A Case for Biased Programming in Flash

*Eitan Yaakobi, **Gala Yadgar**, Nachum Bundak, and Lior Gilon*

Computer Science Department

TECHNION
Israel Institute
of Technology

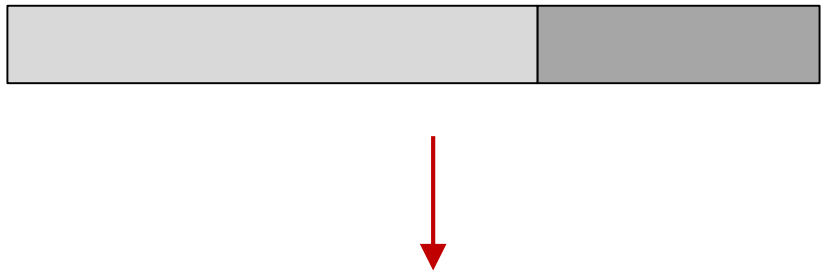


Biased programming: the basic idea

Increase number of 1’s in the data to reduce flash cell wear

10111011101101110110011101110111011001111 0

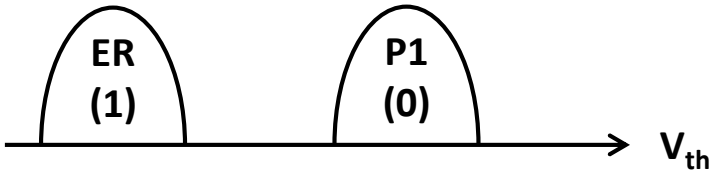
Utilize page spare area for coding overhead



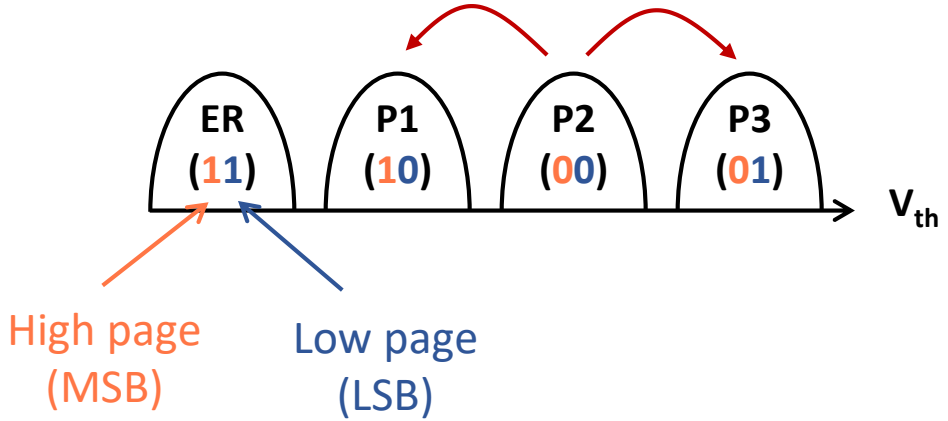
Reduce bit errors and increase lifetime (up to 24%)

Background: bit values and voltage levels

- SLC (single-level cell) flash



- MLC (multi-level cell) flash



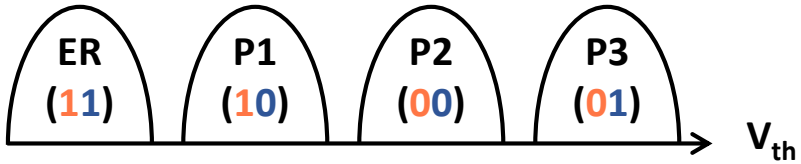
- Bit errors:
  - Program interference
  - Retention

