Super-Drowsy Caches

Single-$V_{DD}$ and Single-$V_T$ Super-Drowsy Techniques for Low-Leakage High-Performance Instruction Caches

Nam Sung Kim, Krisztián Flautner, David Blaauw, Trevor Mudge

nam.sung.kim@intel.com  krisztian.flautner@arm.com  {blaauw, tn}@eecs.umich.edu

ISLPED 2004, August 2004
#1 issue: energy efficiency

What the end-users really want: supercomputer performance in their pockets...
- Untethered operation, always-on communications
- Forget about the battery, charge once a month (or year)
- Driven by applications (games, positioning, advanced signal processing, etc.)

Technology scaling trends are not in our favor
- Need ways of dealing with leakage power
- New processes are expensive
- Diminishing performance gains from process scaling
- Dynamic power remains high

Energy efficient solutions need to cut across traditional boundaries (SW / architecture / microarch / circuits)

[Graph showing normalized total chip power dissipation from 1990 to 2020, with sub-threshold leakage, gate-oxide leakage, and technology node trends marked.]
The drowsy cache philosophy

• **Leakage power reduction with low implementation complexity**
  – Balance complexity between microarchitecture and circuits ➔ small impact on either
  – Low-leakage is achieved using cache line or block-level voltage scaling
  – Simple control policies enabled by low-leakage state-retention in caches

• **Drowsy wake-up policies result in negligible run-time overhead**
  – … even on in-order cores
  – A key requirement is fast wake-up transitions
  – Data caches: periodically putting all lines into drowsy mode yields good results
  – Instruction caches need predictive wake-up for best results

• **Super-drowsy improves on our original techniques**
  – **Simpler** circuit design
  – **More leakage reduction**: ultra-low retention voltage, no pre-charge unless needed
  – **Lower system complexity**: eliminates need for external drowsy voltage source
  – **Faster cache access**: no high-$V_T$ transistors on critical path
  – **Smaller run-time overhead**: simpler, yet better control policy for instruction caches
Single-$V_{DD}$ drowsy voltage controller

- Previous drowsy cache circuits required multiple external voltage levels to be supplied
- Now: no high-$V_T$ transistors required, yielding 20% faster access time
- 165mV is sufficient to preserve state
- 250mV drowsy state reduces leakage by 98% and adds noise margins

- Super-drowsy voltage controller uses feedback through schmitt trigger inverter to generate drowsy voltage
- As $V_{DD}$ is cut off, $VV_{DD}$ floats down
- $V_x$ is supplied through schmitt trigger inverter to stabilize drowsy voltage
To reduce bitline leakage, only one cache sub-bank is precharged at a time
  - Inter sub-bank transitions are predicted to eliminate precharge overhead of drowsy sub-banks
  - Bitline leakage is reduced by 88% using on-demand gated precharge

**Insight:** unconditional branches and sequential accesses cause most transitions
  - The targets of conditional branches are usually within the same sub-bank

Next sub-bank is predicted using the current set and sub-bank indices
  - Even small (64 entry) predictors show significant run-time improvement over no prediction
The predictive technique enables the gating of bit-line precharge for higher leakage savings over the noaccess policy at the cost of modestly increased run-time.

More than half of the SPEC2K workloads show more than 80% leakage reduction at close to zero run-time overhead.

Area overhead of 1K entry next sub-bank predictor (in terms of bits) is 1.2% or a 32K 2-way associative instruction cache.
Conclusions

Super-Drowsy Cache improves on previous techniques in multiple ways:

• **System complexity** of drowsy caches can be reduced by using a simple on-chip drowsy-voltage source
• **Faster cache access** can be achieved by eliminating the need for multiple threshold voltages in the design
• **Pre-charge gating** reduces bitline leakage - an often ignored component of other cache leakage reduction techniques
• **Sub-bank wakeup** latency is mitigated by predictive techniques
Questions?!