Practical Way Halting by Speculatively Accessing Halt Tags

Daniel Moreau, Alen Bardizbanyan, Magnus Själander*, David Whalley**, Per Larsson-Edefors

Chalmers University of Technology

*Norwegian University of Science and Technology

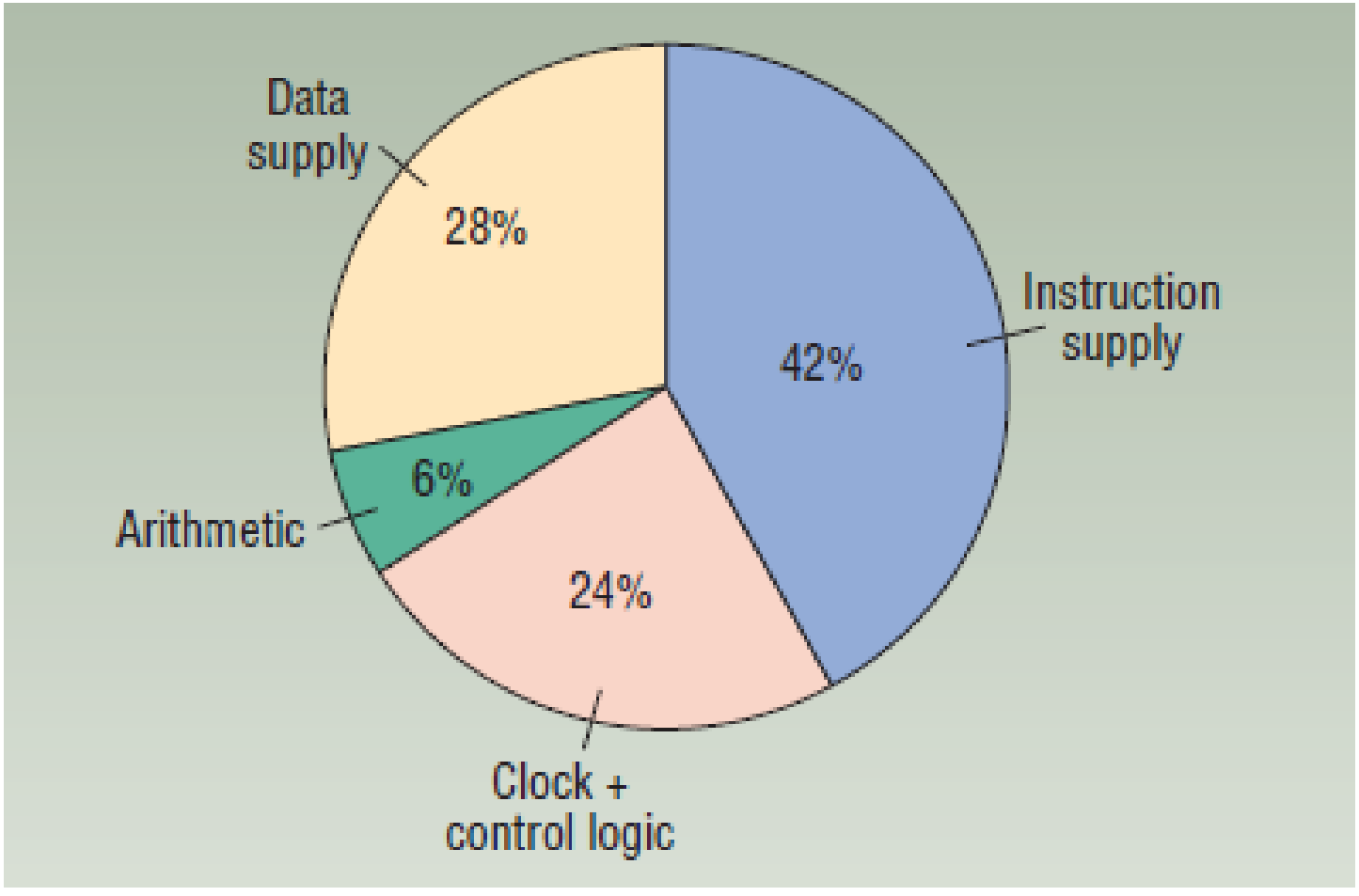
**Florida State University

Energy Efficient Processor Design

- Need for energy efficient processors.
  - Need to extend battery life for mobile systems.
  - Reduce generated heat for general-purpose processors.
  - Electricity cost for computing is increasing.
- Should also not negatively impact performance.
- Architecture features need to be reexamined with respect to energy efficiency, while retaining performance.

Embedded Processor Energy Breakdown

- From *Efficient Embedded Computing* in IEEE Computer 2008.