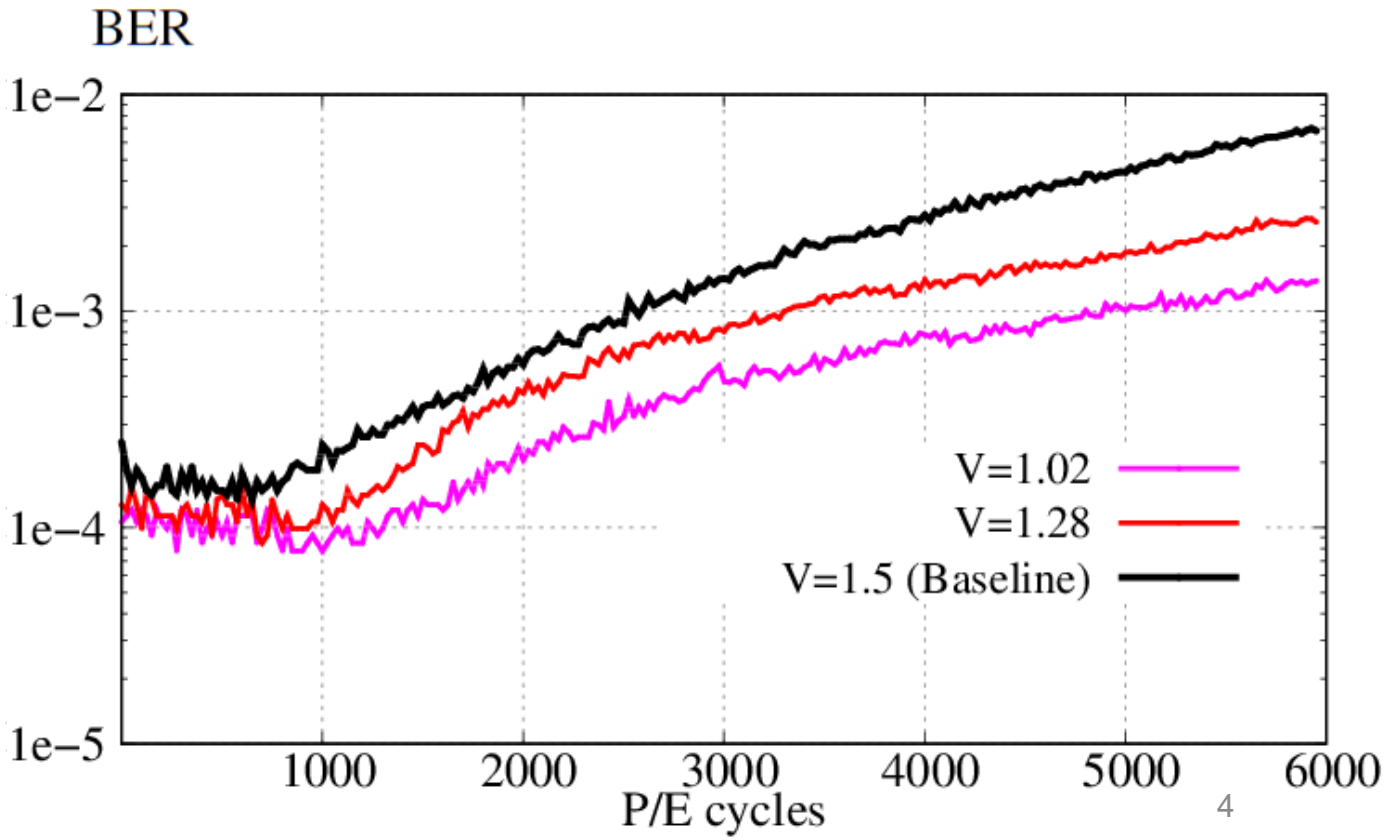
→ECC: error correction code

- Later in this talk, TLC: three level cell

Background: voltage levels and bit errors

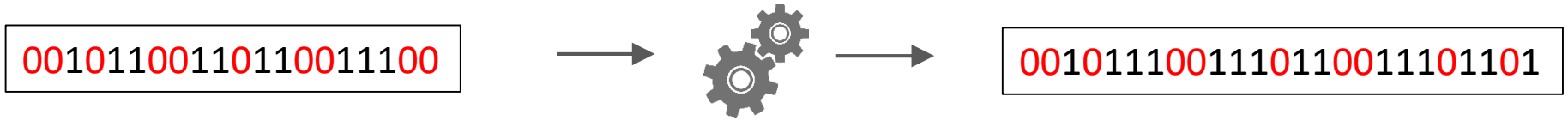
- BER: bit error rate (within page)
- Increases with lifetime
- Increases with average voltage level
  - Short term: more interference
  - Long term: more wear





Background: biased programming

- *Shaping (endurance) codes* can reduce the number of zeros in the data



P(zero) = 0.5

P(zero) = 0.4

- They also increase its overhead:

$$k_{0.5} = 1$$
$$k_{0.425} = 1.13$$
$$k_{0.4} = 1.24$$
$$k_{0.375} = 1.38$$
$$\dots$$

