

# **PRACTICAL NONVOLATILE MULTILEVEL-CELL PHASE CHANGE MEMORY**

**Doe Hyun Yoon**

IBM T. J. Watson  
Research Center

**Jichuan Chang,  
Robert S. Schreiber,  
Norman P. Jouppi**

**Hewlett-Packard Labs**

# MEMORY CAPACITY CHALLENGE IN HPC

- DRAM as main memory
  - Scaling is slowing down
    - Hard to meet ever-increasing capacity demand
- Byte-addressable nonvolatile memory
  - **Phase change memory (PCM)**, memristor, ...
  - Scales better than DRAM
  - Multilevel-cell (MLC) capability
  - **Nonvolatility**
    - Checkpoint, in-situ post processing
    - High-performance file system
- **NV MLC PCM for continued capacity scaling**

# MAJOR CHALLENGE: RESISTANCE DRIFT

- Conventional 4-Level-Cell (4LC) Designs
  - Naïve 4LC is useless
  - Optimized 4LC is only barely usable
  - Still need refresh -- it's volatile memory
- Observation:  
Most errors in 4LC occur in one cell state
- Proposal: **3-Level-Cell (3LC) PCM**
  - Simple, genuinely nonvolatile (>10 years retention)
  - 3-ON-2 and mark-and-spare
    - Low-cost wearout tolerance for 3LC
  - 1.41 bits/cell (vs. 1.52 in 4LC)
    - Only 7% lower capacity than (volatile) 4LC

# PCM AND RESISTANCE DRIFT

# PHASE CHANGE MEMORY

- Best of DRAM and Flash
    - Higher capacity, better scaling (vs. DRAM)
    - Faster, byte-addressable NVM (vs. Flash)
  - MLC (Multilevel-Cell) capability
    - Store more than 1 bits per cell
      - Ex) 2 bits per cell
  - Caveats:
    - Slow, low-bandwidth write
    - Finite write endurance
    - **Resistance drift**
- Common problems  
in both SLC and MLC

# RESISTANCE DRIFT

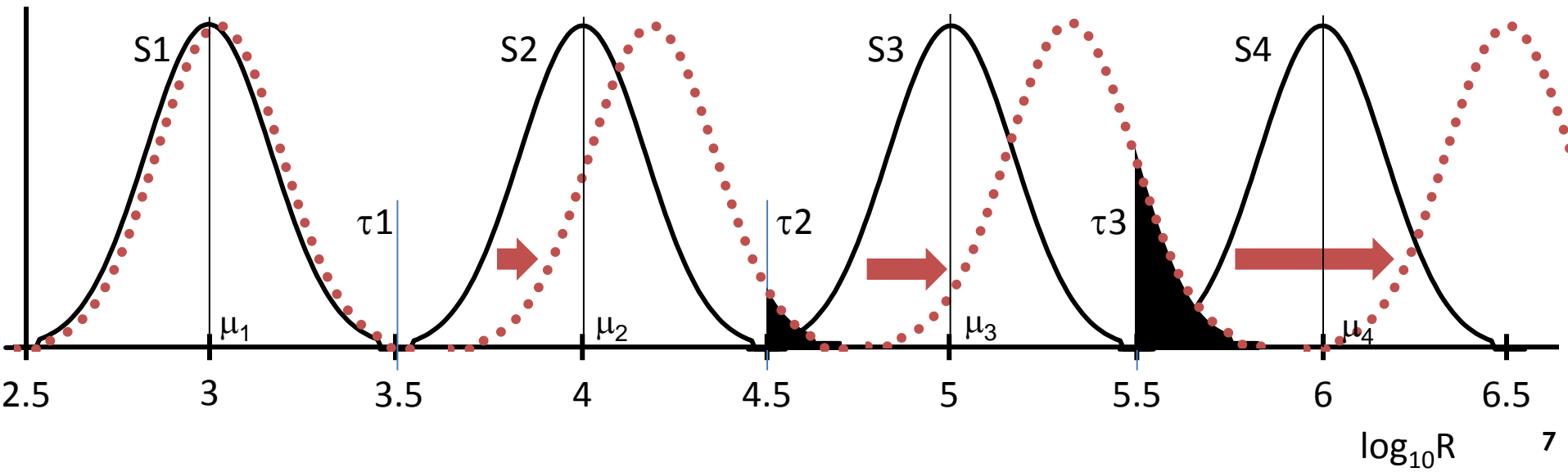
- PCM Cell resistance increases over time
  - $R(t)$ , cell resistance at time  $t$  ( $t > 0$ )
    - A cell is programmed at  $t=0$
    - Sensed as  $R_0$  at time  $t_0$  ( $>0$ )
    - $\alpha$ : drift rate ( $0 < \alpha < 1$ )

$$R(t) = R_0 \times \left( \frac{t}{t_0} \right)^\alpha$$

- Drift errors
  - Negligible in SLC PCM
  - **Major reliability problem in MLC PCM**

