Memory Redundancy Elimination to Improve Application Energy Efficiency

Keith Cooper and Li Xu
Rice University
October 2003

Techniques to Improve Energy

- Circuit and Architecture Level
  - Dynamic Voltage Scaling (DVS)
  - Pipeline gating
  - Cache partitioning
- Application Level Techniques
  - Optimize behavior to improve energy

Approach

- Profile Energy of Application Execution
  - Run SPEC2000 and MediaBench
  - Correlate Execution to Energy Consumption
- Identify and Evaluate Energy Saving Code Transformations

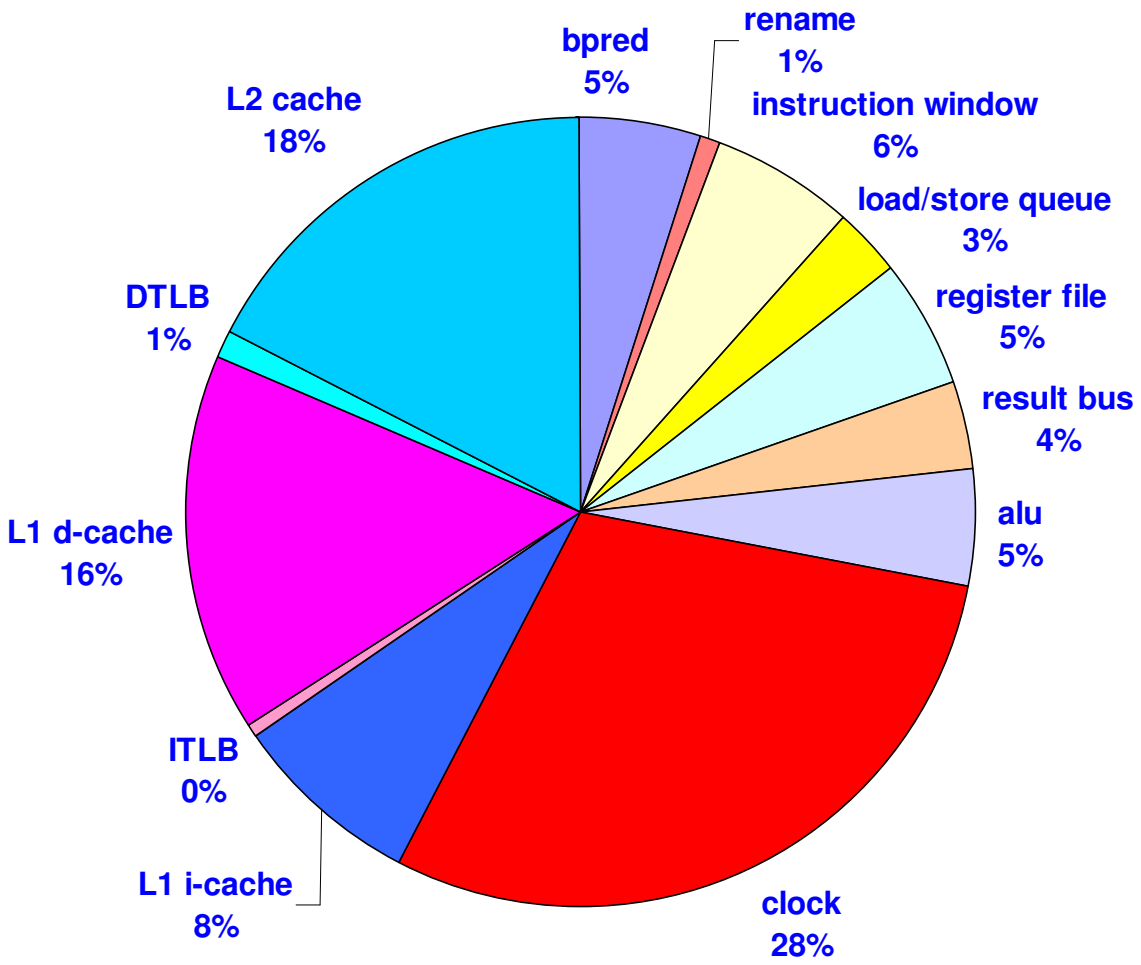
Energy Profiling: Benchmarks

Benchmark	Description	Input
adpcm	16-to-4 bit voice encoding	clinton.pcm
g721	CCITT G.721 voice encoding	clinton.pcm
gsm	GSM speech encoding	clinton.pcm
epic	Pyramid image encoding	test_img.pgm
pegwit	Elliptic curve public key encryption	news.txt
mpeg2dec	MPEG-2 video decoding	child.mpg
181.mcf	Combinational optimization	test input
164.gzip	Compression	test input
256.bzip2	Compression	test input
175.vpr	FPGA placement and routing	test input
197.parser	Link grammar parser of English	test input
300.twolf	Circuit placement and routing	test input

