Error Detection & Vertical LDPC ECC for Reliable 3D-TLC NAND Flash

Shun Suzuki, Toshiki Nakamura,
Kyoji Mizoguchi and Ken Takeuchi
Chuo University, Japan

Outline

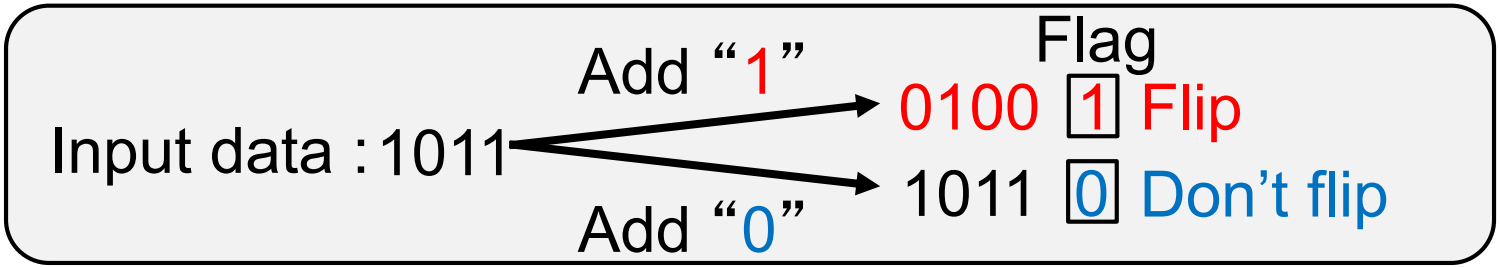
- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

Outline

- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

Conventional Asymmetric Coding (AC)

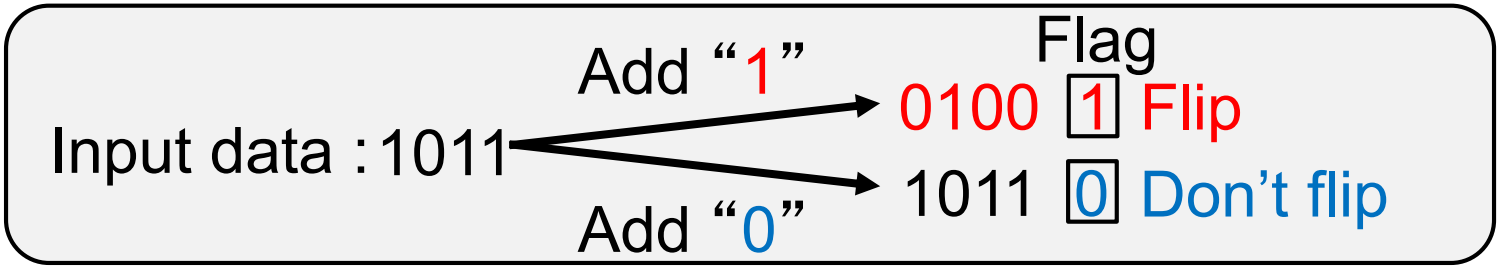
Conventional Asymmetric Coding (AC) [1]



[1] S. Tanakamaru *et al.*, *ISSCC*, 2011, pp. 204-206.

Conventional Asymmetric Coding (AC)

Conventional Asymmetric Coding (AC) [1]



e.g.) Code Length (CL) = 4

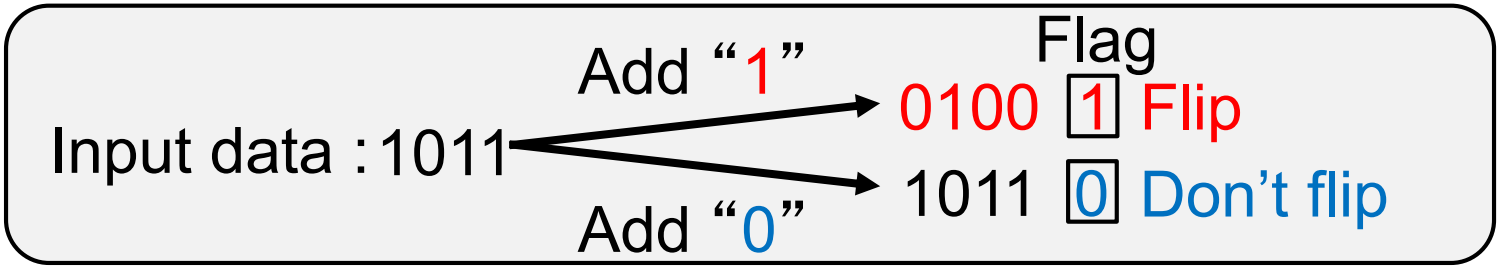
Input data : 101100010011

# of "1"s	# of "0"s
6/12	6/12

[1] S. Tanakamaru *et al.*, *ISSCC*, 2011, pp. 204-206.

Conventional Asymmetric Coding (AC)

Conventional Asymmetric Coding (AC) [1]



e.g.) Code Length (CL) = 4

Input data : 101100010011

Add a flag for increasing "1"

1011 0 1110 1 1100 1

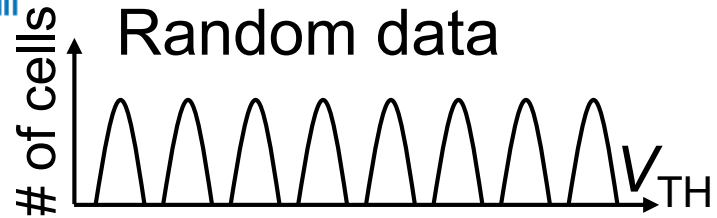
# of "1"s	# of "0"s
6/12	6/12

# of "1"s	# of "0"s
10/15 >	5/15

Proportion of "0" and "1"is changed by AC

[1] S. Tanakamaru *et al.*, *ISSCC*, 2011, pp. 204-206.

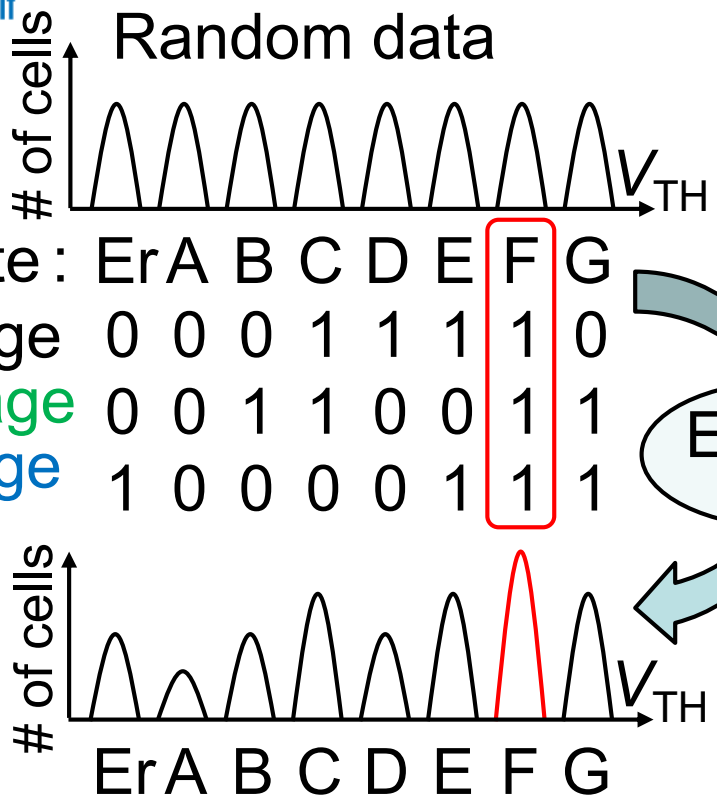
Conventional Asymmetric Coding (AC)



V_{TH}-state :	Er	A	B	C	D	E	F	G
Upper page	0	0	0	1	1	1	1	0
Middle page	0	0	1	1	0	0	1	1
Lower page	1	0	0	0	0	1	1	1

(Er : Erase)

