NAND Flash Basics & Error Characteristics

Oliver Hambrey

Siglead Europe – Coventry, UK

Thanks To…

- slides based on, adapted from and added to previous FMS pre-conference seminars
  - Roman Pletka: IBM Research - Zurich
  - Tom Parnell: IBM Research - Zurich

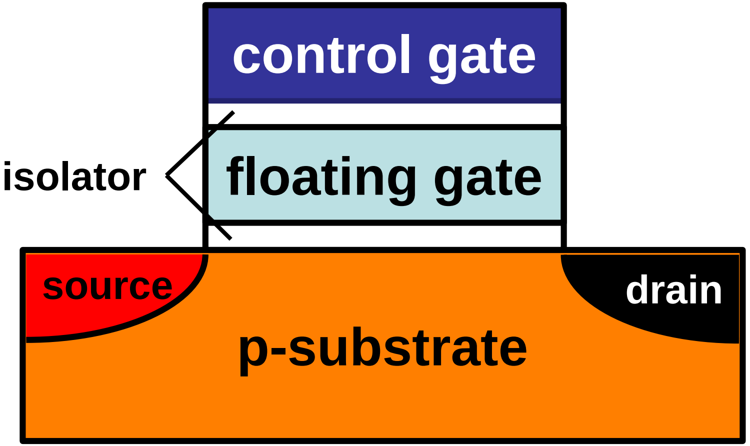
Agenda

- NAND flash basics
  - floating gate transistor
  - read/program/erase mechanisms
  - organization
  - multilevel cell NAND flash
- Error Characteristics
  - program/erase stress
  - data retention
  - read disturb
  - programming error

PART 1: NAND Flash Basics

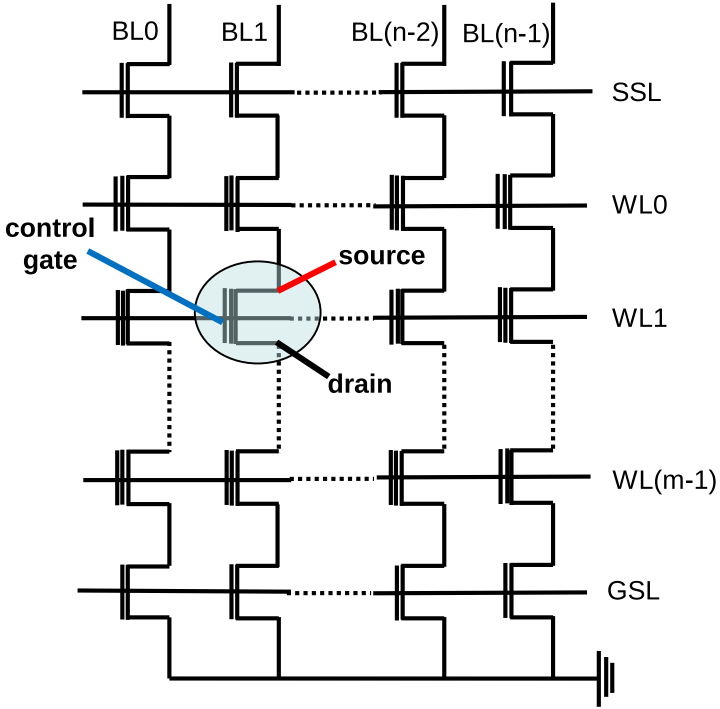
The Floating Gate Transistor (Cell)

- fundamental building block of NAND flash memories



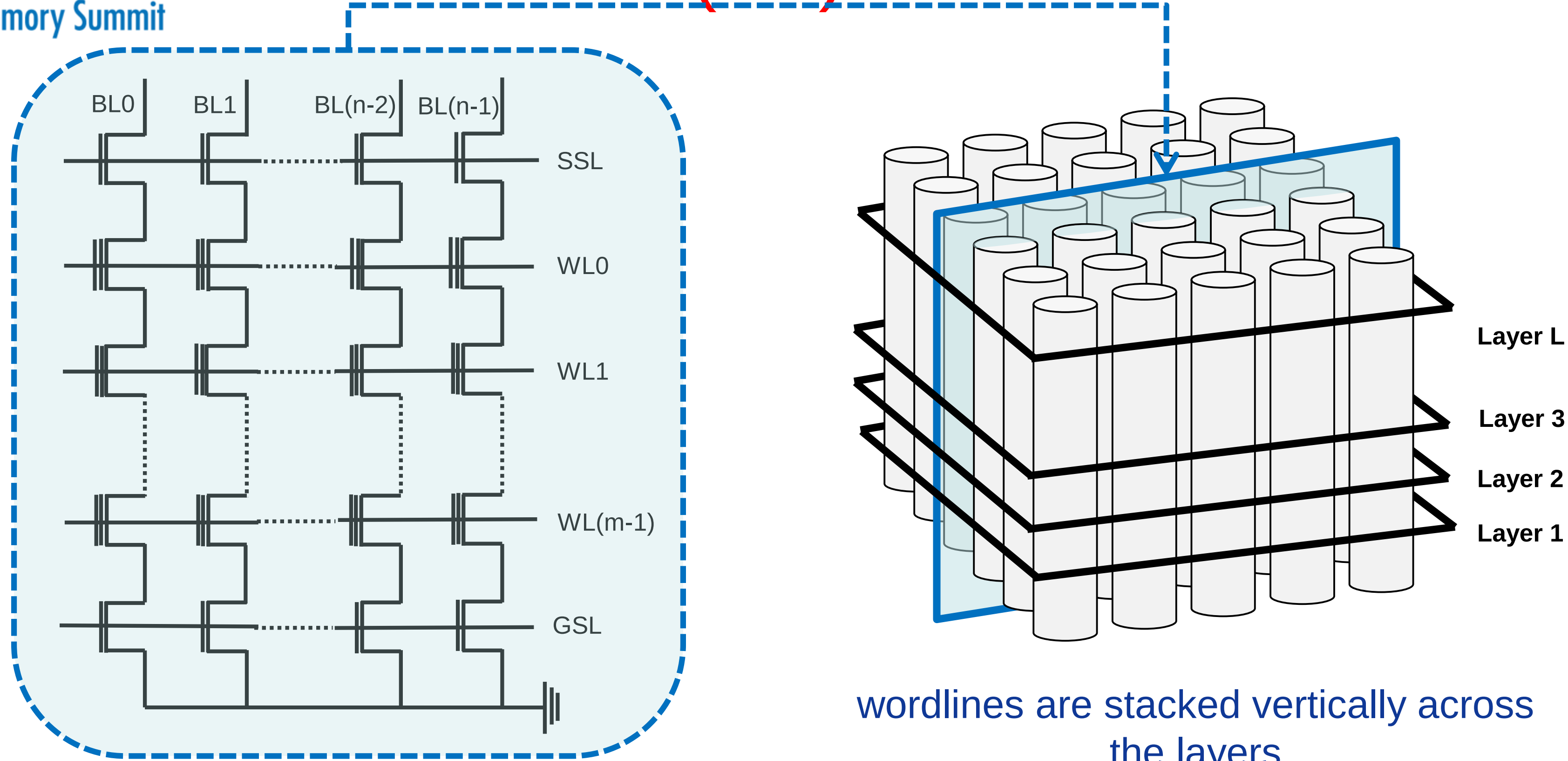
- **readability:** different conductive properties exhibited dependent on the presence of electrons in the floating gate
- **programmability:** electrons can be made to tunnel into and out from the floating gate
- **non-volatility:** electrons remain in the floating gate when the device is powered down
- **durability:** tunnelling electrons into/from the floating gate is destructive (limiting the lifespan of the device)
- **reliability:** electrons can dissipate from floating gate under high stress environmental conditions (ie high temperature)

