Energy-Aware Computing

Lecture 11: Leakage power

Outline

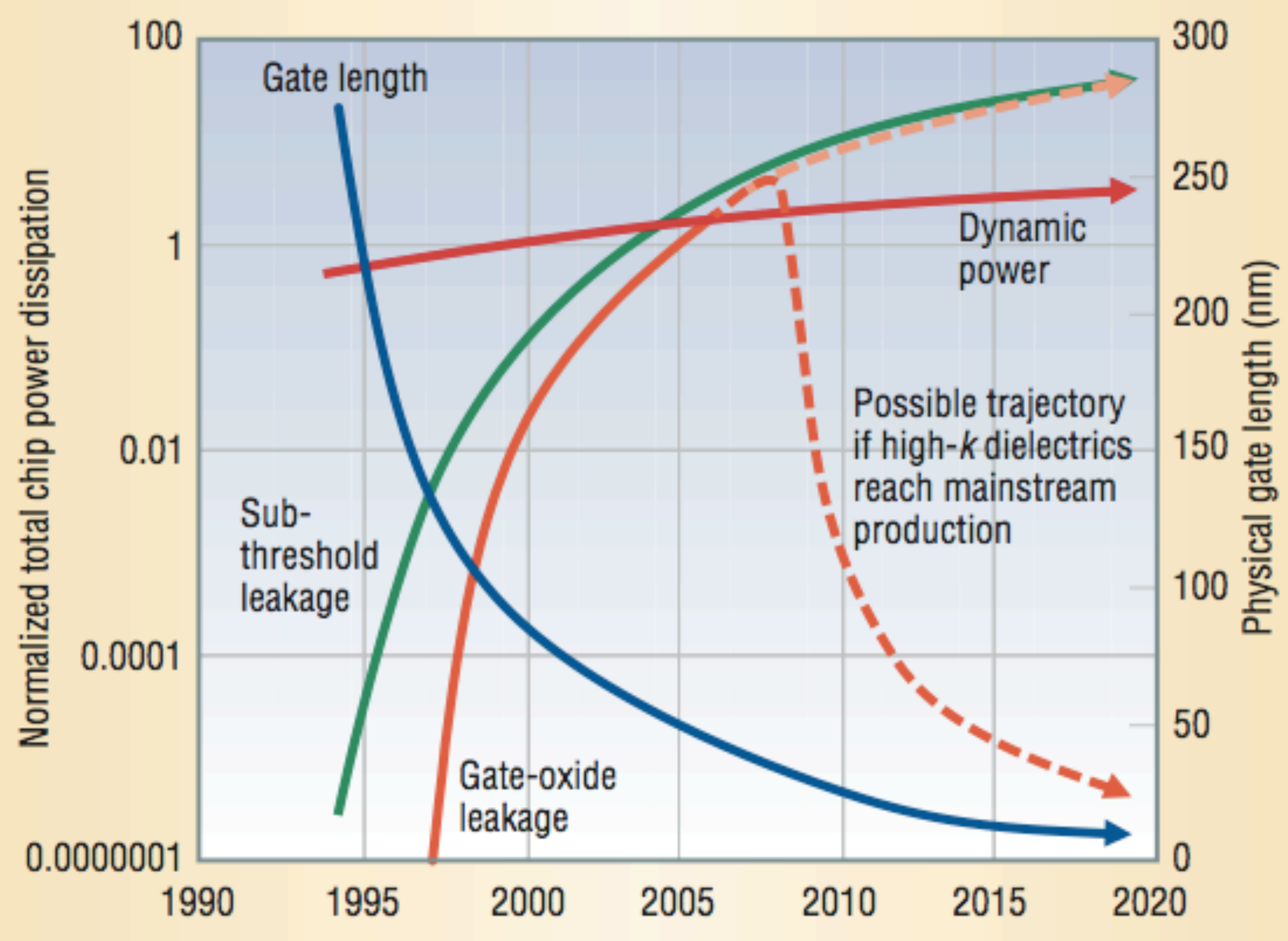
- Sources of leakage power
- Circuit level techniques
- Cache leakage reduction
  - Decay cache
  - Drowsy cache