Conventional Asymmetric Coding (AC)

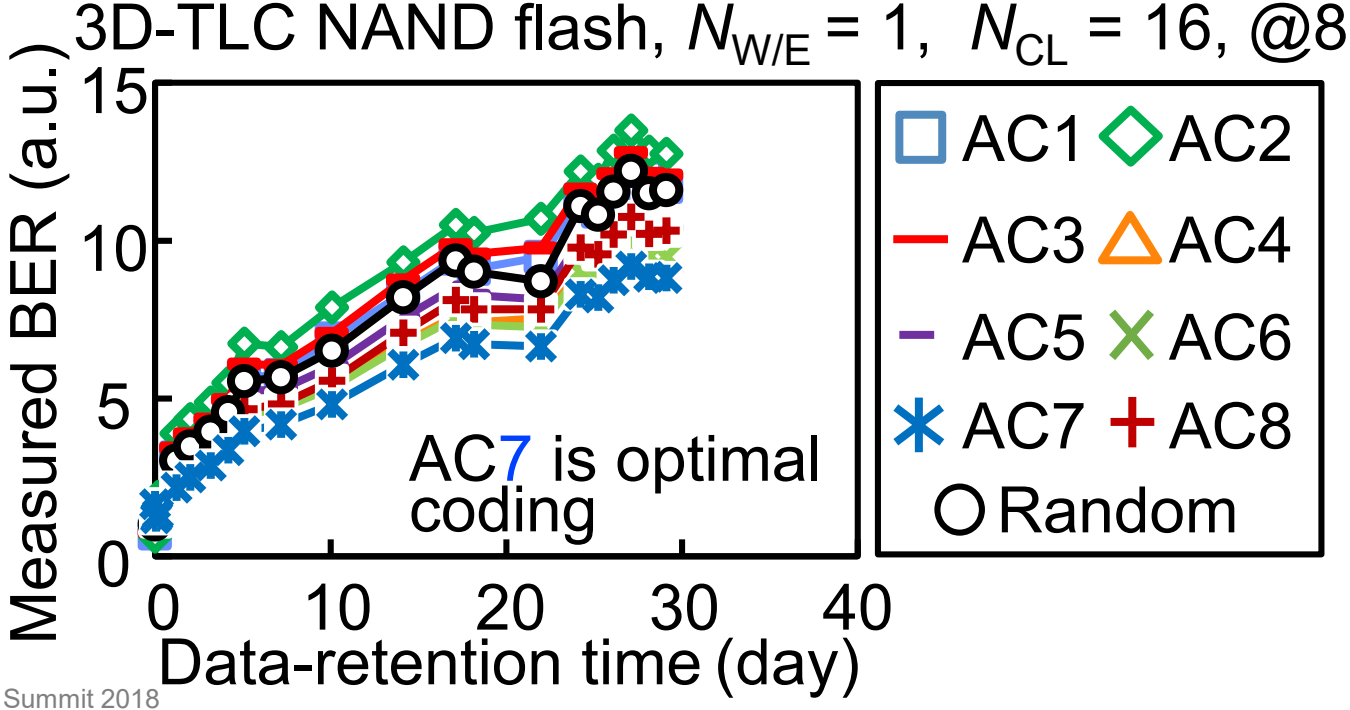


Page	Increased data
Upper page	“1”
Middle page	“1”
Lower page	“1”

Increase Er-state cells : AC1
Increase A-state cells : AC2
⋮ ⋮
Increase G-state cells : AC8

Conventional Asymmetric Coding (AC)

- AC7 decreases BER mostly in 3D-TLC NAND flash

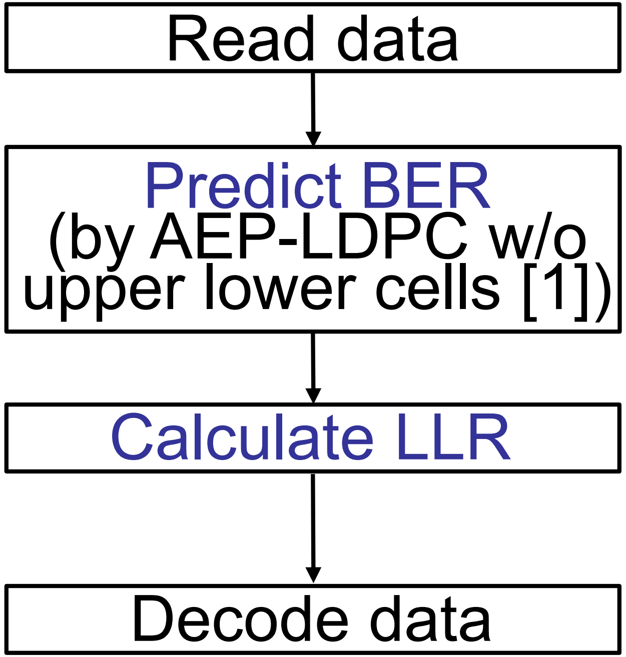


Outline

- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

Low-Density Parity-Check (LDPC) Code

Decoding flowchart

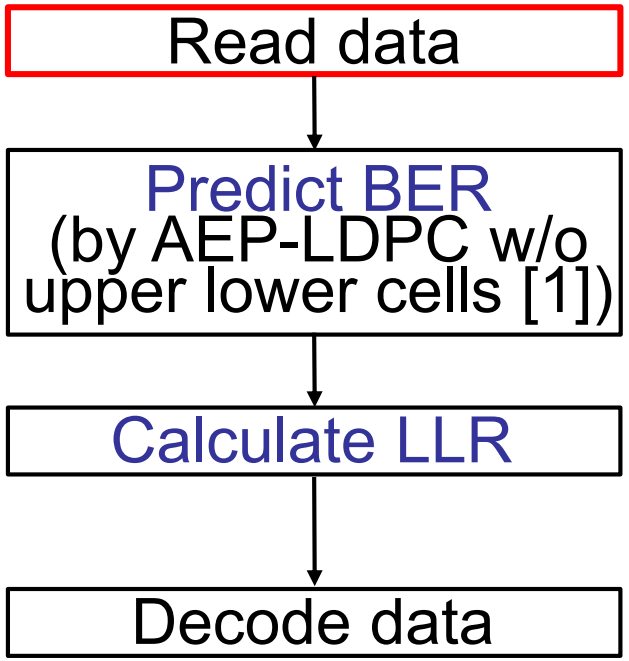


[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

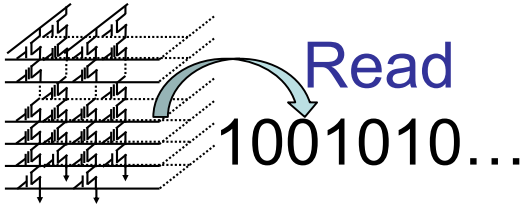
Low-Density Parity-Check (LDPC) Code

Flash Memory Summit

Decoding flowchart



3D NAND flash



[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

Low-Density Parity-Check (LDPC) Code

Flash Memory Summit

Decoding flowchart

Read data

↓

Predict BER (by AEP-LDPC w/o upper lower cells [1])

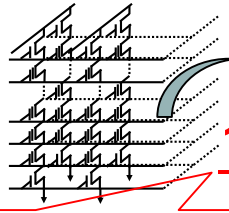
↓

Calculate LLR

↓

Decode data

3D NAND flash



Read

1001010…

Predicted BER is 2.0% based on BER table (shown in next slide)

[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

Low-Density Parity-Check (LDPC) Code

Flash Memory Summit

Decoding flowchart

Read data

↓

Predict BER (by AEP-LDPC w/o upper lower cells [1])