Architecture (2D)



- **scalability:** NAND flash is divided into “blocks” of floating gate transistors (cells)
- cells within a block are connected in a grid
  - bitline (BL) connects via source/drain in series
  - wordline (WL) connects via control gates in parallel
- all cells in a wordline are programmed/read simultaneously
- all cells in the block must be erased simultaneously

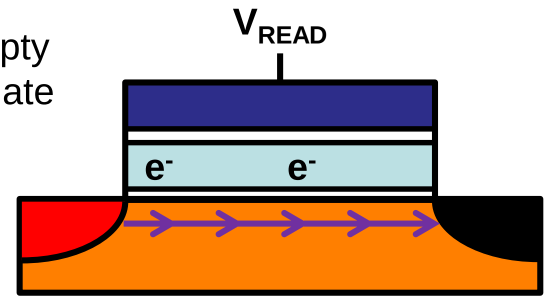
Architecture (3D)



wordlines are stacked vertically across the layers

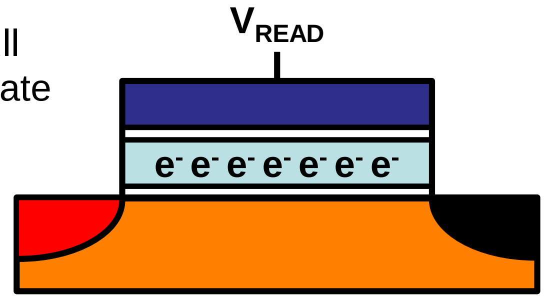
Read Mechanism (1)

“1” - empty floating gate



current flow through p-substrate

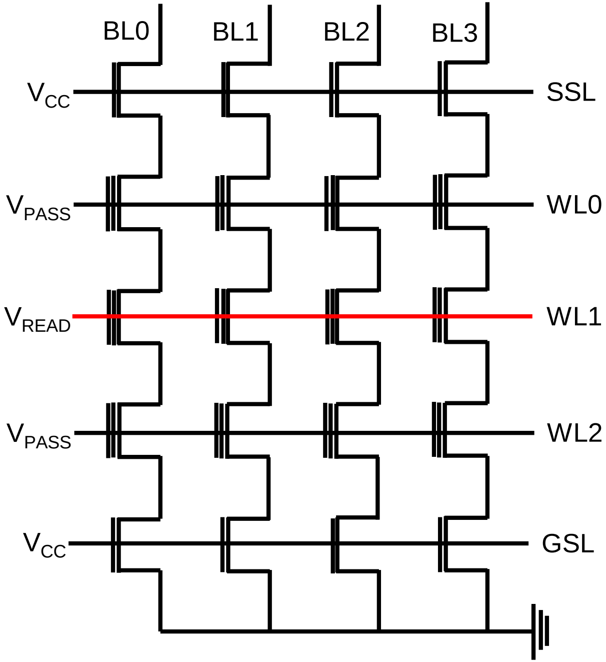
“0” - full floating gate



no current flow through p-substrate

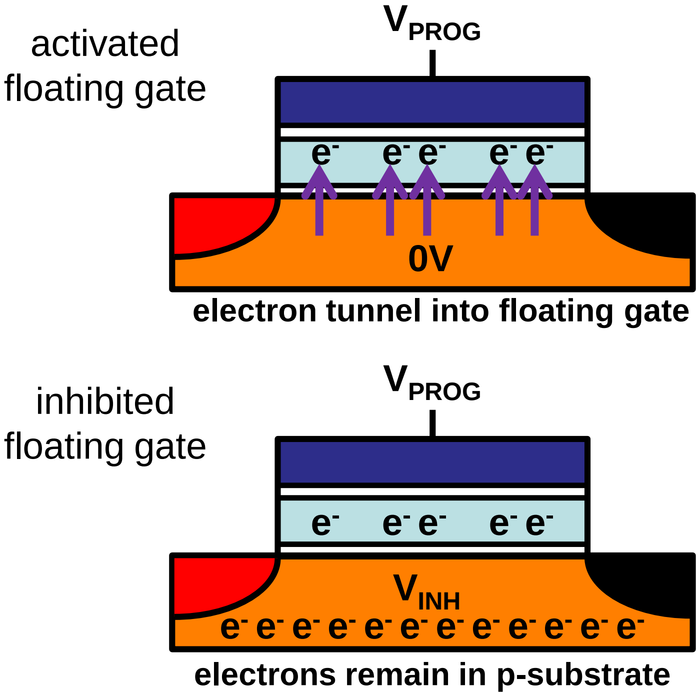
- potential V_{READ} applied to control gate
- if potential between control gate and floating is high enough, p-substrate is conductive
- floating gate “empty”
  - current flow through p-substrate
  - read as “1”
- floating gate “full”
  - no current flow through p-substrate
  - read as “0”

Read Mechanism (2)