Sources of leakage power

Remember, Power = current x Voltage

- Gate oxide leakage current
  - Current flows through gate oxide
- Subthreshold leakage current
  - Transistor does not turn off fully
- A few more leakage currents
  - Not as important as the above

Total power dissipation



Gate oxide leakage

$$I_{ox} = K_2 W \left(\frac{V}{T_{ox}} \right)^2 e^{-\alpha T_{ox}/V}$$

- Parameters
  - Tox - oxide thickness
- New dielectric materials greatly reduce gate oxide leakage so it will not be a problem

Subthreshold leakage

$$I_{sub} = K_1 W e^{V_{gs} - V_{th} - V_{off} / nV_\theta} (1 - e^{-V_{ds} / V_\theta})$$

- Terms:
  - V_θ - thermal voltage increases linearly with temp
  - W - transistor width
  - V_{th} - threshold voltage
- How to control
  - Increase V_{th}
    - Performance, dynamic power (cannot reduce V) loss
  - Turn off voltage (V_{ds}=0)
    - State loss
  - Decrease W
    - Performance loss

Leakage power for architects

- Butts and Sohi (MICRO'00) propose the following equation for leakage
  - Subthreshold only

$$P_{static} = VNk_{design}\hat{I}_{leak}$$

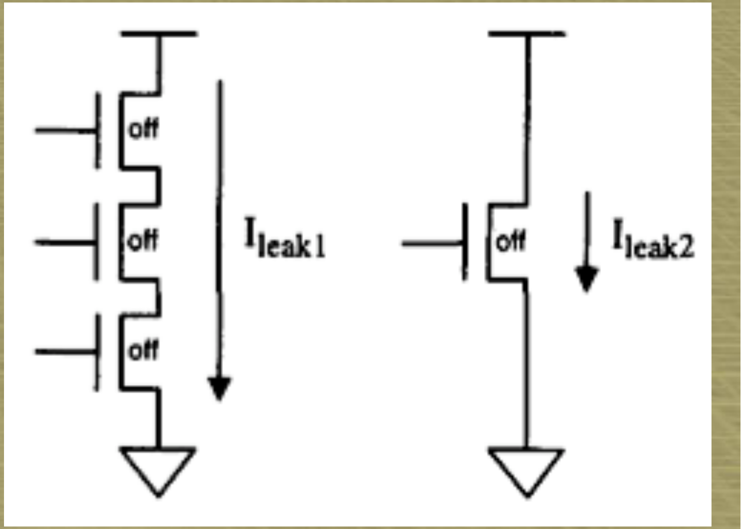
- What can we do?
  - Reduce voltage
  - Reduce number of devices
    - Either use fewer on chip or shut off some
  - Circuit design style (k_{design})
  - Slower, less leaky transistors (I_{leak})

Low-level techniques

- MTCMOS - multi-threshold
  - Use high Vth for non-critical path
- Reverse back-bias
  - Control the 4th transistor terminal - body
  - Effectively increases the threshold voltage
  - Increases circuit size
- Stacking effect
  - Gated Vdd/Vss

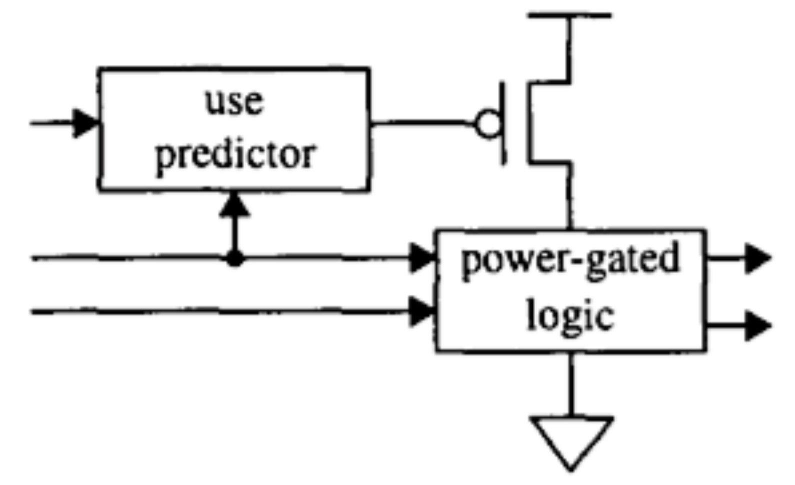
Stacking effect

- Leakage is reduced in stacks of transistors
  - When more than 1 are off
  - Can be an order of magnitude better
  - Even better if one has high V_T