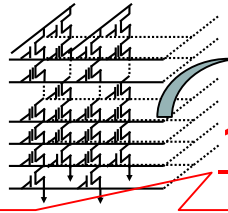
↓

Calculate LLR

↓

Decode data

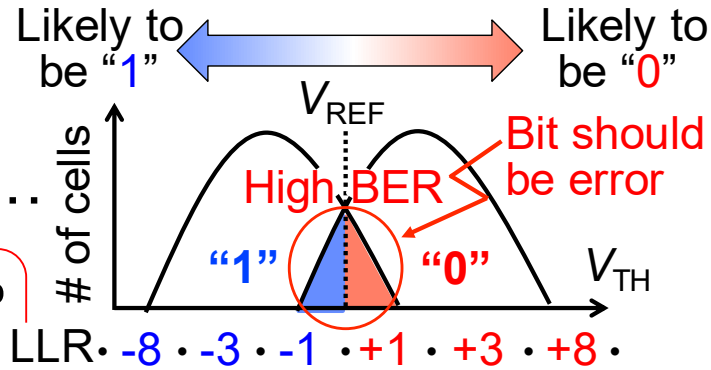
3D NAND flash



Read

1001010…

Predicted BER is 2.0% based on BER table (shown in next slide)



Predicted BER

$$LLR(0) = \ln\left(\frac{1-BER}{BER}\right),\ LLR(1) = \ln\left(\frac{BER}{1-BER}\right)$$

(LLR : Log-likelihood ratio)

[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

Low-Density Parity-Check (LDPC) Code

Flash Memory Summit

Decoding flowchart

Read data

↓

Predict BER (by AEP-LDPC w/o upper lower cells [1])

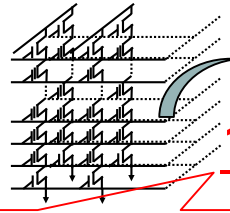
↓

Calculate LLR

↓

Decode data

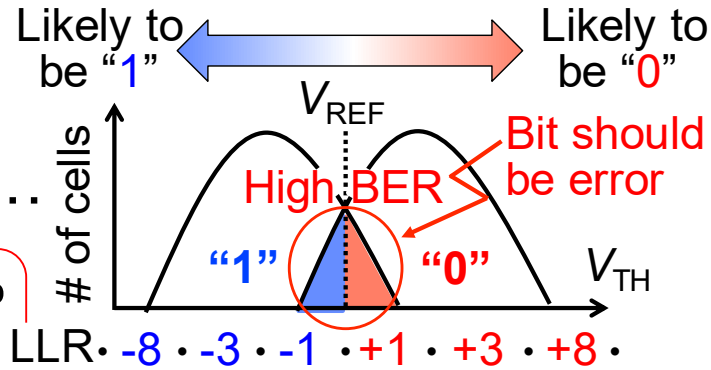
3D NAND flash



Read

1001010…

Predicted BER is 2.0% based on BER table (shown in next slide)



Predicted BER

$$LLR(0) = \ln\left(\frac{1-BER}{BER}\right),\ LLR(1) = \ln\left(\frac{BER}{1-BER}\right)$$

(LLR : Log-likelihood ratio)

→ LDPC corrects errors by using LLR

[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

AEP-LDPC w/o upper lower cells

- BER prediction method “AEP-LDPC w/o upper lower cells [1]”

BER table

Page \ V_{TH}-state	Erase page	A state	…	G
Upper	BER_{UEr}	BER_{UA}	…	BER_{UG}
Middle	BER_{MEr}	BER_{MA}	…	BER_{MG}
Lower	BER_{LEr}	BER_{LA}	…	BER_{LG}

[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

AEP-LDPC w/o upper lower cells

- BER prediction method “AEP-LDPC w/o upper lower cells [1]”

e.g.)

Read data 0011

Which page?

Lower page 0011

BER table

Page \ V_{TH}-state	Erase (page)	A (state)	…	G
Upper	BER_{UEr}	BER_{UA}	…	BER_{UG}
Middle	BER_{MEr}	BER_{MA}	…	BER_{MG}
Lower	BER_{LEr}	BER_{LA}	…	BER_{LG}

Which state?

1 0 1 0
1 0 0 1
0 0 1 1
C A E G

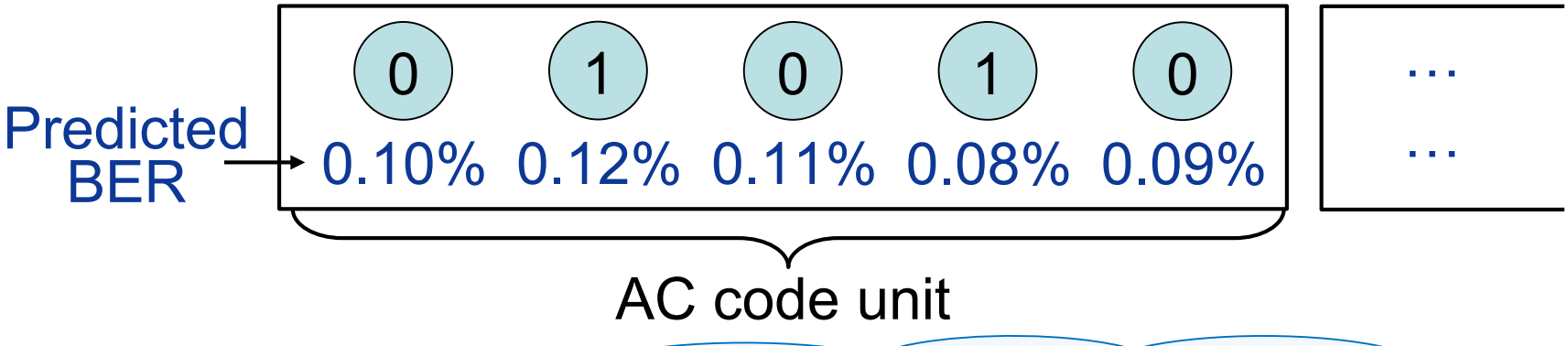
0 0 1 1

Predicted BER is BER_{LA}

[1] T. Tokutomi *et al.*, *IMW*, 2014, pp. 99-102.

Concept of this work

- Conventional AEP-LDPC w/o upper lower cells can not correct errors because predicted BER is too small



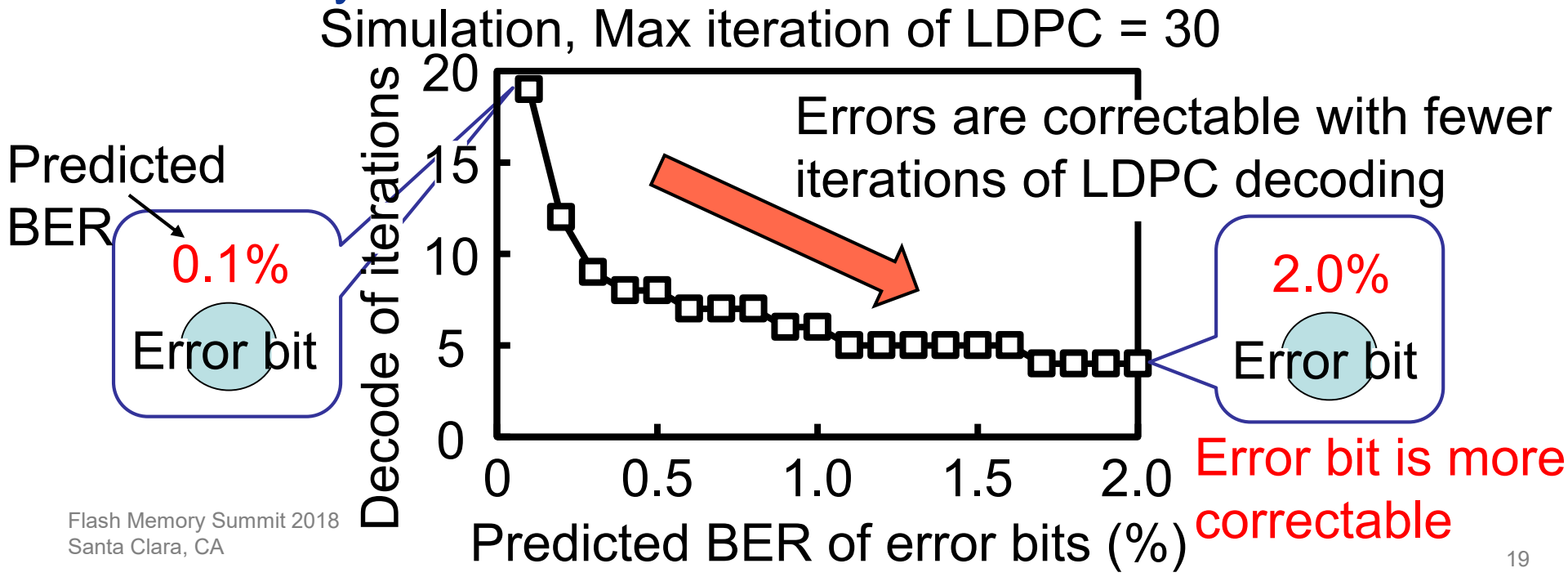
Conv. LDPC : Error include in these bits