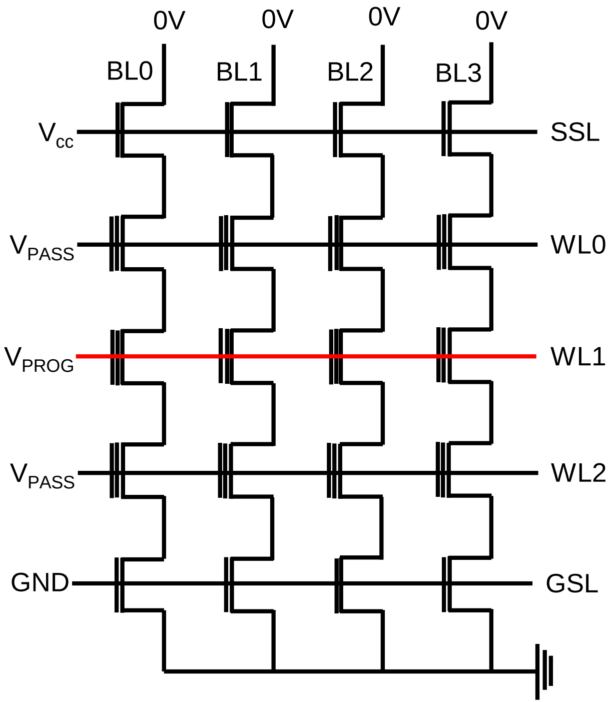
- **selected wordline:** V_{READ} is applied to control gates
  - current flows through p-substrate only if “level” of electrons in floating gate is less than V_{READ}
- **unselected wordlines:** $V_{PASS}>V_{PROG}$ is applied to control gates
  - V_{PASS} large enough so that current flows through p-substrate even if floating gates are “full”
- **result:** current flows through bitline if and only if “level” of electrons in floating gate on selected wordline is less than V_{READ}
  - read as “1” if floating gate is “empty”
  - read as “0” if floating gate is “full”
- Typical time for read: ~50μs

Program Mechanism (1)



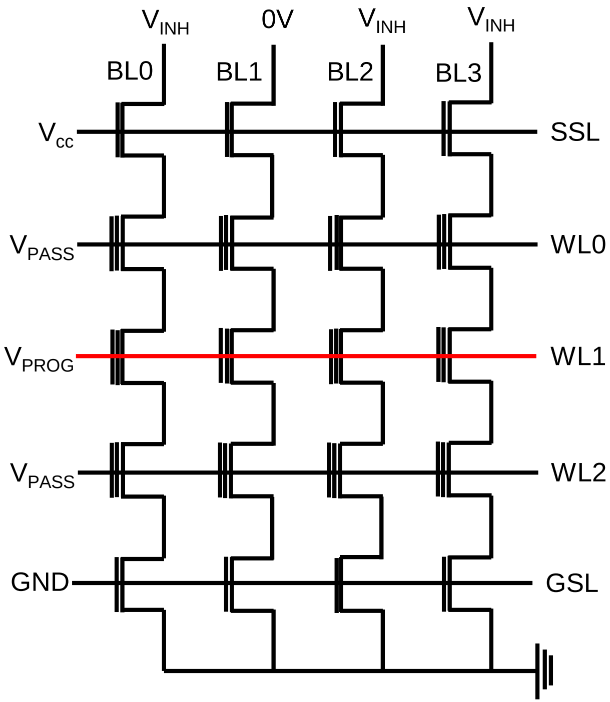
- potential V_{PROG} is applied to control gate
- if potential difference between control gate and p-substrate is large enough (~18V), electrons tunnel into floating gate
- to program (activate):
  - 0V is applied to p-substrate
  - electrons tunnel into floating gate
- to prevent program (inhibit)
  - V_{INH} is applied to p-substrate
  - electrons remain in p-substrate

Program Mechanism (2)



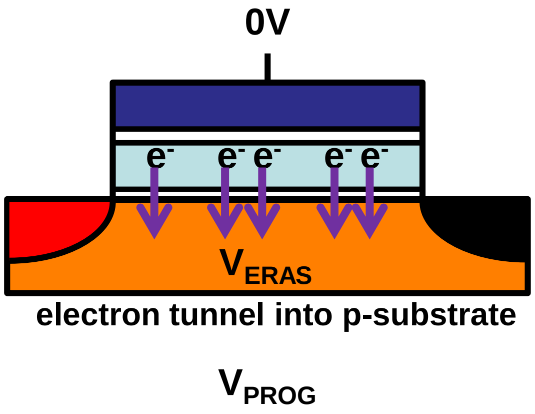
- **selected wordline:** $V_{PROG}>V_{PASS}$ is applied to control gates
  - electrons tunnel into activated floating gates on selected wordlines
- **unselected wordlines:** V_{PASS} is applied to control gates
  - V_{PASS} low enough to prevent tunnelling of electrons into floating gate
- **pulse programming:** programming occurs in short pulses
  - prevent over programming of floating gate
  - pulse duration of order microseconds

Program Mechanism (3)



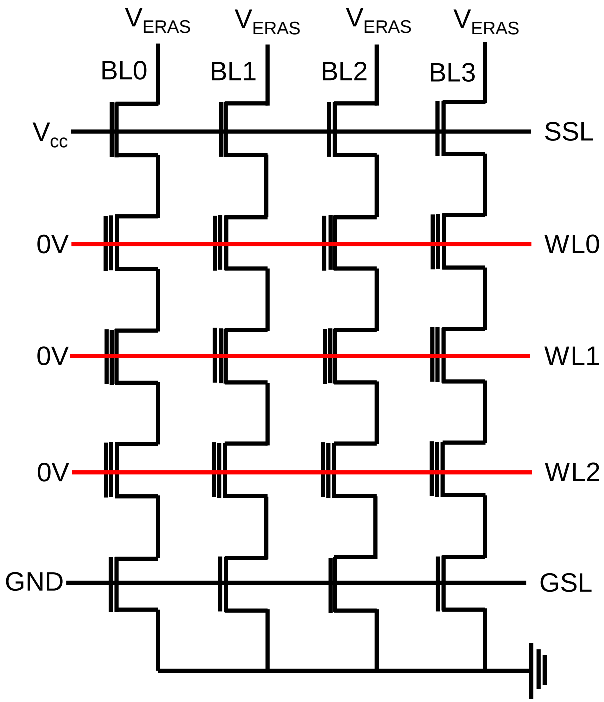
- **read verify:** before program, target wordline is read
    - if a floating gate is “full”, its bitline is inhibited
    - prevent over programming of floating gate
- **activated bitlines:** 0V is applied across bitline
    - electrons tunnel into floating gate of selected wordline
- **inhibited bitlines:** V_{INH} is applied across bitline
    - electrons remain in floating gate of selected wordline
- pulse programming continues until all cells are “full” or maximum iterations have occurred
- typical time for program: 1-10ms

Erase Mechanism (1)


0V
e$^-$ e$^-$ e$^-$ e$^-$ e$^-$
V_{ERAS}
electron tunnel into p-substrate
V_{PROG}

- potential V_{ERAS} is applied to p-substrate
- if potential difference between p-substrate and control gate is large enough (~20V), electrons tunnel out of floating gate
- to erase
  - 0V is applied across the control gate
  - V_{ERAS} is applied to p-substrate
  - electrons tunnel into p-substrate

Erase Mechanism (2)