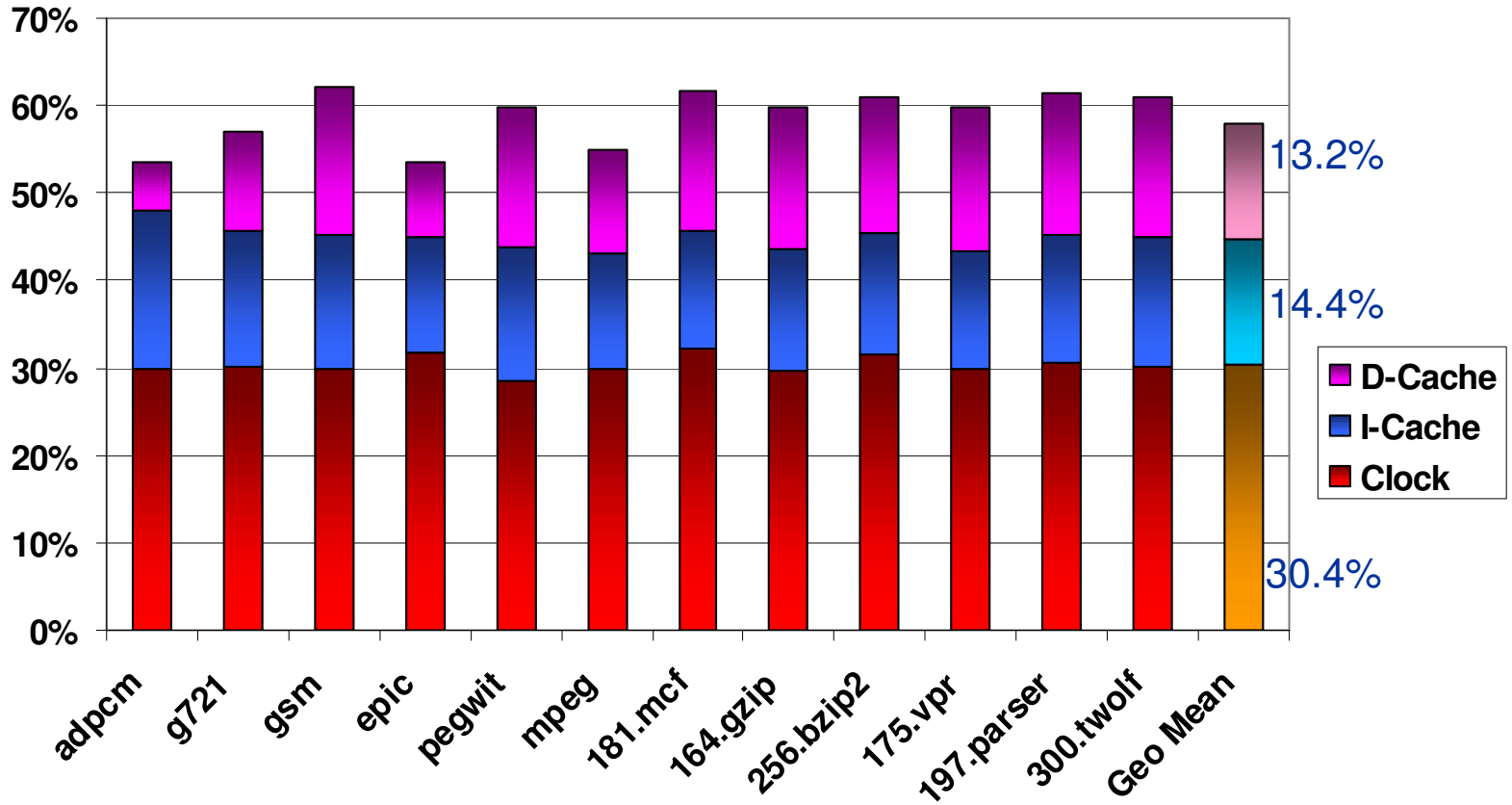
Energy Profiling: Testing Setup

- Use SimpleScalar and Wattch
  - Compiled with SimpleScalar gcc –O4
  - Run on out-of-order superscalar simulator with Wattch module
  - Configuration
    - Architecture: models Alpha 21264
    - Wattch: 0.35μm, 600MHz, Vdd=2.5V