Which bit is error

Error information is poor

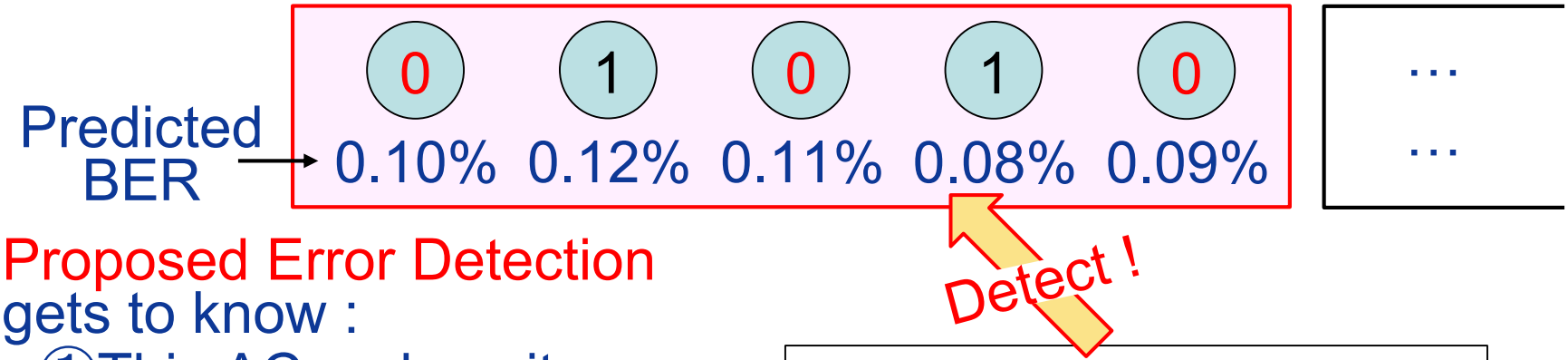
Concept of this work

- If error bits have large predicted BER, LDPC corrects errors effectively



Concept of this work

- In our work, proposed Error Detection detects errors based on AC algorithm



Proposed Error Detection gets to know :

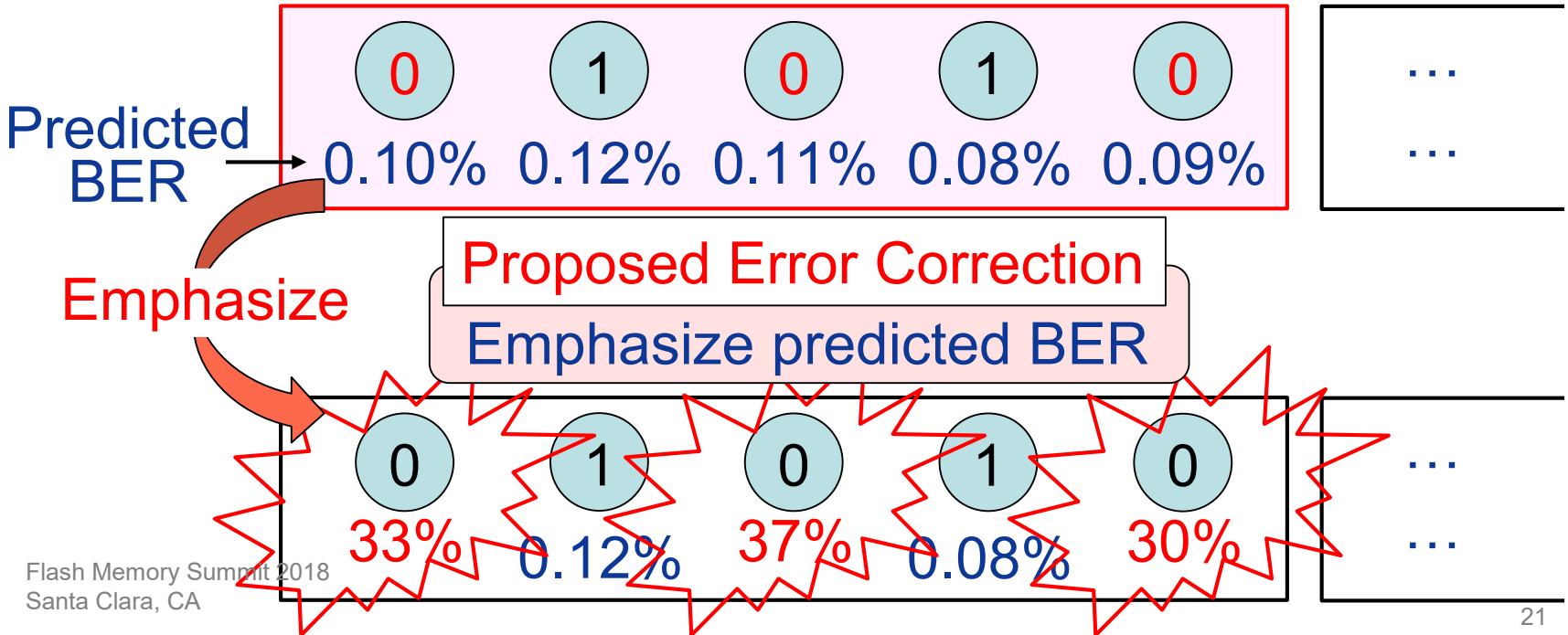
①This AC code unit includes error bit

②Error exists in “0”s of this unit

Detect that at least one error exists in “0”s

Concept of this work

- Our proposed Error Correction intentionally increases predicted BER of detected bits → LDPC ECC effectively corrects errors



Outline

- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

Concept of Proposed Error Detection

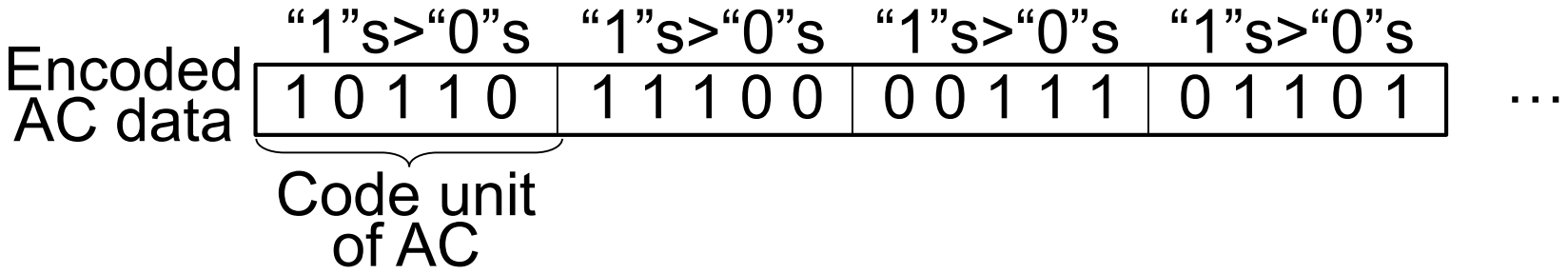
- Proposed Error Detection

①Horizontal Error detection (HED)
Proposed HED detects AC code unit including error bit
②Vertical-LDPC (V-LDPC)
Proposed V-LDPC averages detected errors among decode units of LDPC