$$k_p = \frac{k}{h(p)}$$

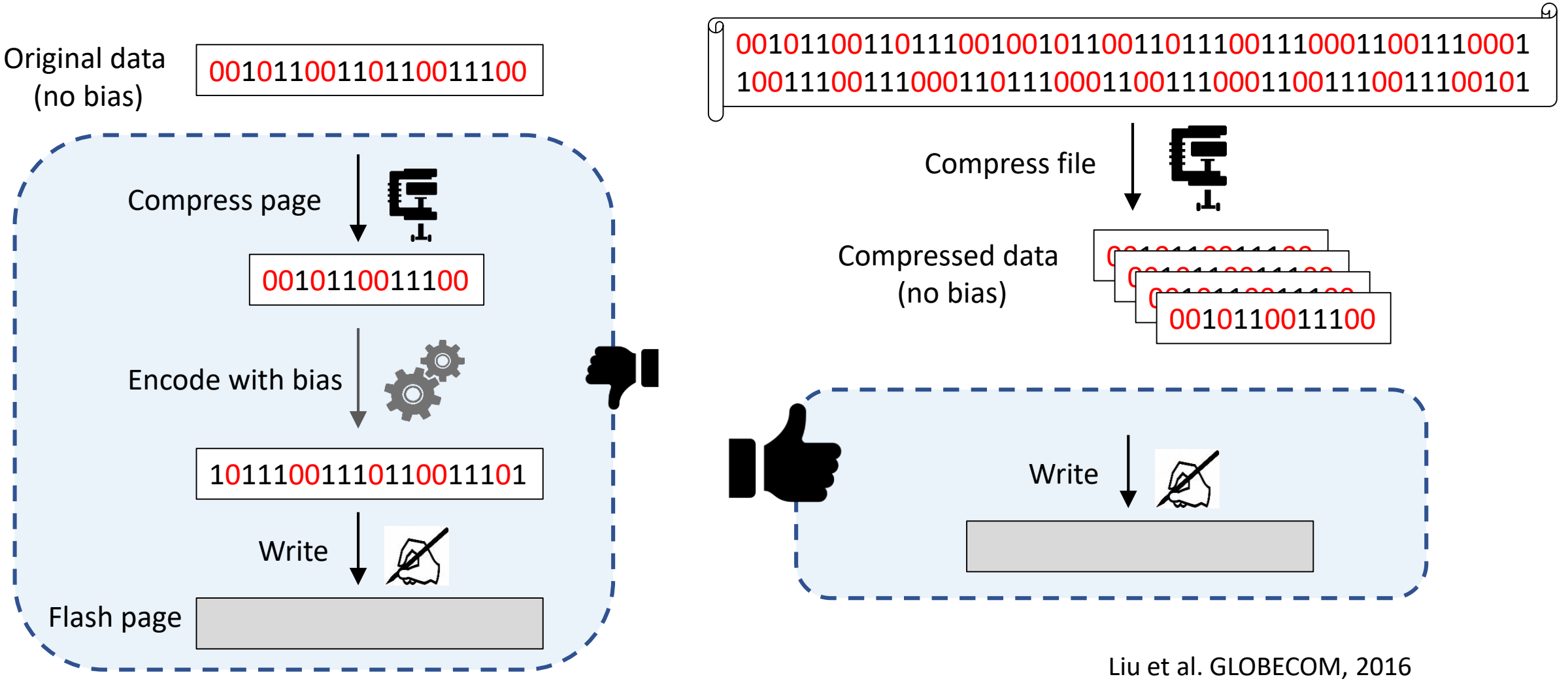
information bits

physical bits

binary entropy function

Jagmohan et al. GLOBECOM Workshops, 2010
Cover. IEEE Trans. on Info. Theory, 1973

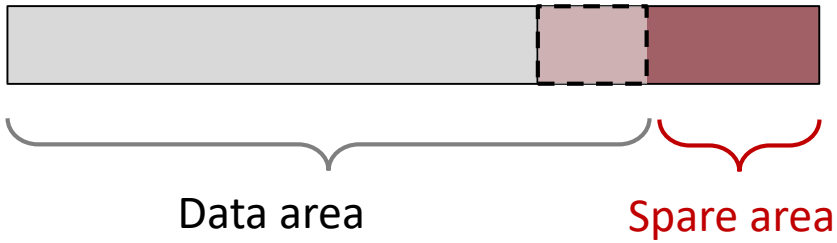
Previous approach: device-level compression

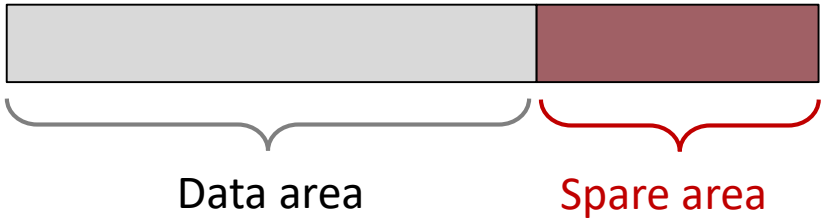


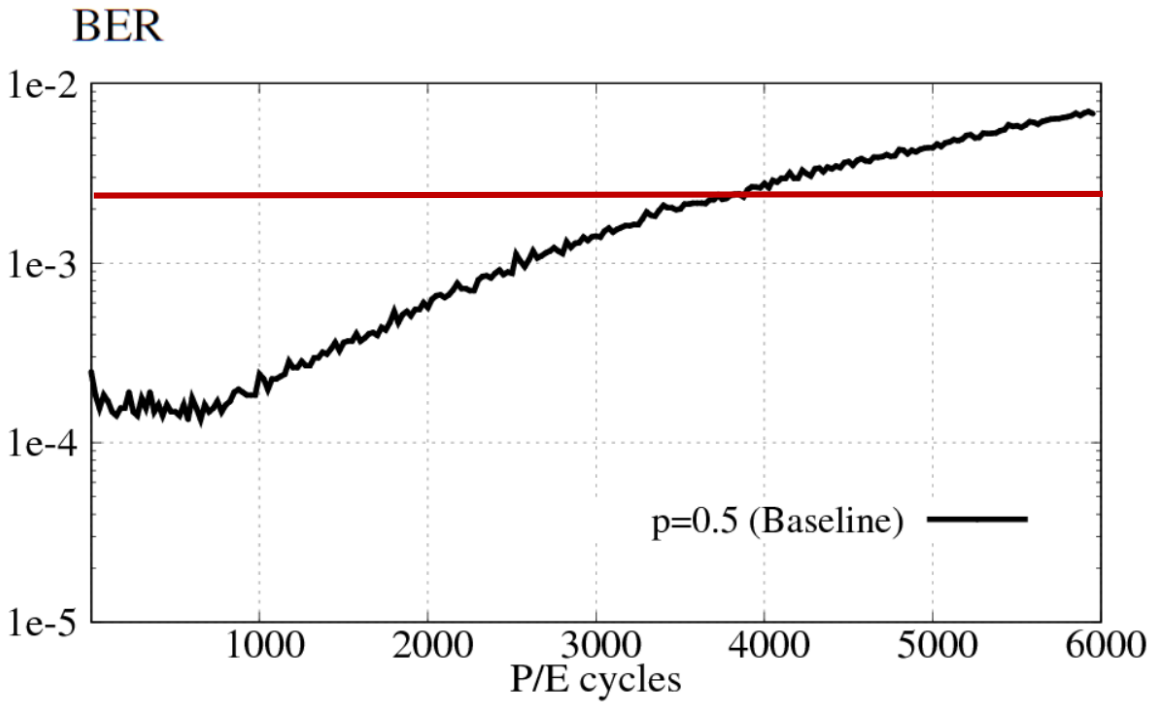
Liu et al. GLOBECOM, 2016
Li et al. USENIX FAST, 2015

Our approach: reallocation of page spare area

- The page spare area is used for ECC bits
  - Logical distinction
  - Read/written simultaneously
- Spare area size designed for worst-case scenario →end of lifetime
- In the meantime, we reallocate