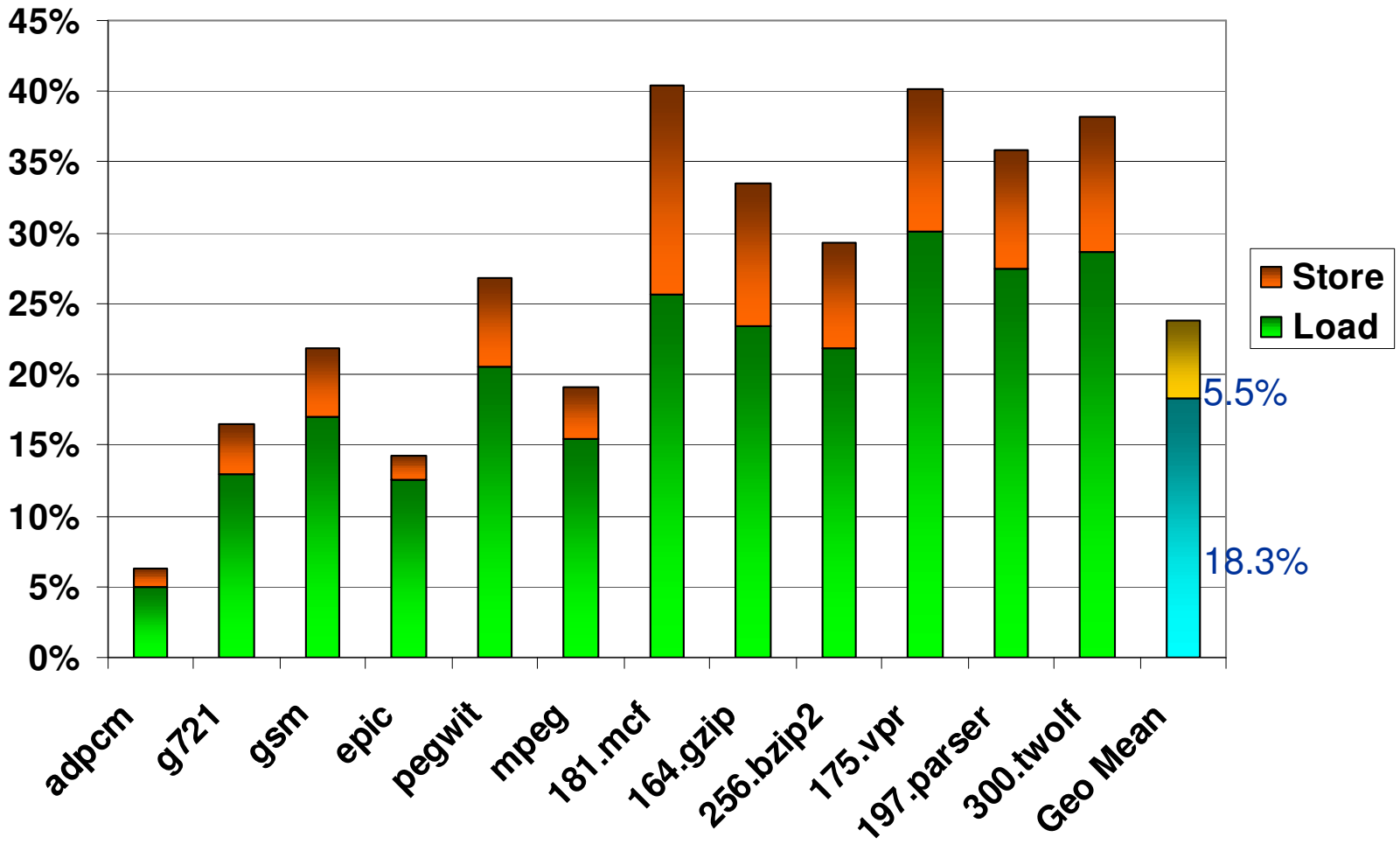
Dynamic Power of Components



Energy of Clock and Caches



Dynamic Load and Store Count



Memory Redundancy

- Redundant loads and stores

```
void foo(X* p)                foo_asm:
{
    …p->field_a…                  ld (p+offset) =>r
                                      ……
    …p->field_a…                  ld (p+offset) =>r
                                      ……
    …p->field_a…                  ld (p+offset) =>r
}
```

Memory Redundancy Elimination to Improve Energy Efficiency

- Reduce execution cycle count
  - Save energy in clocking network
- Reduce I-Cache accesses
  - Save energy in I-Cache