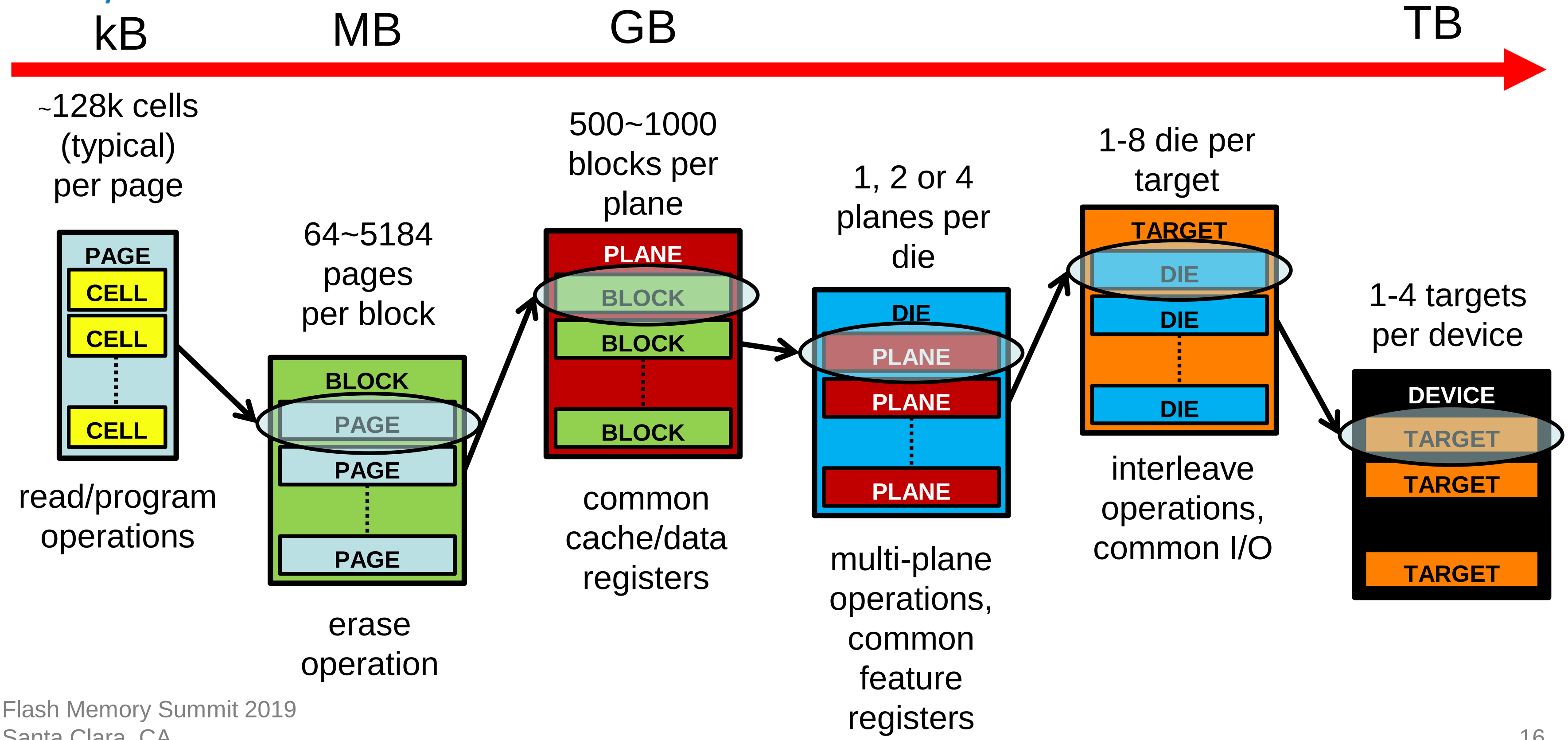


- **read verify:** before erase, block is read
  - if “most” cells are erased, erase is complete
  - otherwise, pulse erase
- **pulse erase:**
  - V_{ERAS} is applies across bitlines
  - 0V must be applied across all wordlines
  - After pulse erase, perform read verify
- **result:** entire block is erase
  - not possible to erase single cells/wordlines
- typical time for erase: 5~10ms

Program/Erase Granularity Mismatch

- an individual cell can only have its program level increased
    - electrons can be added to floating gate without “significantly” affecting other cells
    - electrons cannot be removed from floating gate without erasing all cells in the block
- “overwriting” not possible
    - must erase entire block first, then program wordline
    - but what happens to all of the wordlines I don’t want to rewrite?

Device Organization



kB MB GB TB

~128k cells (typical) per page

PAGE
CELL
CELL
CELL

read/program operations

64~5184 pages per block

BLOCK
PAGE
PAGE
PAGE

erase operation

500~1000 blocks per plane

PLANE
BLOCK
BLOCK
BLOCK

common cache/data registers

1, 2 or 4 planes per die

DIE
PLANE
PLANE
PLANE

multi-plane operations, common feature registers

1-8 die per target

TARGET
DIE
DIE
DIE

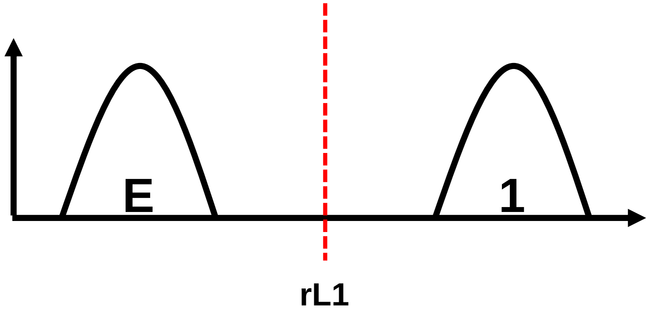
interleave operations, common I/O

1-4 targets per device

DEVICE
TARGET
TARGET
TARGET

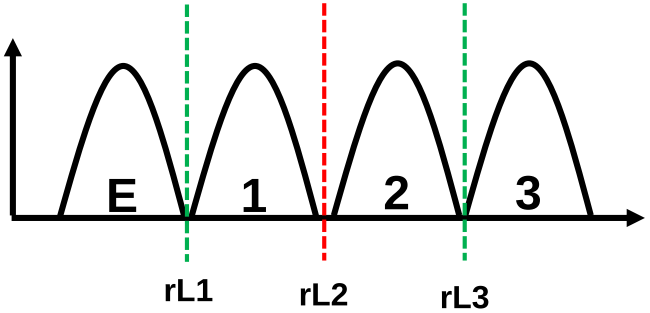
Multi Level NAND Flash (1)