Error detection information is used for proposed Error Correction

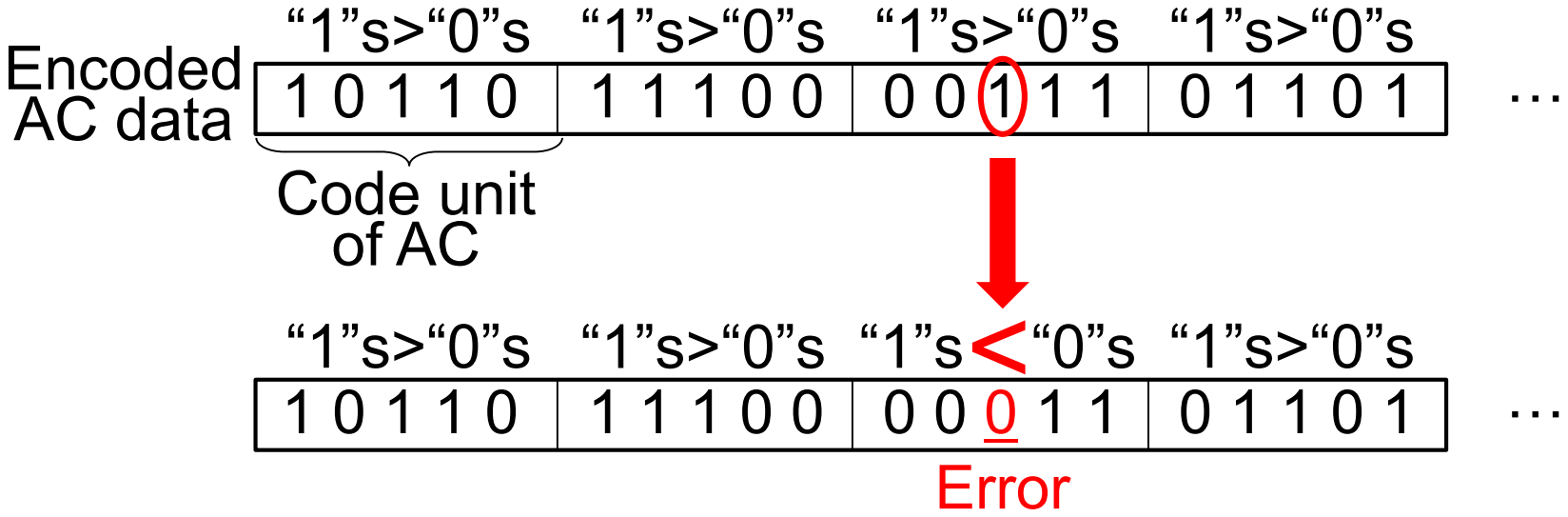
Proposed Error Detection
①Horizontal Error Detection (HED)

- HED detects AC code unit including error bit



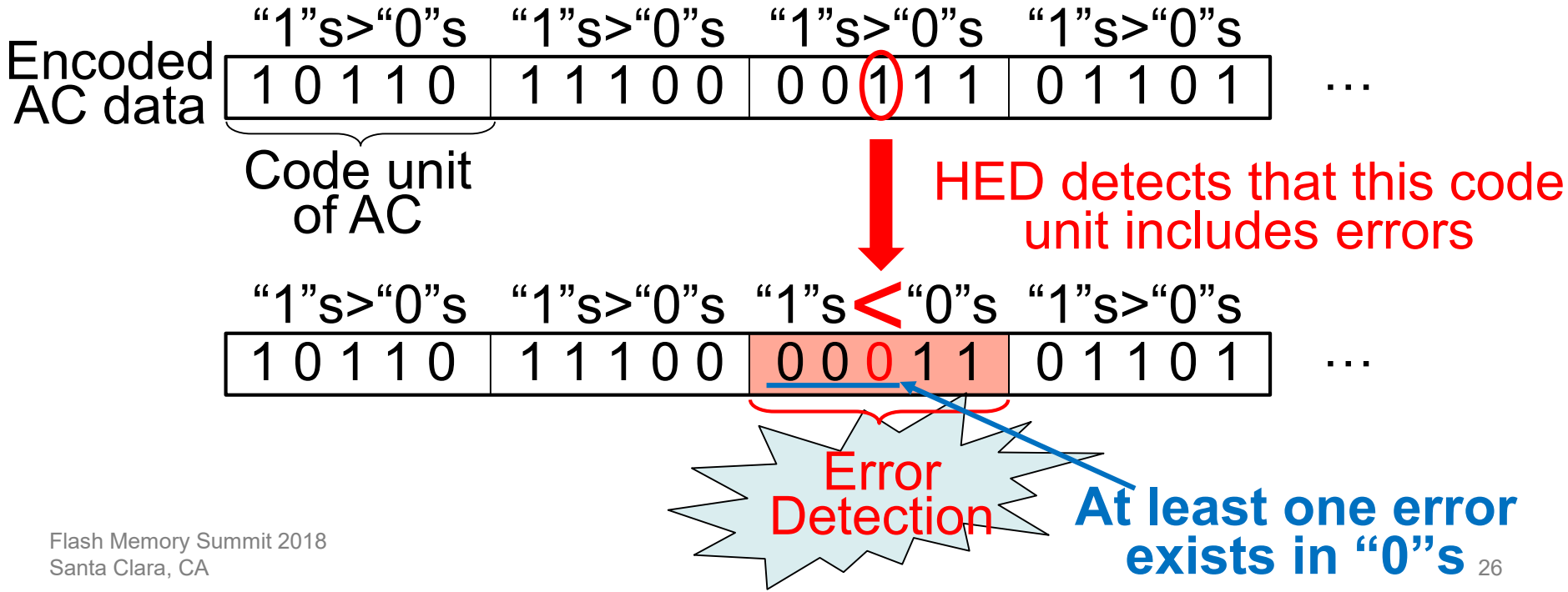
Proposed Error Detection
①Horizontal Error Detection (HED)

- HED detects AC code unit including error bit



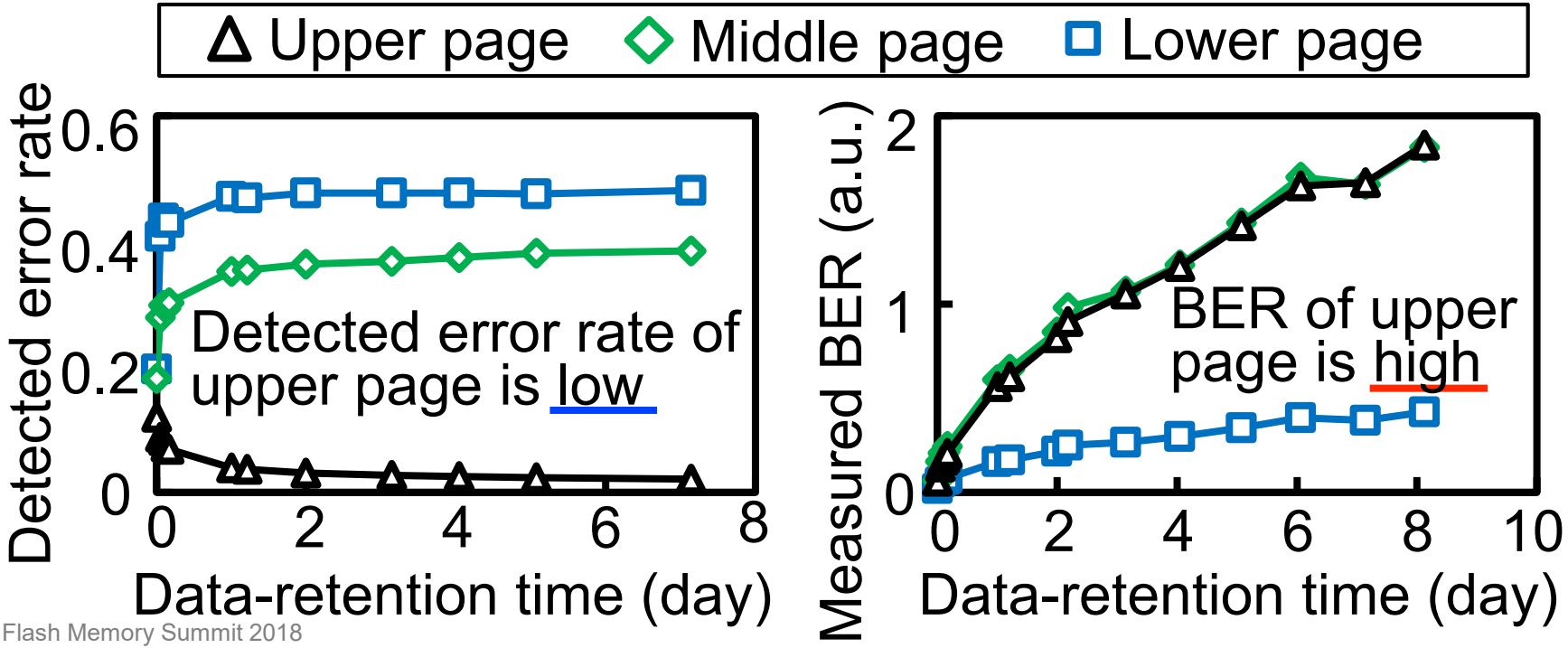
Proposed Error Detection
①Horizontal Error Detection (HED)

