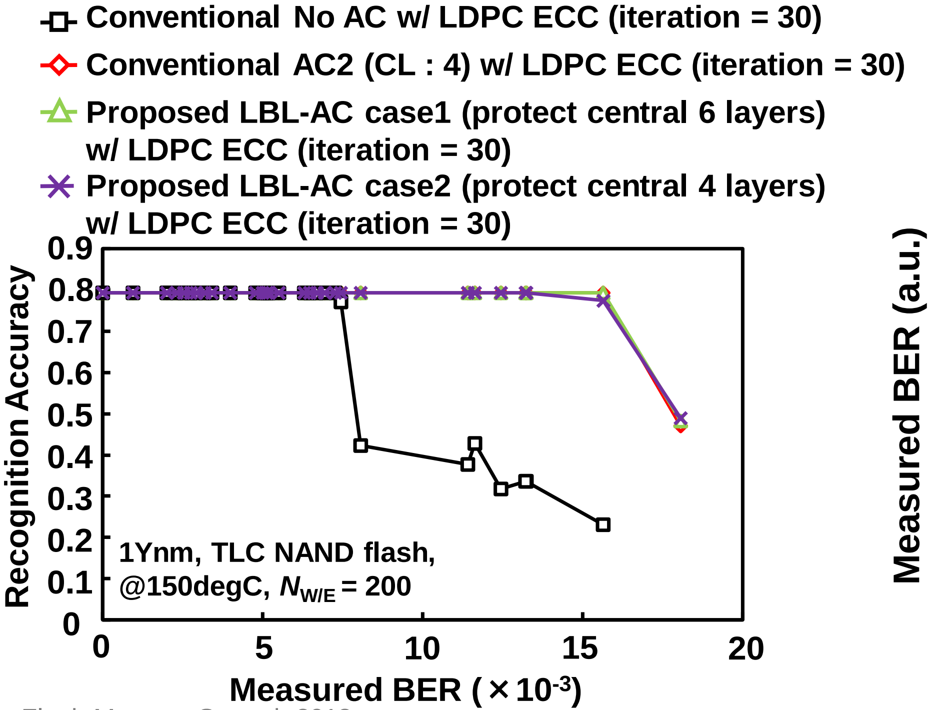
- Proposed Layer-by-layer AC (LBL-AC) adjusts AC code length