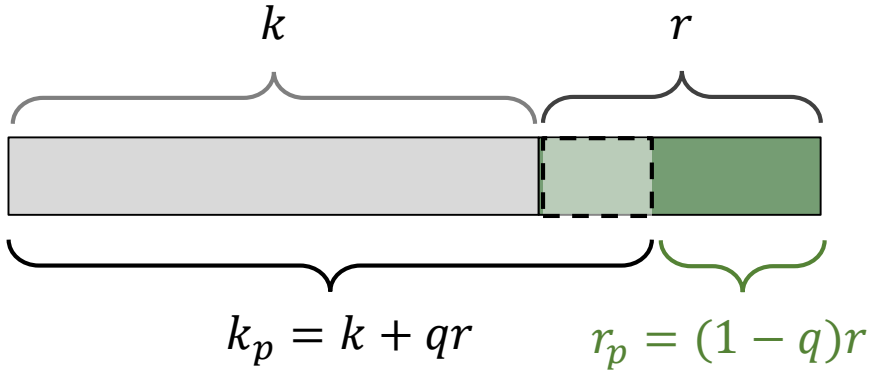


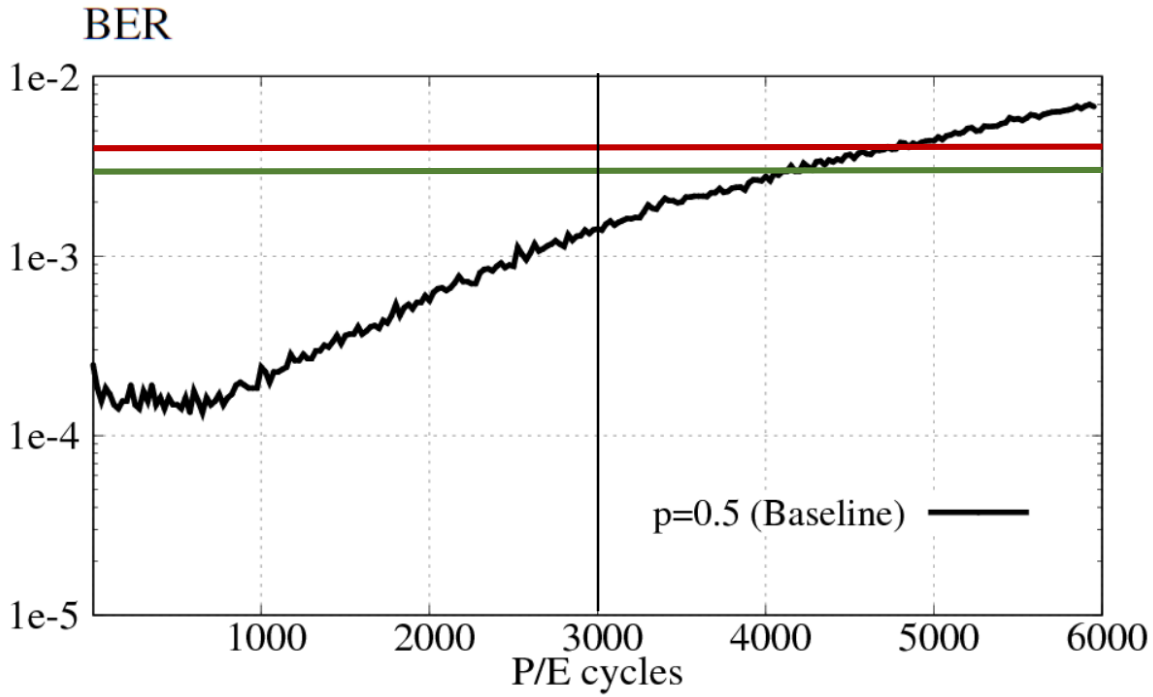


Safe reallocation (I)

- ECC must match the BER
  - # correctable bit errors: $\mathrm{t} = \left\lfloor \frac{r}{\lceil \log(k+r) \rceil} \right\rfloor$
  - Tolerated BER: $\mathrm{TBER} = \frac{t}{r+k}$
- After reallocation for bias p
  - New #correctable bit errors:

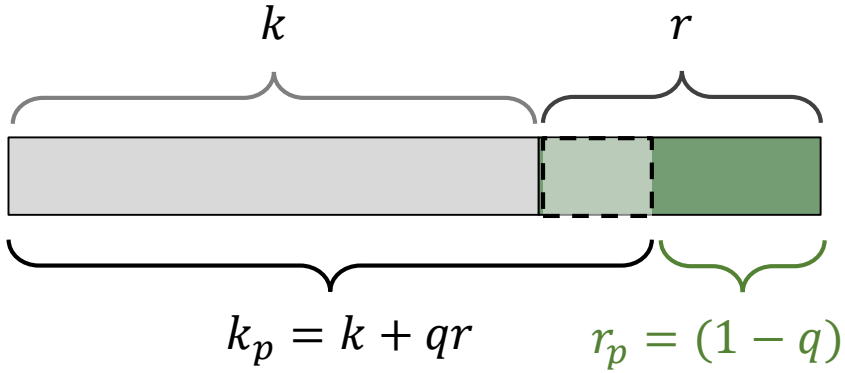
$$t_p = \left\lfloor \frac{(1-q)r}{\lceil \log(k+r) \rceil} \right\rfloor = (1-q) \times \mathrm{t}$$

