Exploiting Program Hotspots and Code Sequentiality for Instruction Cache Leakage Management

J. S. Hu, A. Nadgir, N. Vijaykrishnan, M. J. Irwin, M. Kandemir
Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802, USA
Outline

- Motivation
- Hotspots based leakage management
- Just-in-time activation
- Experimental results
- conclusions
Motivation

- Leakage is projected to account for 70% of the cache power budget in 70nm technology.
- Instruction caches are much more sensitive (performance impact) to leakage control mechanisms.
- Objective: exploiting application dynamic characteristics for effective instruction cache leakage management.
- Two major factors that shape instruction access behavior:
  - Hotspot execution
  - Code sequentiality
- Exploit program hotspots for further leakage savings.
- Exploit code sequentiality for performance penalty reduction.
Hotspot based Leakage Management

- Management cache leakage in an application-sensitive manner
- Track both spatial and temporal locality of instruction cache accesses
- Observations:
  - Program execution occurs in phases
  - Instructions in a phase do not need to be clustered
- Program phases produce hotspots in the cache
- Hotspot based leakage management (HSLM)
  - Prevent hotspots from inadvertent turn-off
  - Detect phase change for early turn-off
Dynamic Hotspot Protection

Original Scheme

HSLM Scheme

mask I-Cache

Global Set

PC

PC
HSLM Detecting Phase Changes

Original Scheme

I-Cache

Global Set

Drowsy Window

HSLM Scheme

mask I-Cache

Global Set

New hotspots detected

Drowsy window expires
Hotspot based Leakage Management

- Tracking program hotspots through the branch predictor (BTB)
- Augmenting BTB entries to maintain the access history of each basic block
- Adding a voltage control mask bit for each instruction cache line
  - Set mask bit indicates a hotspot cache line
- HSLM performs two main functions:
  - Keep hotspot cache lines from being turned off by the Global Set counter
  - Track phase changes and allow early turn-off of cache lines not in a newly detected hotspot
Augmented Cache Microarchitecture

Set: drowsy
Reset: active

to prevent accessing drowsy lines
BTB Microarchitecture for HSLM

Branch Target Buffer

- vbit
- tgt_cnt
- fth_cnt

Branch taken

BTB hit

Global Mask Bit

Leakage Control Circuitry

VCM

ICache

Global Reset

PC
Just-In-Time Activation (JITA)

- Sequentiality is the norm in the execution of many applications (e.g., >80% static sequential code, 50% branches are not taken -> 90%)
- Take advantage of the sequential access pattern to do just-in-time activation
  - The status of the next cache line is checked and pre-activated if possible
  - In sequential access, performance penalty due to activation of drowsy lines is eliminated
- JITA may fail:
  - target of taken branch is beyond next cache line
  - or next instruction is beyond current bank
Just-In-Time Activation (JITA)
Augmented Cache Microarchitecture

to prevent accessing drowsy lines

Set: drowsy
Reset: active

---

row decoder
word line drivers
word line

Preactivate
Global Set
Set
IQ Q
Reset

0.3V (drowsy)
1V (active)

power line
SRAMs
word line
wordline gate

---

to prevent accessing drowsy lines
Leakage Energy Breakdown (overhead)
DHS-Bank-PA achieved a leakage energy reduction of 63% over Base, 49% over Drowsy-Bank, and 29% over Loop (Compiler)
Energy Delay Product (EDP)

DHS-Bank-PA achieved the smallest EDP: 63% over Base, 48% over Drowsy-Bank, and 38% over Loop.
Effectiveness of JITA

Applying JITA, DHS-PA removes 87.8% performance degradation of DHS
Conclusions

- HSLM and JITA can be implemented with minimum hardware overhead
- Energy model takes the energy overhead of introduced hardware and other processor components into account
- Applying both HSLM and JITA helps
  - Reduce leakage energy by 63% over Base, 49% over Drowsy-Bank, and 29% over Loop
  - Reduce the (leakage) energy*delay product by 63% over Base, 48% over Drowsy-Bank, 38% over Loop
- Evaluation shows that application characteristics can be exploited for effective leakage control
Thank You!