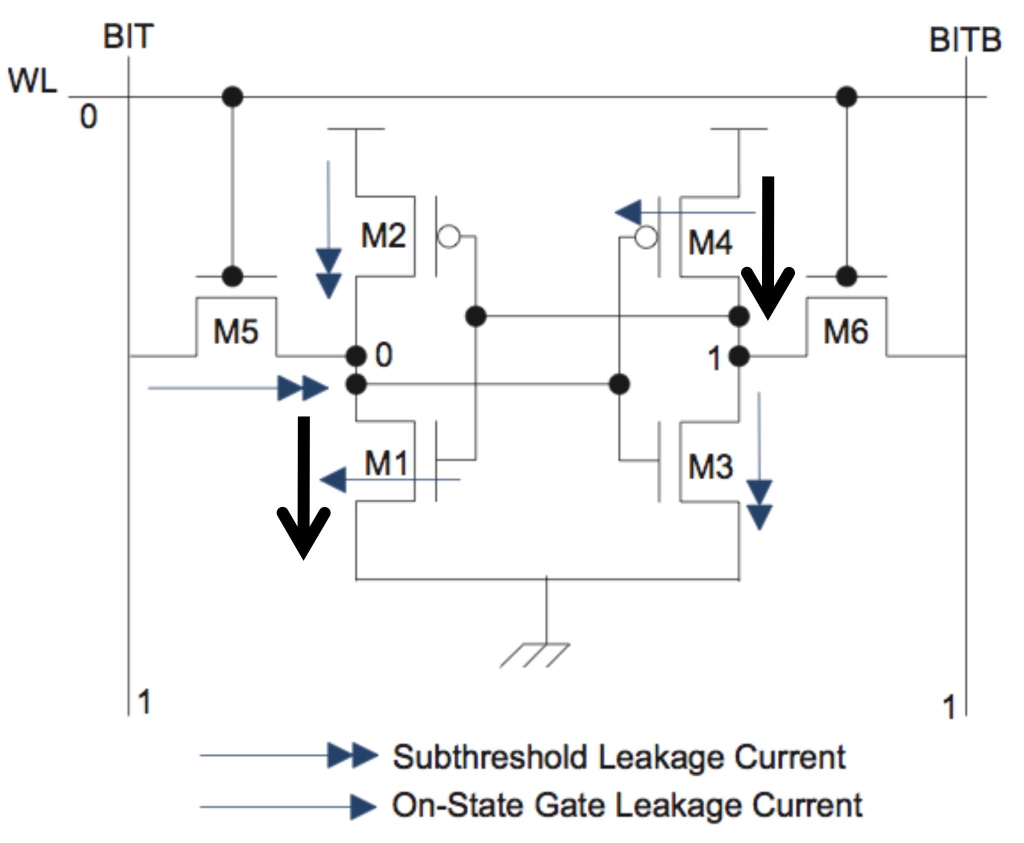
- Rationale
  - Drain-source voltage less than V_{dd}
  - Non-zero body bias (body effect)

Power gating



- Shut-off the power (or ground) of a unit
  - Preferably using a high V_{th} transistor
  - Safe for combinational circuits
  - State loss for sequential circuits
- Requires large power switching transistor
  - Consumes dynamic power, may cause slow down