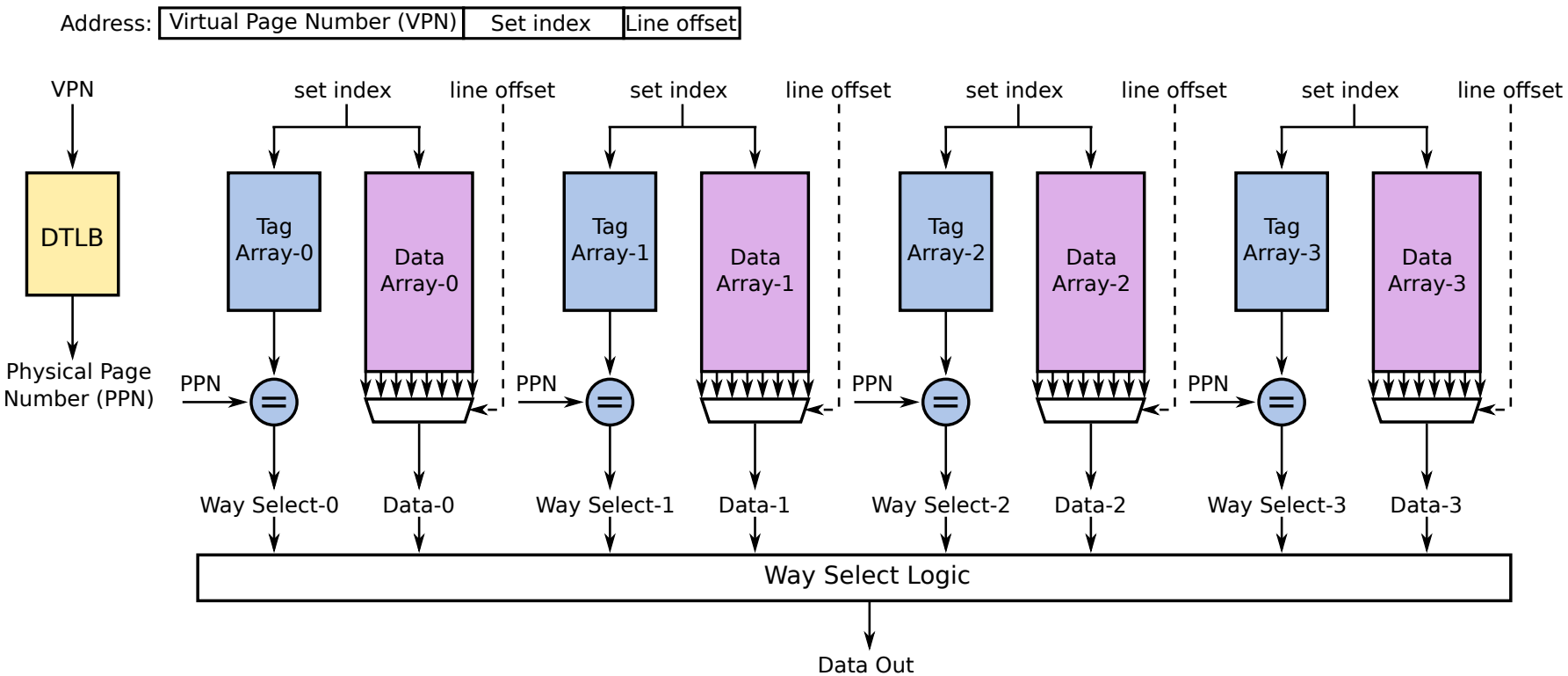


My Energy Efficient Processor Research

- instruction fetch
  - instruction register file (IRF) (ISCA, PAC^2, MICRO, CASES, LCTES)
  - tagless hit instruction cache (THIC) (MICRO, LCTES, ODES, TACO)
- pipeline execution
  - static pipelining (SP) (INTERACT, CAL, LCTES, LCTES, CASES)
- data access
  - tagless access buffer (TAB) (CGO)
  - practical data filter cache (PDFC) (ODES, TACO)
  - speculative tag access (STA) (ICCD)
  - early load data dependence detection (ELD^3)
  - context aware loads and stores (CALS) (LCTES)
  - **speculative halt tag access (SHA) (DATE)**

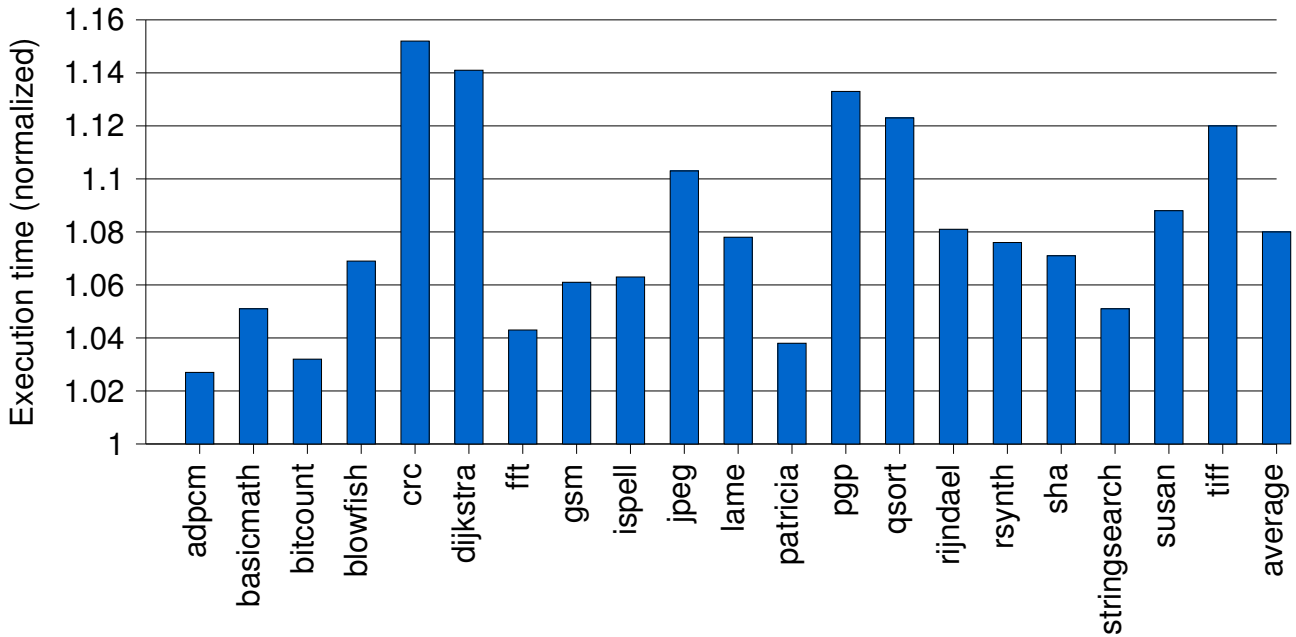
Set-Associative L1 DCs

- Most level-one data caches (L1 DCs) use a set associative organization to decrease the miss rate.
- L1 DC data arrays require more power to access than L1 DC tag arrays since the data arrays are much larger.