- Reduce D-Cache accesses
  - Save energy in D-Cache

Memory Redundancy Detection

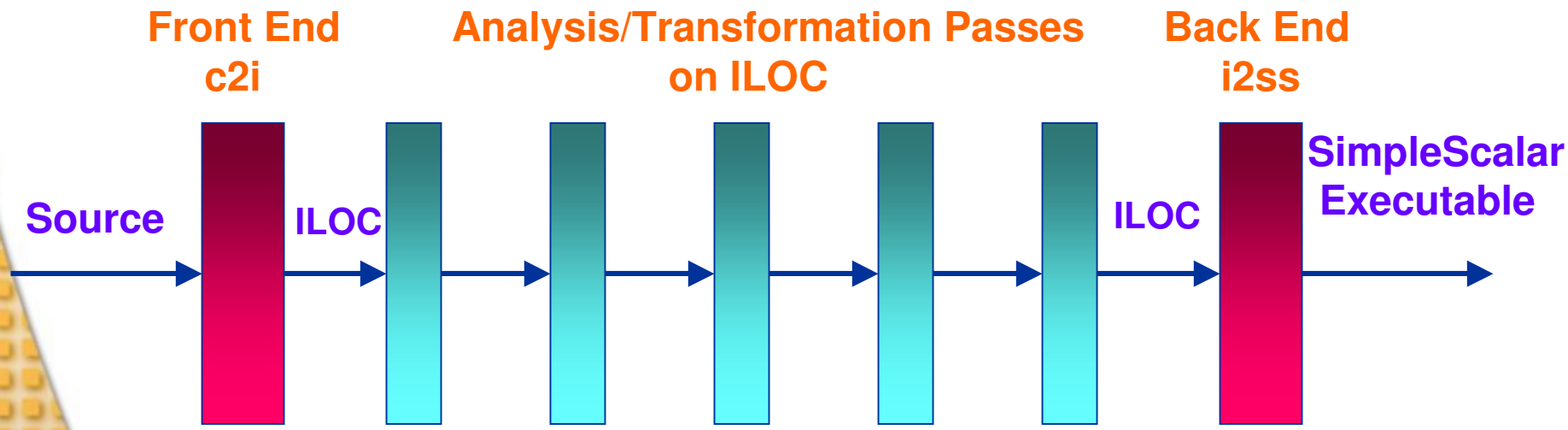
- Want to know P(adr, v) =?= Q(adr', v')
- Global value numbering on memory operations [MSP '02]
- Annotate P,Q with mem state info
- Unified analysis for both scalar and memory redundancy
  - Detect more redundancies due to interaction of scalar and memory values

Memory Redundancy Elimination

- Recast scalar CSE (common sub-expression elimination) and PRE (partial redundancy elimination)
- Solve data flow system to remove memory redundancy
  - Treat loads the same way as scalar
  - Model dependence using mem state info
  - Details in paper

