Accelerating Flash Memory Performance by Speculative Early Sensing Decision

Amit Berman and Yitzhak Birk

Technion – Israel Institute of Technology
{bermanam@tx, birk@ee}.technion.ac.il

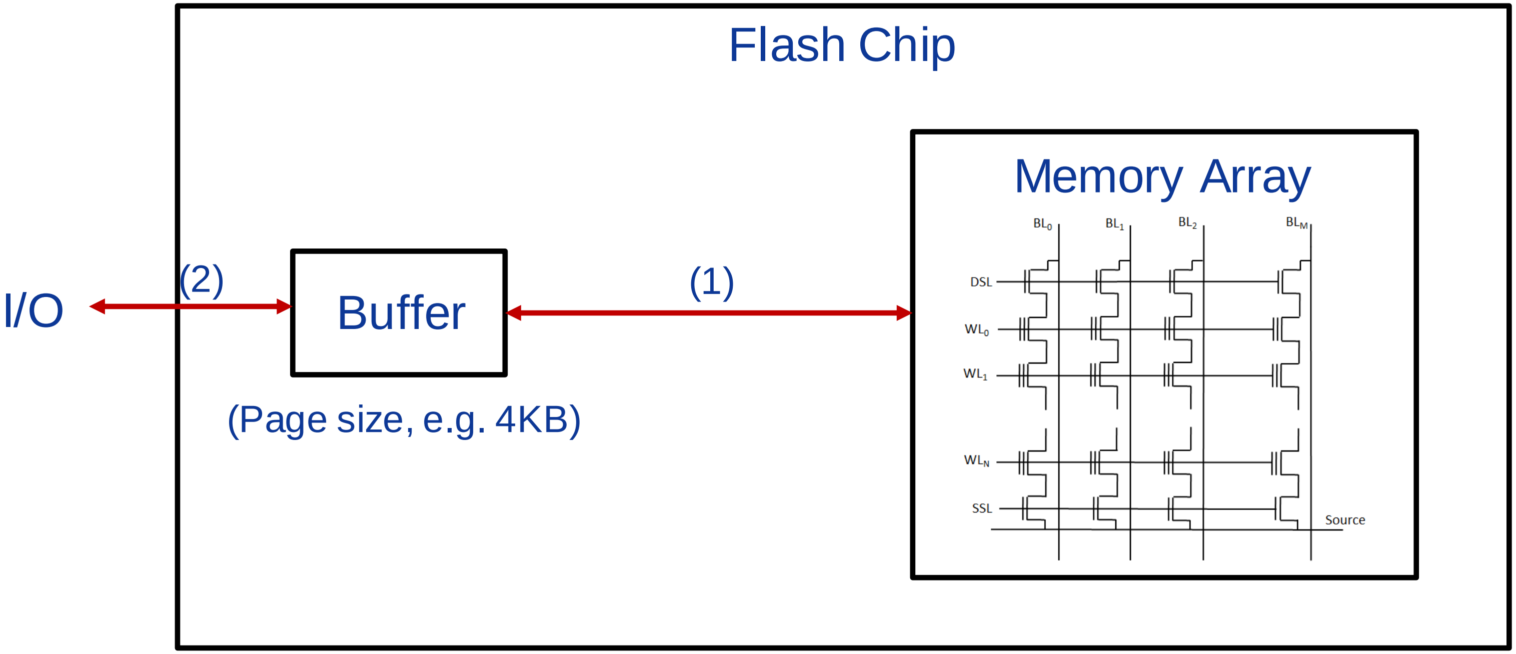
Goal

- Reduce the duration of:
  - Page Read
  - Page Program
  - Block Erase

for both NAND and NOR Flash Memories

Flash Memory Read

- Read comprises two stages:
  1) Data transfer from memory array to buffer
  2) Data transfer from buffer to I/O pins



Flash Chip

Memory Array

I/O

(2)