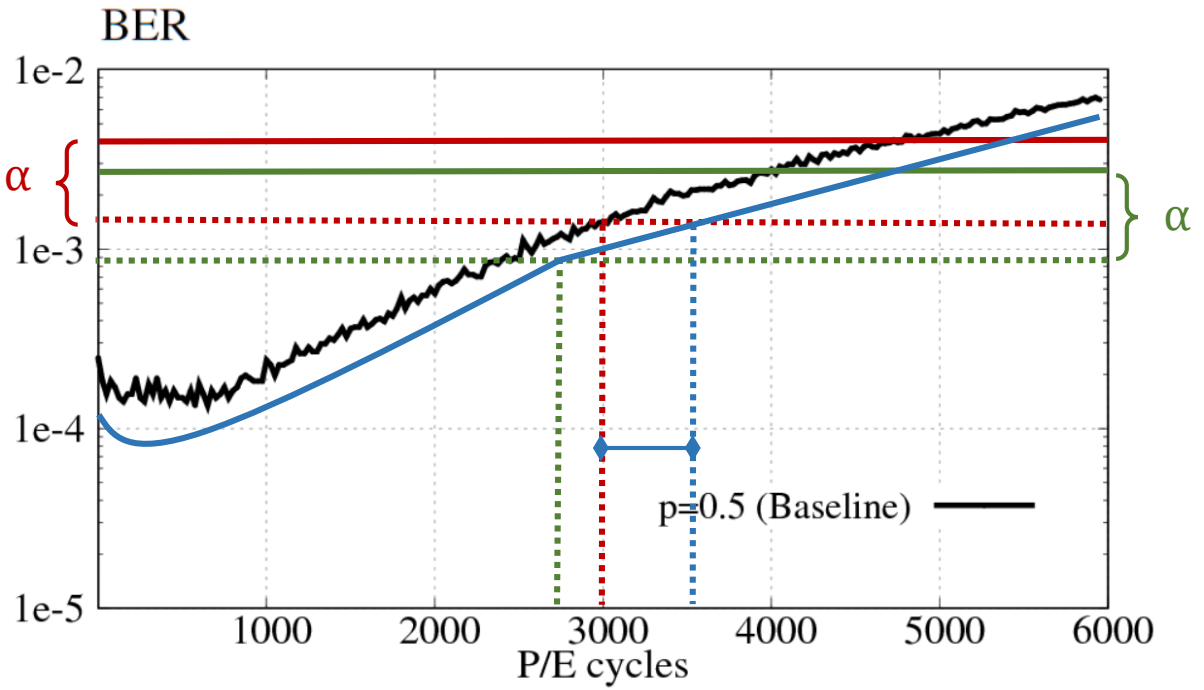




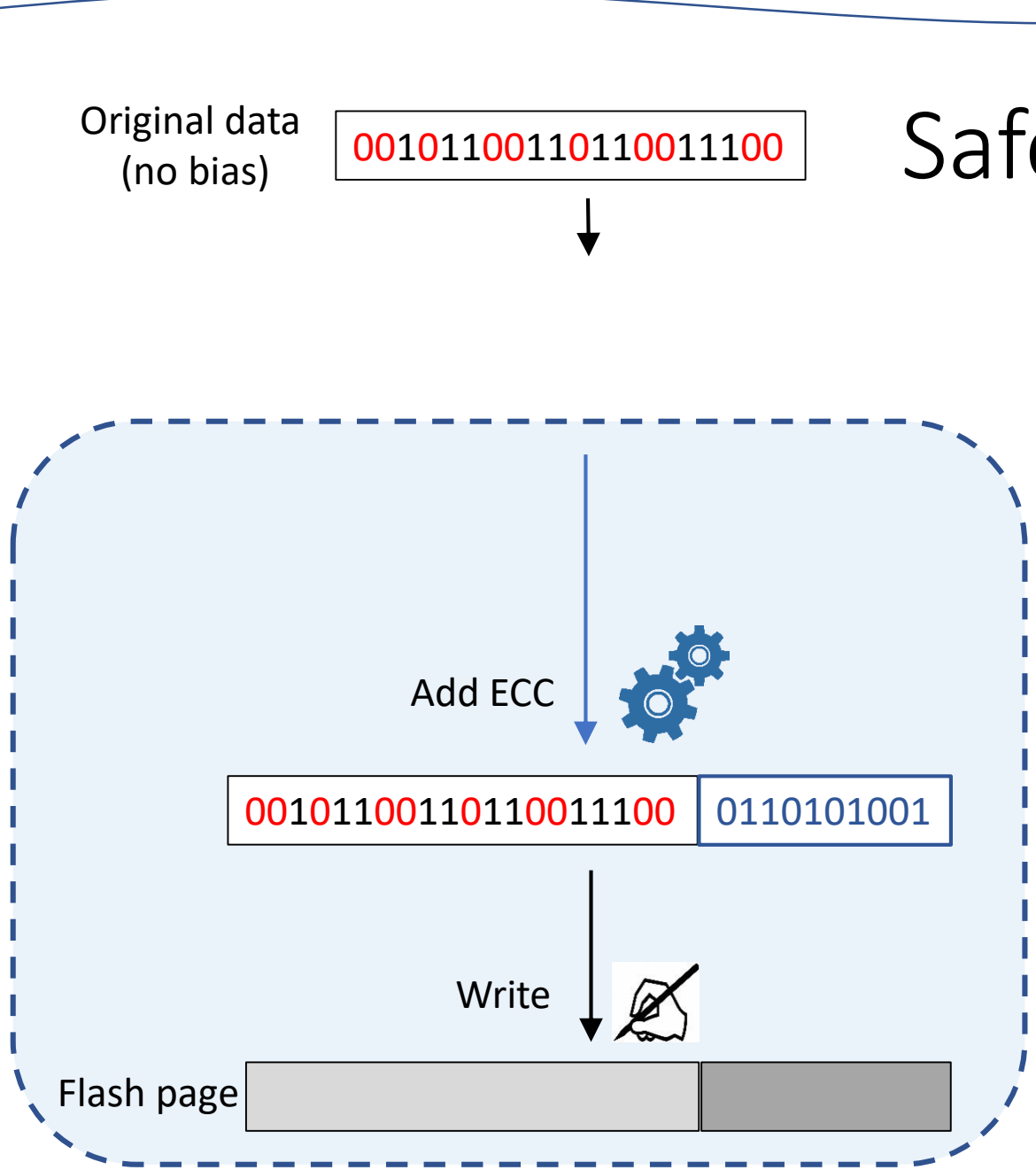
Safe reallocation (II)