Experimental Setup

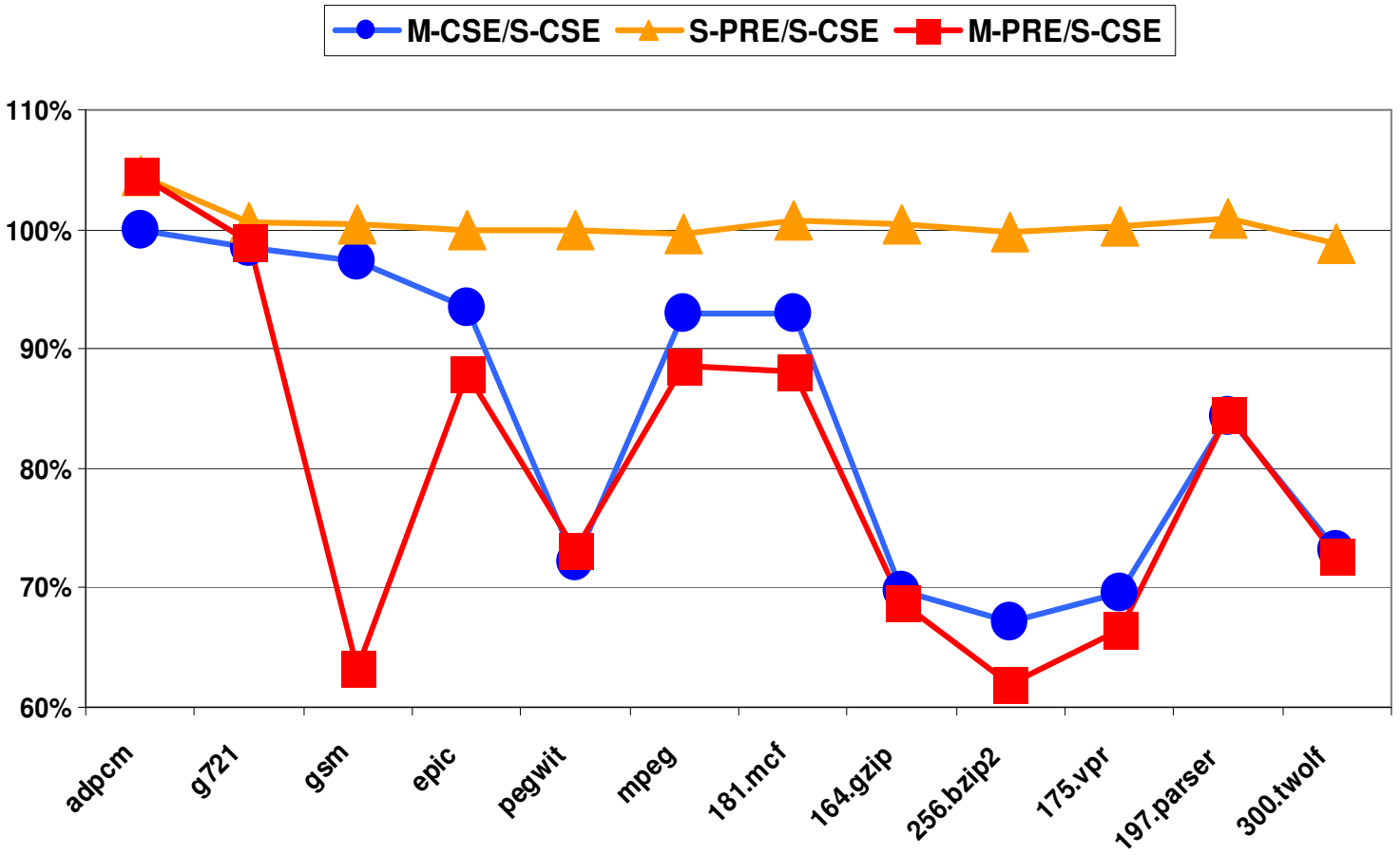
- Use Rice ILOC compiler
- Backend creates SimpleScalar binaries