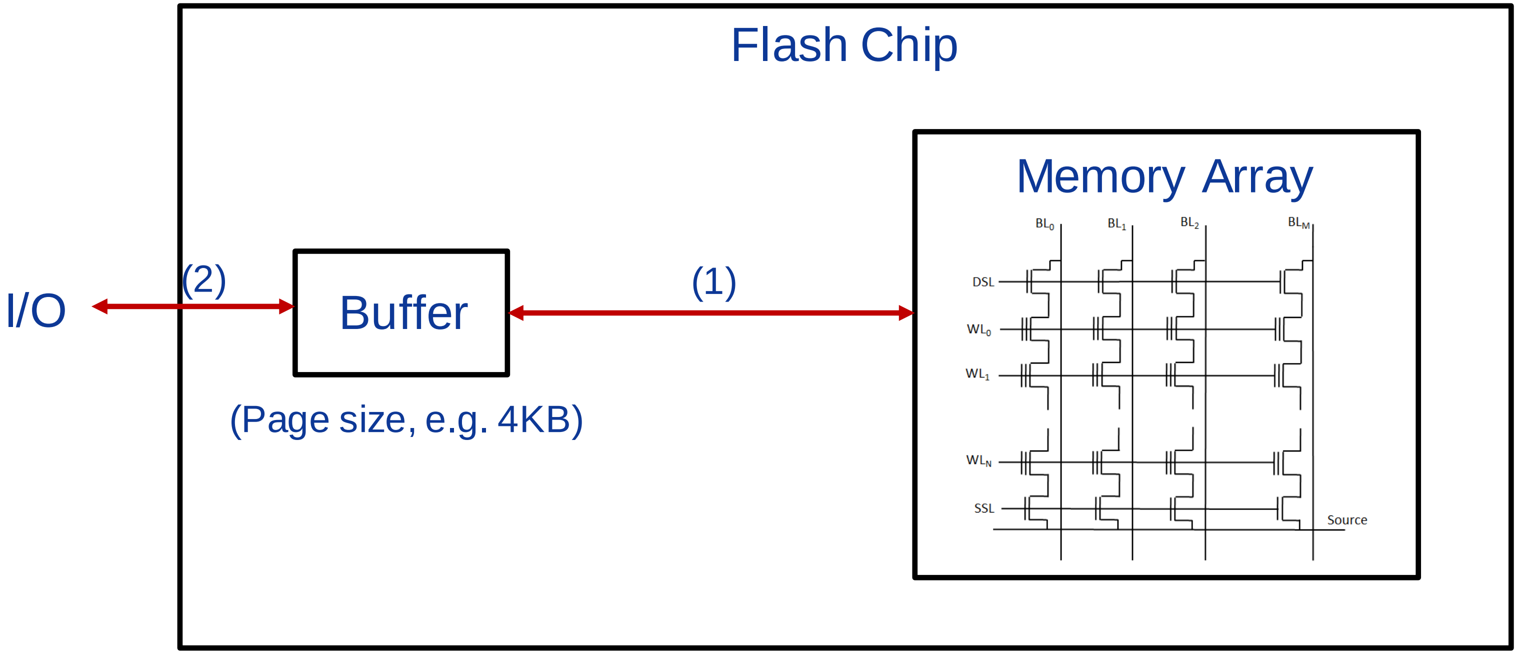
Buffer

(1)

(Page size, e.g. 4KB)

Read Duration

- Duration of those stages:
  1) Mem array → buffer: 25µs - 50µs
  2) Read cycle time: t_{RC}=20ns (50 MB/s with 8-bit interface)



Flash Chip

Memory Array

I/O

(2)

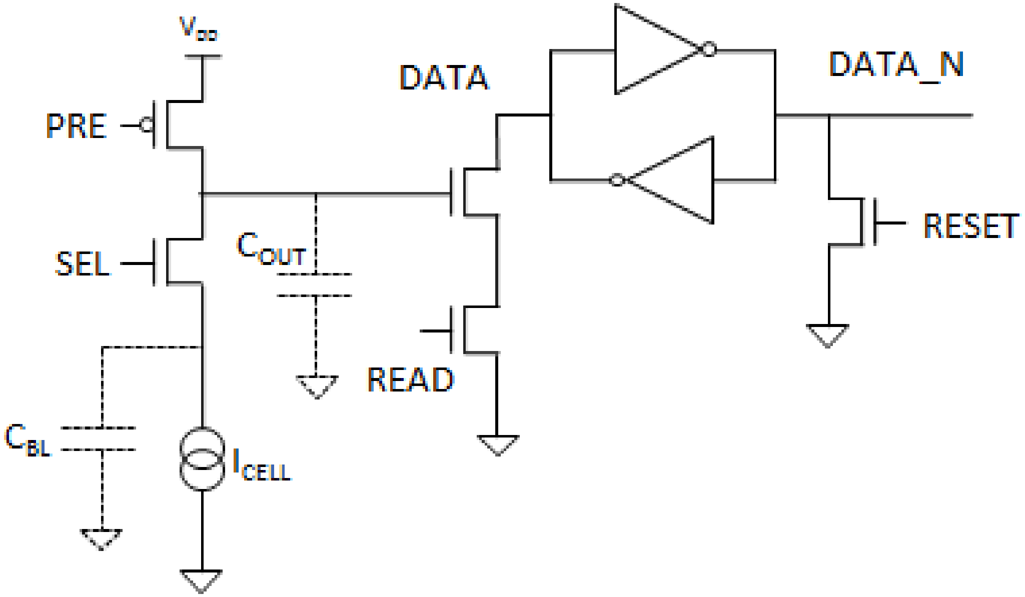
Buffer

(1)