- Chip characteristics
  - T = original lifetime (#P/E cycles)
  - $BER_{max}(T)$ = maximal BER at T cycles
  - *Safety ratio*: $\alpha = \frac{BER_{max}(T)}{TBER}$
- After reallocation, ensure
  $$\frac{BER_{max}(T_p)}{TBER_p} \leq \alpha$$
- "Return" reallocated cells, use until
  $BER_{max}(T') \leq \alpha TBER$
- Lifetime extension = $\frac{T'-T}{T}$

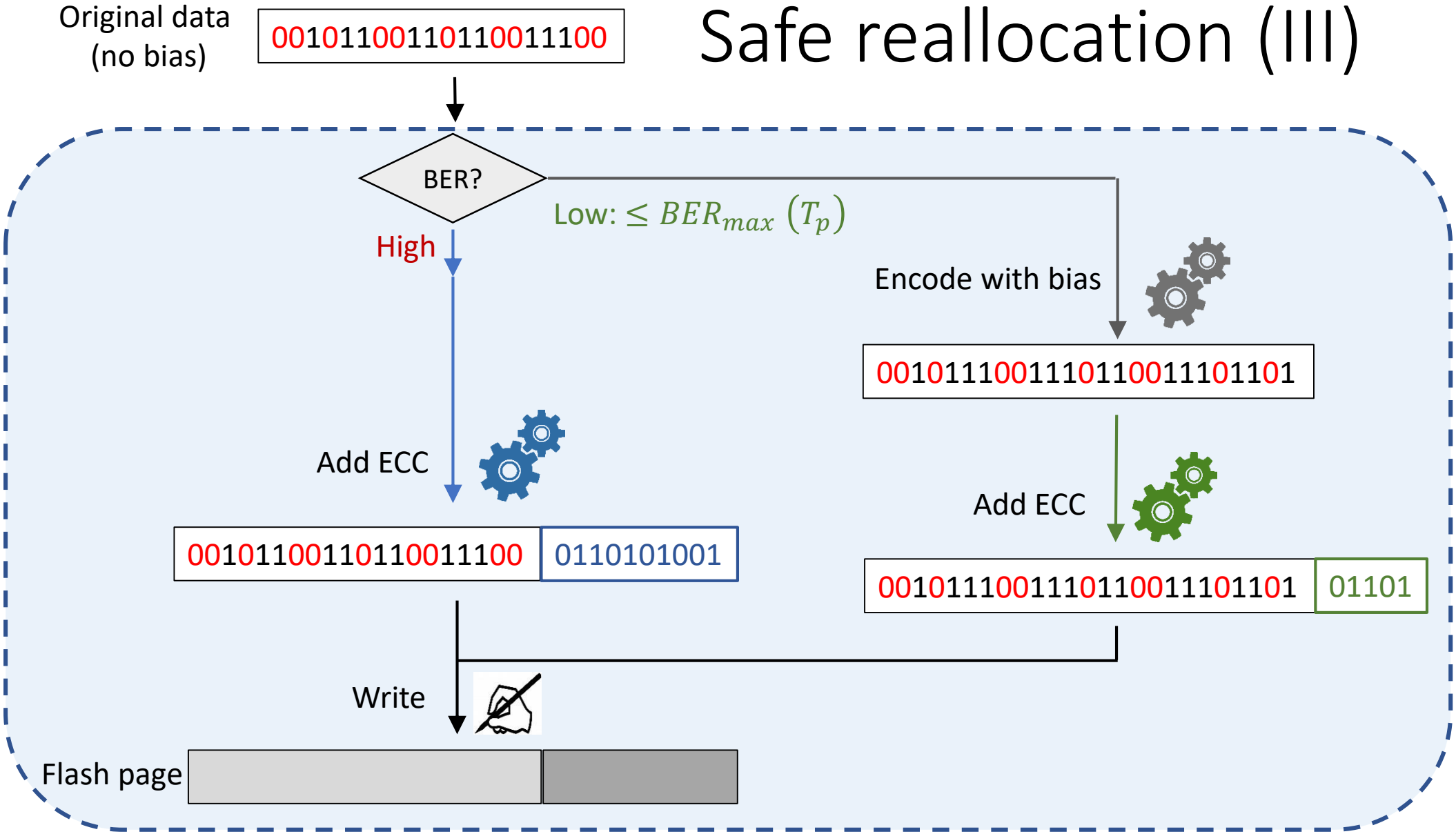




Safe reallocation (III)



Safe reallocation (III)



