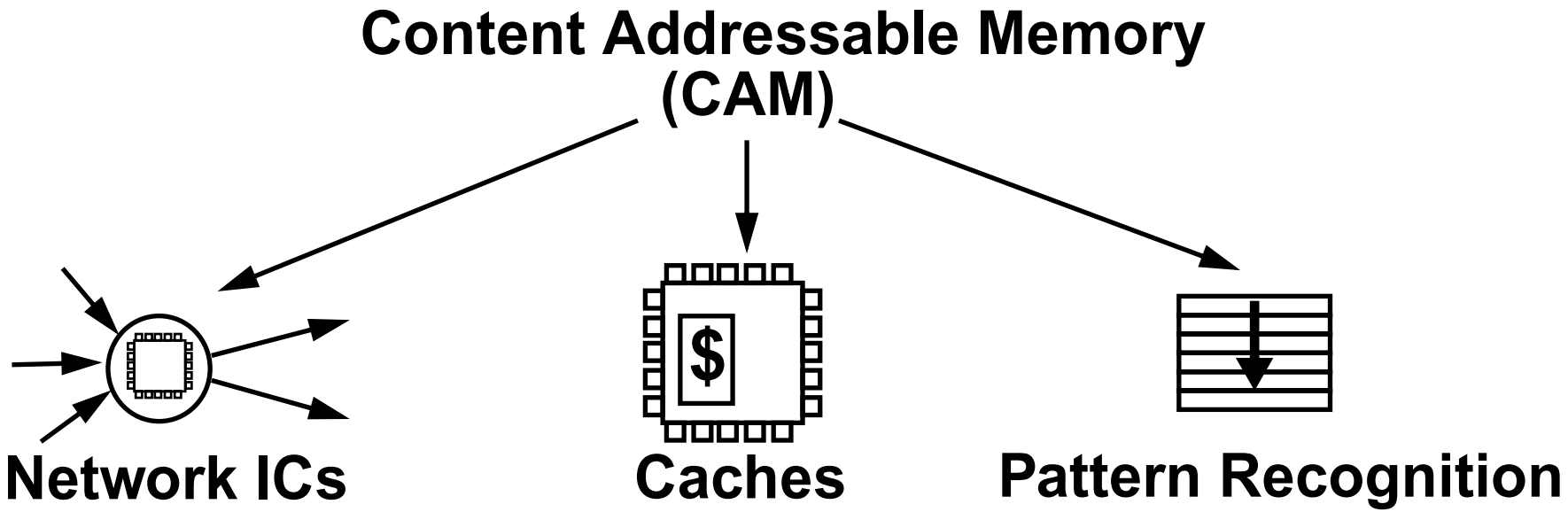
Current-Saving Match-Line Sensing Scheme for Content Addressable Memories