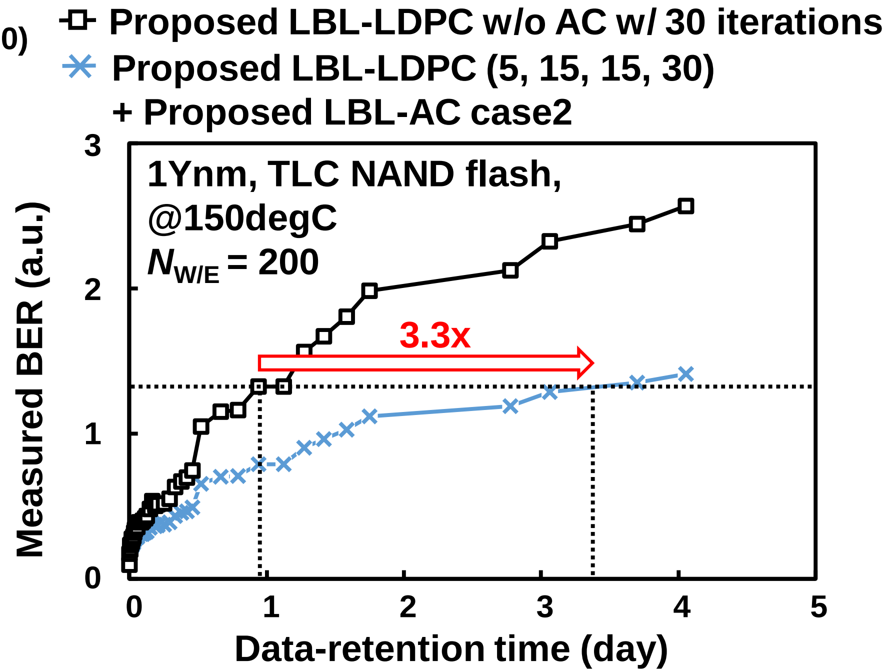




Performance of Proposed LBL-AC

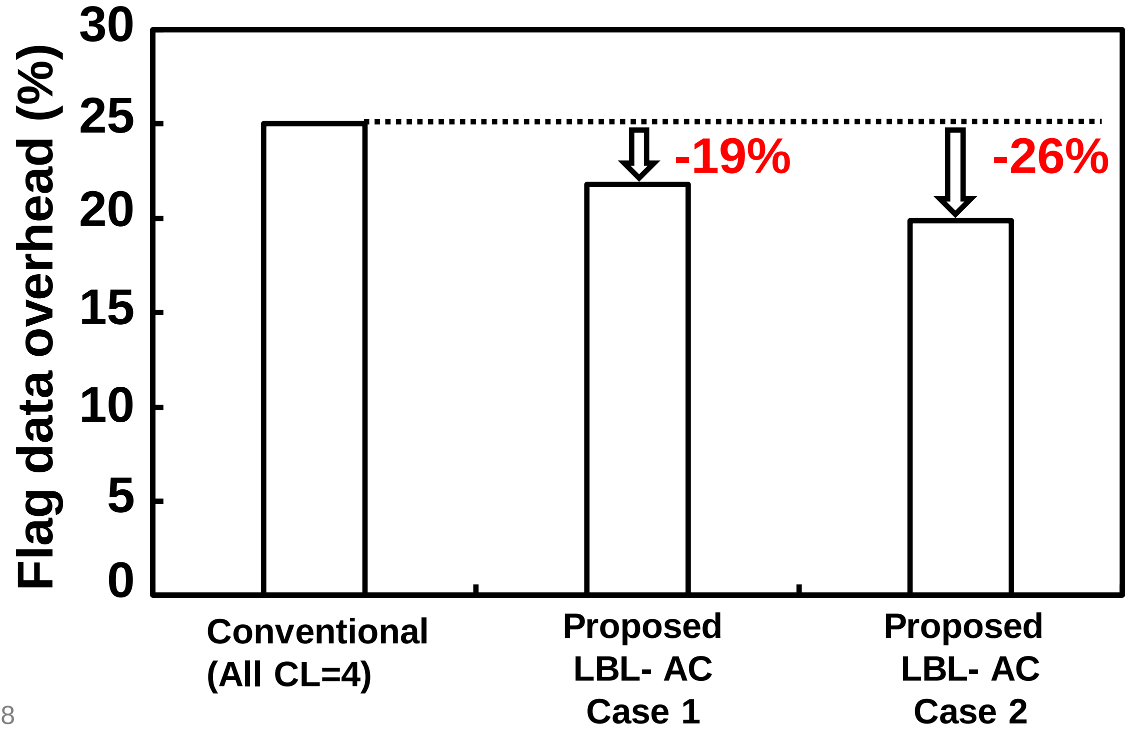
- Acceptable data-retention time increases by 3.3 times





Performance of Proposed LBL-AC

- LBL-AC reduces data-overhead by 26%



Result of Proposed LBL-AC

- Proposed LBL-AC decreases data-overhead by 26% compared with conventional AC

	Conventional LDPC ECC	Conventional AC w/ Conventional LDPC	Proposed LBL-AC w/ Conventional LDPC
LDPC decoding time	96 µs	96 µs	96 µs
Flag data overhead rate	0 %	25 %	19 %
Acceptable data-retention time	1.0 day	3.3 days	3.3 days