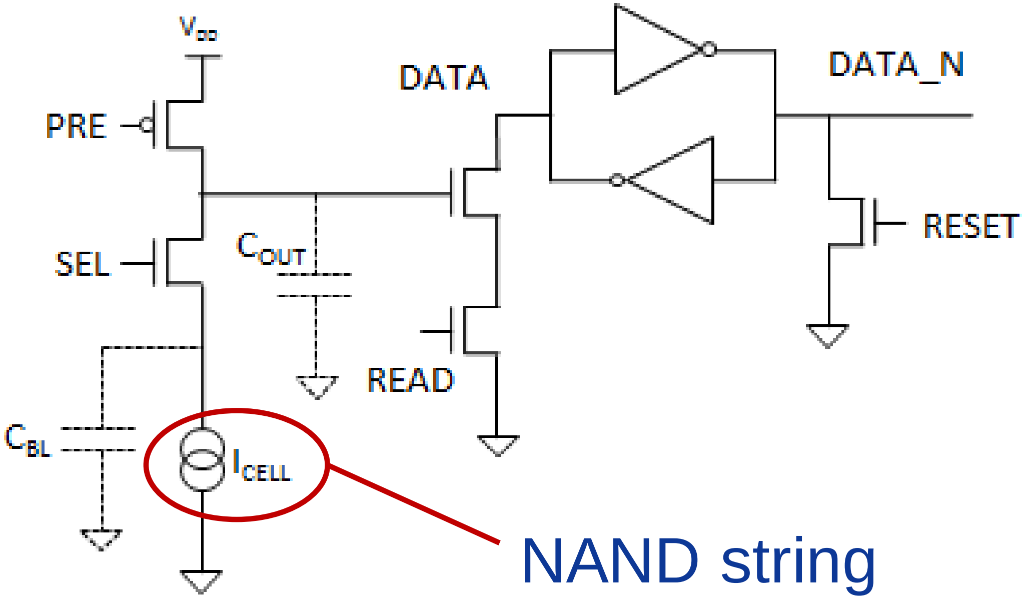
(Page size, e.g. 4KB)

Accelerating Array Access

Flash Read Circuitry



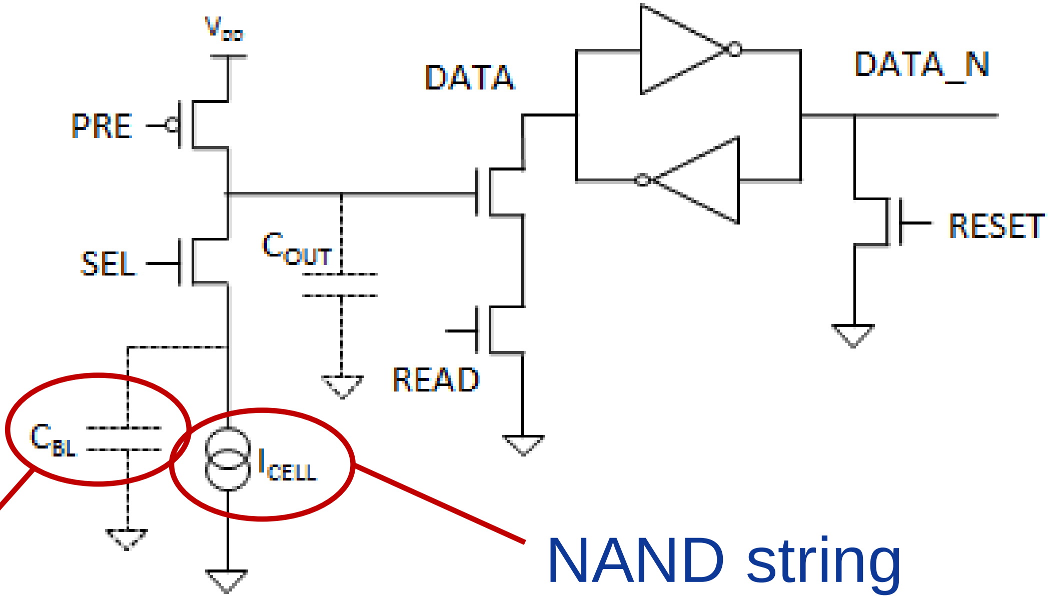
Flash Read Circuitry



NAND string

- NAND string is modeled as a current source:
  - I_{CELL} depends on cell voltages
  - Other cells act as pass transistors