Igor Arsovski and Ali Sheikholeslami

Electrical and Computer Engineering Department
University of Toronto

February 12, 2003

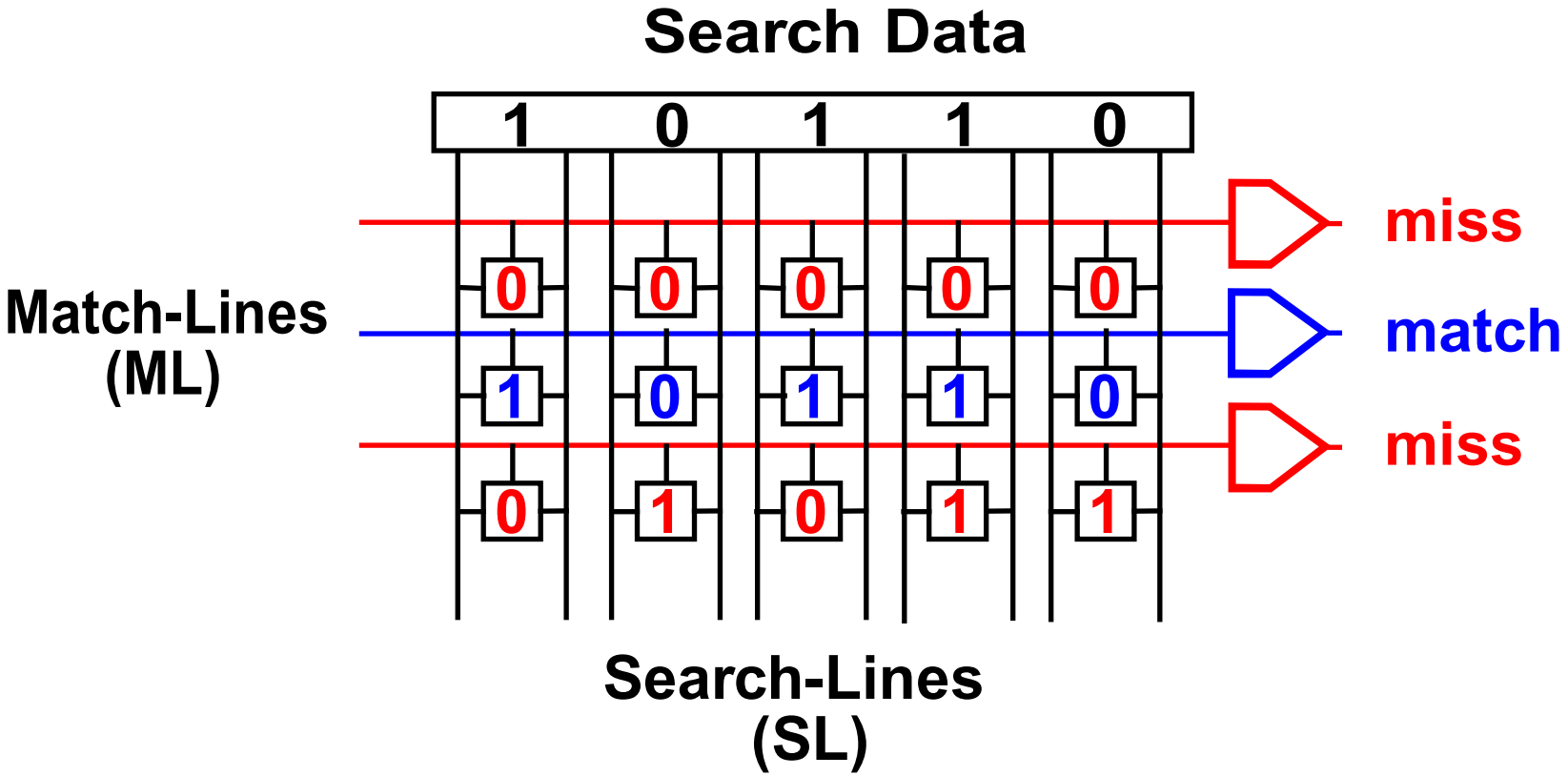
Motivation



CAMs search entire memory contents within a single clock cycle

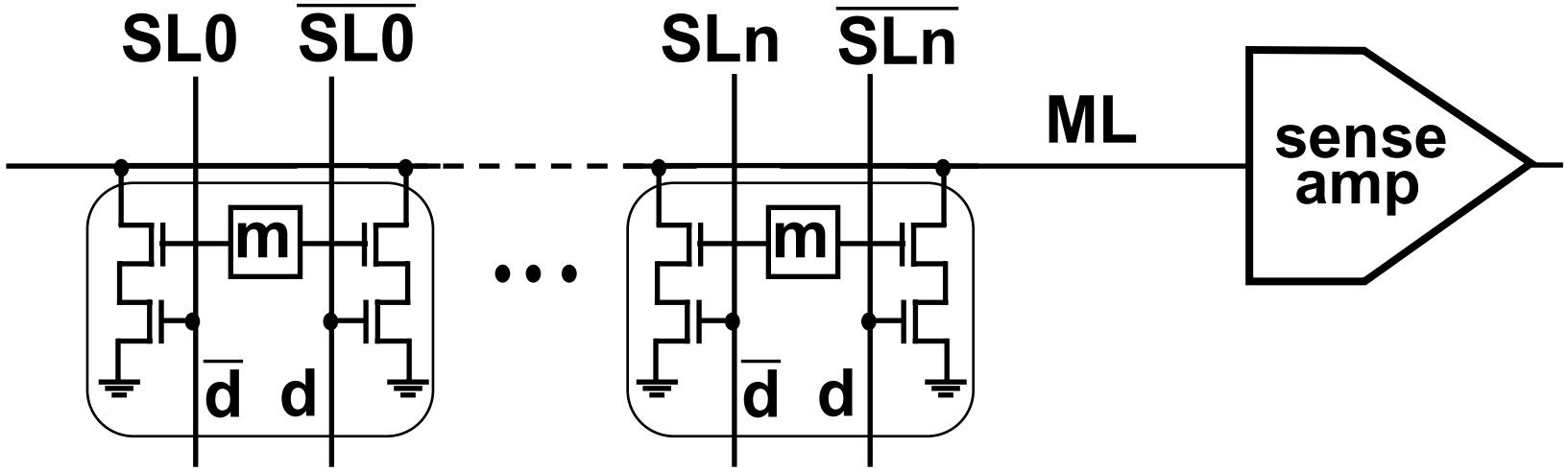
- **Problem:** CAMs parallel search causes high power dissipation.
- **Solution:** Proposed scheme reduces power by 60% without sacrificing search speed.

Basic CAM operation



- Search data is applied to Search-Lines (SL)s in parallel
- Search results develop on Match-Lines (ML)s in parallel

Conventional ML Sensing



- Reset SLs to GND
- Precharge MLs to VDD
- Apply Search Data to SLs
- Mismatched MLs discharge to GND
- Matched MLs remain at VDD