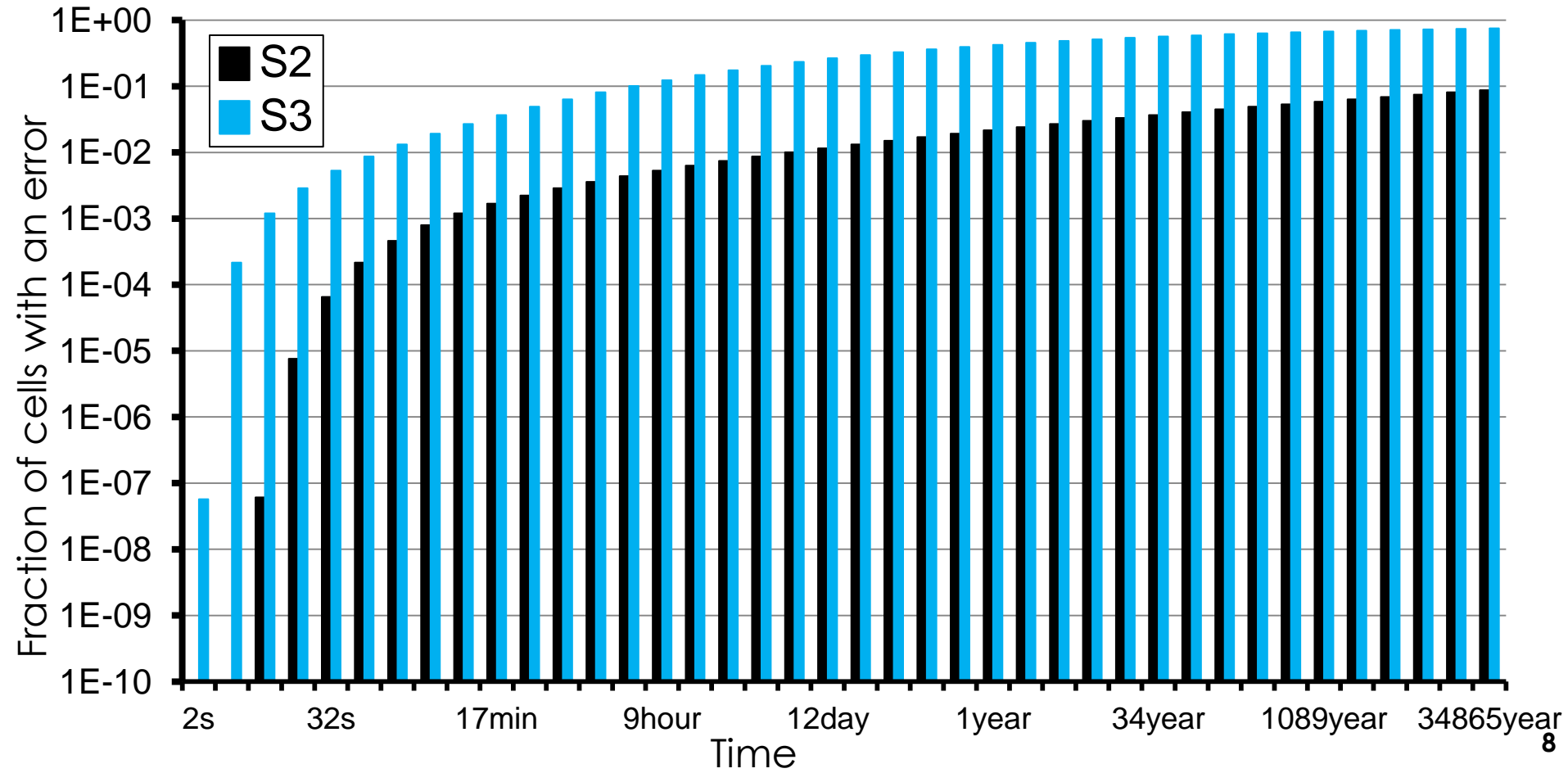
# DRIFT ERRORS IN 4LC PCM

- 4 cell states: S1, S2, S3, S4
  - PDF is truncated Gaussian
    - $\pm 2.75 \sigma$  around mean values
    - Mean resistance values:  $\mu_1, \mu_2, \mu_3, \mu_4$
  - Threshold between states:  $\tau_1, \tau_2, \tau_3$
- **Drift rate ( $\alpha$ ) increases with cell resistance**



# DRIFT ERROR RATES

- Monte-Carlo simulation
- Errors only in S2 and S3





# REFRESH

- Refresh before cells lose their data
  - Consume already limited PCM write BW
  - Too frequent refresh will make PCM unavailable to users
- What PCM refresh interval is acceptable?
  - At least 50% of write BW should be available to users
  - Refresh interval >17 minutes
- **Caveat: PCM w/ refresh is no longer nonvolatile**