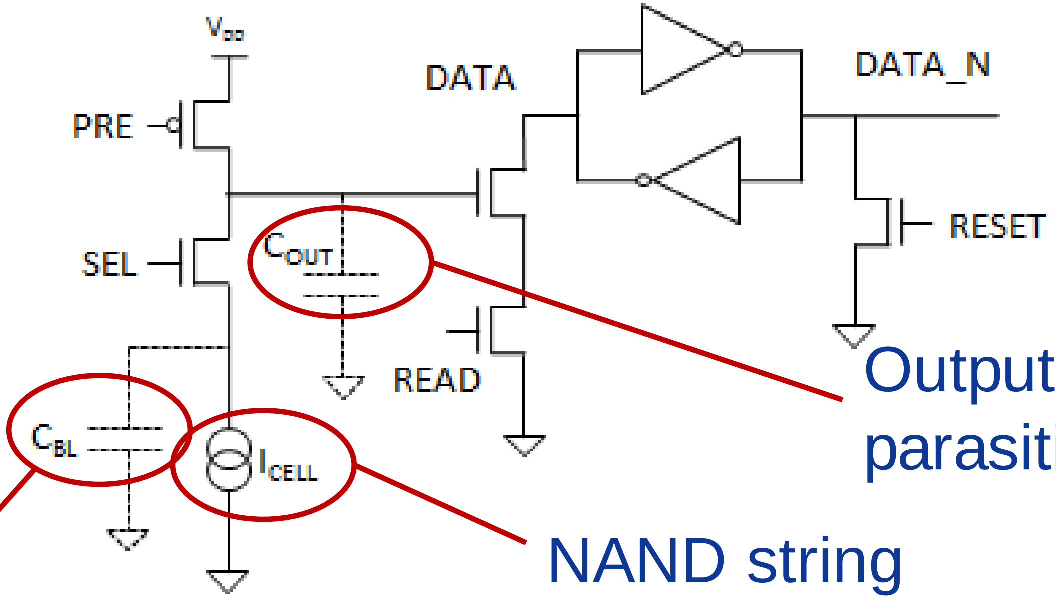
Flash Read Circuitry



NAND string

Bitline parasitic capacitance

Flash Read Circuitry



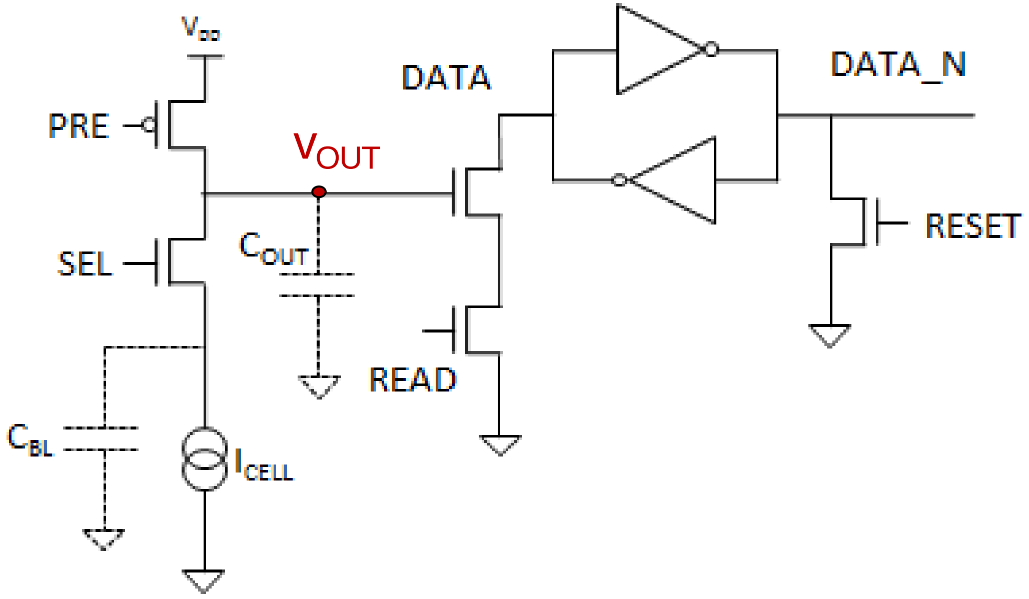
Output node parasitic capacitance