Loads from a Set-Associative L1 DC Are Energy Inefficient

- For stores all L1 DC tag ways are checked first and then a single data way is updated on the following cycle.
- For loads all L1 DC tag and data ways are accessed in parallel since the loaded value may be used in subsequent instructions.
- The requested data can at most reside in one of the n ways!
- We found an 8% execution time overhead on average when the L1 DC tag and data memories are sequentially accessed.

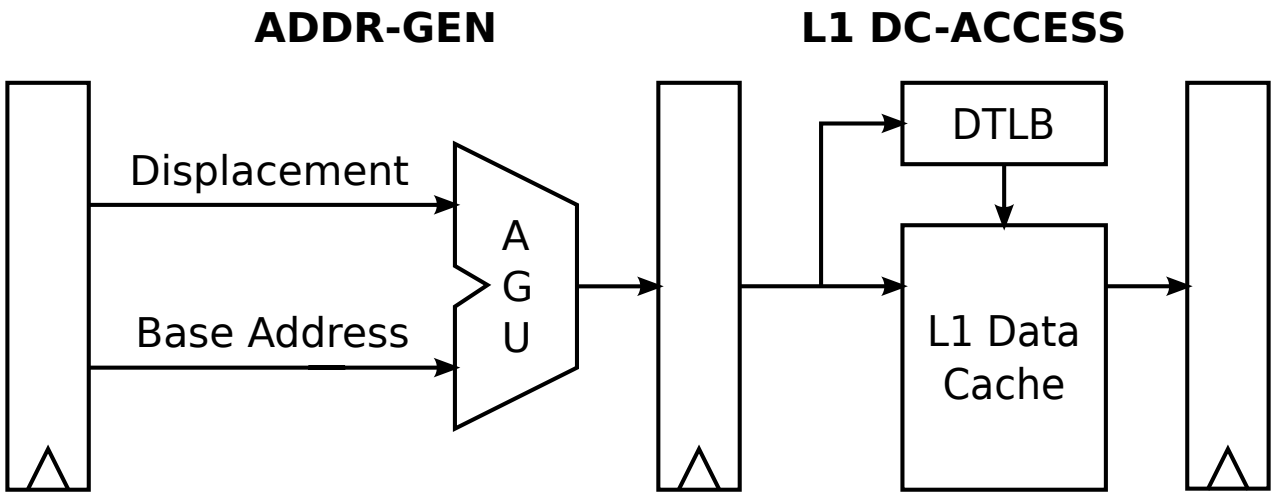


Way Halting

- Way halting has been proposed to reduce energy usage.
- L1 DC tags are split into two parts.
  - A few low-order bits called the *halt tag*.
  - The remaining higher-order bits.
- Halt tags are checked first and only the ways of the halt tags that match the corresponding bits in the address are accessed for the remaining bits of the tag and the data.
- The insight is that the low-order bits are the ones most likely to differ.
- Conventional SRAMs are synchronous and can only be controlled at the start of a clock cycle.
- The halt tag approach would either require an extra access cycle or a custom SRAM implementation, which would be costly to implement.