- HED detects AC code unit including error bit



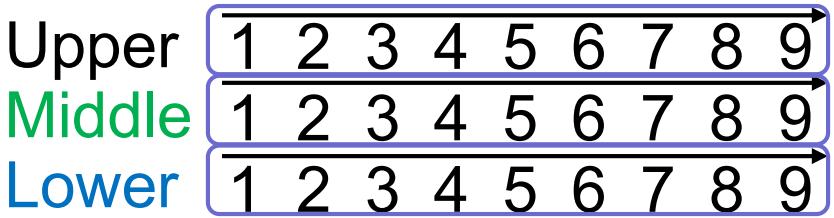
Proposed Error Detection ~Result of HED~

3D-TLC NAND flash, N_{CL} = 8, $N_{W/E}$ = 400, AC7, @85degC

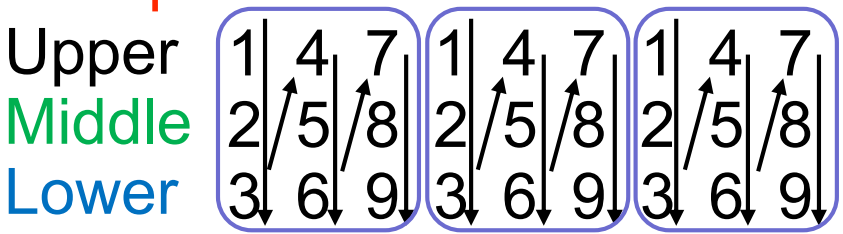


Proposed Error Detection ②Vertical-LDPC (V-LDPC)

Conventional LDPC ECC order



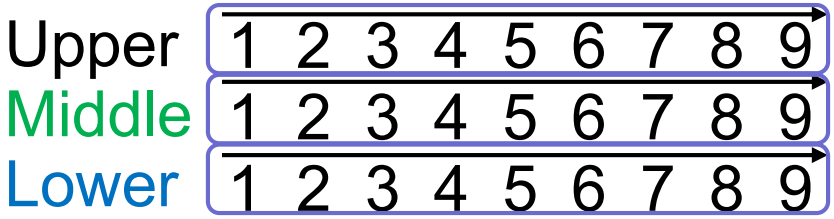
Proposed V-LDPC order



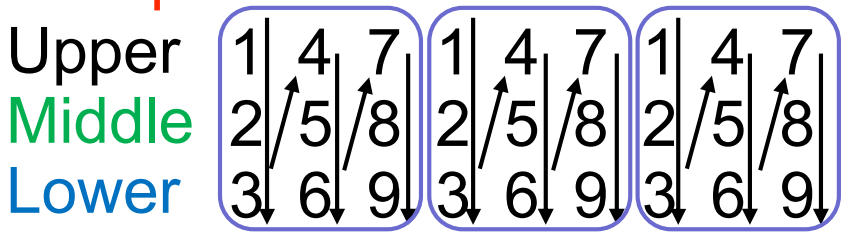
: Encode/decode unit

Proposed Error Detection ②Vertical-LDPC (V-LDPC)

Conventional LDPC ECC order

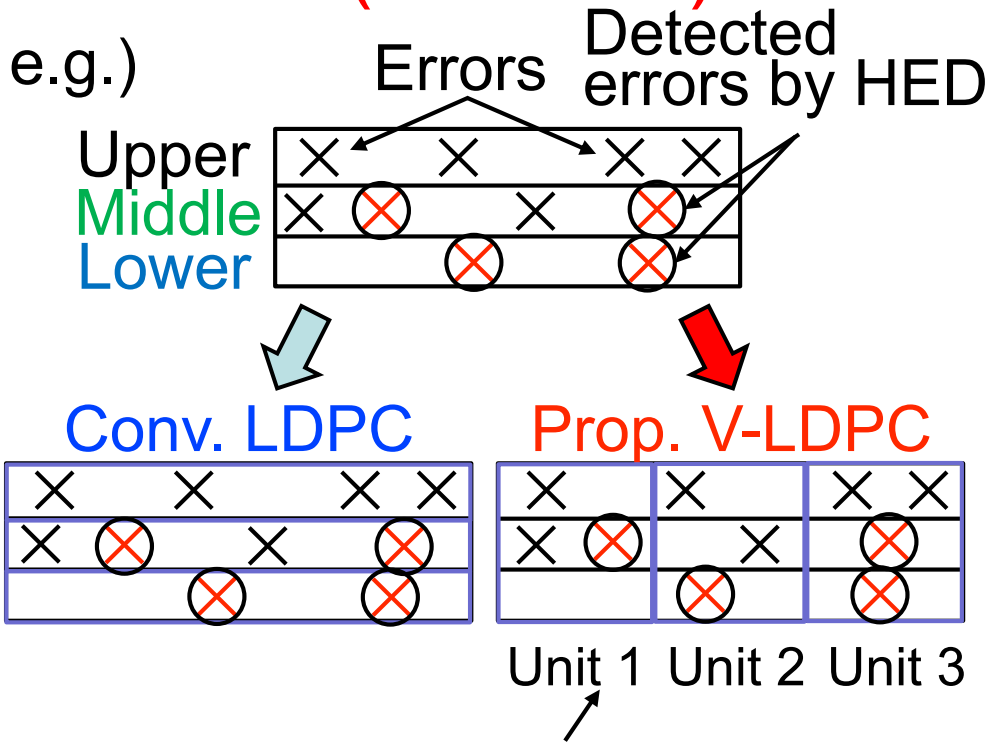


Proposed V-LDPC order



: Encode/decode unit

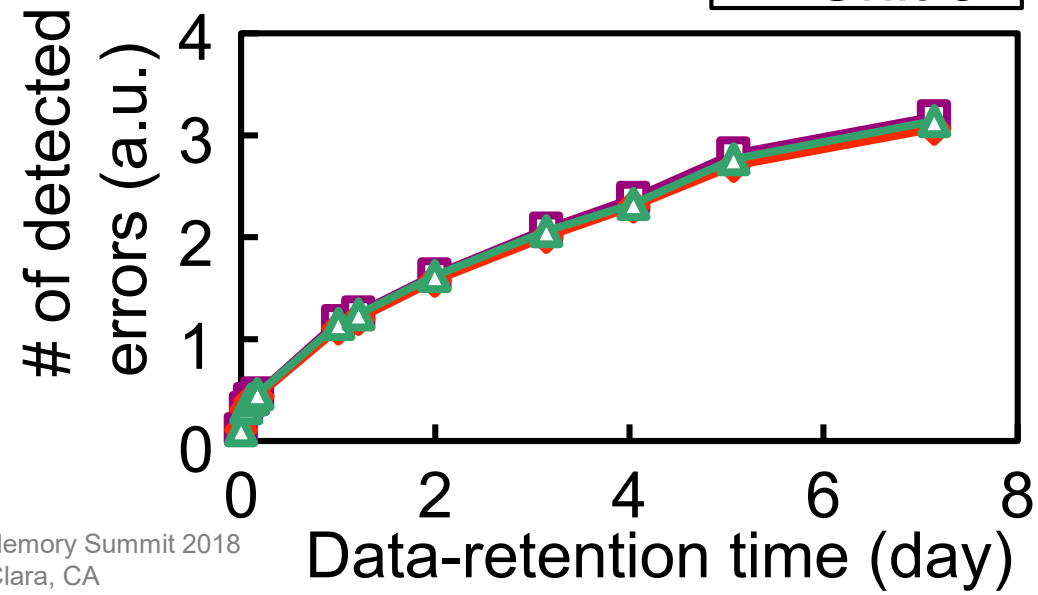
e.g.)