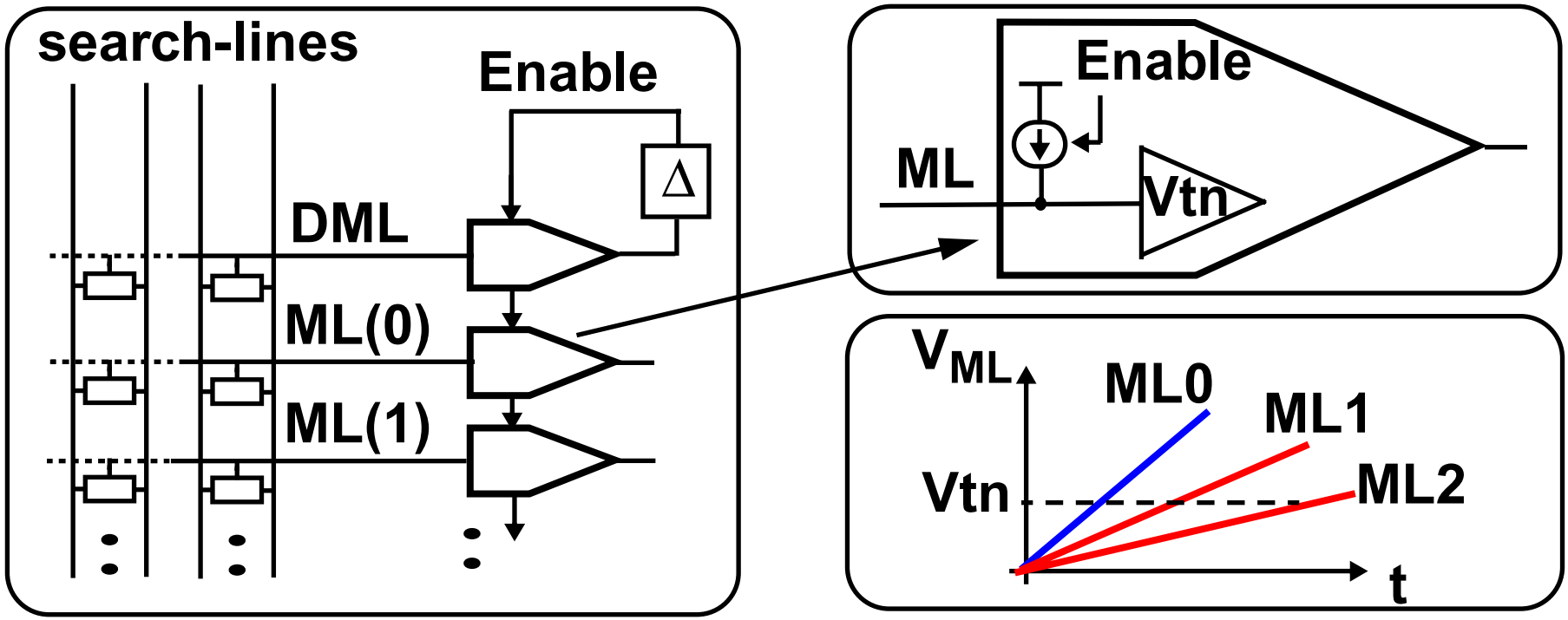
Improved ML Sensing Schemes

Conventional ML Sensing Scheme

Current-Race ML Sensing Scheme

Current-Saving ML Sensing Scheme

Current-Race ML Sensing Scheme

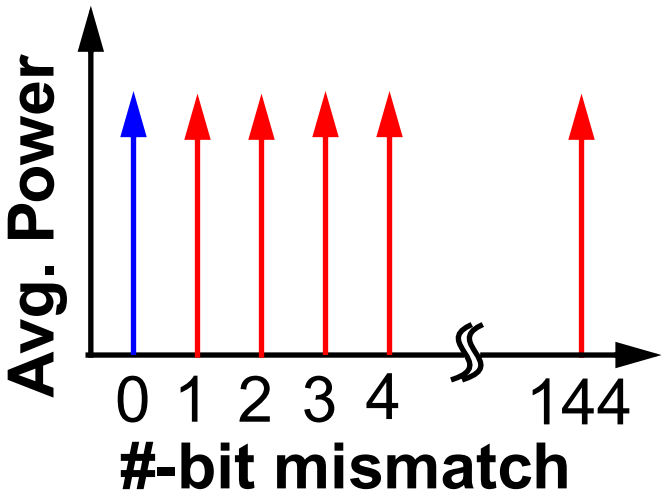


- Reset MLs to GND, then use equal currents to charge them up.
- Matches (ML0) ramp up faster then x-bit mismatches (MLx)
- The current-race stops when matched DML reaches Vtn.
- JSSC publication (January 2003)

Current Saving Idea

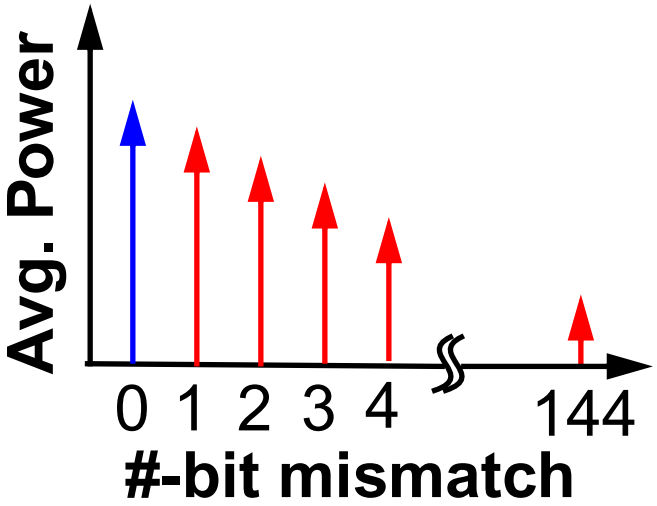
Current-Race ML Sensing Schemes

- Uniform power consumption on all MLs regardless of the number of mismatches. (i.e. same power for ML10 and ML1)

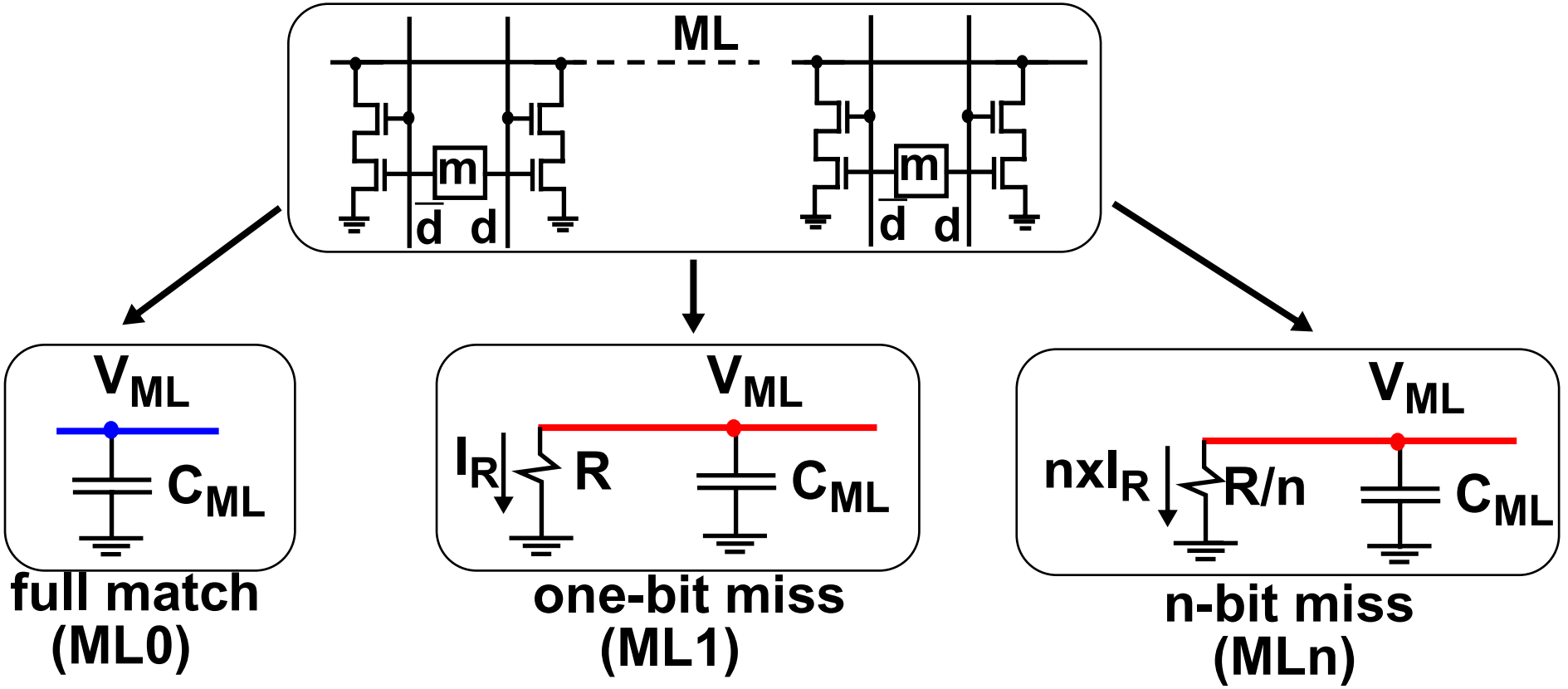


Current Saving ML Sensing Scheme

- Dynamically allocate less power to more mismatched MLs (i.e. less power for ML10 then ML1)