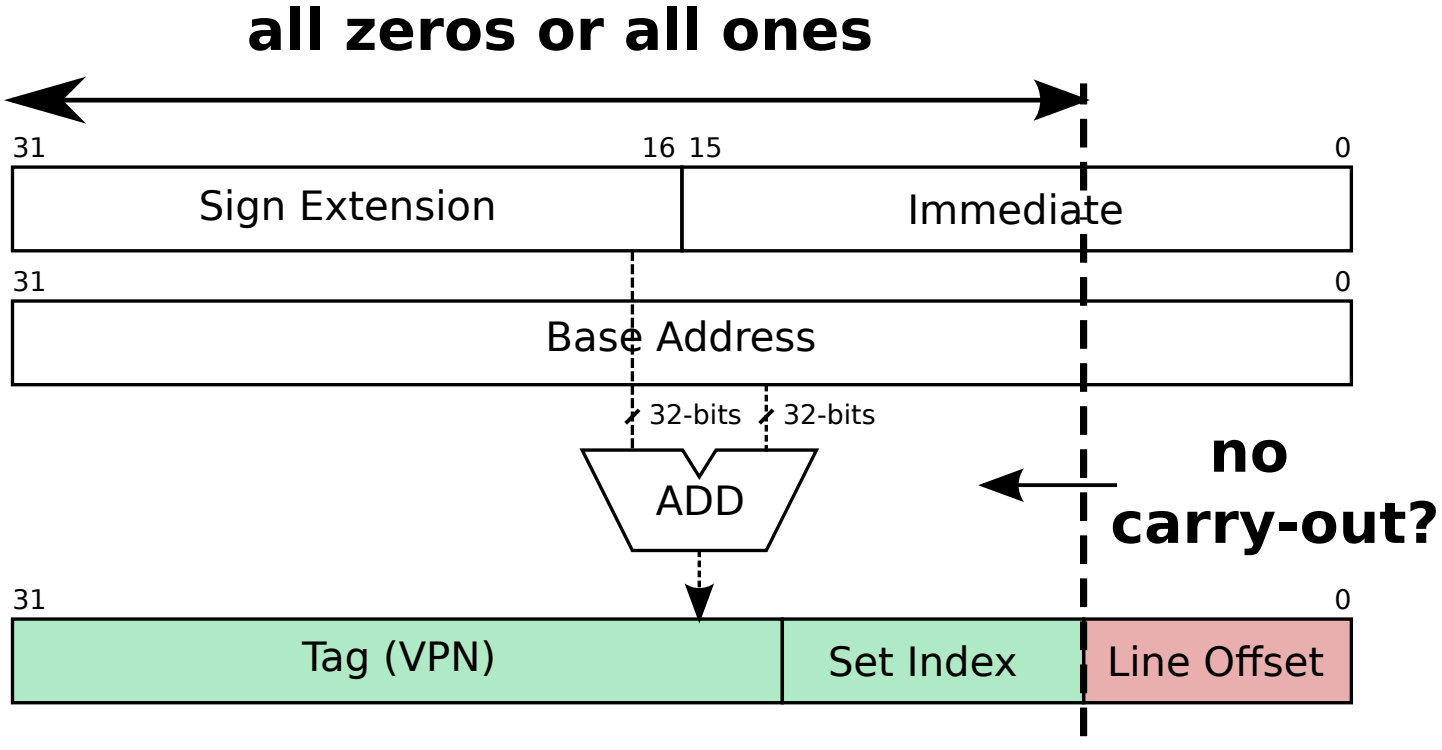
Address Generation and L1 DC Access

- The address generation unit (AGU) calculates the effective address in a stage before the L1 DC is accessed.
- The AGU takes as input:
  - base address from a register value
  - displacement from an immediate value in the instruction
- The figure below assumes a virtually-indexed, physically-tagged (VIPT) organization.



Data Memory Address Calculation

If the displacement is small, then it is possible that the tag and set index portions of the memory address will be the same as these fields in the base address.

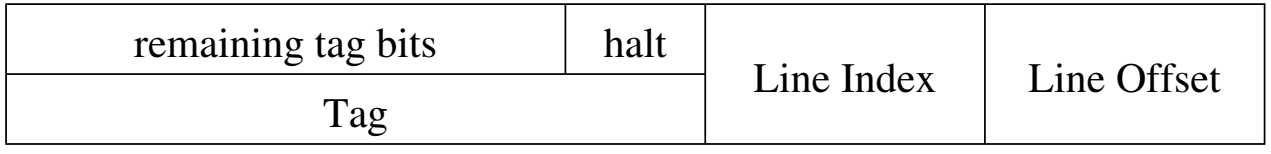


Speculative Tag Access (STA)

- The speculative tag access (STA) approach speculatively accesses the L1 DC tags and the DTLB in the address generation stage when the displacement is no larger than half the magnitude of the L1 DC line size.
- If the index and tag fields of the address are not affected, then at most only a single L1 DC data array need be accessed.
- May lead to structural hazards as the DTLB and the L1 DC tag arrays could be accessed in both the address generation and SRAM access stages.
- *Speculative Tag Access for Reduced Energy Dissipation in Set-Associative L1 Data Caches* by A. Bardizbanyan, M. Sjalander, D. Whalley, P. Larsson-Edefors in IEEE International Conference on Computer Design (ICCD) 2013

Speculative Halt Tag Access (SHA)

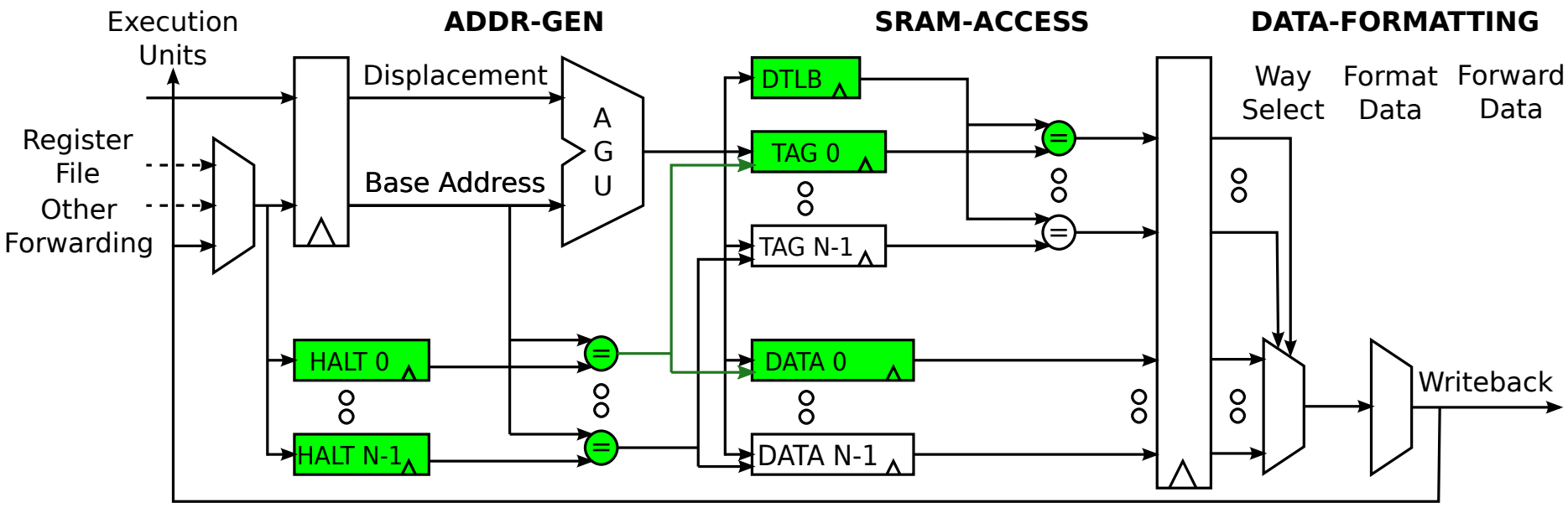
- We propose to speculatively check halt tags in the address generation stage when the displacement is small.
- The speculative halt tag access succeeds when the index and tag fields do not change after adding the displacement.
- For large displacements the L1 DC is conventionally accessed without a halt tag comparison.
- We use the virtual address to avoid speculatively accessing the DTLB and the OS performs page coloring so that the halt tag bits for the virtual and physical address are identical.



