Energy Characterization of Hardware Data Prefetching

Y. Guo, S. Chheda, I. Koren, M. Krishna, C. Andras Moritz
Electrical & Computer Engineering
University of Massachusetts, Amherst

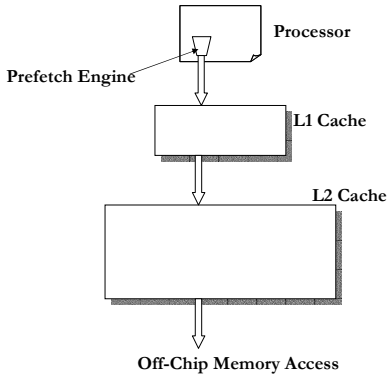
Jan 30th, 2004

Motivation

- Data Prefetching has been successful in hiding memory access latency.
  - Different techniques have been proposed
    - Software: Mowry '94, Lipasti *et al* '95, Luk & Mowry '96
    - Hardware: Smith '78, Baer '91, Roth *et al* '98, Cooksey *et al* '02.
- Power and energy consumption becomes more and more important in recent years.
- How does prefetching affect on-chip energy consumption?
  - The scope of my presentation today
  - On longer term we are interested in developing new energy-aware prefetching solutions.

How Does Prefetching Work?

- The Prefetch Engine decides which data (address) to be prefetched.
- No prefetching if data is already in L1 Cache.



Sources of Prefetching Energy

- Extra Tag-checks in L1 cache
  - When a prefetch hits in L1.
- Extra memory accesses to L2 Cache
  - Due to useless prefetches from L2 to L1.
- Extra off-chip memory accesses
  - When data cannot be found in the L2 Cache.
- Prefetching hardware: data (history table) and control logic.

Prefetching Techniques Used

- **Prefetching-on-miss** (POM) - basic technique
- **Tagged Prefetching** - A variation of POM.
- **Stride Prefetching** [Baer & Chen]– Effective on array accesses with regular strides
- **Dependence-based Prefetching** [Roth & Sohi]– Focuses on pointer-chasing relations
- **Combined** Stride and Pointer Prefetching [**new**] – Applied on general-purpose programs

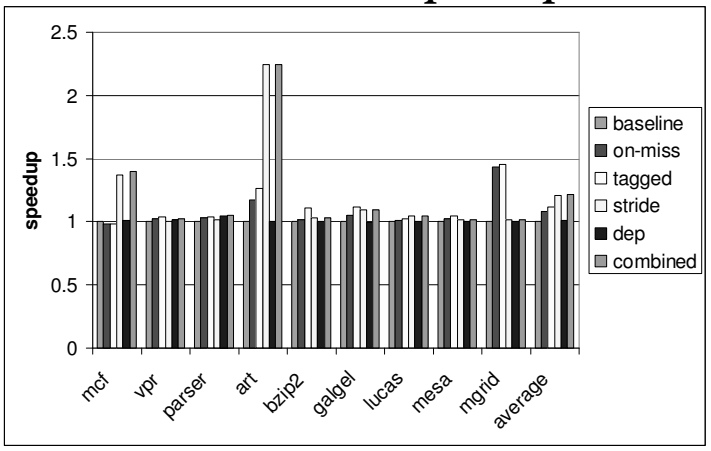
Experimental Setup

- SimpleScalar
  - Implementation of prefetching techniques
  - Gather statistics which will be used for energy estimation.
- Energy Estimation for L1 & L2 cache accesses
  - Spice simulation with 100-nm BPTM technology
- Benchmark Suites
  - SPEC2000 – Array-intensive benchmarks
  - Olden – Pointer-intensive benchmarks

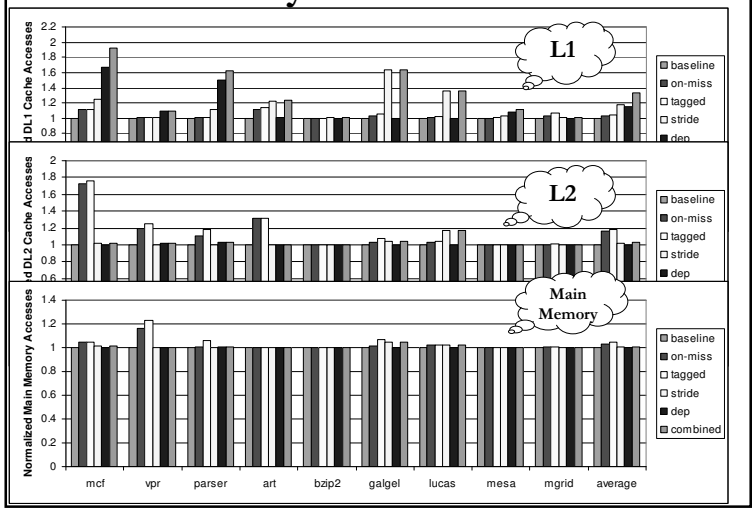
Cache Configuration & Power

Parameter	L1	L2
size	32KB	256KB
tag array	CAM-based	RAM-based
associativity	32-way	4-way
bank size	2KB	4KB
# of banks	16	64
cache line	32B	64B
Power (mW)		
P_tag	6.5	6.27
P_read	9.5	100.52
P_write	10.3	118.62
P_leakage	3.1	23.0
P_reduced_leakage	0.62	1.15