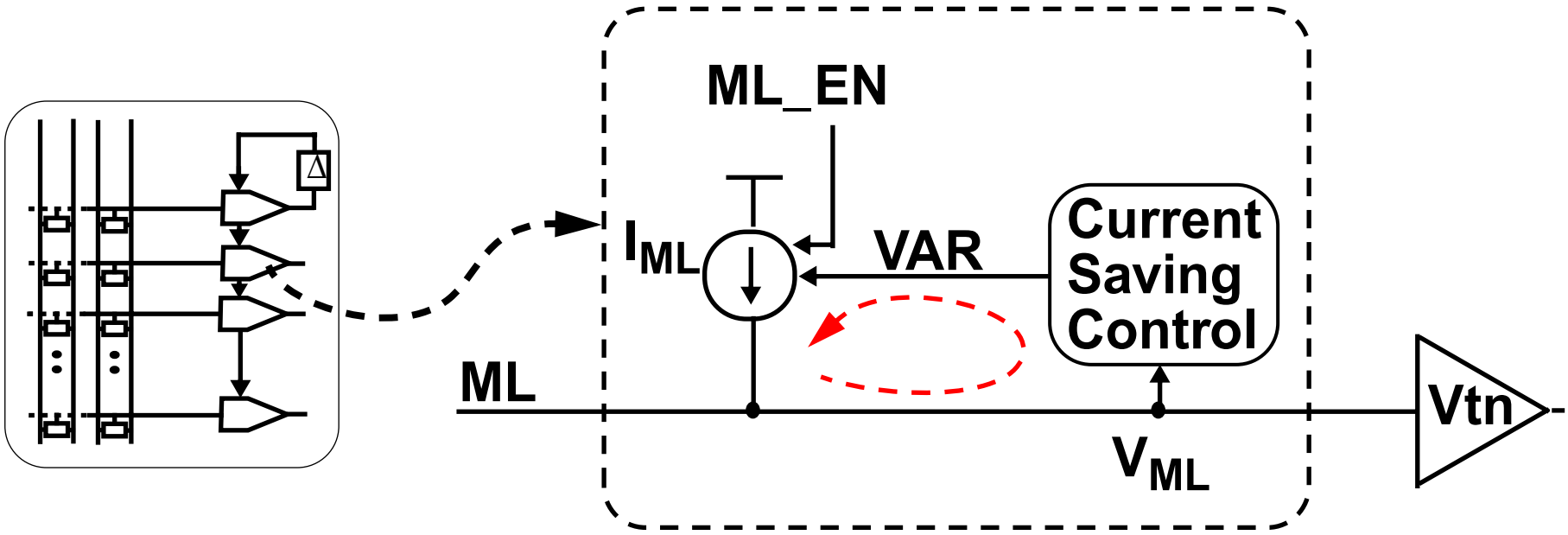


Dynamic Current Allocation



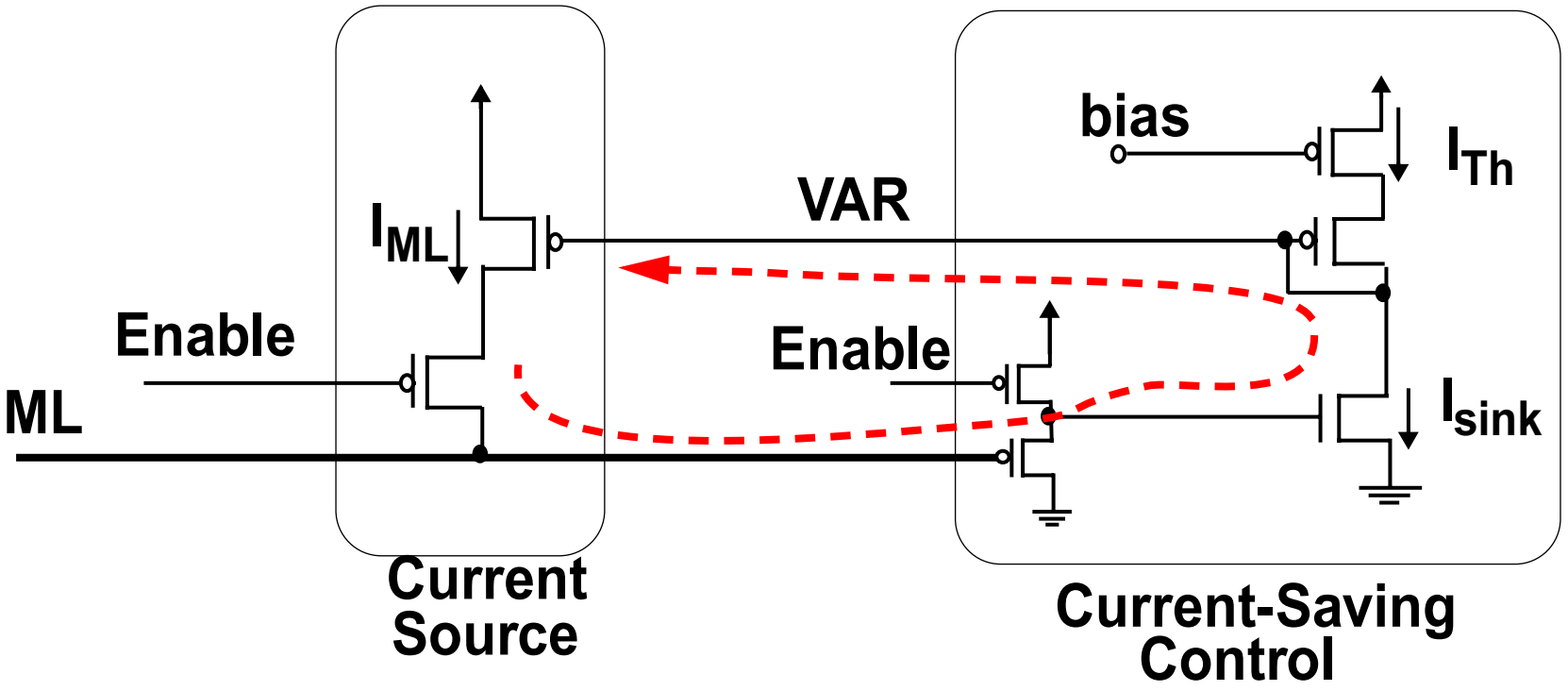
- Initial uniform current results in different voltages on MLs
- Voltages are then used to dynamically reduce current
- Hence, Save Power!!!