NAND string

Bitline parasitic capacitance

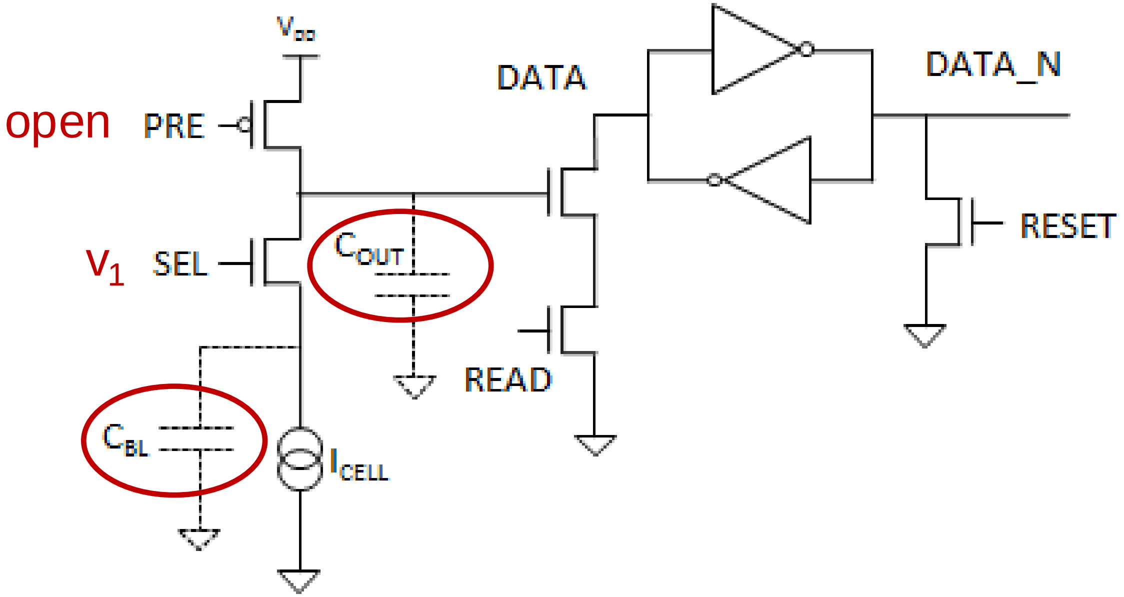
Flash Read Circuitry

- Array access has two phases:
  1) Precharge of parasitic bitline capacitance
  2) Evaluation of V_{OUT}