SRAM leakage



Gate leakage for ‘on’ transistors at least an order of magnitude greater

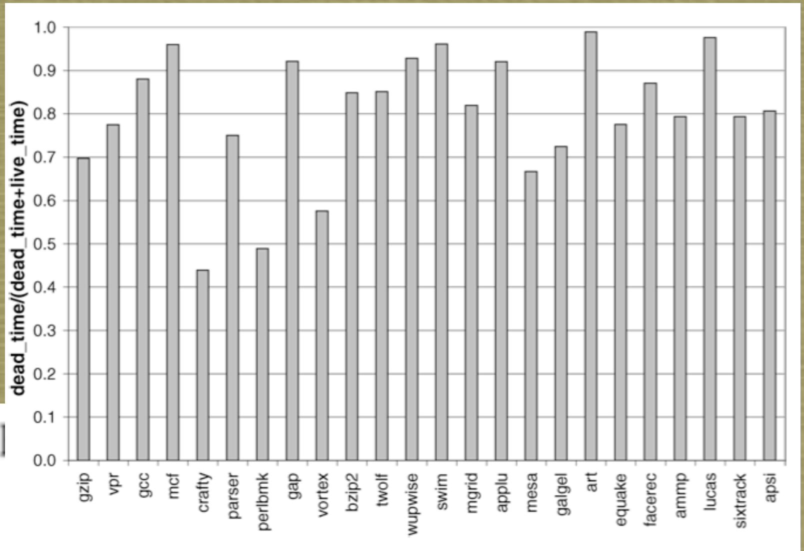
- Asymmetric, low leakage cell when storing 0 (as in fig)
  - Change M5, M2, M3 to high Vt

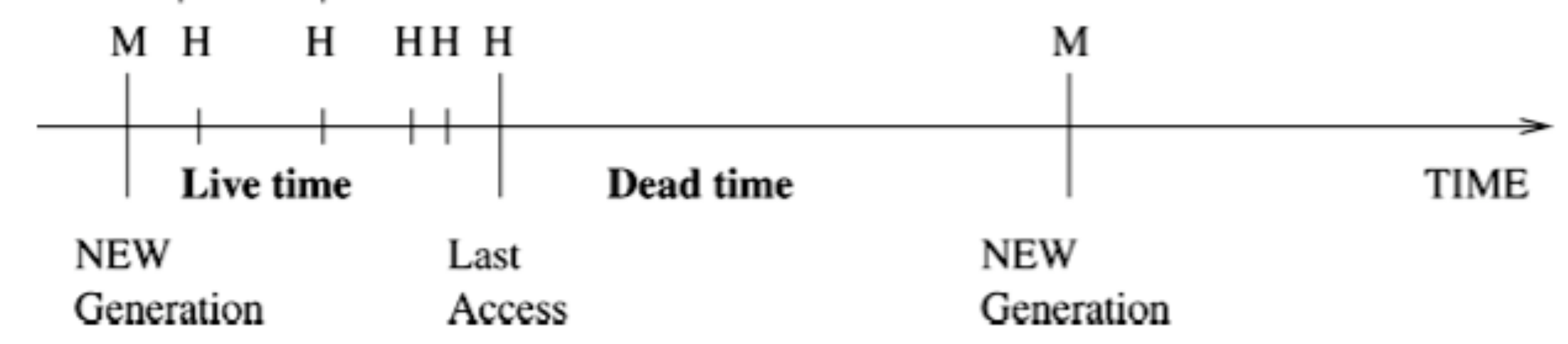
Cache leakage power

- On-chip caches account for most of the "real-estate" of a chip
- Most of the cache transistors are not switching most of the time
- Two types of leakage reduction methods:
  - State-destructive, e.g. cache decay
  - State-preserving, e.g. drowsy cache

Cache behaviour

- For a fixed period of time the activity in a cache is only centered on a small subset of the lines