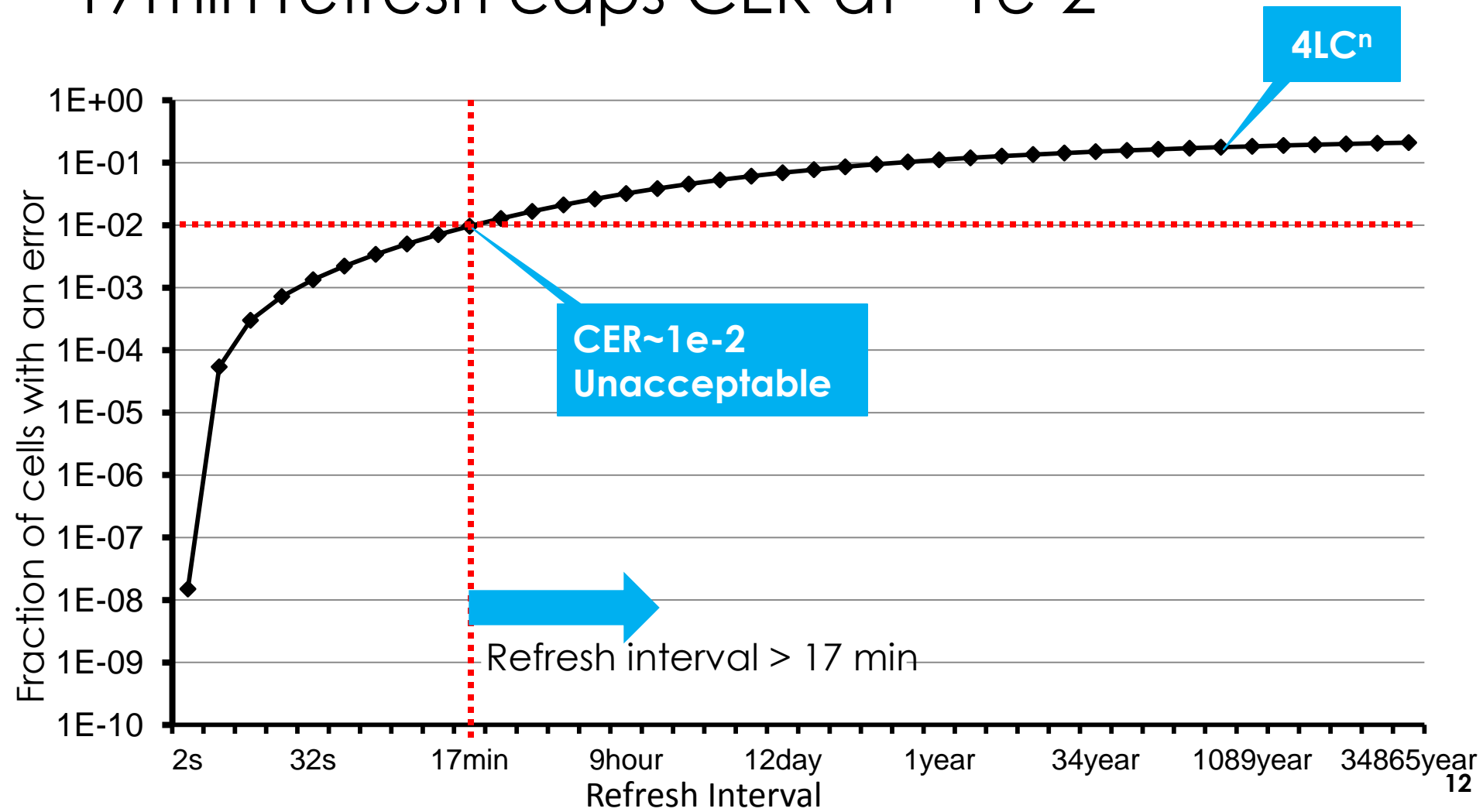
# CELL ERROR RATE

- What cell error rate is tolerable?
  - Goal: 10-year device MTBF
    - Fewer than 1 erroneous 64B block in a 16GB device for 10 years
  - **CER > 1e-2**
    - Impossible to achieve the goal even with unrealistically strong ECC
  - **CER ~ 1e-3 @ 17min refresh**
    - Barely meets the goal with BCH-10
- More analysis in the paper