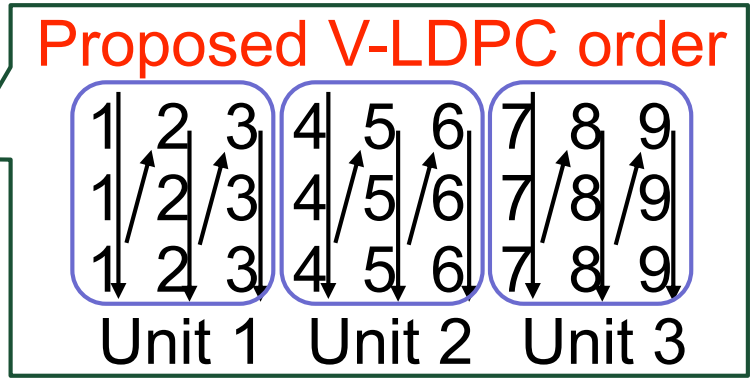


Detected errors are dispersed among decode units

Proposed Error Detection ~Result of V-LDPC~

Proposed V-LDPC,
$N_{W/E}$ = 1, N_{CL} = 8, @85degC





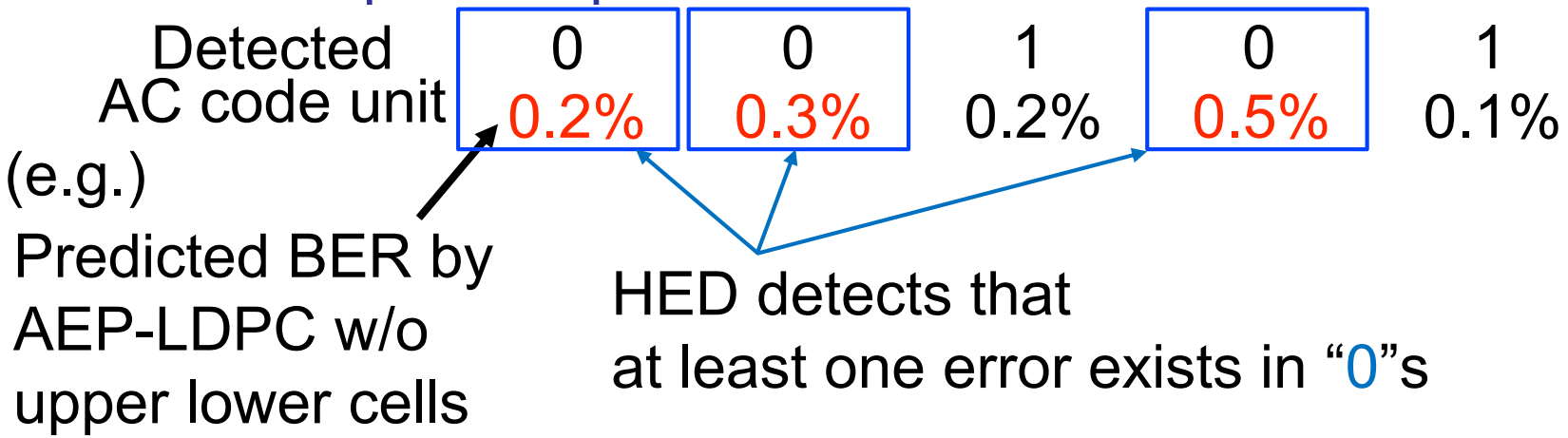
- All units have almost same number of detected errors