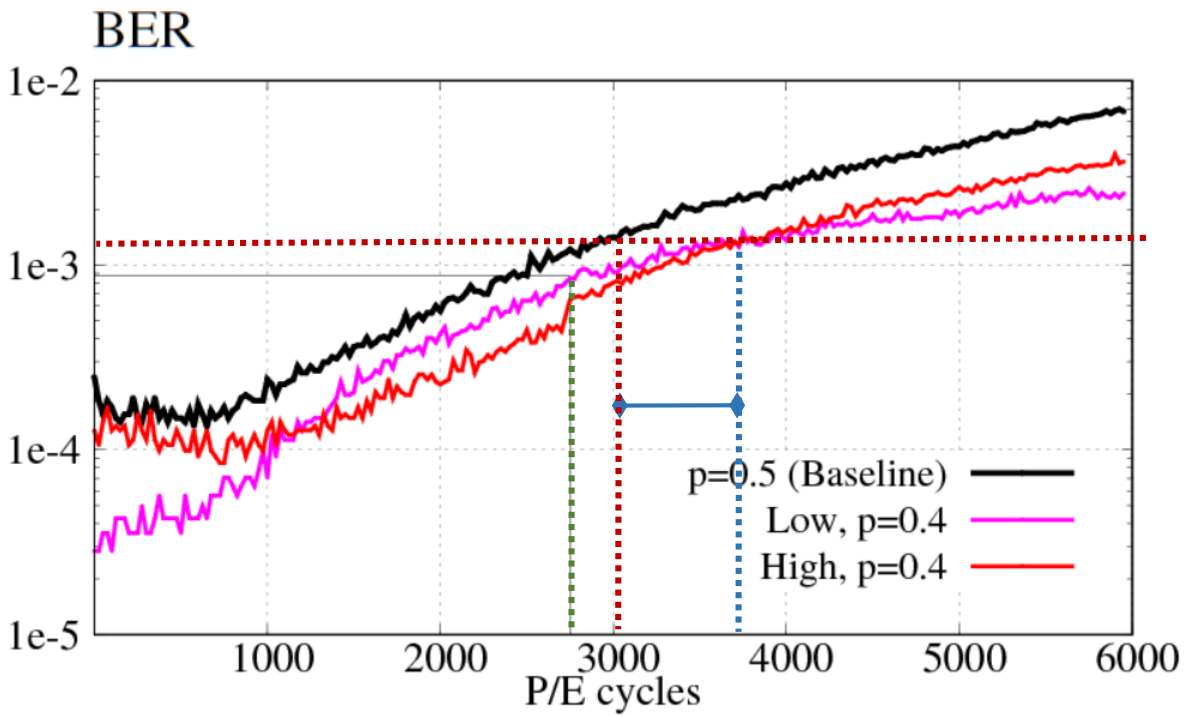
Example: Chip MLC-A

- Original lifetime $T = 3000$

1.415×10^{-3}

- Safety ratio $\alpha = \frac{BER_{max}(T)}{TBER} = 0.35$

4.019×10^{-3}



8.745×10^{-4}

- After reallocation $\frac{BER_{max}(T_{0.4})}{TBER_{0.4}} \leq 0.35$

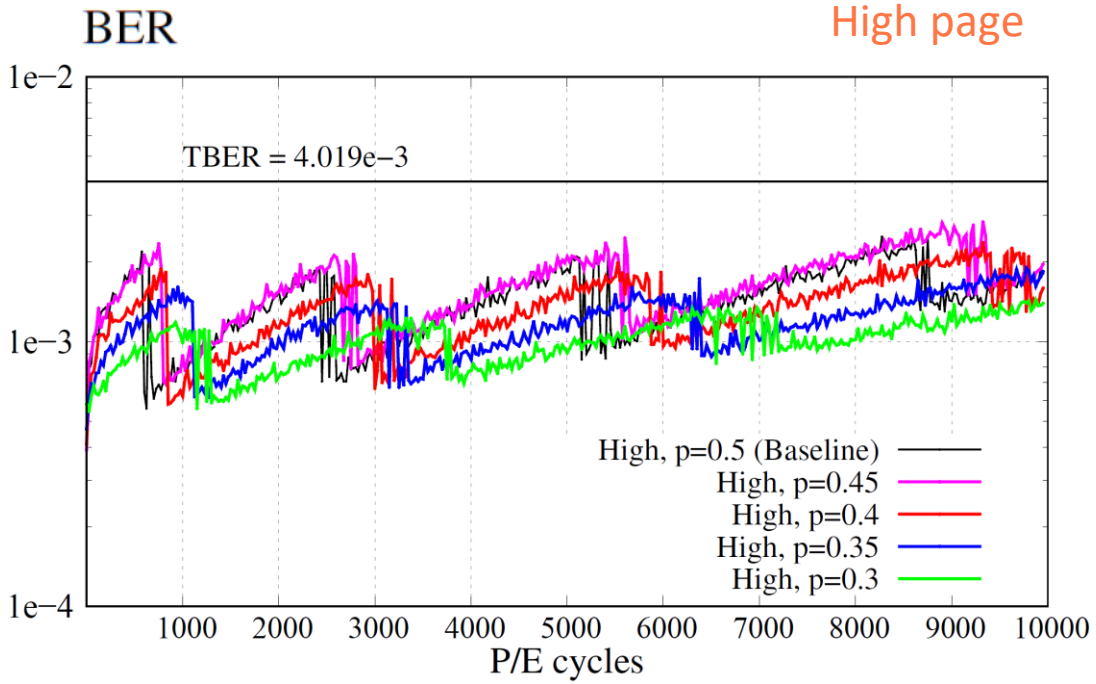
$T_{0.4} \leq 2750 \longrightarrow T' = 3725$