# **BASELINE 4LC PCM**

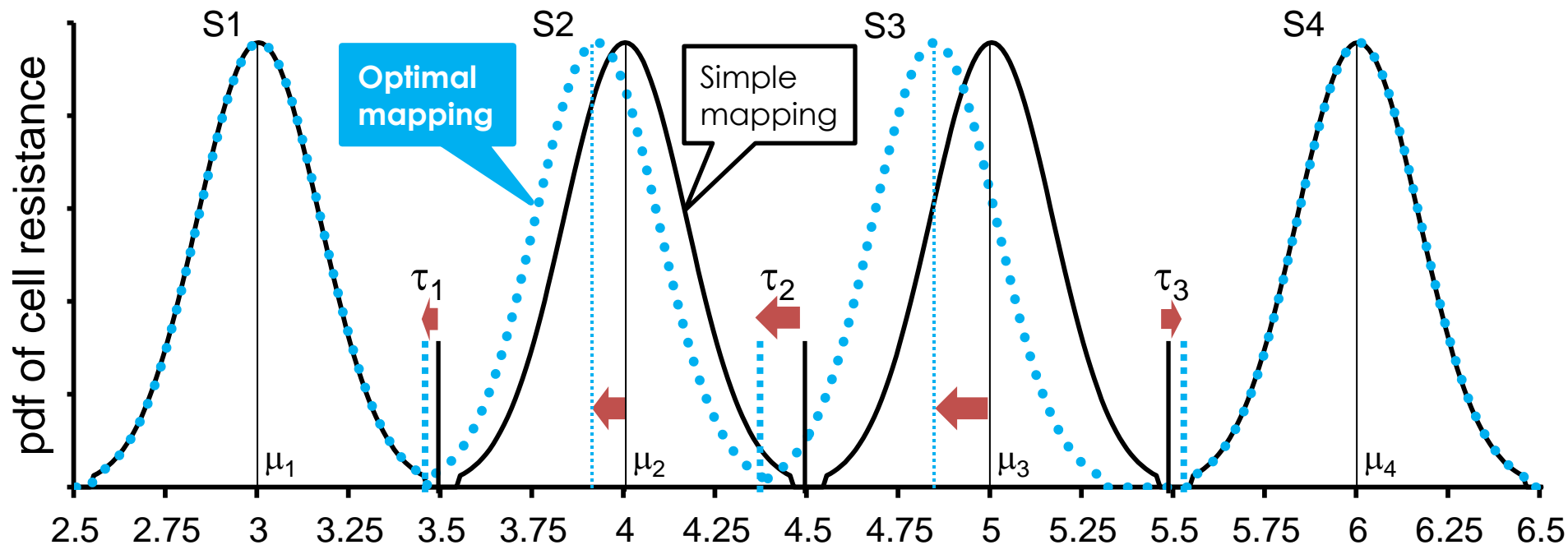
# NAÏVE DESIGN: $4LC^N$

- Equal probability for all 4 states
- 17min refresh caps CER at  $\sim 1e-2$



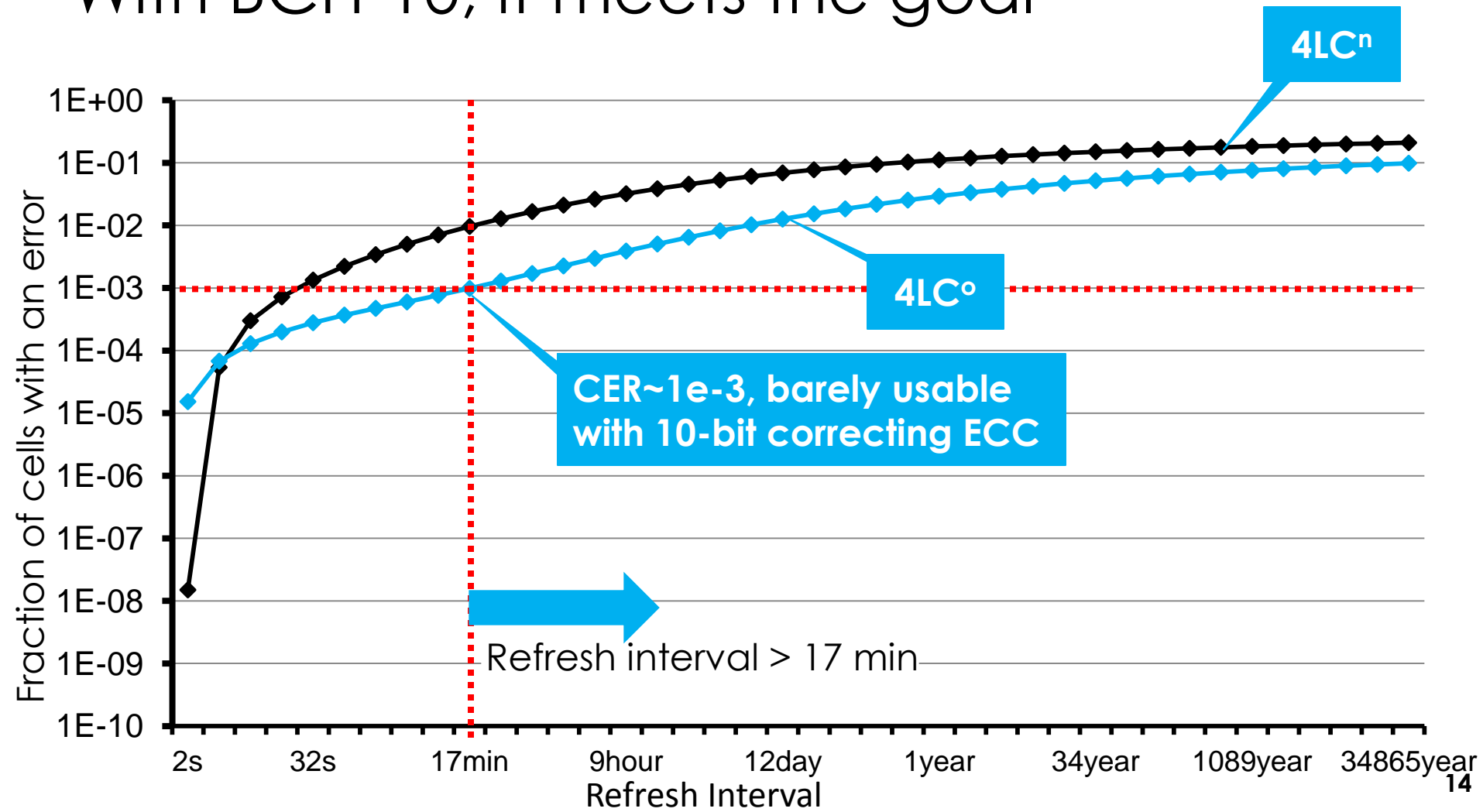
# OPTIMAL STATE MAPPING

- Drift only increases cell resistance
- Optimize  $\mu_2, \mu_3, \tau_1, \tau_2, \tau_3$  to minimize CER
  - minimize  $\text{CER}(\mu_2, \mu_3, \tau_1, \tau_2, \tau_3)$
  - subject to  $\mu_i + 2.75\sigma + \delta < \tau_i < \mu_{i+1} - 2.75\sigma - \delta$
  - for  $i=1,2,3$