Performance Speedup

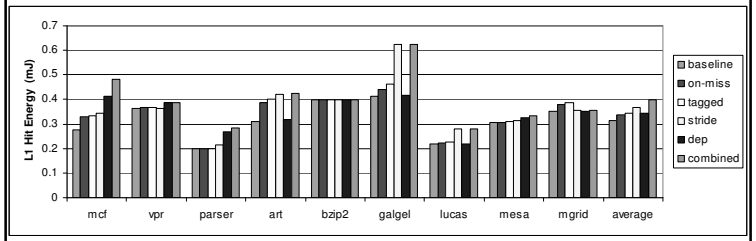


Memory Traffic Increase



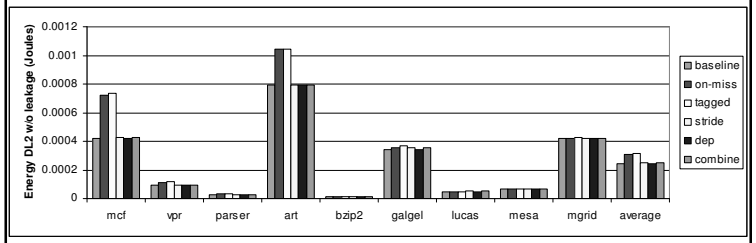
Dynamic Energy Consumption - L1

- L1 Cache Hit Energy increased significantly due to extra tag lookups.



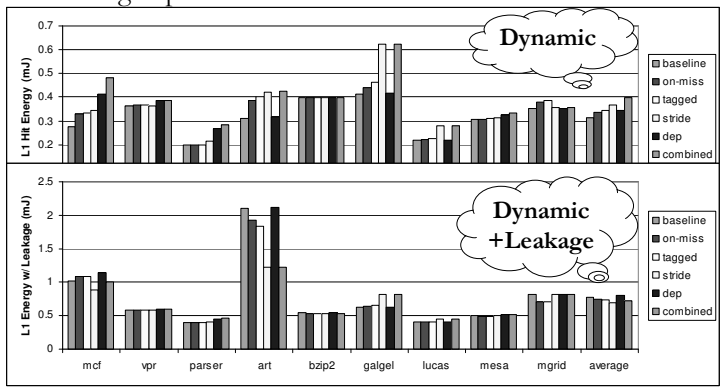
Dynamic Energy Consumption – L2

- **L2 Cache Hit Energy increased slightly in most situations.**



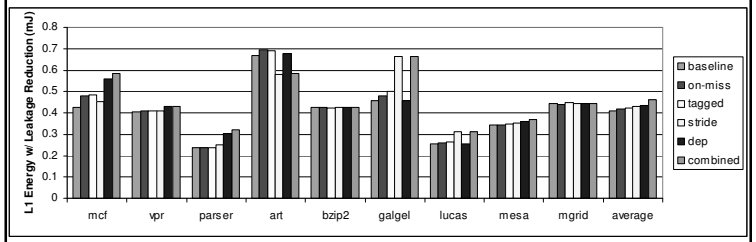
Adding Leakage Energy for L1

- Leakage dominates the total energy consumption without leakage optimization.

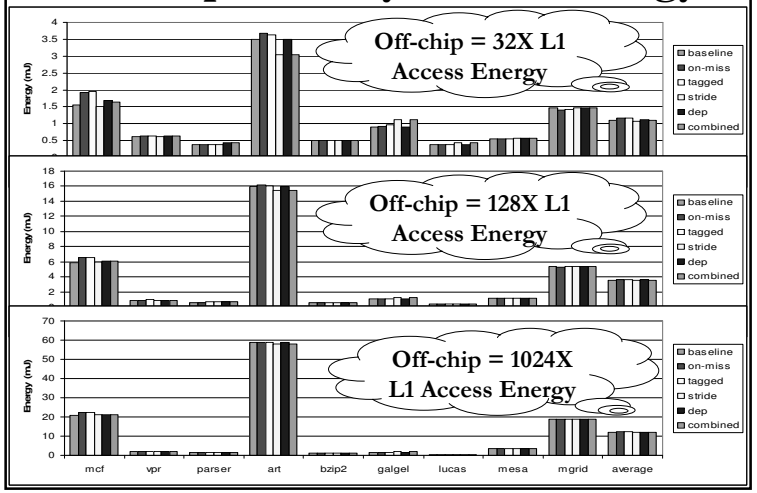


Leakage Reduction Techniques

- **Many leakage optimizations proposed: body biasing, dual Vt, VTCMOS, MTCMOS, asymmetric cells, etc**
  - **E.g., leakage can be reduced by 7X for writes and 40X for reads in cells [Azizi et al ISLPED 2002, evaluated for 130-nm]**
- **We assume that cache leakage could/will be reduced by 80% with circuit techniques.**

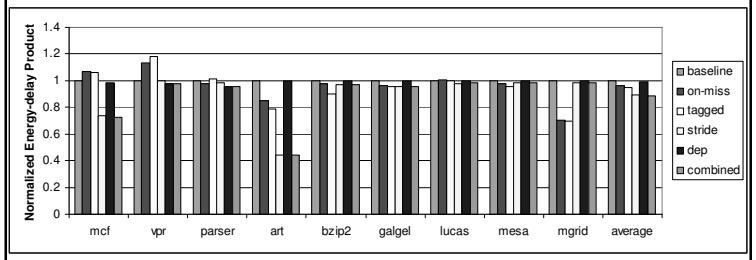


Off-Chip Memory Access Energy



Energy-Delay Product

- **Energy-delay product improves with prefetching in most cases.**



Conclusion

- Prefetching can be considered as an energy reduction technique as well, esp. in deep submicron tech. where leakage becomes dominate.
- Aggressive prefetching techniques increase L1 access energy significantly due to extra tag-checks.
  - We are working on a new technique to improve it.
- Effective prefetching techniques consistently improve energy-delay products (EDP) due to performance improvements.