Outline

- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

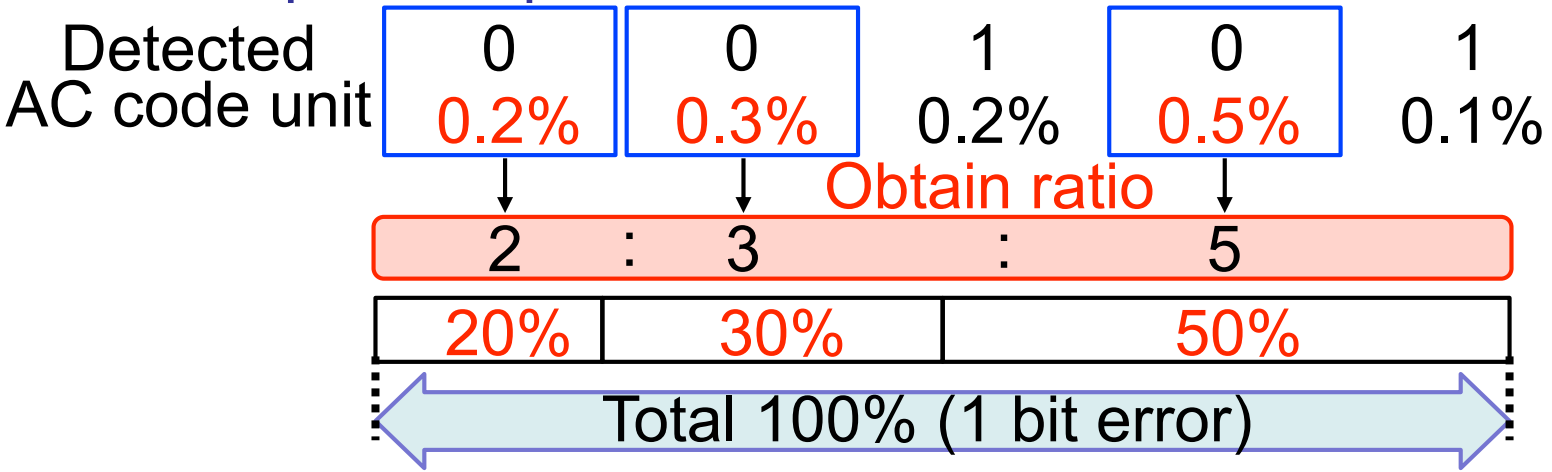
Proposed Error Correction

- Emphasize predicted BER of detected bits



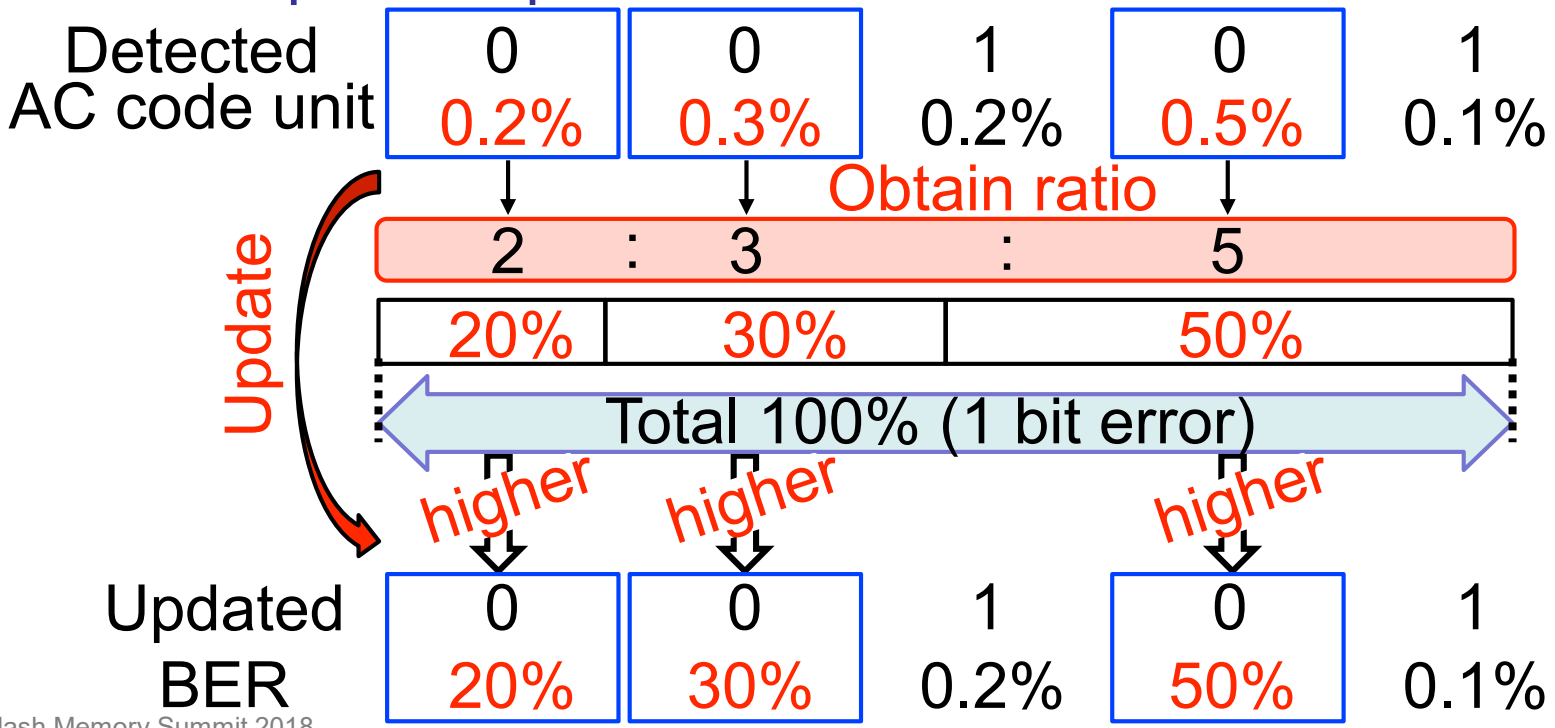
Proposed Error Correction

- Emphasize predicted BER of detected bits



Proposed Error Correction

- Emphasize predicted BER of detected bits



Result of Prop. Error Detection & Correction

Flash Memory Summit

N_{CL} = 8, Max iteration = 30, AC7, @85degC

Conventional AEP-LDPC w/o upper lower cells

Data-retention time (day) \ $N_{W/E}$	100	200	300	400	500
0	3	4	4	3	4
0.093	5	5	6	8	6
0.17	5	5	7	9	11
1.0	10	10	16		
1.2	11	11	18		
2.0	15	27			
2.2	20				
3.1			Fail		
4.0					
5.1					
6.1					

Proposed Error Detection & Correction

Data-retention time (day) \ $N_{W/E}$	100	200	300	400	500
0	3	3	3	3	3
0.093	4	5	4	4	4
0.17	4	4	4	4	5
1.0	5	7	6	8	10
1.2	6	7	8	8	11
2.0	8	9	10	12	19
2.2	8	10	12	17	
3.1	10	11	15	25	
4.0	13	14	26		
5.1	22			Fail	
6.1					

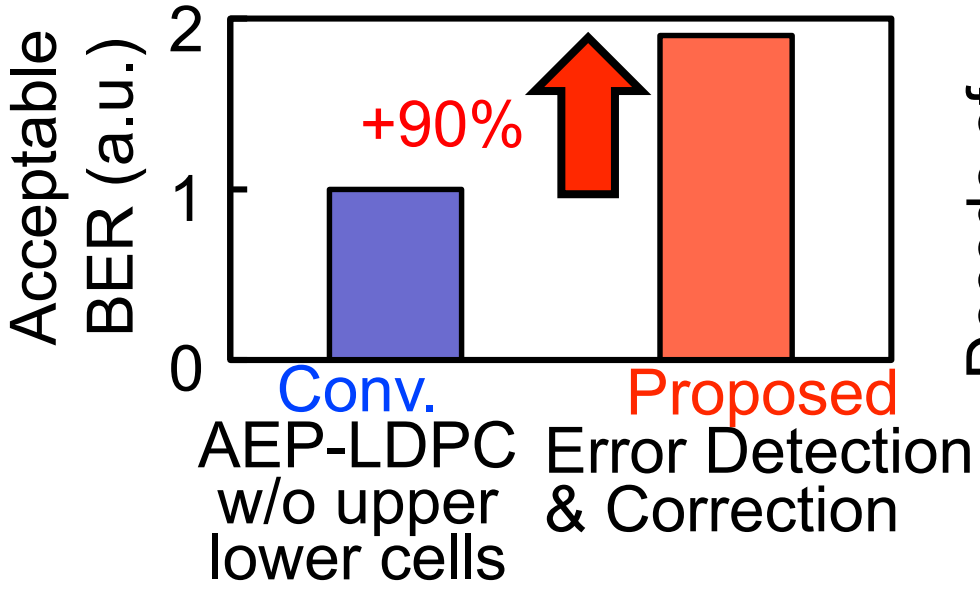
※Numbers in blue frames show number of iterations

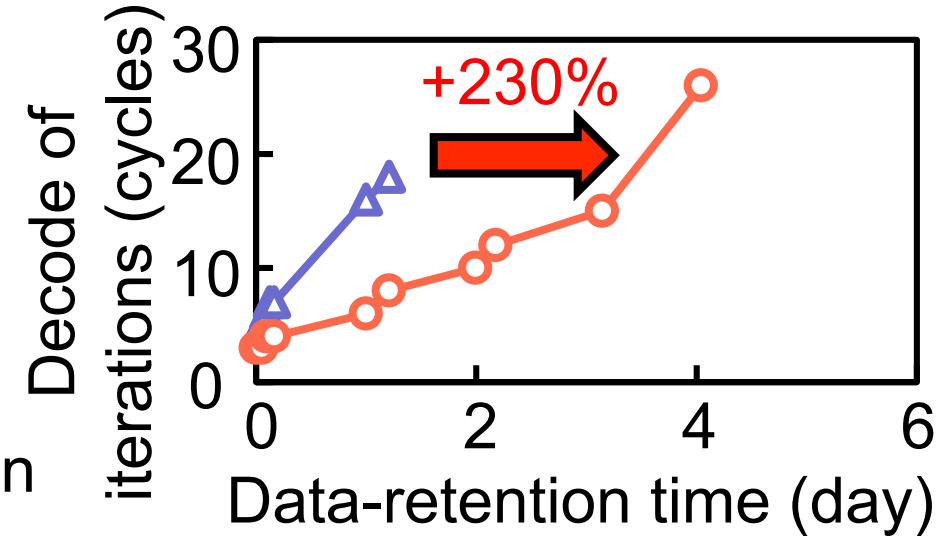
Result of Prop. Error Detection & Correction

Flash Memory Summit

3D-TLC NAND flash, N_{CL} = 8, $N_{W/E}$ = 300, Max iteration = 30, AC7, @85degC

△ Conv. AEP-LDPC w/o upper lower cells
○ Prop. Error Detection & Correction





Outline

- Background
  - Conventional Asymmetric Coding (AC)
  - Low-Density Parity-Check (LDPC) Code
- Proposals
  1) Error Detection (HED + V-LDPC)
  2) Error Correction
- Conclusion

Conclusion

Proposed techniques

①Error Detection (HED + V-LDPC)

- HED detects errors and V-LDPC averages detected errors

②Error Correction

- LLR values of detected error bits are emphasized

	Conv. LDPC ECC	Prop. Error Detection & Correction
Acceptable BER	Baseline	+90%
Acceptable data-retention time	Baseline	+230%

Thank you for your attention

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