# OPTIMAL STATE MAPPING: 4LC<sup>o</sup>

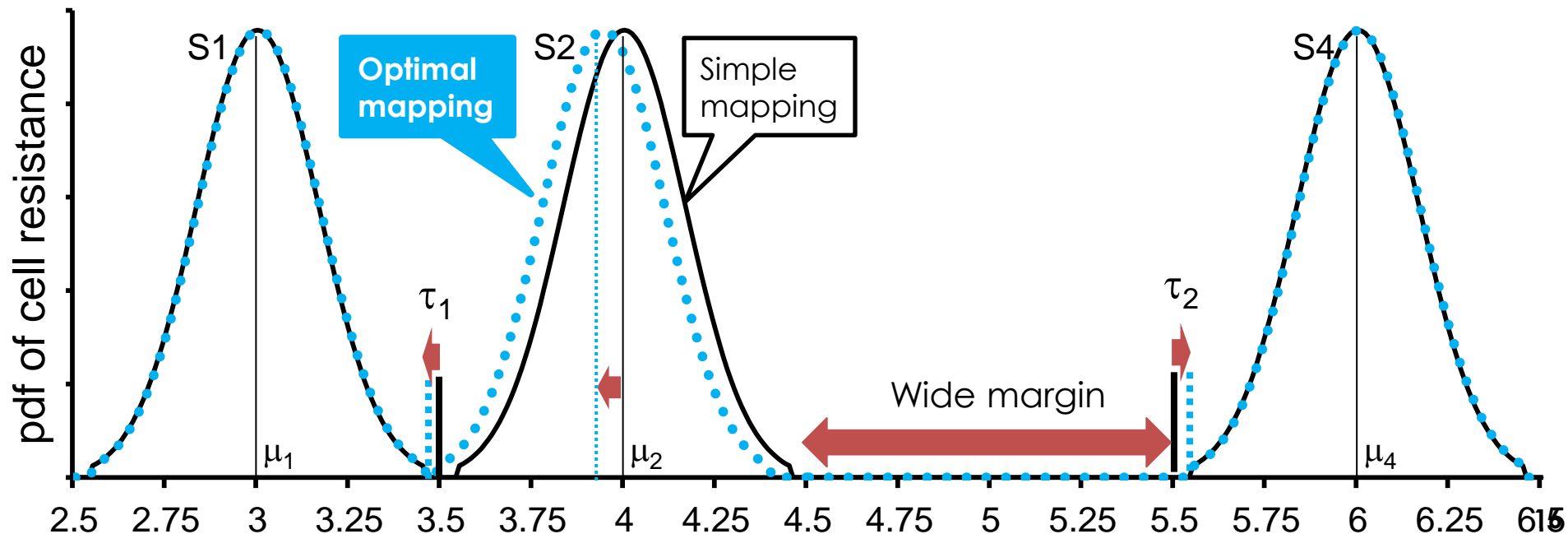
- CER  $\sim 1e-3$  @ 17-min refresh
- With BCH-10, it meets the goal



# **PROPOSAL: 3LC PCM**

# PROPOSAL: 3LC PCM

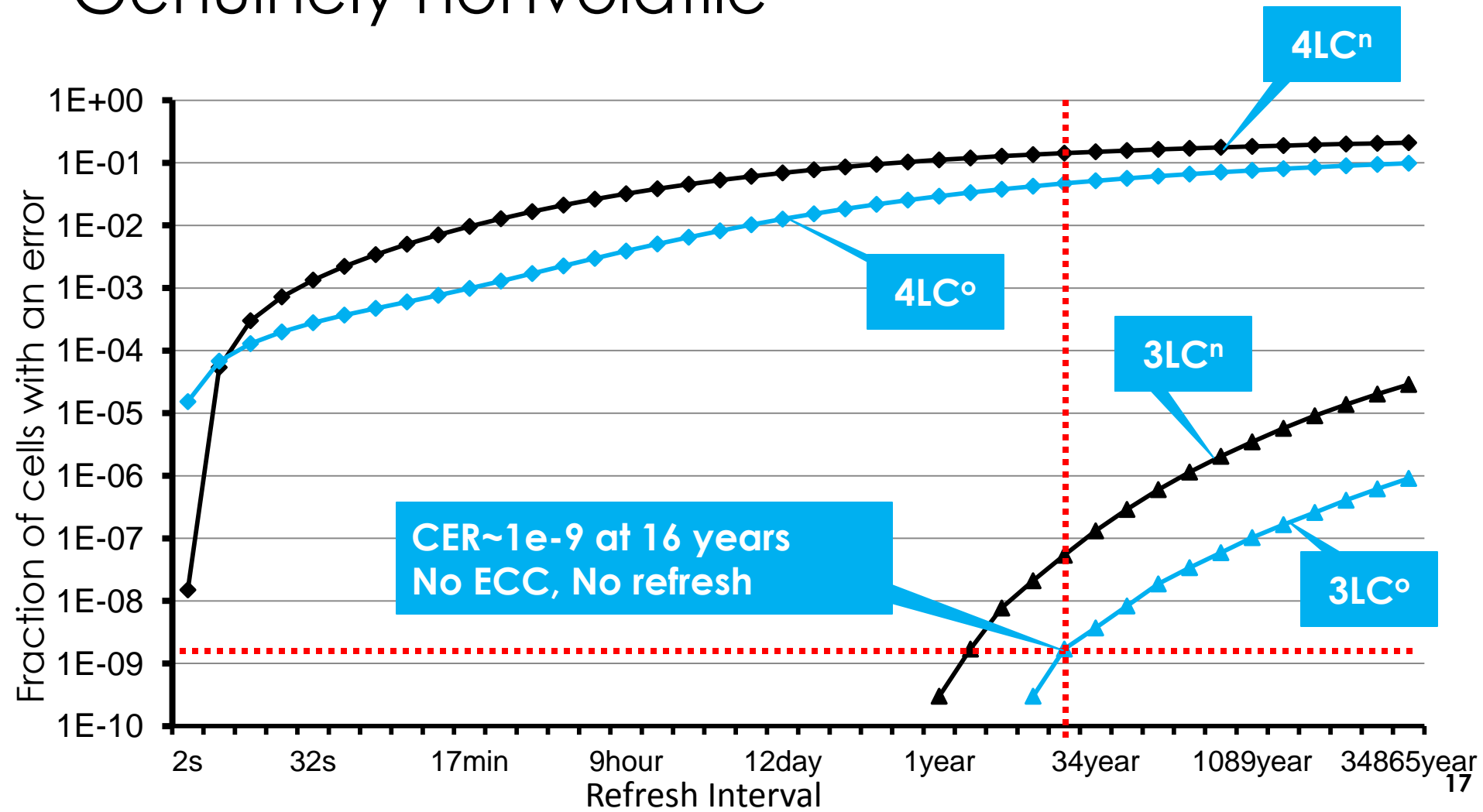
- Observation:
  - Most errors occur in one state (S3)
- DO NOT USE IT
  - Wide Margin for S2
- Simple and optimal mapping ( $3LC^n$  &  $3LC^o$ )