- Single Level Cell (SLC)
  - erase state
  - 1 program state



- 1 read needed to get 1 bit of information per cell

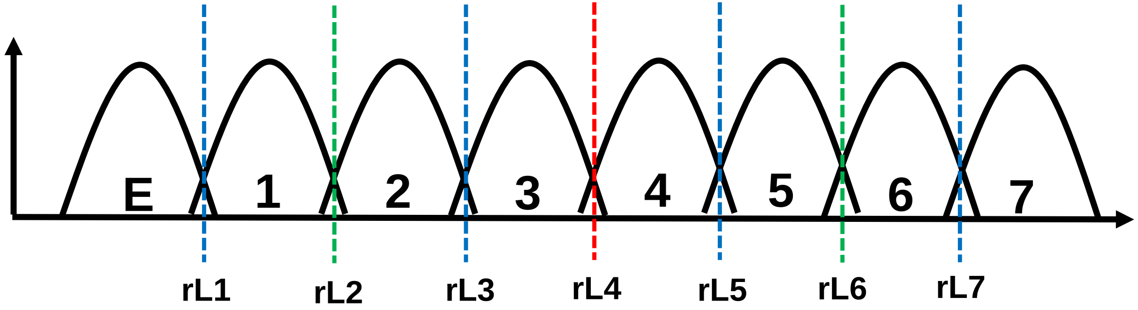
- Multi Level Cell (MLC)
  - erase state
  - 3 program states



- 3 reads needed to get 2 bits[10] of information per cell

Multi Level NAND Flash (2)

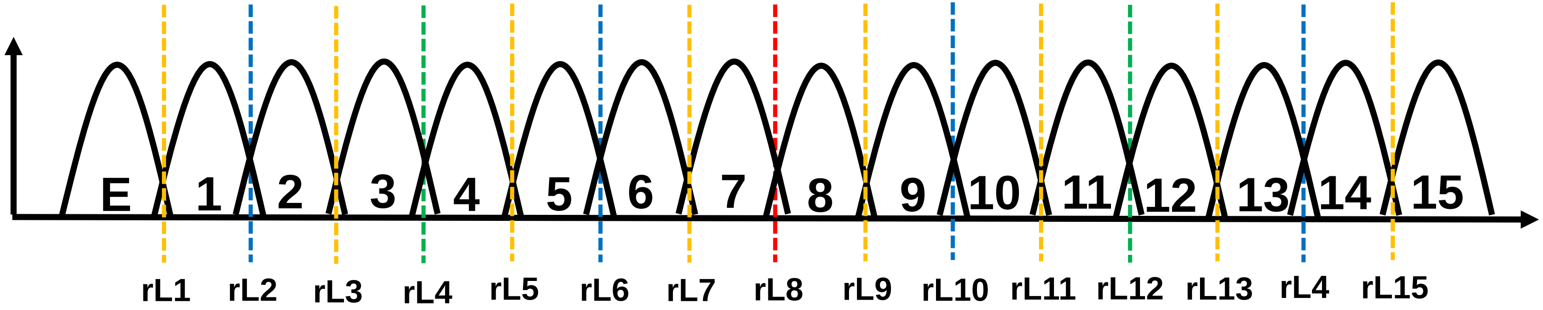
- Triple Level Cell (TLC)
  - erase state
  - 7 program states



- 7 reads needed to get 3 bits of information per cell

Multi Level NAND Flash (3)

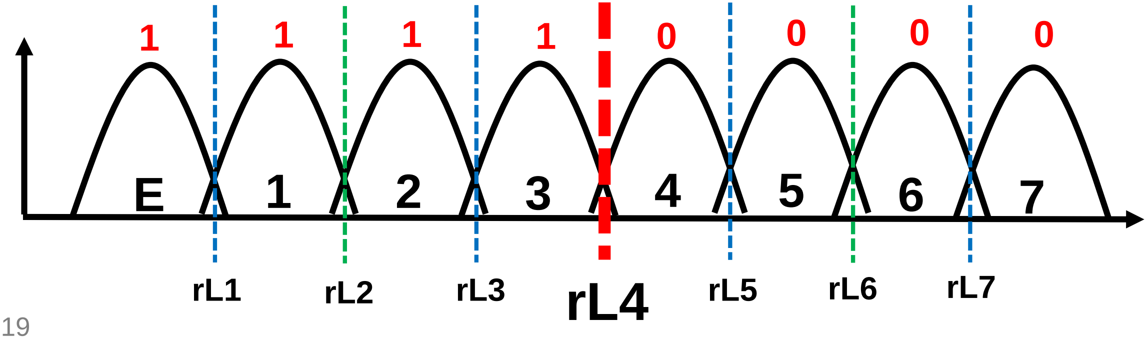
- Quadruple Level Cell (QLC)
  - erase state
  - 15 program states



- 15 reads needed to get 4 bits of information per cell

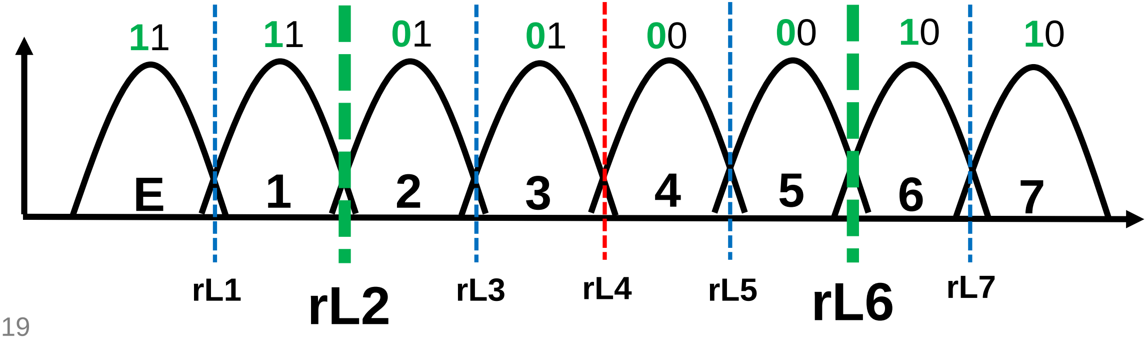
Gray Code (1)

- how many reads needed to get 1 bit of information from M/T/QLC NAND?
- **traditional approach:** Gray Code
  - lower page: rL4 – 1 read



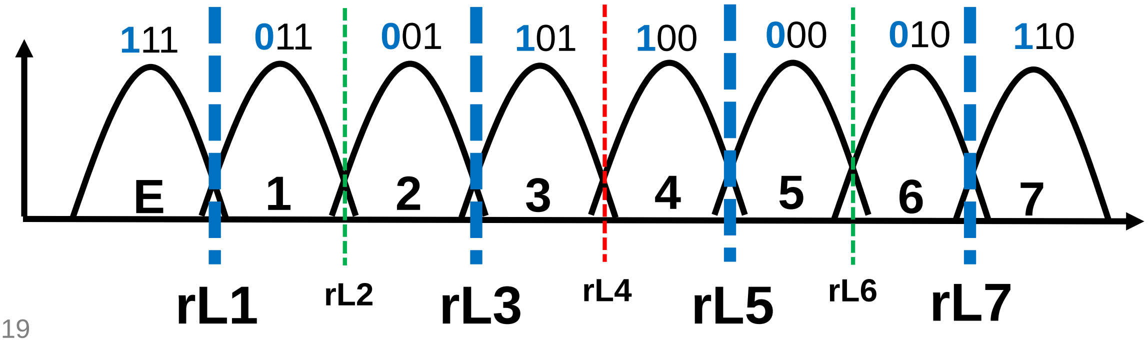
Gray Code (2)