Decay cache

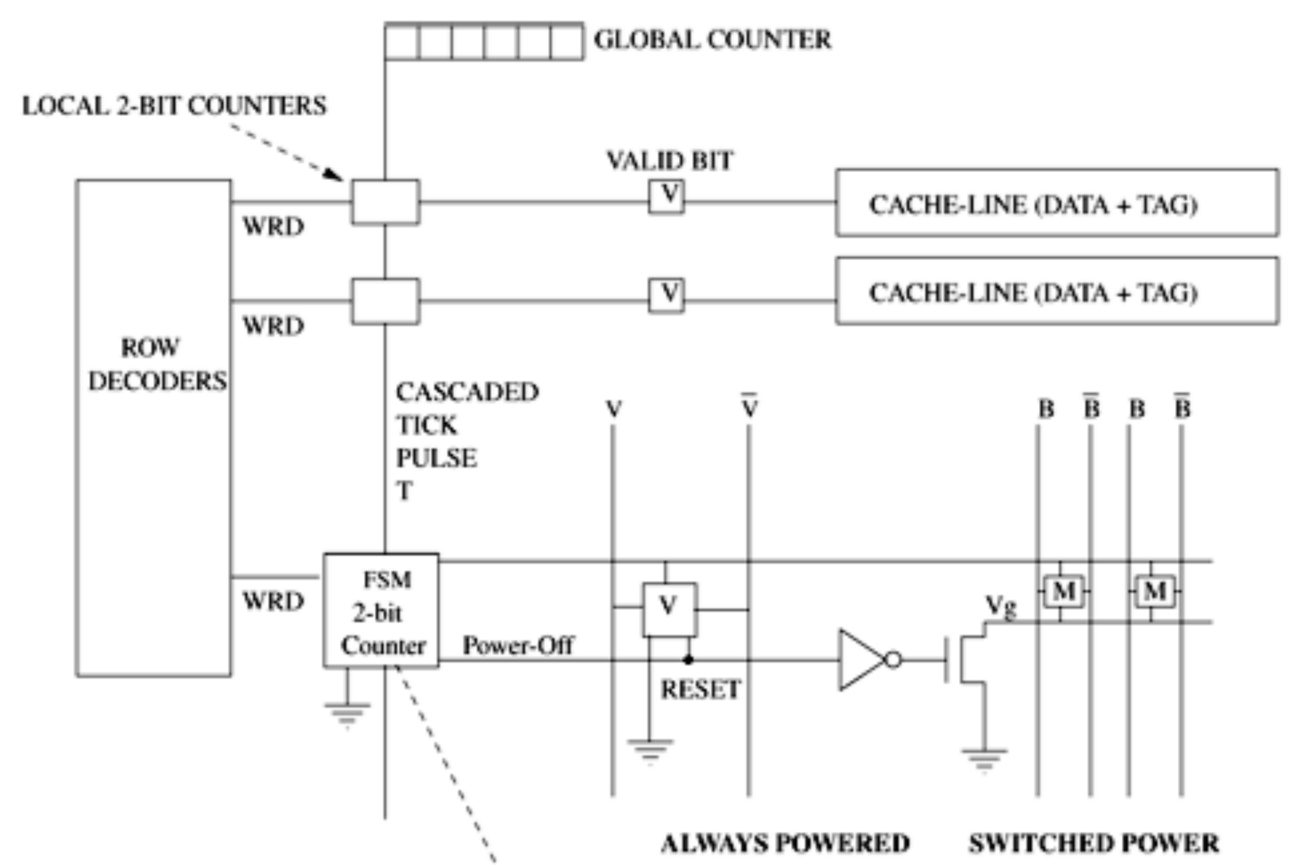
- Switch off the supply of a line/block, including tags, that is (predicted) “dead”
  - State destructive
  - Need to store it to lower level mem, if dirty
- Turn on again if line accessed
  - Treated as a miss
- How to safely predict a cache line is dead?

Decay policy

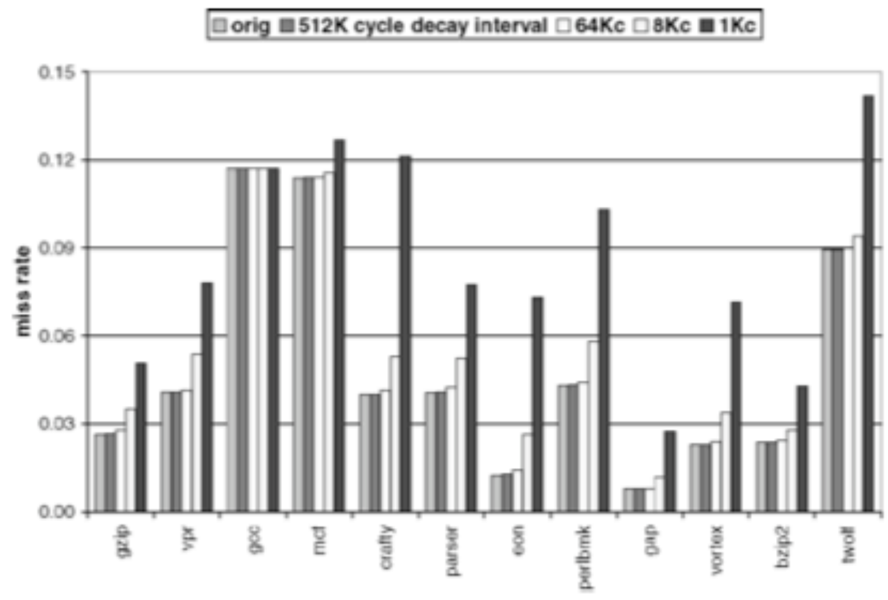
- Leave lines on for ~10K cycles from last access
  - Based on competitive algorithms
  - Switch off when leakage already dissipated equal to dyn energy of a miss
- Adaptive approaches may be better
  - Change this 10K according to application
  - Harder to implement