# 3LC DESIGNS (3LC<sup>N</sup> AND 3LC<sup>0</sup>)

- Reliable for >10 years w/o ECC & refresh
- Genuinely nonvolatile



# 3LC PCM DESIGN ISSUES

- How to store information?
  - Binary information in ternary cells
- What about wearout failures?
- How to compensate for the reduced cell density?
  - 3LC's ideal capacity is 1.58 bits/cell ( $\log_2 3$ )
  - vs. 2 bits/cell in 4LC

# HOW TO STORE BINARY INFO IN TERNARY CELLS?

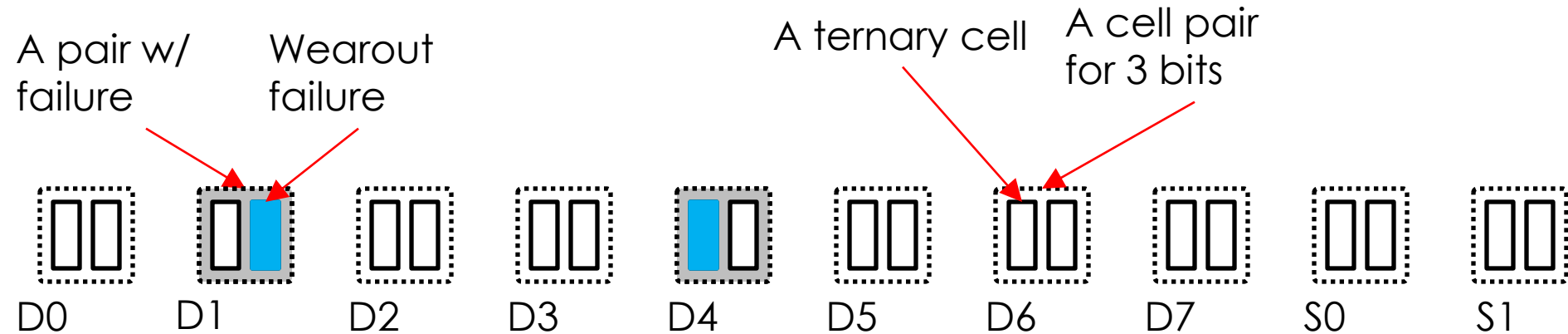
- 3-ON-2
  - Store three bits in two ternary cells
  - 64B (512-bit) data block in 342 cells
- 9 states in 2 ternary cells
- 8 states for 3-bit data
- **INVALID state**
  - (S4, S4)
  - Use this for tolerating wearout failures

First cell	Second cell	3-bit data
S1	S1	000
S1	S2	001
S1	S4	010
S2	S1	011
S2	S2	100
S2	S4	101
S4	S1	110
S4	S2	111
S4	S4	INVALID