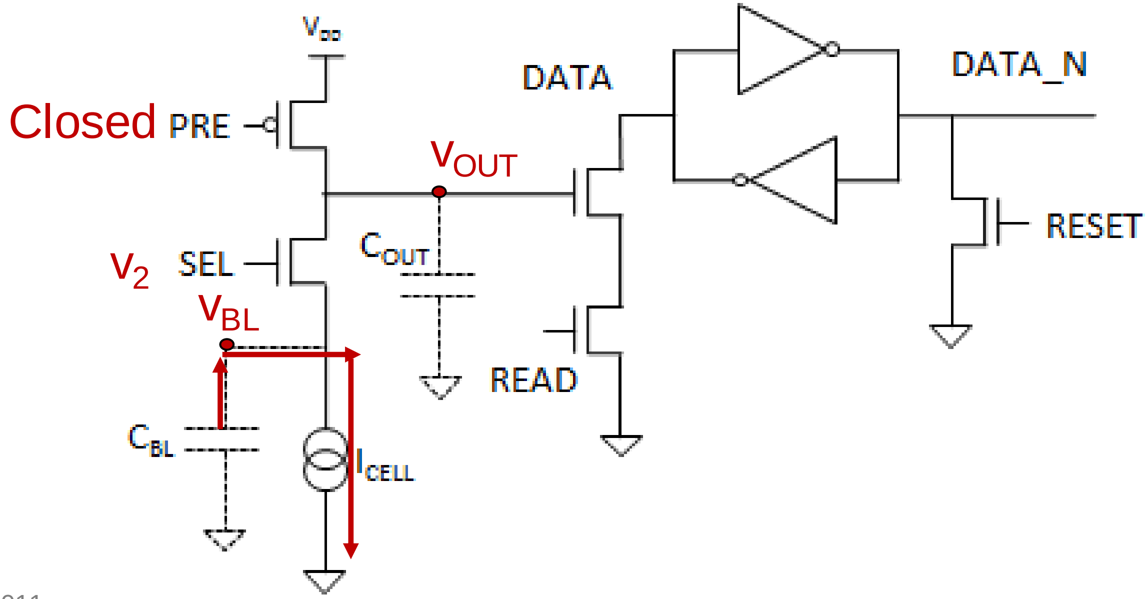
Precharge

- C_{BL} and C_{OUT} are charged
  Takes about 5μs



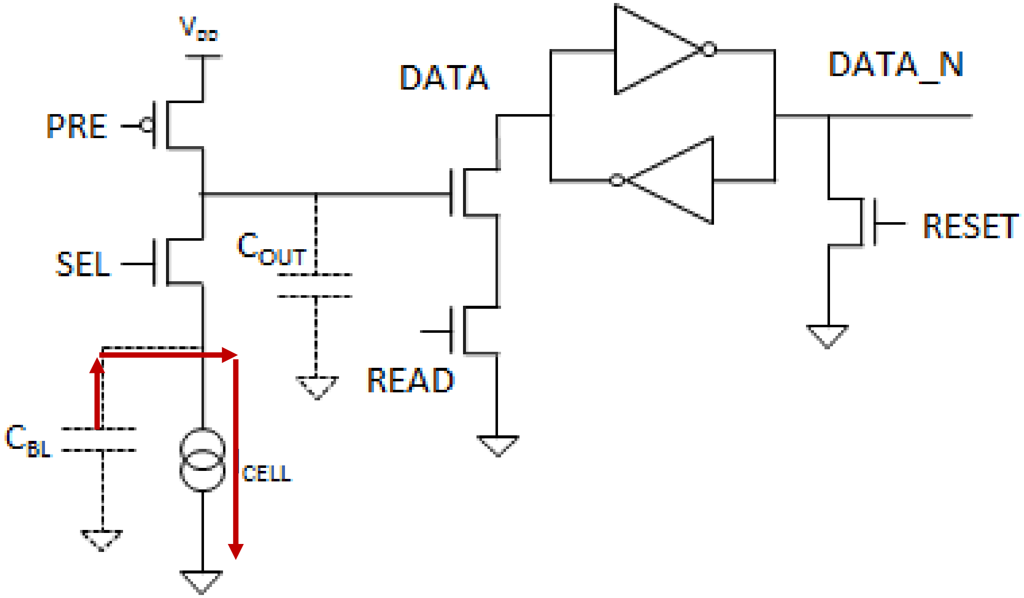
Evaluation

- Read voltage V_{REF} is applied to the cell
  - If $V_t < V_{REF}$ then C_{BL} is discharged (I_{CELL}=100-200nA) and V_{OUT} changes to V_{BL} (eventually ~0)
  - Else, C_{BL} retains its charge (I_{CELL}=0, V_{OUT} unchanged)
  - Decision based on V_{OUT}



Duration of the Evaluation phase

- If $V_t < V_{REF}$ then $$T_{EVAL} = \frac{C_{BL}(V_1 - V_2)}{I_{CELL}}$$
- Else – Immediate (V_{OUT} is unchanged)



Do we really need all this time?

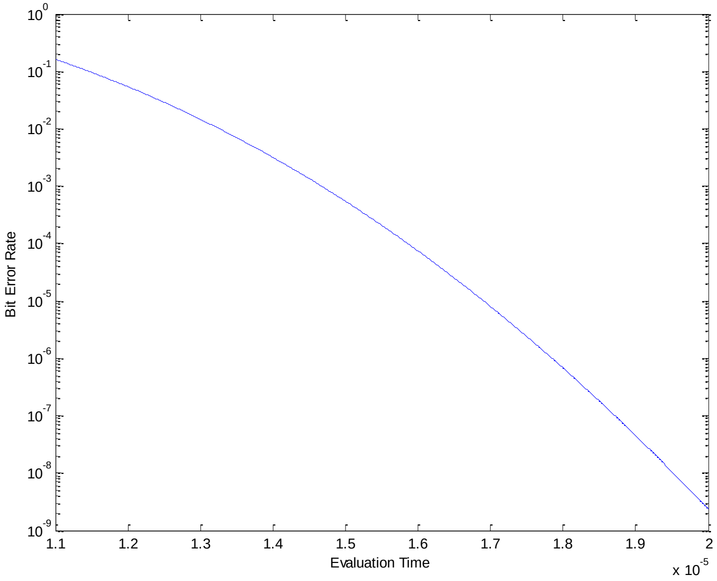
Evaluation Time – a Closer Look

- I_{CELL} and C_{BL} vary due to process variations
  - T_{EVAL} is determined by the worst-case cell
  - Typical worst case: 15µs - 45µs