- how many reads needed to get 1 bit of information from M/T/QLC NAND?
- **traditional approach:** Gray Code
  - middle page: NXOR(rL2,rL6) – 2 reads



Gray Code (3)

- how many reads needed to get 1 bit of information from M/T/QLC NAND?
- **traditional approach:** Gray Code
  - upper page: NXOR(rL1,rL3,rL5,rL7) – 4 reads

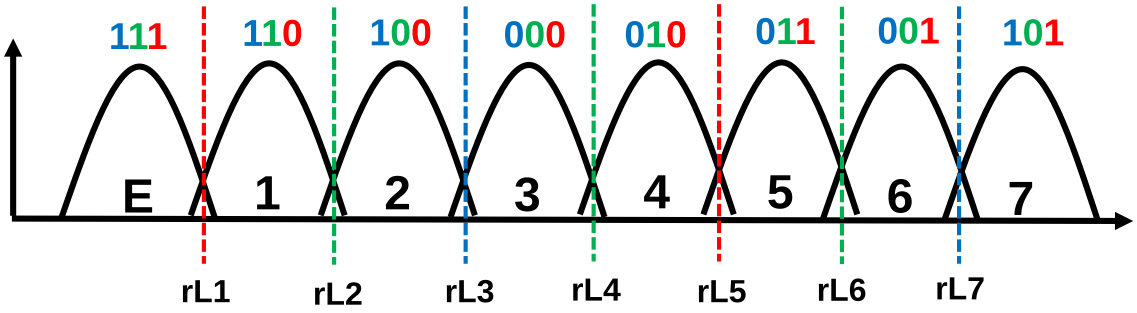


Gray Code (4)

- how many reads needed to get 1 bit of information from M/T/QLC NAND?
- **traditional approach:** Gray Code
- **minimum:** 1 read
- **maximum:** 2^{k-1} reads (for 2^k level NAND flash)
  - this does not scale well
  - significant variation in read times between page types

Beyond Gray Code

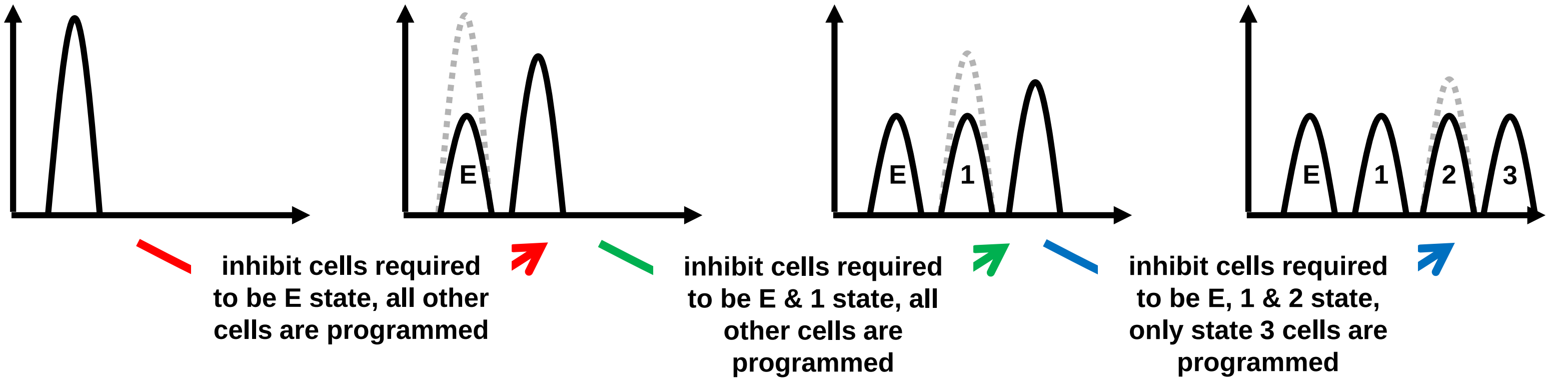
- other assignments exist that look to use similar number of reads for all page types
  - lower page: NXOR(rL1,rL5) – 2 reads
  - middle page: NXOR(rL2,rL4,rL6) – 3 reads
  - upper page: NXOR(rL3,rL7) – 2 reads



- in such assignments, all pages scale like $\sim\frac{2^k}{k}$ reads for 2^k level NAND

Multilevel Cell Programming (1)

- one shot programming:
  - pass in data for all pages (lower/middle/upper/top)
  - program once



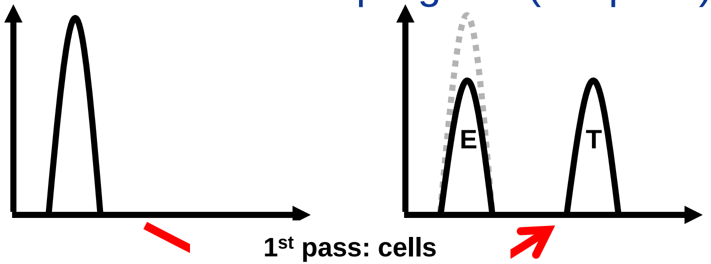
inhibit cells required to be E state, all other cells are programmed

inhibit cells required to be E & 1 state, all other cells are programmed

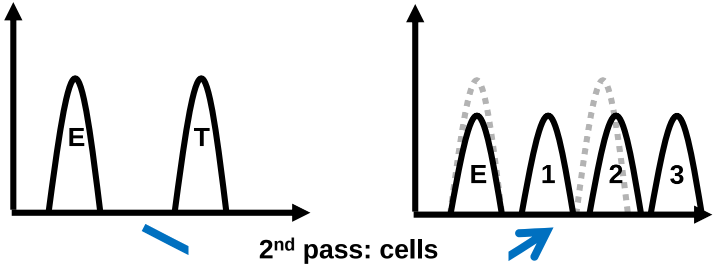
inhibit cells required to be E, 1 & 2 state, only state 3 cells are programmed

Multilevel Cell Programming (2)

- multipass programming
  - pass in data for some (or all) pages
  - make intermediate program (1st pass)
  - pass in data for remaining (and previous) pages
    - (possibly) read intermediate program to “fill in the gaps”
  - make final program (2nd pass)