Current-Saving ML Sensing Scheme



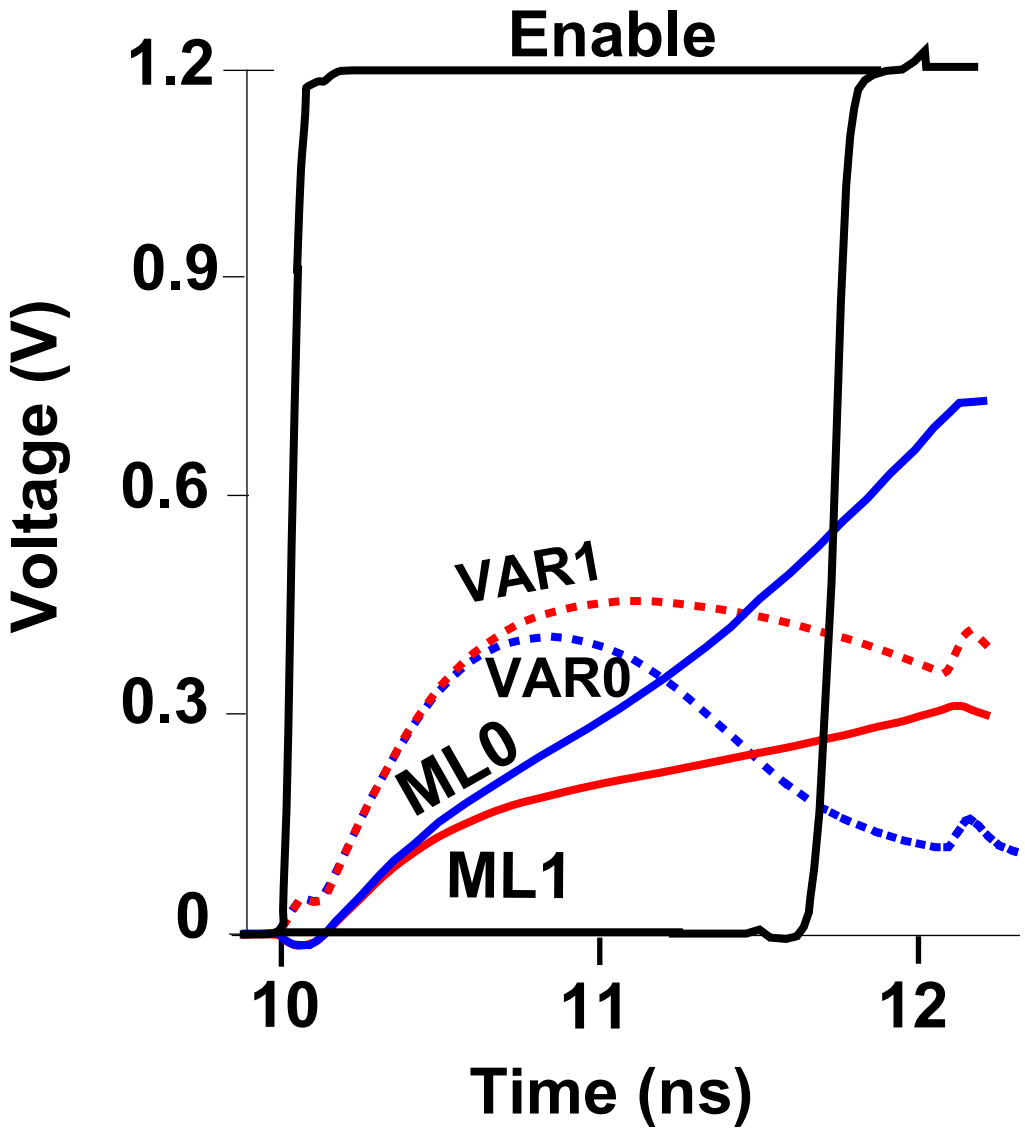
- Allocate current based on individual ML voltage (V_{ML})
- Dynamically allocate less current to slower rising MLs
- Matched MLs ramp fastest and receive most current
- Mismatched MLs ramp slower and receive less current

Current Saving Control



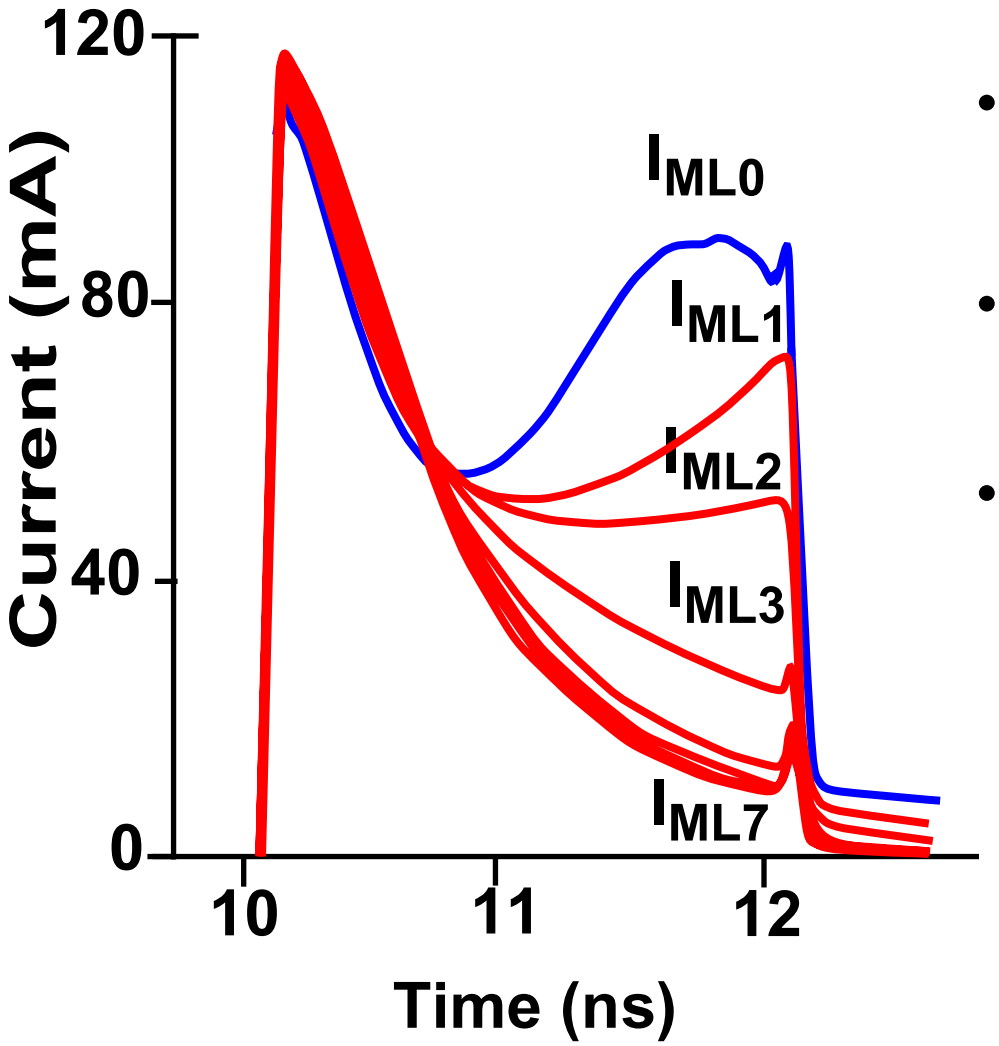
- VAR node controls the ML current (I_{ML})
- VAR node is pulled up by I_{Th} or pulled down by I_{sink}
- I_{sink} is proportional to the ML voltage