$$T_{EVAL} = \frac{C_{BL}\left(V_1 - V_2\right)}{I_{CELL}}$$

Our Approach: Early Decision

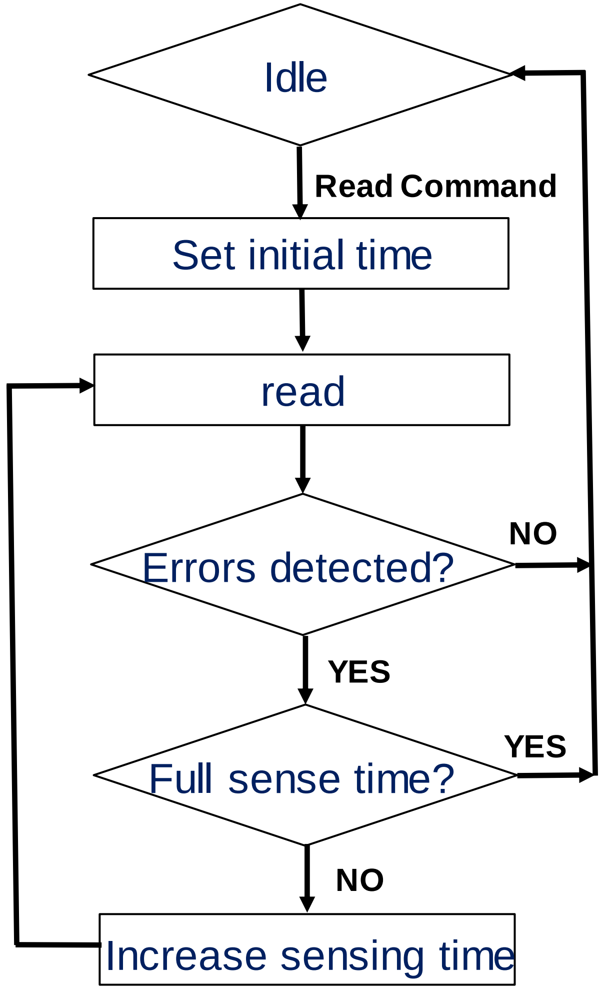
- Problems:
  - Speculation! may result in read errors
  - Retry penalty
- Challenges:
  - Reduce mean read time
  - Detect and correct resulting read errors



Simulation with various I_{CELL} and C_{BL}

Read Algorithm

A good choice of access time reduces mean Read time



Example

- Full read time:

$$T_{READ} = T_{PR} + T_{EVAL} = 25\mu S$$

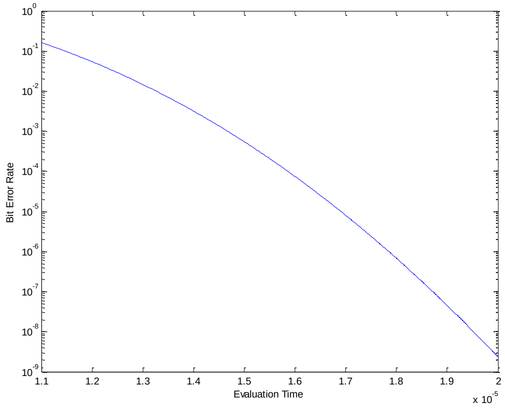
- Speculative early sensing
  - Reduces T_{EVAL} from 18µs to 13µs
  - Error probability P_e increases from 10^{-6} to 10^{-2}

$$T_{SER} = T_{PR} + \left(T_{PEVAL} + T_{ED}\right)\left(1 - P_e\right) + T_{EVAL} P_e$$

$$7 + 13 \cdot 0.99 + 18 \cdot 0.01 = 20\mu S$$

$$Speedup = \frac{T_{READ}}{T_{PS}} = 1.25$$

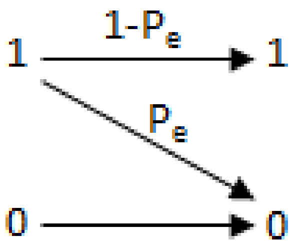
Improvement of 25%

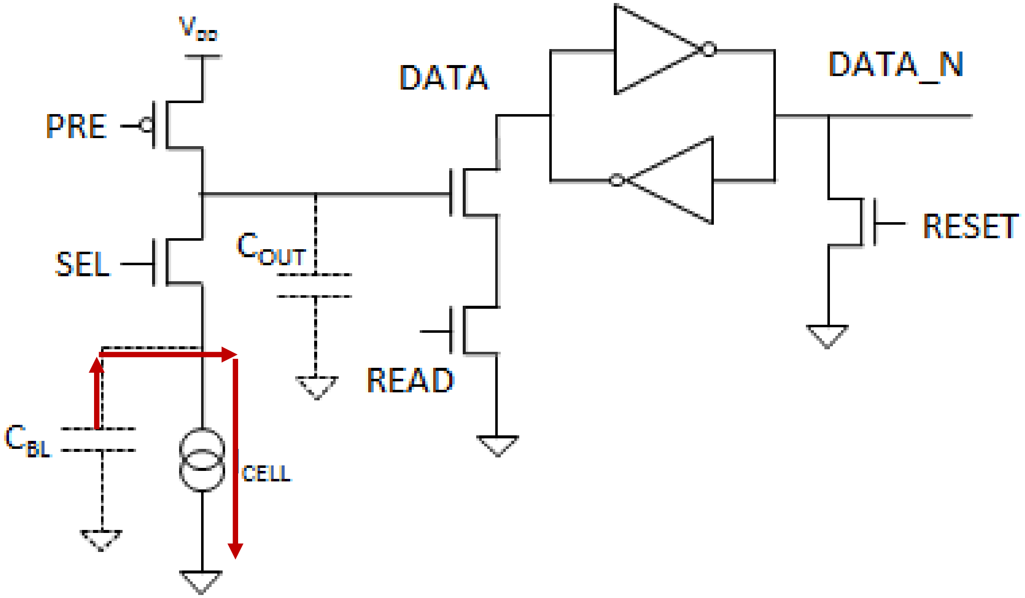


What about errors?