Effective Address

Speculative Halt Tag Access (SHA) (cont.)

- Only when a halt tag matches in the address generation stage are the corresponding L1 DC tag and data ways enabled in the SRAM access stage.



Pipeline with Speculative Halt-Tag Arrays and N-Way Cache

Evaluation Framework Used for This Study

- Simulated an in-order processor with a classical 5 stage pipeline.
- L1 DC is 16kB, 4-way set-associative, and has a 32B line size.
- Data translation lookaside buffer (DTLB) has 16 entries and is fully associative, with a 4KB page size.
- RTL implementation synthesized (Synopsys Design Compiler) and placed and routed (Cadence Encounter) to obtain energy values for various events.
- Used the SimpleScalar simulator to count events and estimate total energy usage.

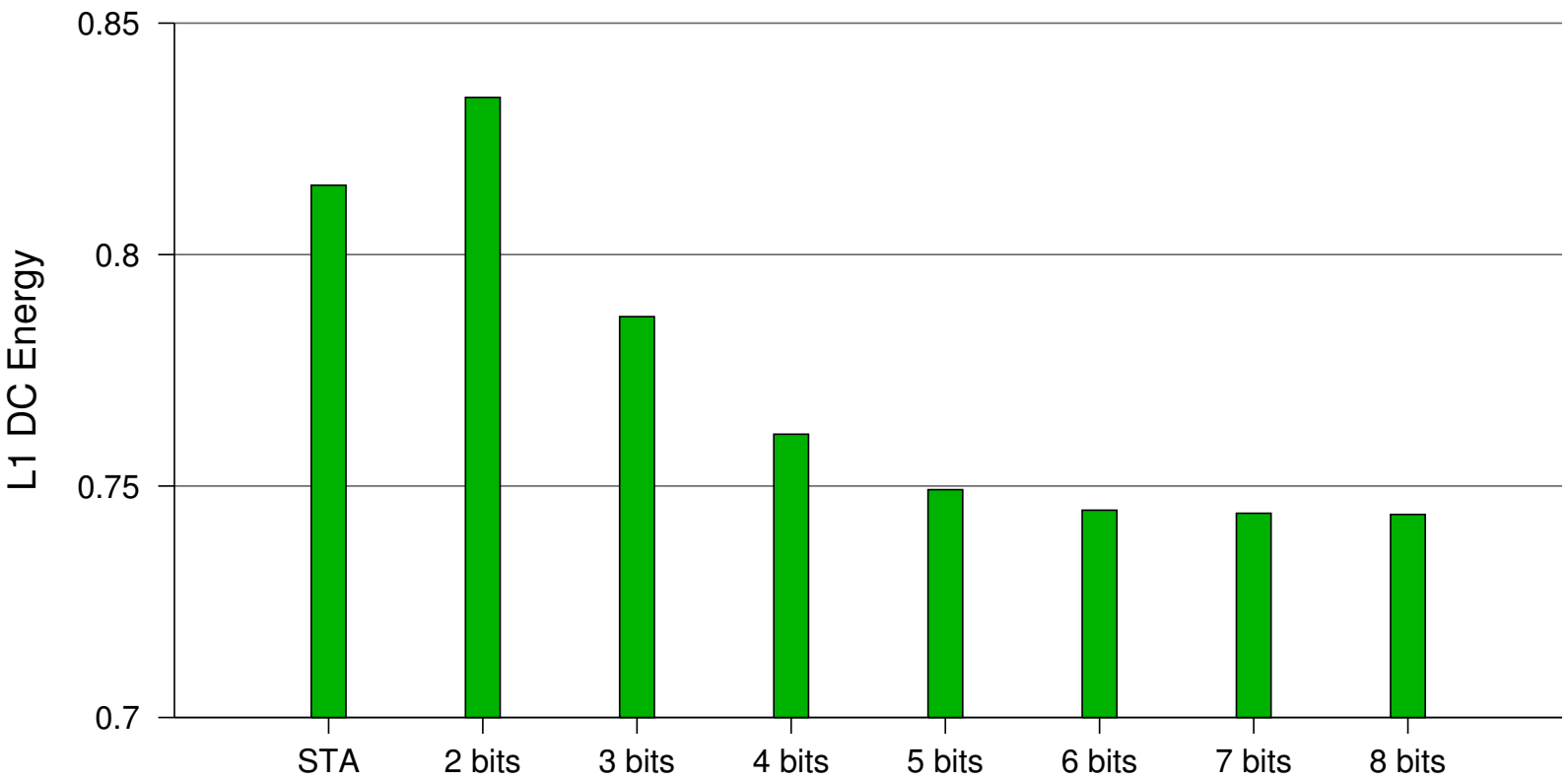
Benchmarks

- 20 benchmarks simulated from the MiBench benchmark suite.