Simulation Results



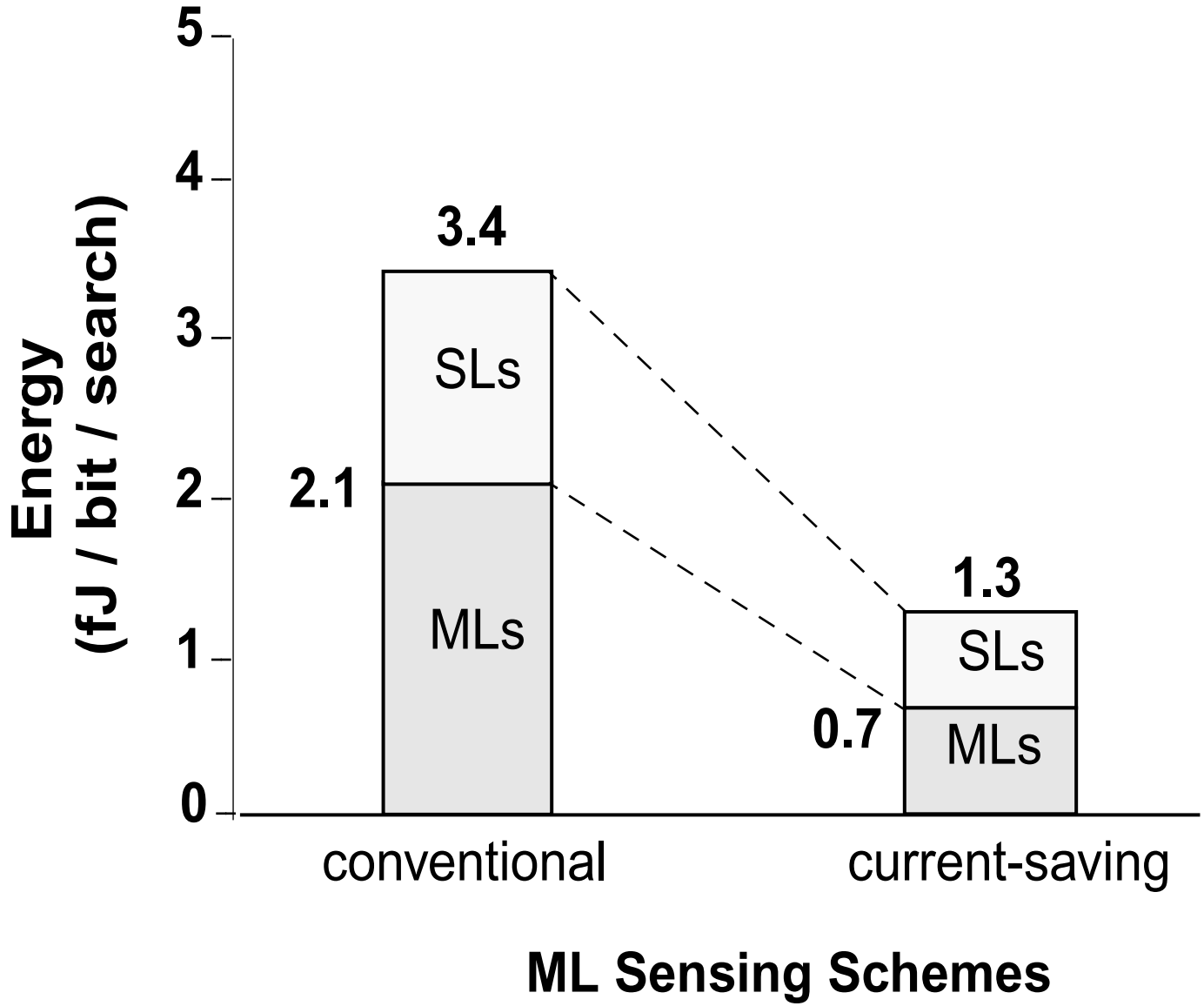
- VAR node precharged low to kick-start sensing.
- Initially, all MLs receive identical currents.
- The fully matched ML (ML0) rises faster and lowers VAR0
- The one-bit mismatched ML (ML1) rises slower increasing VAR1