Unique Error Behavior

- Challenge: detect and correct resulting read errors
- Errors resulting from premature decisions are uni-directional: over estimation of cell’s charge
- If SLC (erased=1):





- In MLC: true with respect to any reference voltage

Guaranteed Error Detection

- Required: An error detection code that detects any number of errors
- For SLC (Berger Code)
  - During Program
    - Compute signature: number of zeros in the data
    - Store data and signature
  - During read
    - Read data and signature (estimates!)
    - Compute (accurate) signature of estimated data
    - Compare signatures
    - If equal - done, else error detected
- For MLC: apply same for each V_{REF}

Error Detection

- Error Detection Code
  - Systematic
  - Signature = number of zeros in the data
- (Reminder: all errors are 1→0)
- Data errors – more zeros in data
- Error in signature – more zeros in signature (→ fewer zeros in data → Contradiction)
- Signature matches data iff both are error free

$$\underbrace{11010101}_{\text{Data}} \qquad \underbrace{011}_{\text{Signature: Log}_2\text{(N)+1 bits}}$$

Program Acceleration

- Shorten the verify operation for each program increment

- Cell charge is never underestimated
  ⇒ We never over-charge

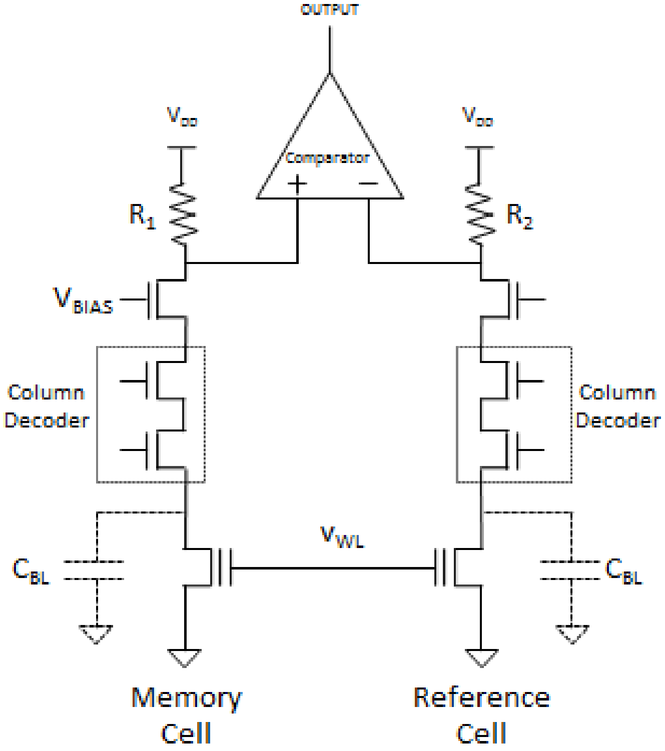
- Once target level has apparently been reached, perform full-time verify
  ⇒ resulting charge level is accurate

Erase Acceleration

- Erase stages:
  - Program all cells to high charge (apply Program acceleration)
  - Apply Erase Pulses (pulse+verify)
  - Post-erase repair (apply Program acceleration)
- Challenge: accelerate Erase-pulse phase without causing “over erase” (We may over-estimate charge and apply extra erase pulses!)
- Solution:
  - Set erase target higher (more charge)
  - Once reached, do full-duration verify
  - …

NOR Flash

- The concept is also valid for NOR Flash
  - Read has precharge and evaluation phases
  - Errors are uni-directional



Conclusions

- Key idea: speculative early decision + guaranteed error detection (and correction by retry)
- Applicable to Read, Program and Erase
- Advantages:
  - No change in manufacturing process
  - Added circuitry is negligible
  - Low-complexity feature integration
  - Significant improvement (over 25% in simulations)

Observation:

- Read speed improves on average, but worst-case may be slower

Extensions

- Save chose decision time for each page/block etc.
- Shorten the precharge phase
- Other signatures
- Apply Uni-directional error correction codes
- ...

- Covered by Patent Application