Category	Applications
Automotive	Basicmath, Bitcount, Qsort, Susan
Consumer	JPEG, Lame, TIFF
Network	Dijkstra, Patricia
Office	Ispell, Rsynth, Stringsearch
Security	Blowfish, Rijndael, SHA, PGP
Telecomm	ADPCM, CRC32, FFT, GSM

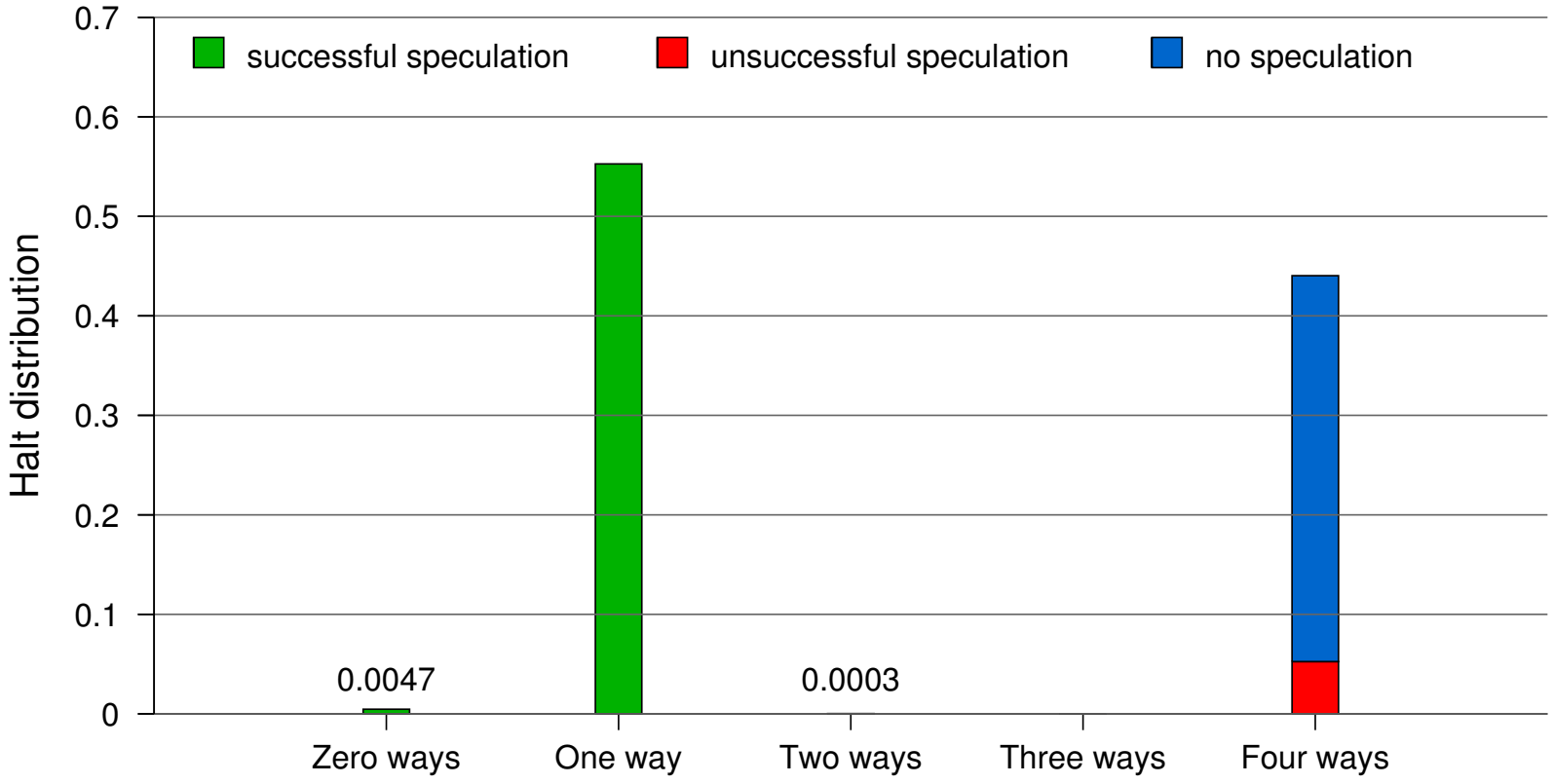
Energy Usage for Different Halt Tag Widths

- STA stands for speculative tag access, where the regular L1 DC tags are speculatively accessed.
- Only had access to a 32-bit wide SRAM macro for the halt tag array where we could store up to four 8-bit halt tags.