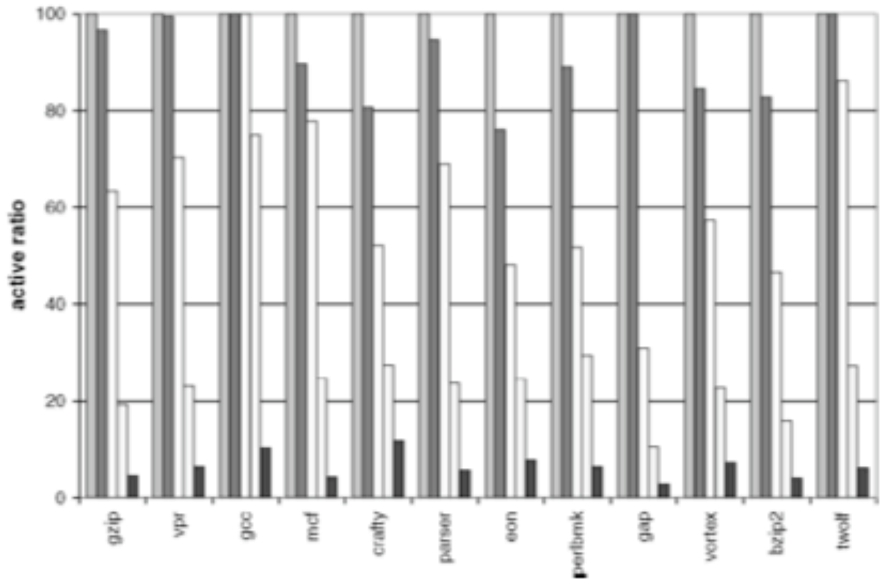
Cache decay circuits

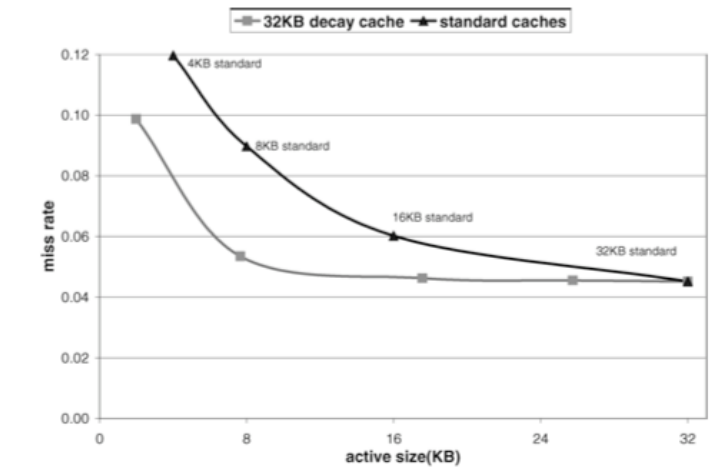
Hierarchical counter-based implementation