Simulation Results: Current Saving

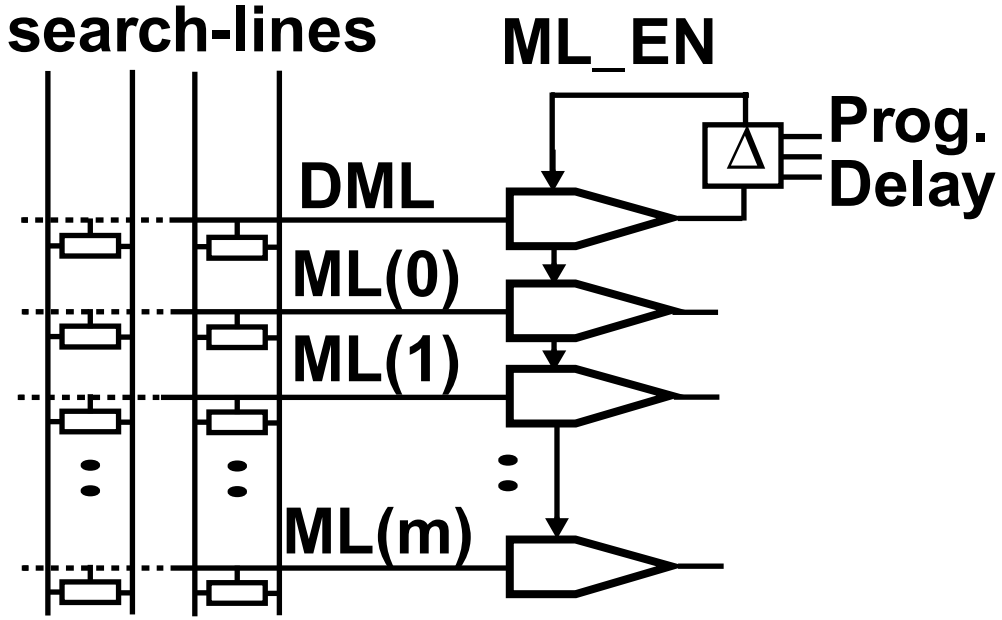


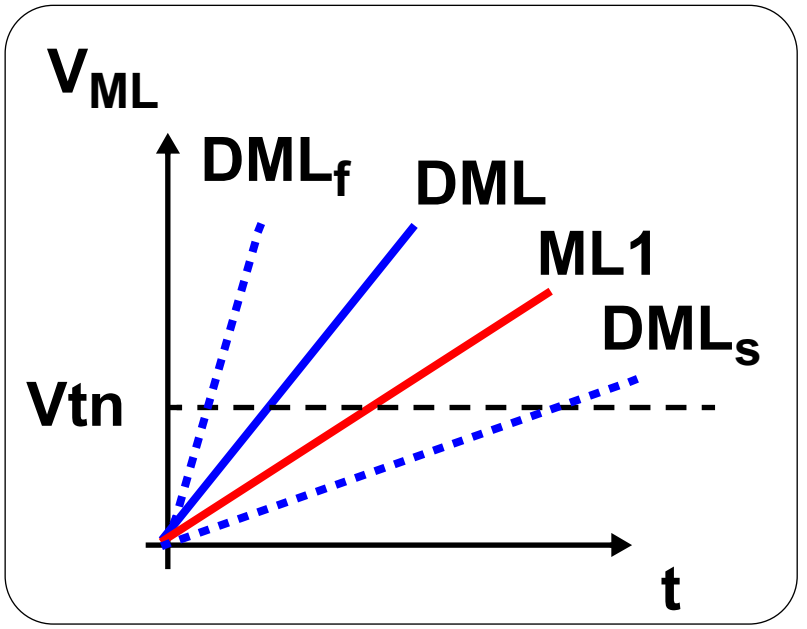
- Initially all MLs receive identical currents (I_{ML0},..I_{ML7})
- MLs with more mismatches receive less current
- MLs with seven or more bit mismatches receive similar reduction in current.

Simulation Results: Energy Saving



Process Variation Sensitivity

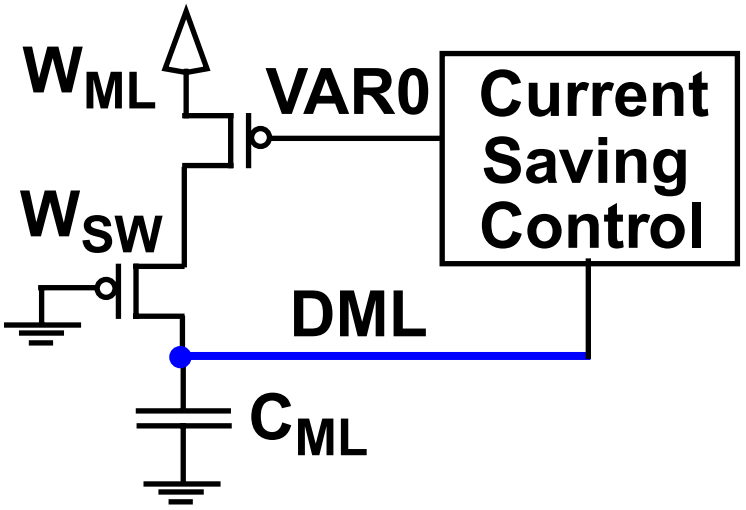




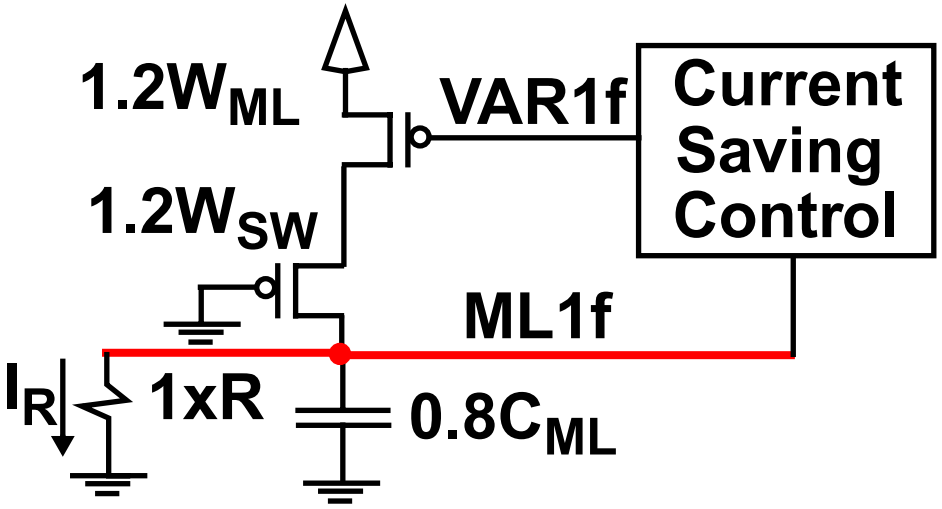
Worst case process variations:

- DML is faster than other matches - causing matches to look like misses
- DML is slower than other misses - causing misses to look like matches