2.484×10^{-3}

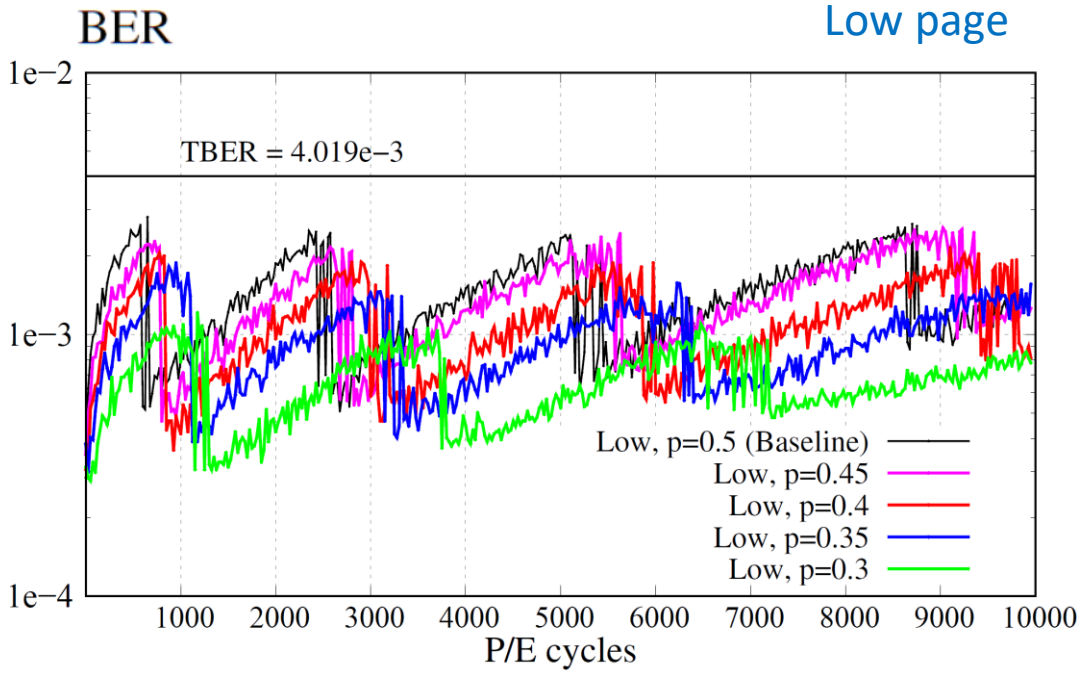
→ Biased programming increases lifetime by $\frac{T'-T}{T} = 24.17\%$

Example: Chip MLC-B

High page



Low page

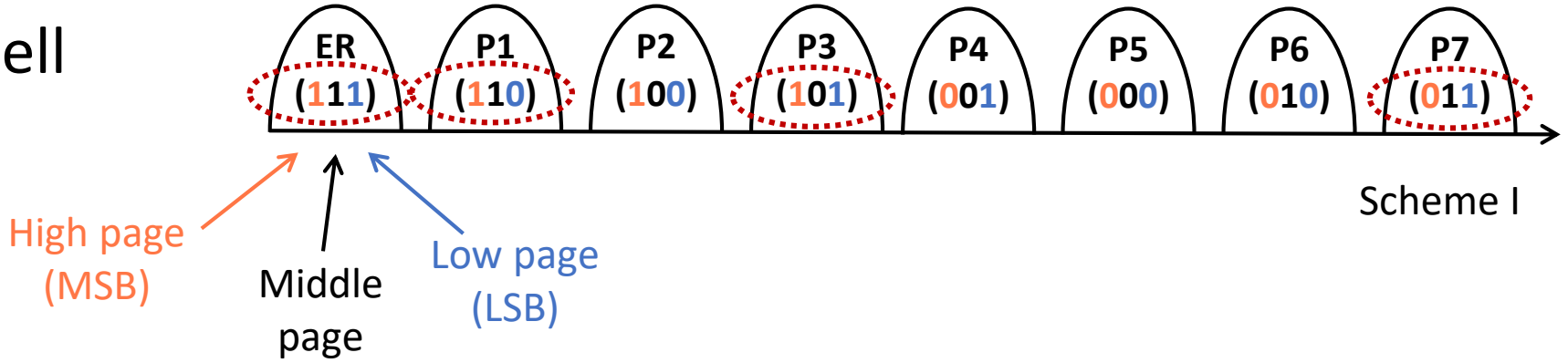


- “Deep erase” confirmed by manufacturer

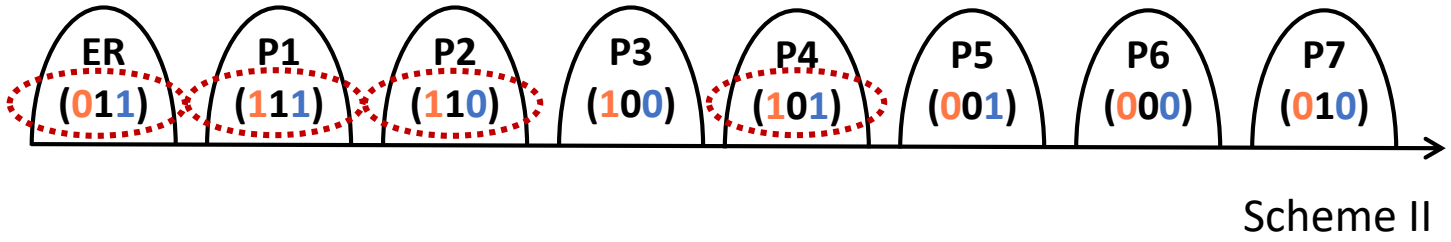
→Biased programming is applicable → future research

Example: TLC chip

- TLC: three level cell



- More mapping options
  - Some are better



- More bias combinations

→ Future research

[I] Yaakobi et al. *ICNC* 2012.
[II] Doi et al. *Japanese J. of App. Physics* 2016.

Conclusions

- Spare area can serve ECC and biased programming

→ interchangeably

- Requires tuning to chip characteristics
- Next:
  - TLC, 3D-NAND
  - LDPC
  - Bias combinations
  - Dynamic bias
  - System-level implications



Thank you!

Vector graphics by Starline, Katemangostar and Freepik from www.flaticon.com