1st pass: cells required to be state E & 1 are inhibited. Other states are programmed to some intermediate (T) state

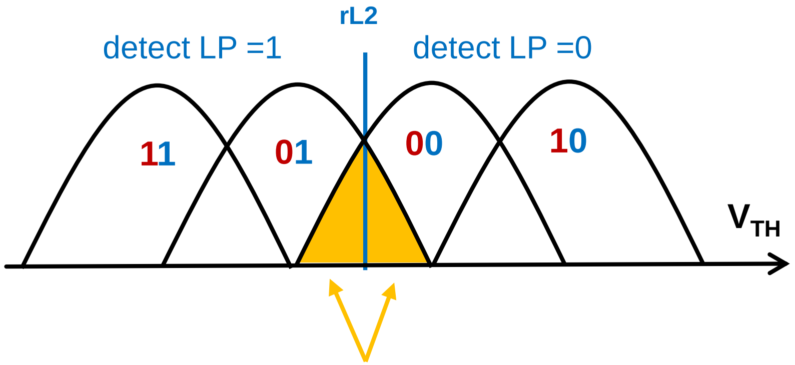


2nd pass: cells required to be E & 2 are inhibited. Cells required to be 1 & 3 are programmed

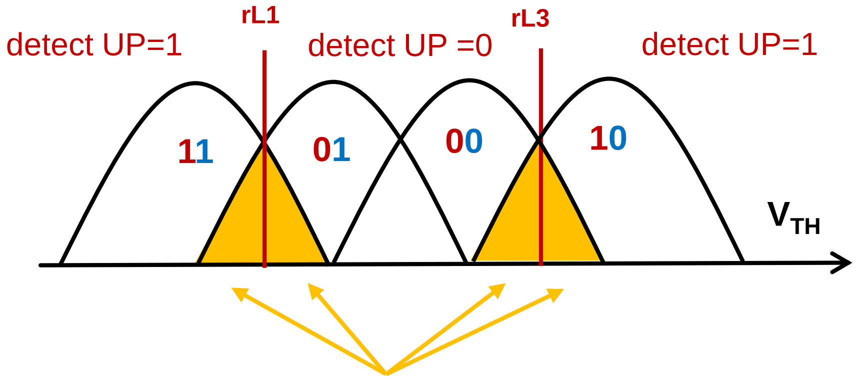
PART 2: Error Characteristics

Read Errors

- broadening of V_{TH} distributions due to noise can lead to read error
- what are the main sources of noise?



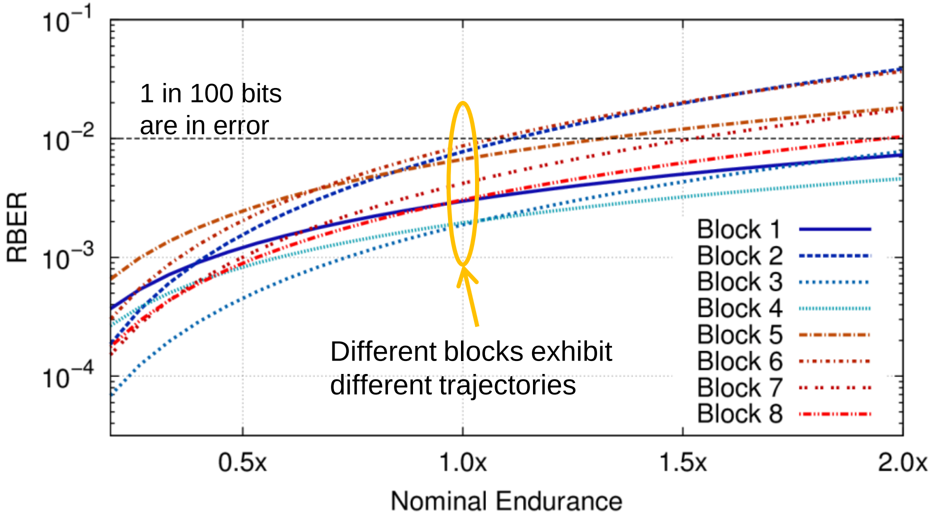
Lower page read errors



Upper page read errors

Program/Erase Cycling Stress

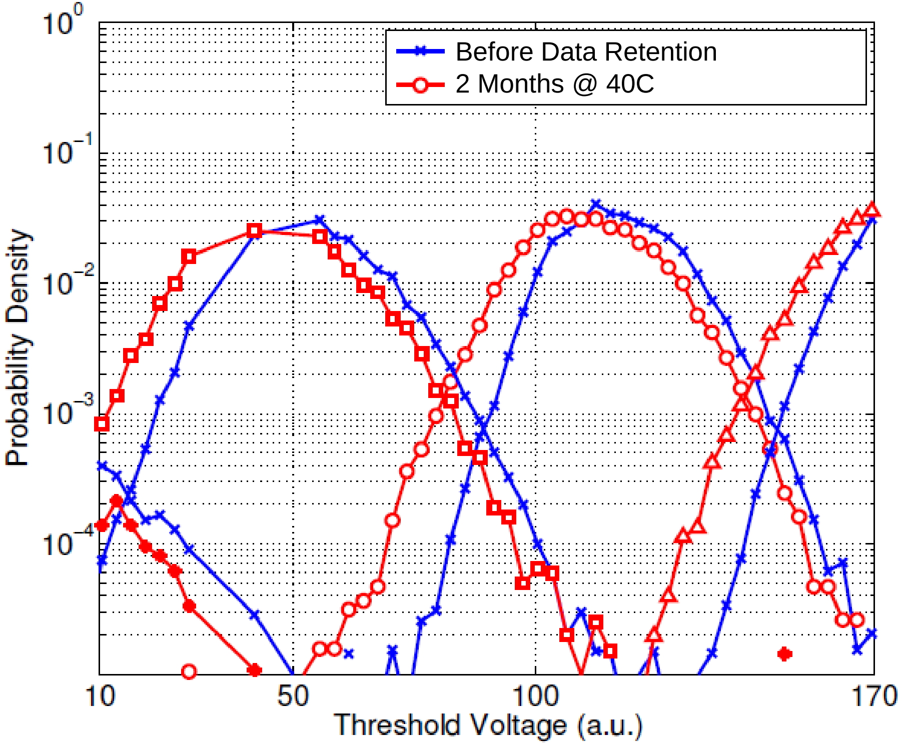
RBER of different flash blocks in the same device as a function of P/E cycles