# TOLERATING WEAROUT FAILURES IN 3LC

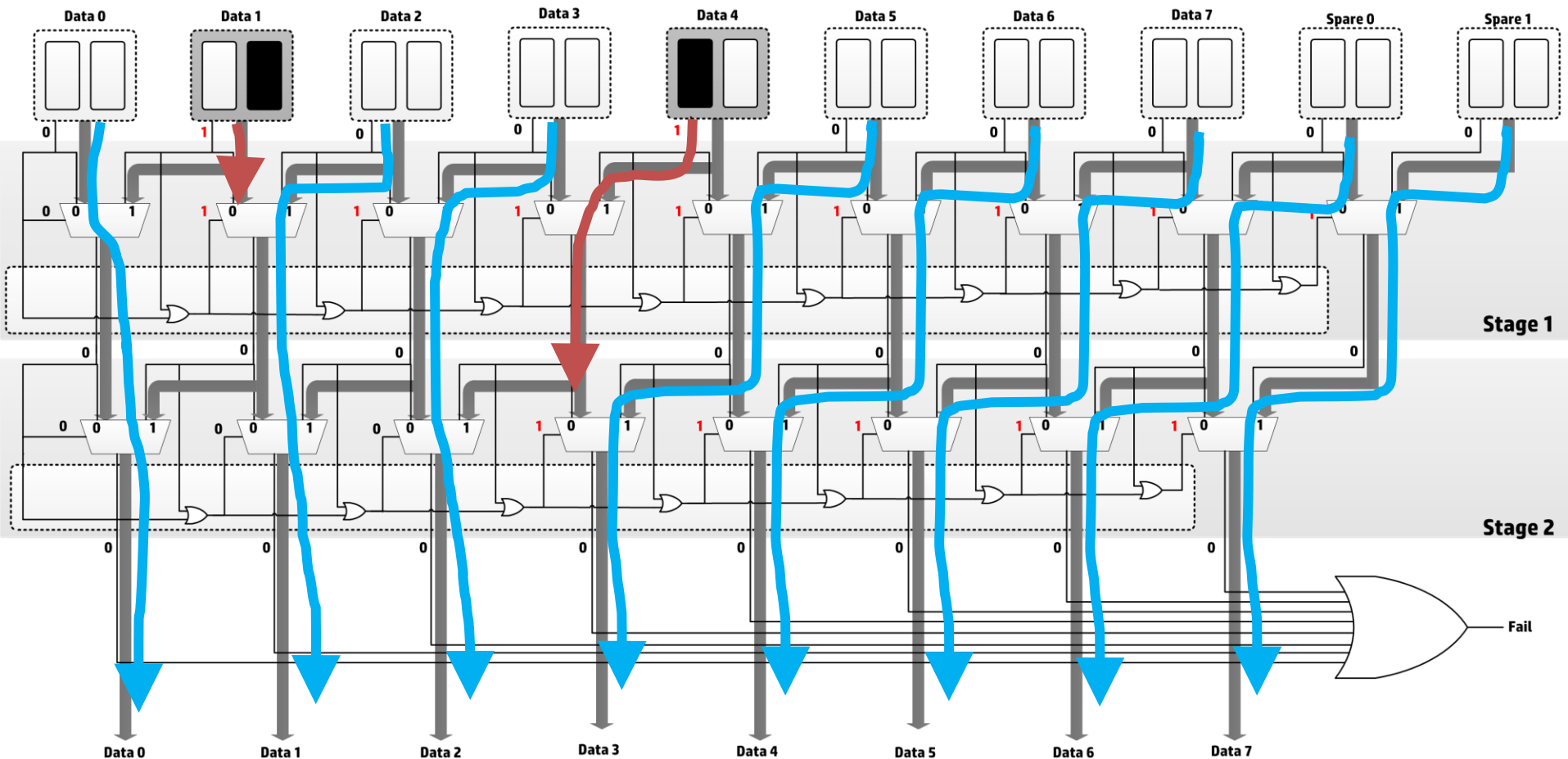
- PCM has only finite write endurance
  - $\sim 10^8$  writes per cell
- **Mark-and-spare**
  - A low-cost wearout failure tolerance for 3LC
  - Use 3LC's INVALID state for marking a cell pair with a failure
  - No need to store failed-cell location
  - 2 spare cells per failure
- c.f. ECP [Schechter+ ISCA'10 ]
  - Need a pointer and a spare cell for a failure
  - 5 cells per failure with 512-bit data block and 4LC

# MARK-AND-SPARE EXAMPLE



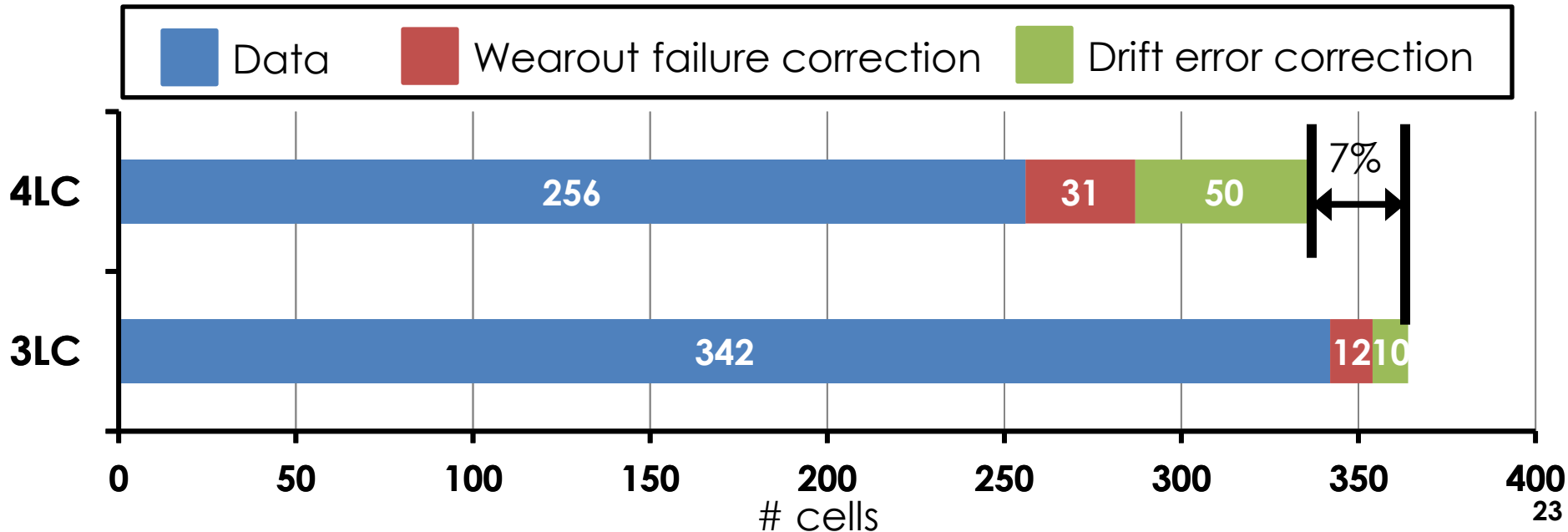
- Use INVALID (S4, S4) to mark a cell pair w/ failure
  - A stuck-at cell stuck can be revived by applying reverse current [Goux+ IEEE TED'09]
- Need a spare pair for tolerating a failure