-26%

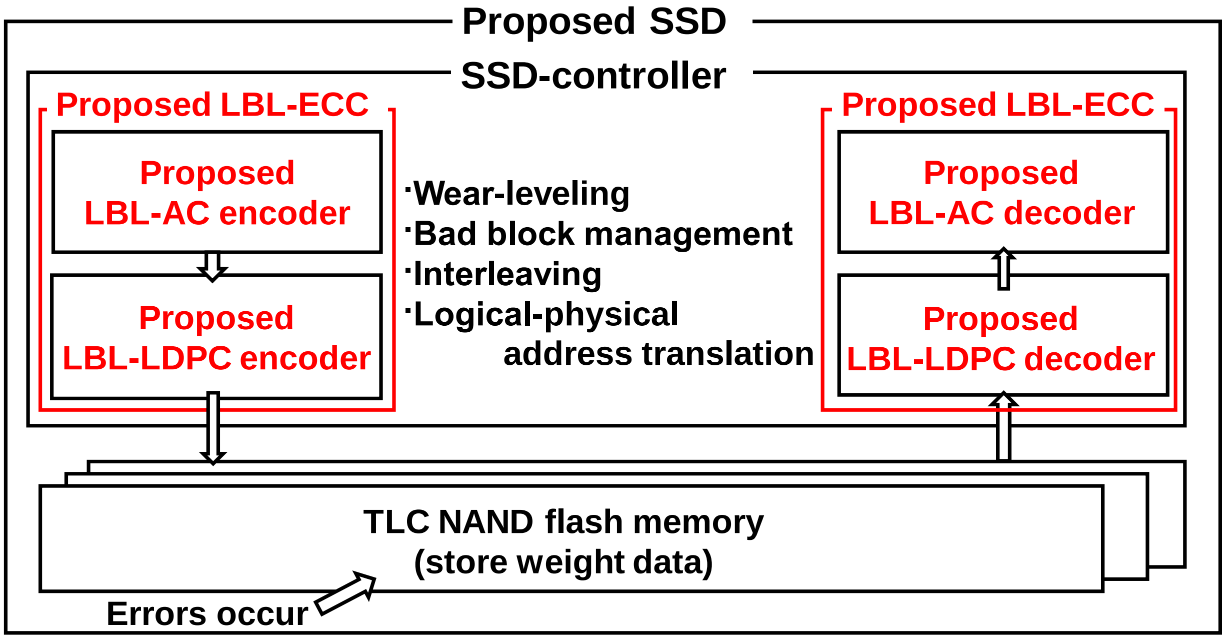
3.3x

Outline

- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

Summary of Proposed LBL-ECC

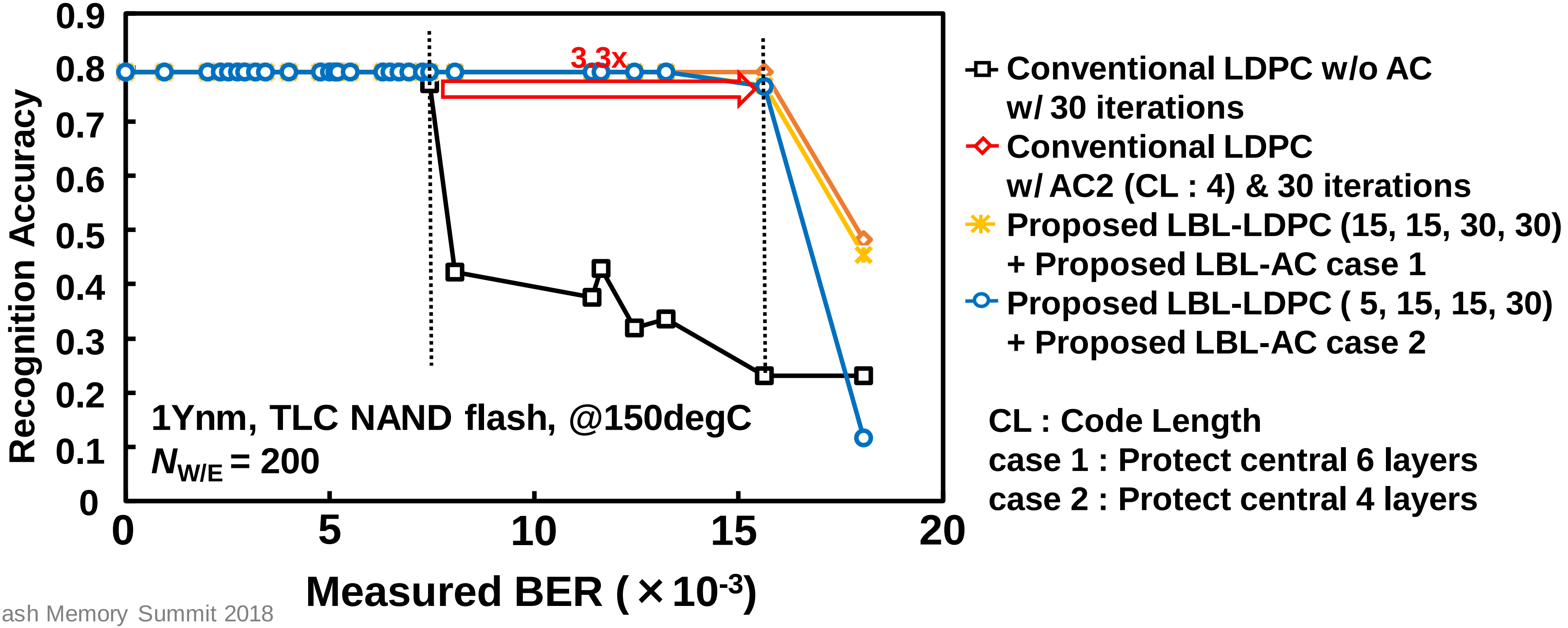
- Summarize proposed LBL-ECC method



Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
Layer-by-layer Iteration-Optimized LDPC (LBL-LDPC)
Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)