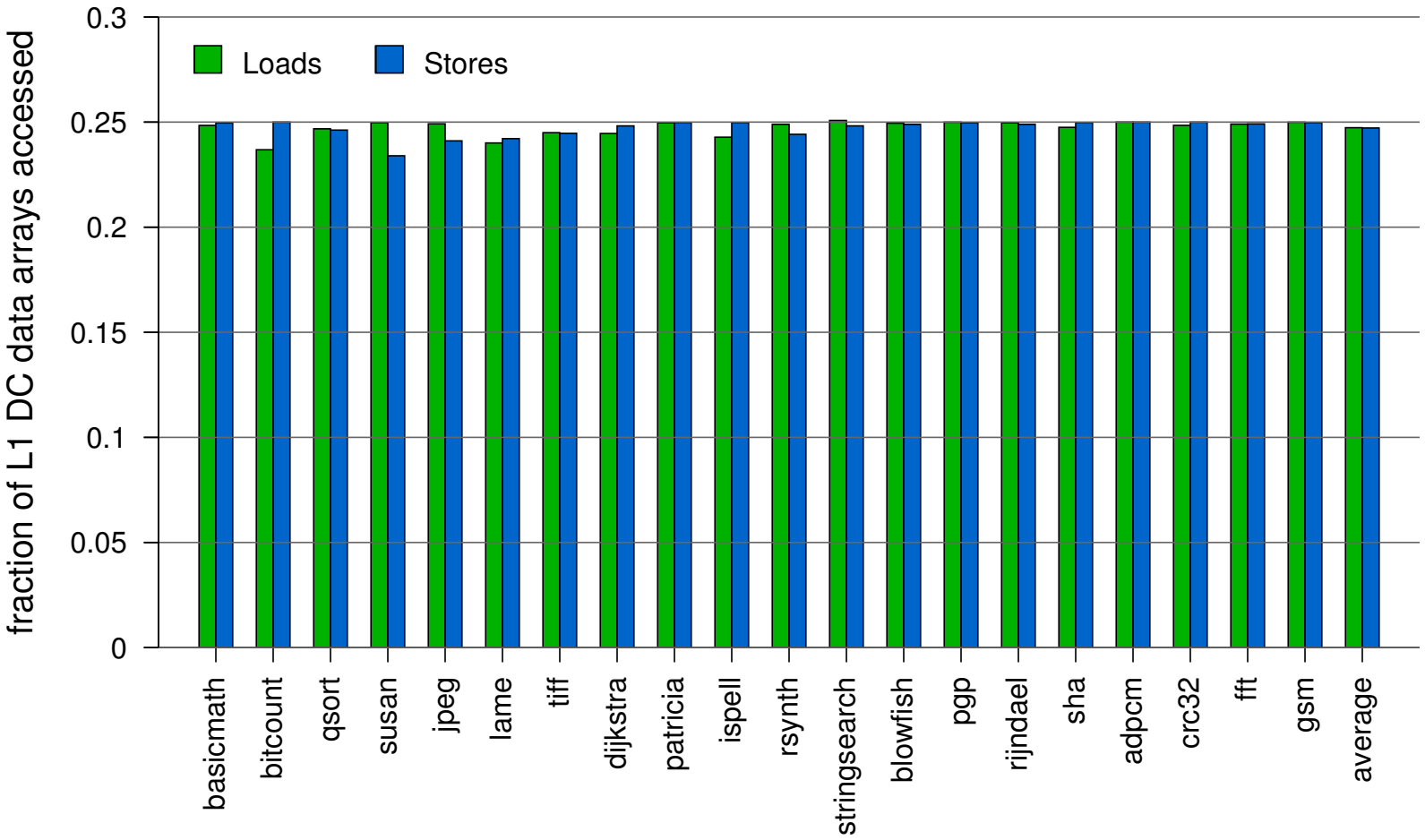
Halt Distribution Cases

- # ways means only # L1 DC tag and data arrays are enabled.
- No speculation means the displacement was too large.
- Over 55% of the accesses enable one L1 DC tag array and one L1 DC data array.



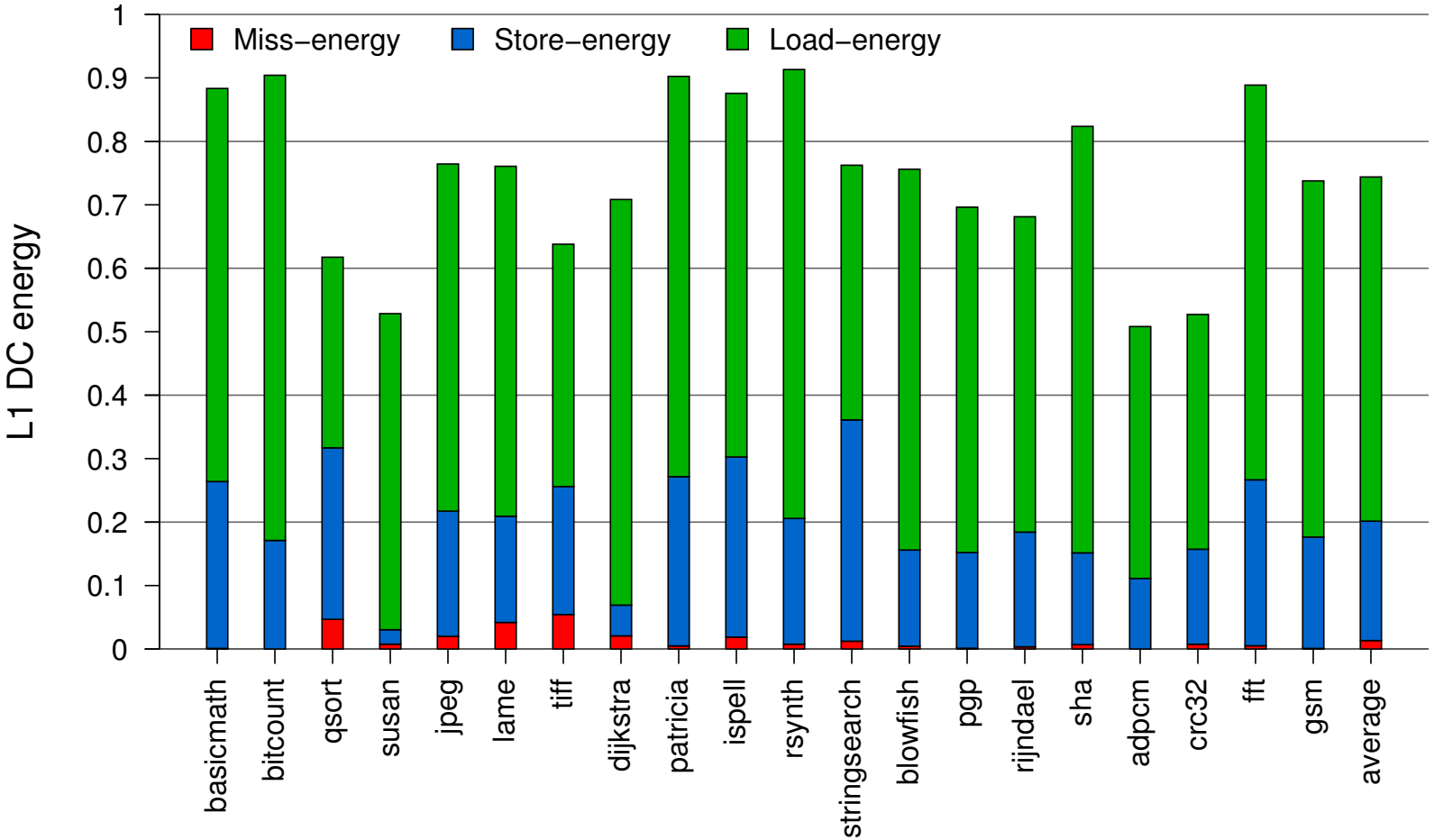
Fraction of L1 DC Data Arrays Accessed per Benchmark

- The average fraction of L1 DC data arrays accessed on successful speculations is slightly less than 0.25.