# HOW TO CORRECT WEAROUT FAILURES?



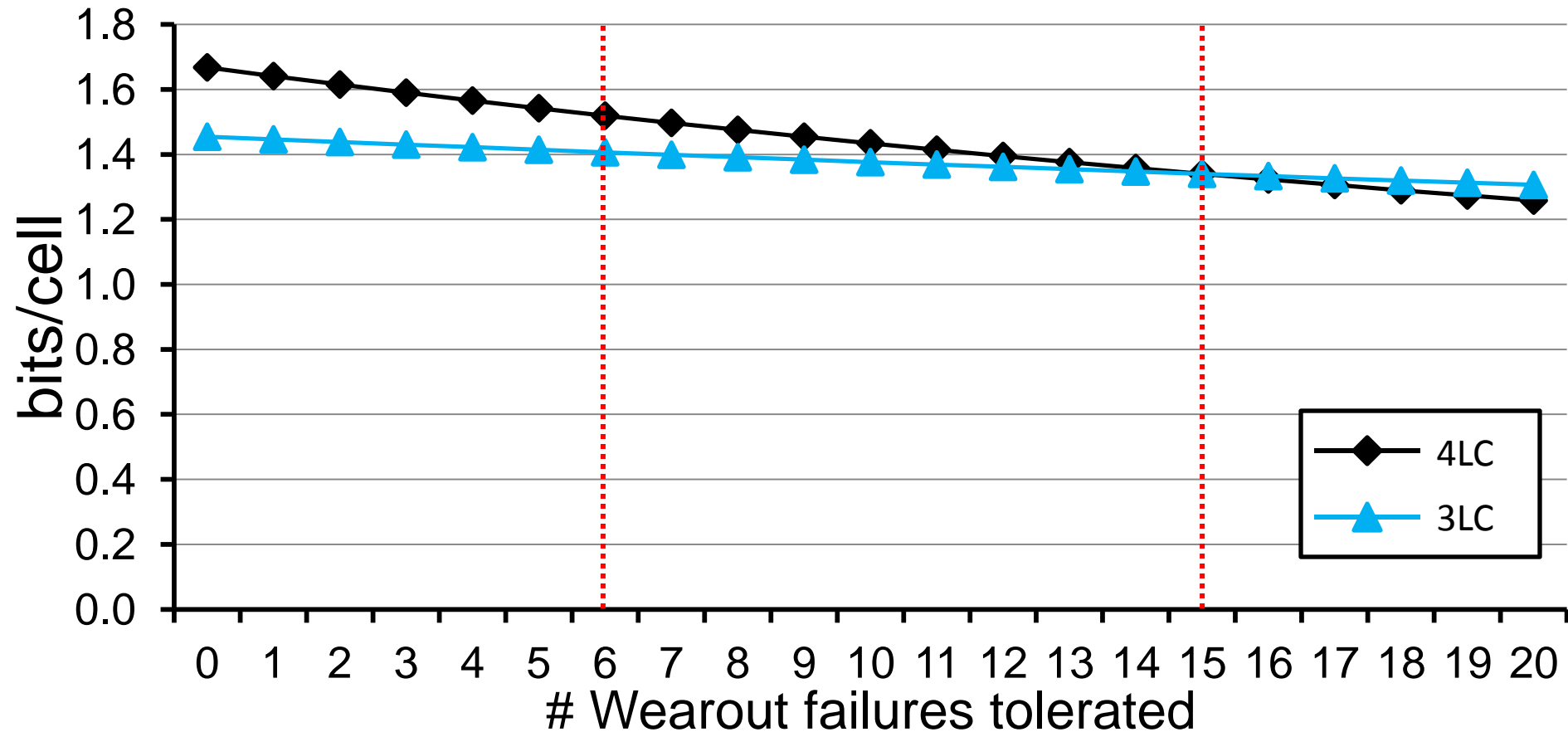
# CAPACITY: 3LC vs. 4LC

- 64B (512-bit) block
- 3LC needs fewer bits than 4LC for error correction
  - 6 wearout failures:  
Mark-and-spare (2cells/failure) vs. ECP (5cells/failure)
  - Drift errors: BCH-1 vs. BCH-10
- 3LC: 1.41 bits/cell, 4LC: 1.52 bits/cell
- Besides, 3LC is nonvolatile



# CAPACITY VS. # WEAROUT FAILURES

- MLC has worse endurance than that of SLC
- May need to tolerate more than 6 wearout failures





# COMPARISON TO TRI-LEVEL-CELL PCM

- Recent work on MLC drift errors [ISCA'13]
  - Same observation
    - Most errors occur in the S3 state
  - Same solution
    - Use 3 levels instead of 4 levels
- TLC paper does not address
  - Wearout failures
  - Optimal resistance/threshold mapping
    - Baseline 4LC is overly pessimistic – not usable at all
- Unique feature in TLC paper
  - Bandwidth-Enhanced writes

# MLC PCM FOR CONTINUED CAPACITY SCALING

- Major challenge: resistance drift
- Conventional 4LC PCM is not practical
  - Strong ECC and frequent refresh:
    - Performance/power penalty
    - Loose nonvolatility
- Proposal: 3LC PCM
  - Simple, genuinely nonvolatile
  - 3-ON-2 & Mark-and-spare
    - Low-cost wearout tolerance mechanism for 3LC
  - Only 7% lower capacity than (volatile) 4LC
- Generalized non-power-of-two level cells
  - 5LC, 6LC, ...