Performance of Proposed LBL-ECC

- Acceptable data-retention time increases by 3.3 times



CL : Code Length
case 1 : Protect central 6 layers
case 2 : Protect central 4 layers

Result of Proposed LBL-ECC

- Proposed LBL-AC decreases data-overhead by 26% compared with conventional AC

	Proposed LBL-LDPC	Conventional LBL-AC w/ Conventional LDPC	Proposed LBL-AC w/ Proposed LBL-AC
LDPC decoding time	96 µs	96 µs	83 µs
Flag data overhead rate	0 %	25 %	19 %
Acceptable data-retention time	1.0 day	3.3 days	3.3 days

-14%

-26%

3.3x

Outline

- Introduction
- Proposed Layer-by-layer Iteration-optimized Low-Density Parity-Check Error Correcting Code (LBL-LDPC)
- Proposed Layer-by-layer Code-length Adjusted Asymmetric Coding (LBL-AC)
- Proposed Layer-by-layer Adaptively Optimized Error Correcting Code (LBL-ECC)
- Conclusion

Conclusion

- Proposed LBL-ECC extends data retention time by 3.3 times

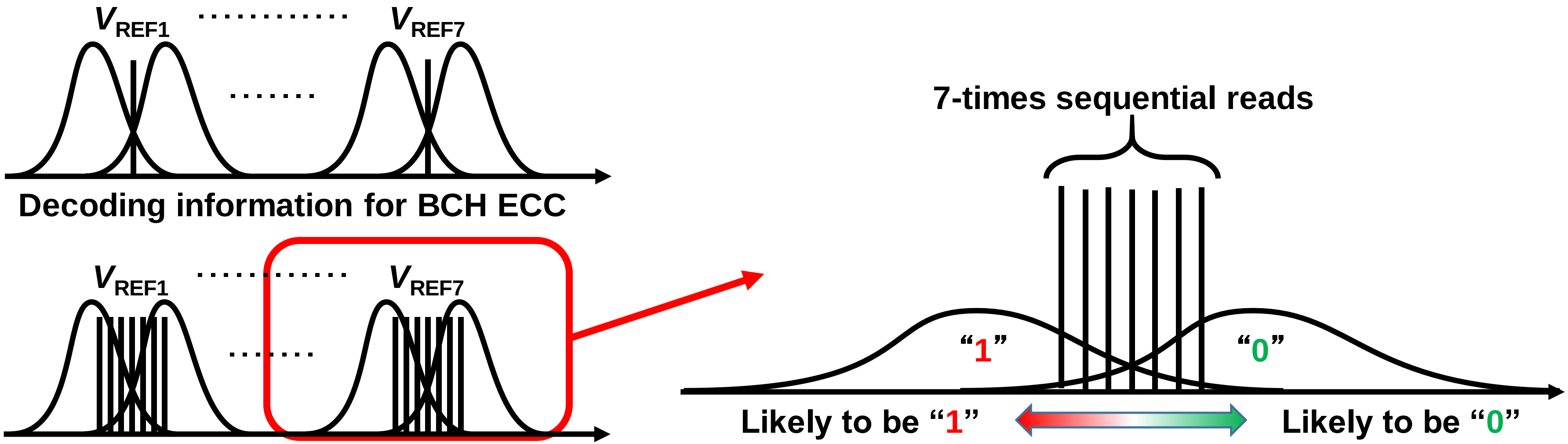
	Conventional LDPC	Proposed LBL- LDPC (5, 15, 15, 30)	Proposed LBL- LDPC (5, 15, 15, 30) + Proposed LBL- AC Case 2
LDPC decoding time	96 µs	83 µs	83 µs
Flag data overhead rate	0	0	19%
Acceptable data-retention time	1.0 day	1.0 day	3.3 days

Thank you for your attention

This work was supported by JST CREST Grant Number JPMJCR1532, Japan

Soft-decoding LDPC

- Log-likelihood ratio (LLR) is required for LDPC decoding.