Process Variation Sensitivity



full match

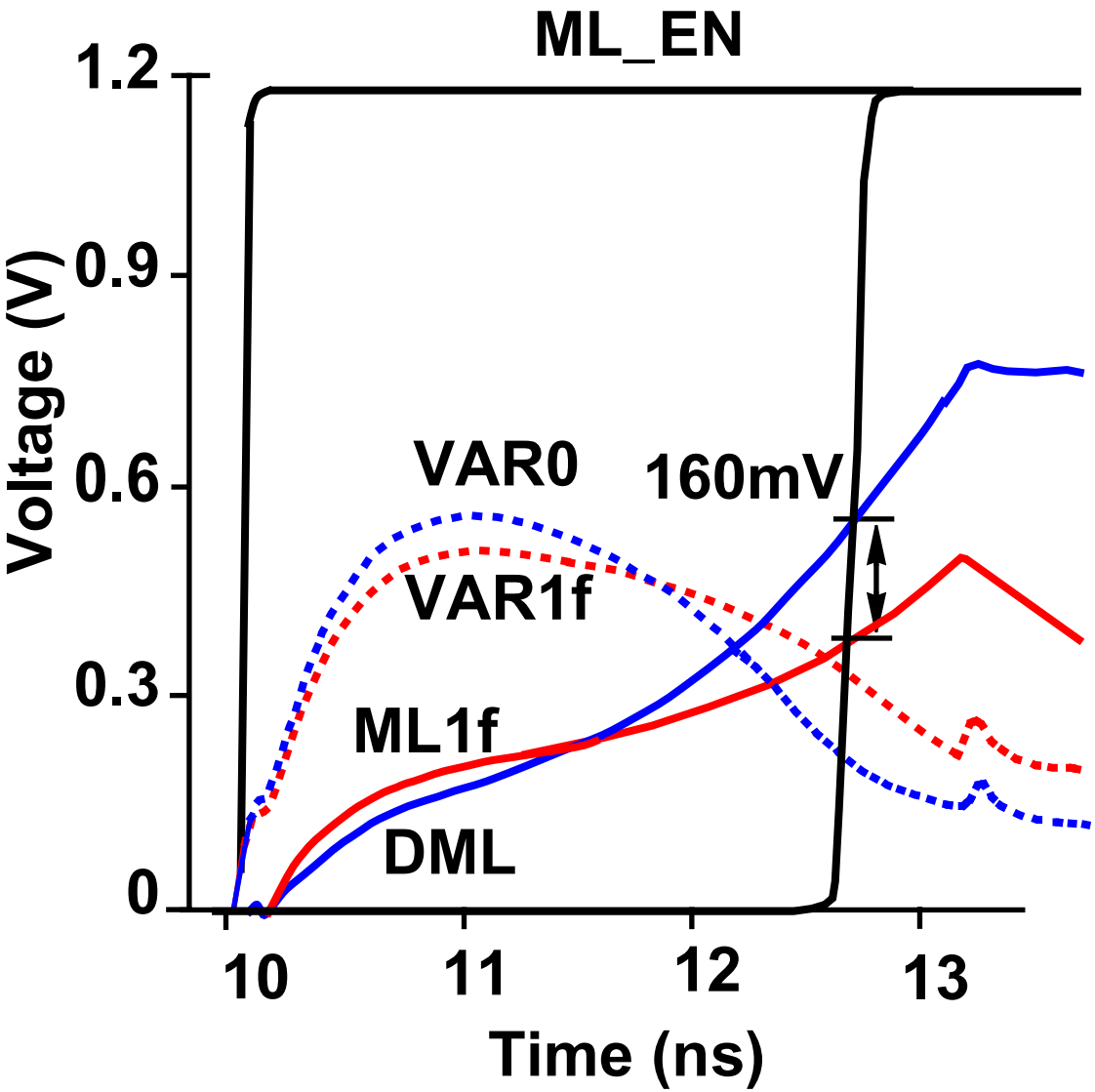


one-bit miss (fast mismatch)

Process Sensitivity Test:

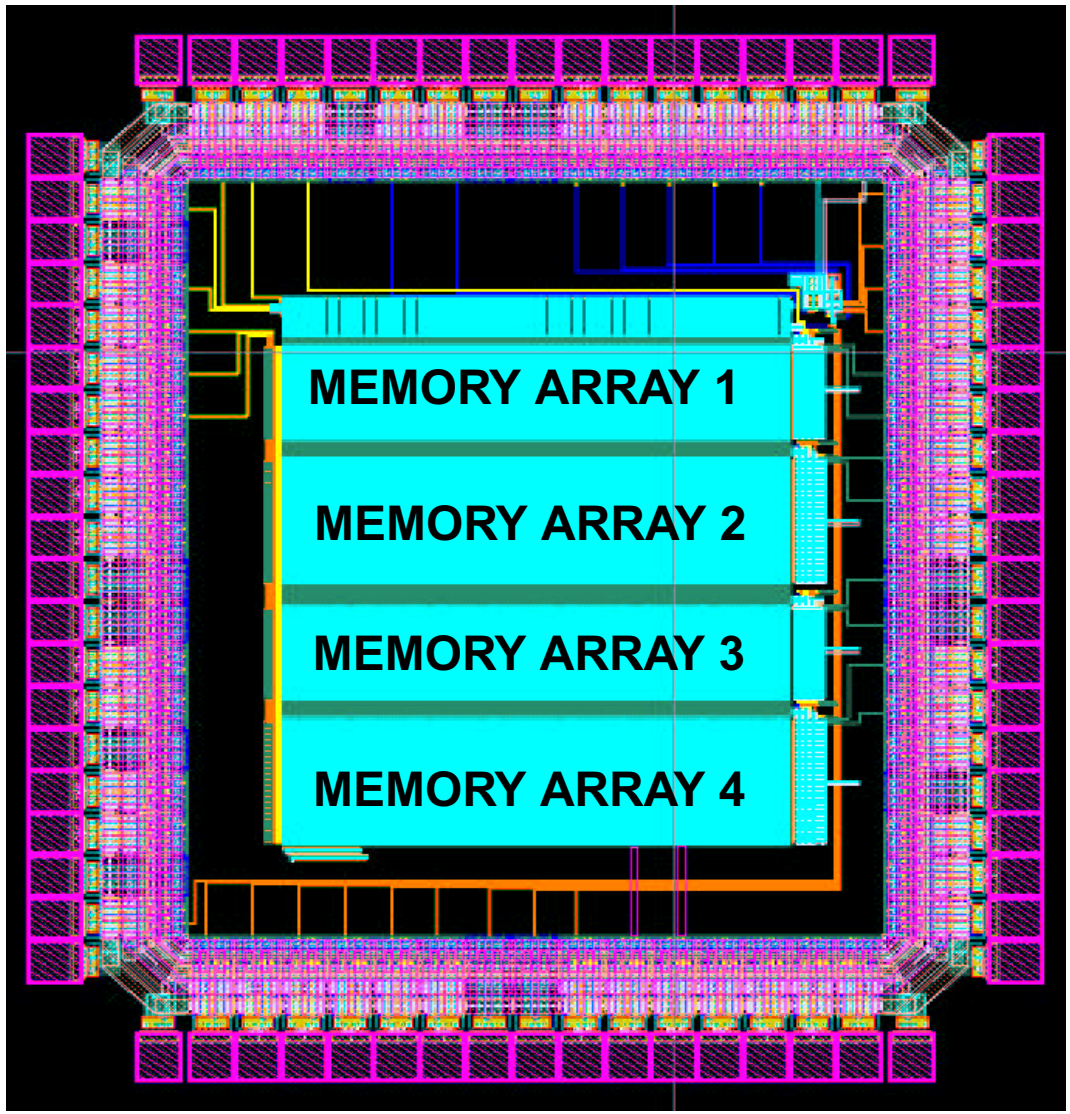
- Full match (ML0) compared to fast-mismatched 1-bit miss (ML1f)
- Fast mismatch has 20% lower capacitance and 20% larger current source

Voltage Development on ML0 and ML1f



- Initially ML1f develops higher voltage
- Current sunk from ML1f slows down its ramping
- 160mV ML voltage difference at sense time

CAM Test Chip



Specifications:

- 1.2V supply
- 0.13µm CMOS process
- 1600µm x 1800µm

Current Saving Sensing:

Area Overhead < 1%

Current-Saving Scheme Summary

The proposed scheme dynamically allocates power to achieve:

- 60% power reduction
- 2 ns match-time on a 144 bit (ternary) word
- Robust implementation insensitive to process variations