Decay evaluation







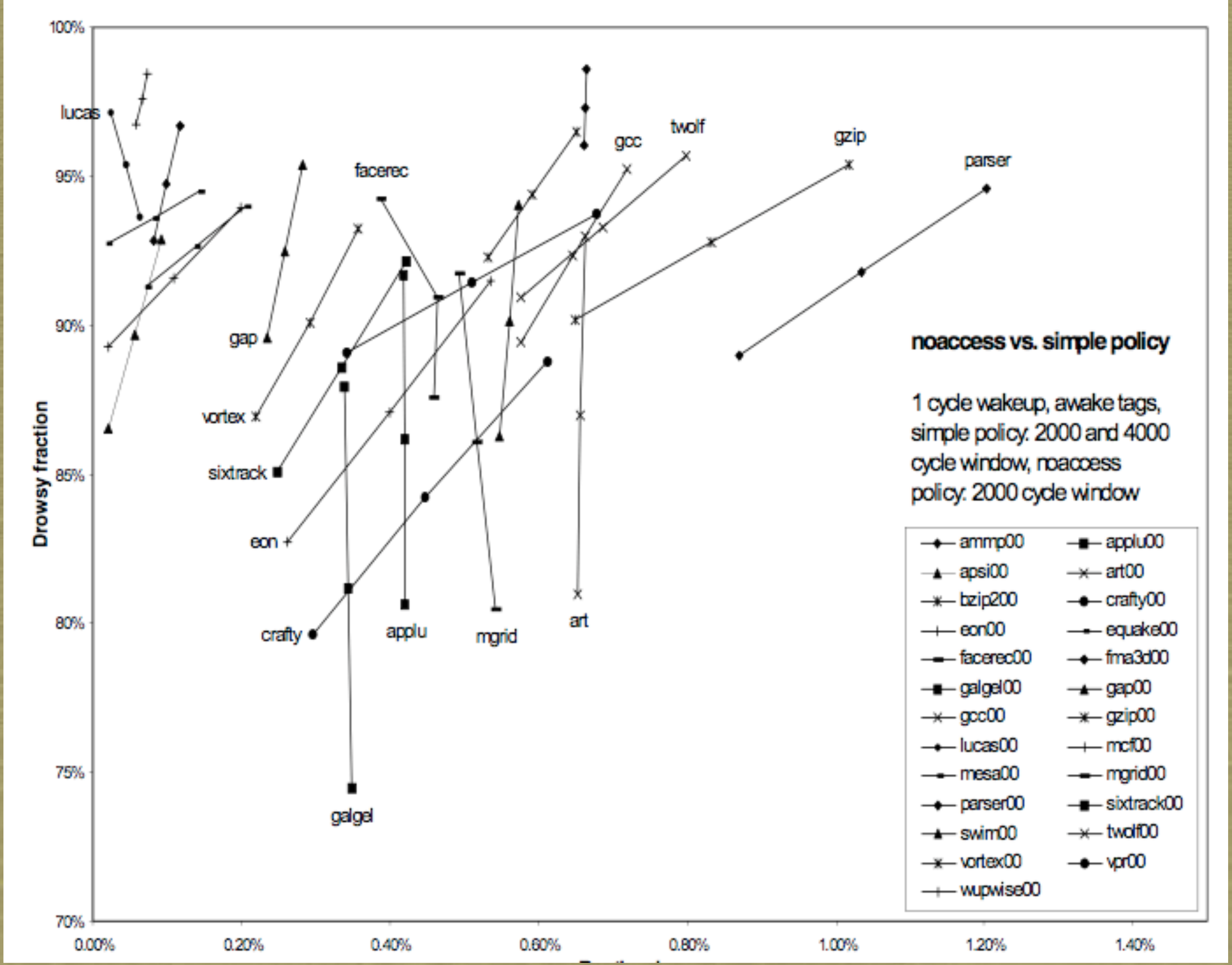
Drowsy cache

- Put a “dead line” into drowsy mode
  - Line is driven at lower supply voltage
    - Around 1.5x the transistor threshold voltage
  - Leaks more than decay on bit per bit comp.
  - State is preserved but inaccessible
- Relatively quick return to normal mode
  - 1-2 cycles should be enough
  - No need for a lower level mem access

Drowsy cache policy

- Allows for simpler policies than decay
  - Cost of wrong prediction is small
- Periodically put all lines in drowsy mode (ignoring access history) and wake up when accessed
- Drowsy tags?
  - Performance penalty for drowsy tags
  - Important only for associative caches