Decoding information for BCH ECC

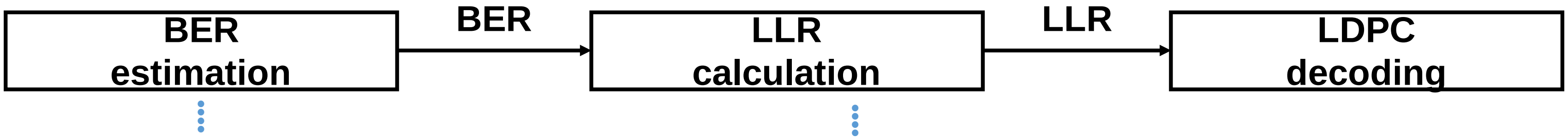
Decoding information for Soft-decoding LDPC ECC (LLR)

Likely to be “1” ⟷ Likely to be “0”

[6] C. Kim et al., *Symp. VLSI Circ.*, pp. 196-197, 2011.

Error Prediction (EP-) LDPC

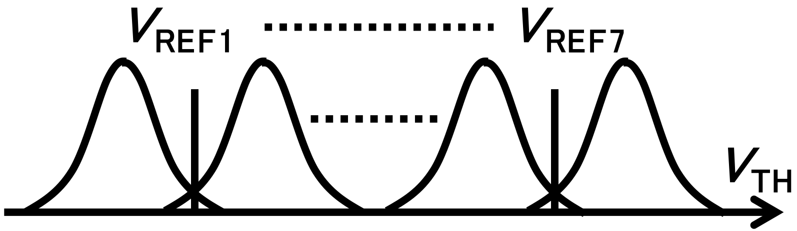
- EP-LDPC is 7-times faster reads than Soft-decoding LDPC



BER is estimated from some factors of error

- V_{TH} information(×7)
- Inter-cell coupling info.
- Write/Erase cycles ($N_{W/E}$)
- Retention time

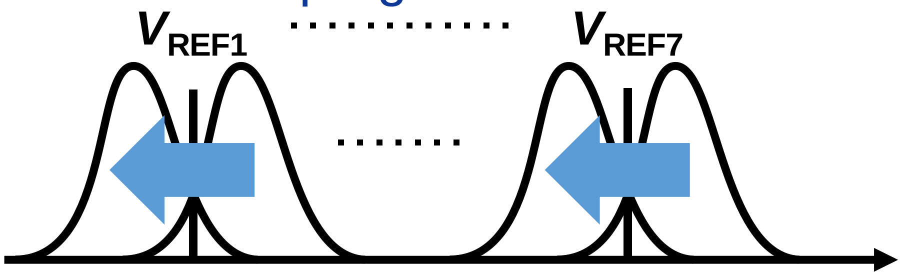
$$LLR(1) = \ln\{BER / (1\text{-}BER)\}$$
$$LLR(0) = \ln\{(1\text{-}BER) / BER\}$$



[7] S. Tanakamaru et al., *JSSCC,* pp. 2920-2934, 2013.

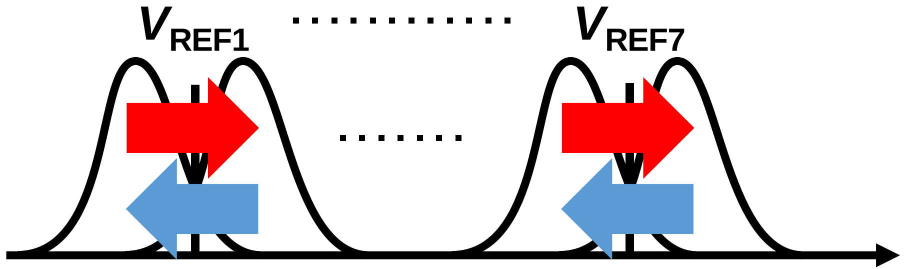
Advanced Error Prediction (AEP-) LDPC

- TLC NAND Flash memory is sensitive to program disturb errors
- AEP-LDPC can correct more accurate and efficiency by considering with program disturb



EP-LDPC[7]

Considering with only data-retention error



AEP-LDPC[8]

Considering with program disturb and data-retention error

[7] S. Tanakamaru et al., *JSSCC*, pp. 2920-2934, 2013.
[8] T. Tokutomi et al., *IEEE IMW*, pp. 99-102, 2014.