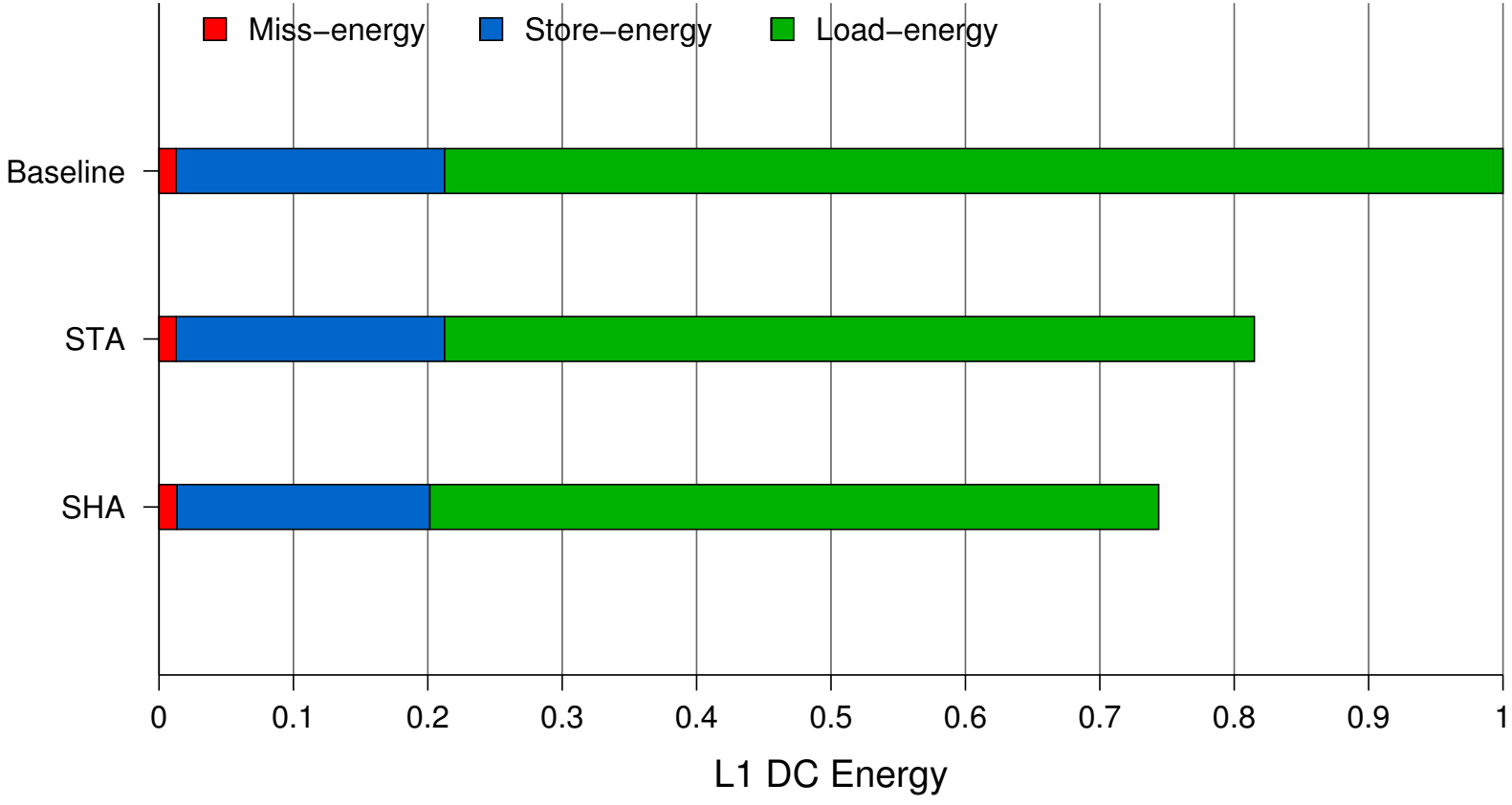
SHA Miss, Store, and Load Energy per Benchmark

- The average energy usage reduction is 25.6% compared to an L1 DC baseline with no halt tags.



Average Energy Distribution

- The speculative halt tag access (SHA) reduces energy usage by 6% more than the speculative tag access (STA) approach.



Conclusions

- Average energy usage reduction of the speculative halt tag access (SHA) approach is 25.6% as compared to a baseline L1 DC that has no halt tags.
- The SHA design is simpler than the speculative tag access (STA) approach since halt tags are accessed in the address generation stage and the DTLB, L1 DC tags, and L1 DC data are accessed in the SRAM access stage.
- Unlike previously proposed way halting techniques, the SHA approach is practical since it can use conventional SRAM chips to implement the L1 DC.
- *Practical Way Halting by Speculatively Accessing Halt Tags* by D. Moreau, A. Bardizbanyan, M. Sjalander, D. Whalley, P. Larsson-Edefors in the IEEE/ACM Design Automation and Test in Europe (DATE) Conference 2016