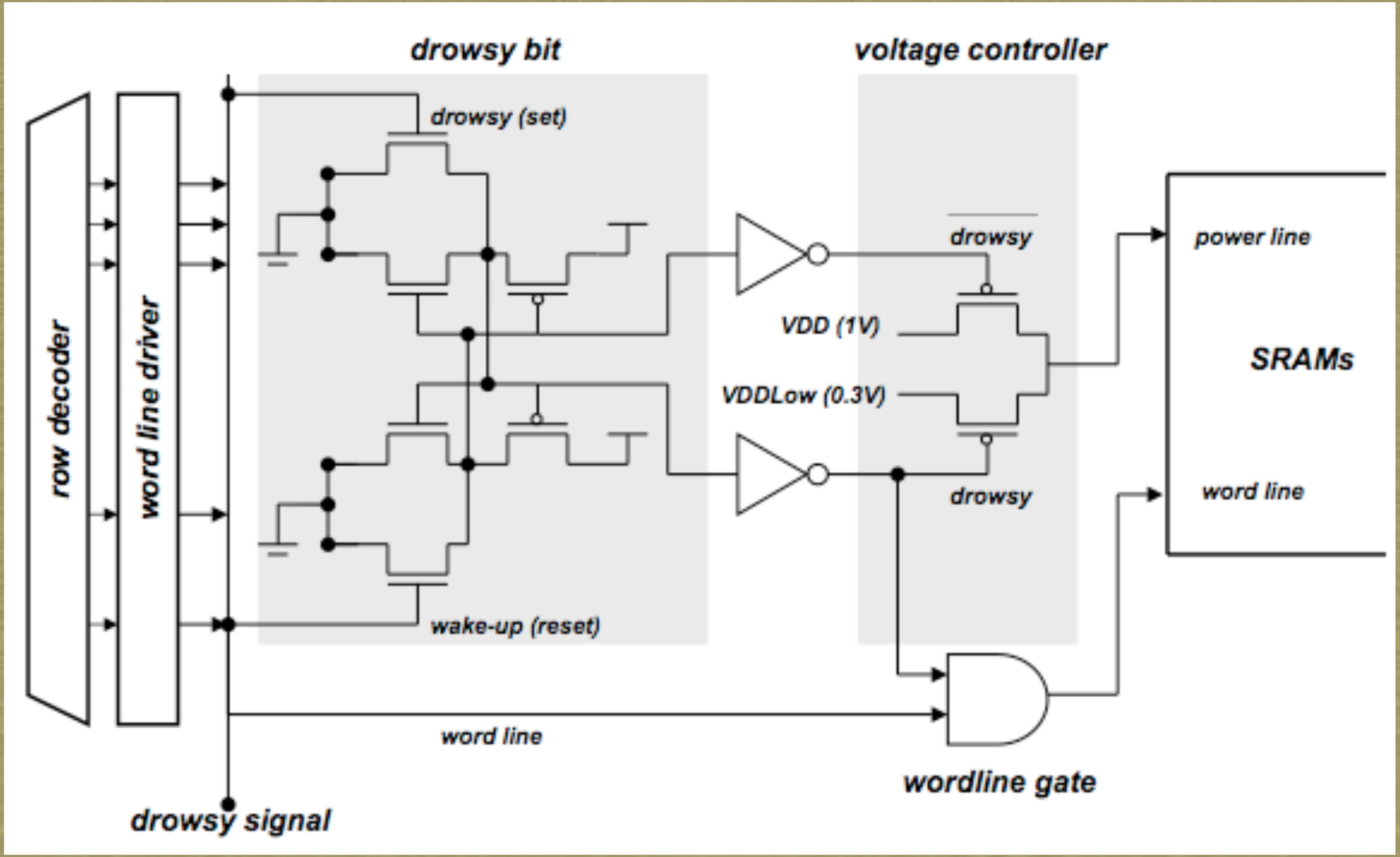
Policy comparison



noaccess vs. simple policy

1 cycle wakeup, awake tags, simple policy: 2000 and 4000 cycle window, noaccess policy: 2000 cycle window

Drowsy circuits



Drowsy energy

- Average total energy reduction 50%
- Leakage reduction 71% with active tags
  - 76% with drowsy tags
- EDP with drowsy tags always better
  - Decision based on acceptability of performance degradation