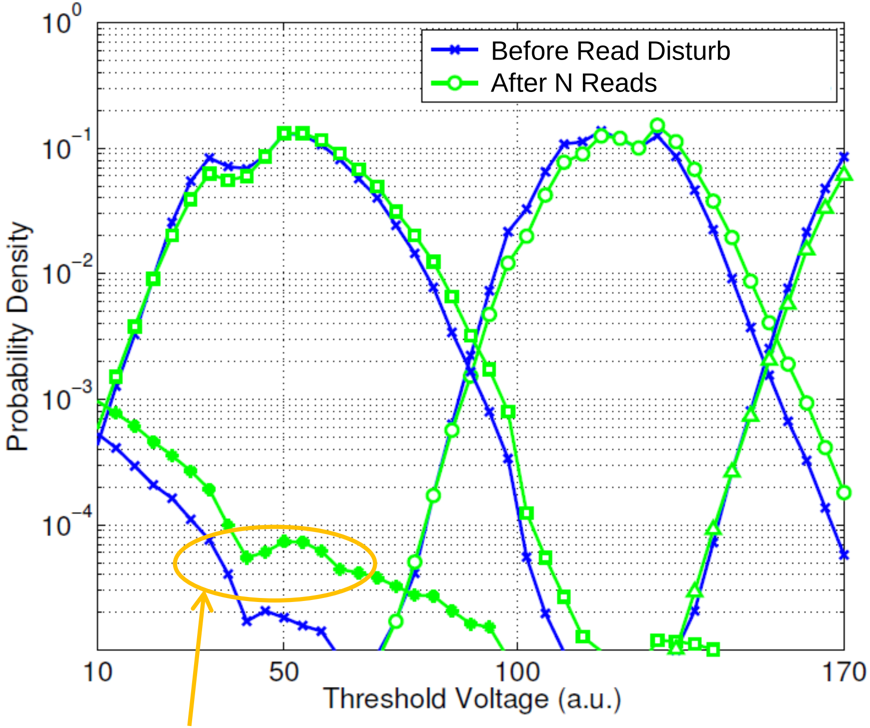
- repeated application of program/erase (P/E) pulses leads to degraded reliability of the underlying NAND flash cells
- the measured raw bit error rate (RBER) increases as a function of P/E cycles
- low RBER at early life does not indicate a good block, and an early high RBER not a weak one!
- strong error-correction codes must be implemented on the controller to be able to deal with increased RBER

Data Retention

- over time electrons can "escape" from the programmed flash cells, causing a loss of threshold voltage
- this can cause a large increase in RBER unless the read voltage is shifted to compensate for charge loss
- the data retention effect is temperature dependent (charge escapes faster at higher temperature)



Read Disturb

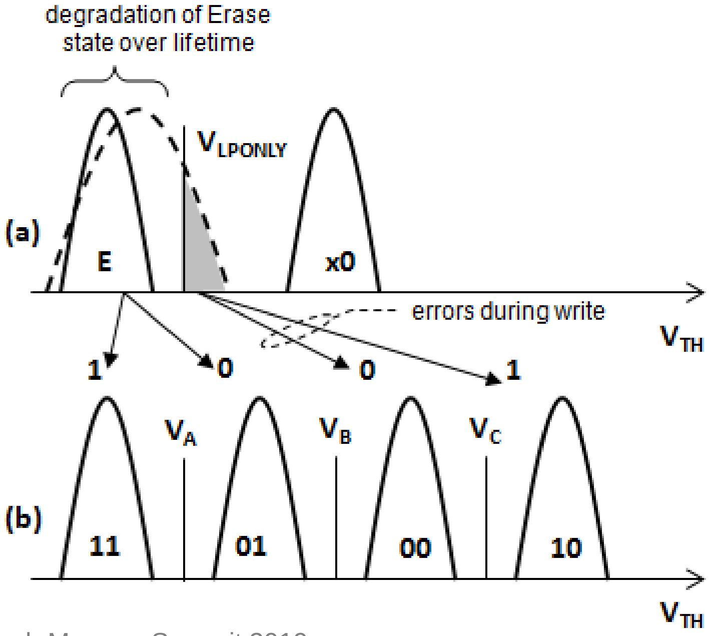


Dominant effect of read disturb is seen on Erase state

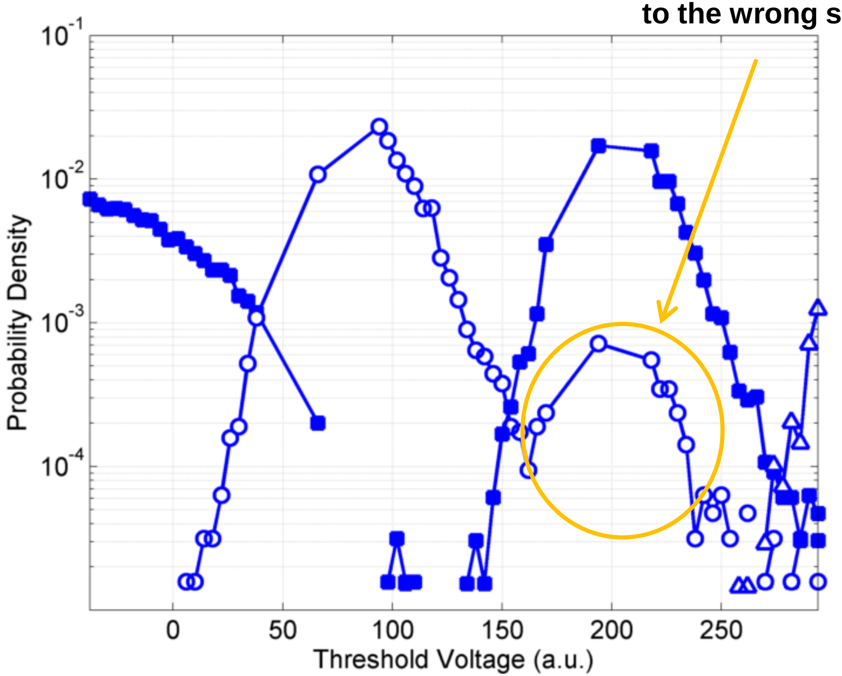
- When reading a particular page in a block of NAND Flash, a pass through voltage is applied to all other WL in order to “deselect” them
- This applied voltage can affect the V_{TH} distributed of the unselected WLs
- If a block is read from too many times, the RBER will increase to a point that the ECC is no longer able to correct
- The controller must be able to manage such effects

Programming Errors

- Degradation of erase state can cause error propagation during the two-pass programming procedure → switch to 1-pass?



Cells are programmed to the wrong state!



Want to do your Own Investigations?

- Siglead NAND Analyzer series
- compatibility up to & including latest 128 layer TLC/QLC 3D NAND devices
- test multiple packages in parallel
- suitable for
  - NAND error characterization
  - NAND busy time characterization
  - NAND sorting, acceptance/rejection
  - data recovery
  - erasure verification
  - and more…
- visit www.siglead.com for more info

SigNAS3

SigNASII

end