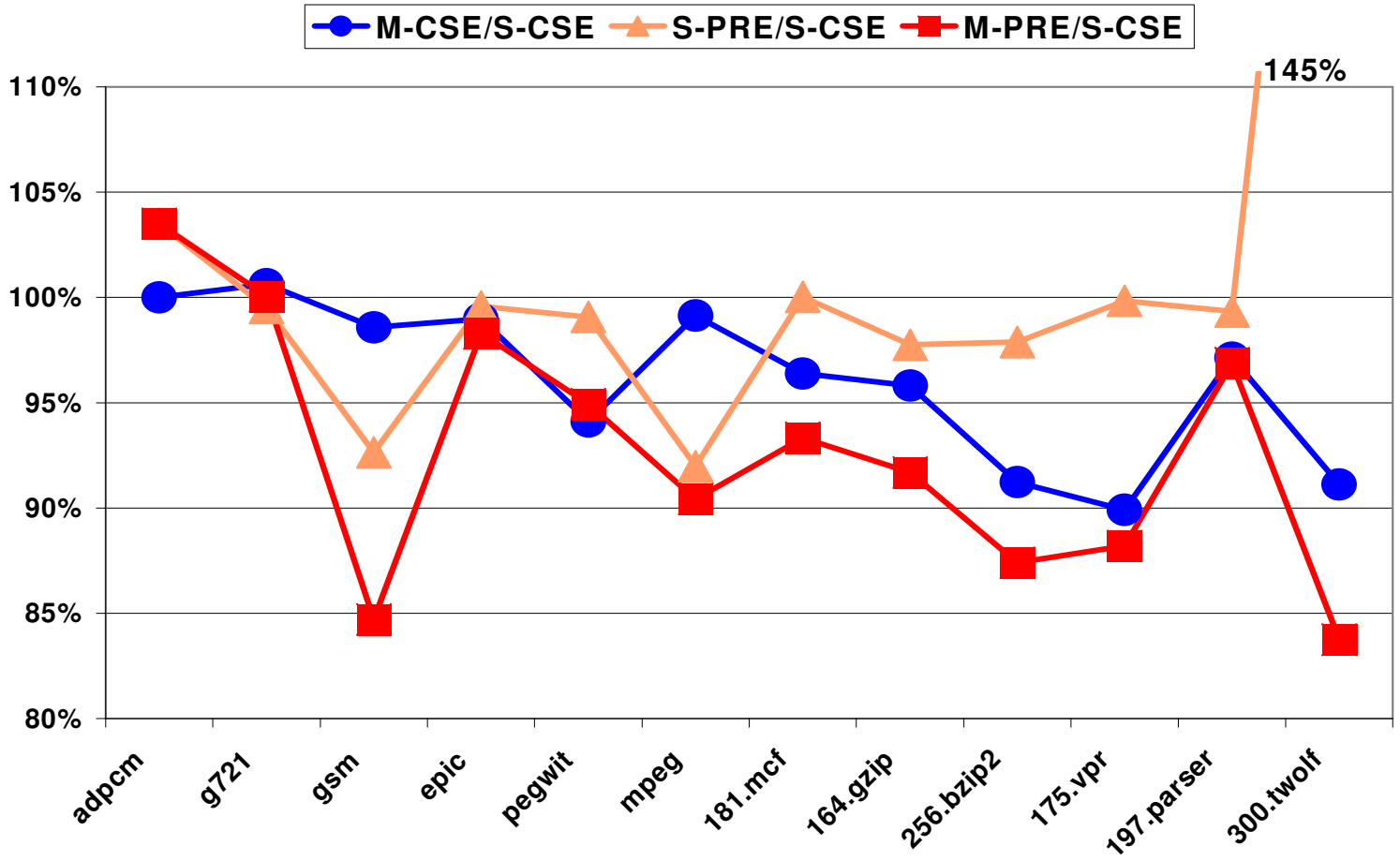
Experimental Setup, Cont'd

- Compare scalar CSE (S-CSE) and scalar PRE (S-PRE) against memory CSE (M-CSE) and PRE (M-PRE)
  - Implement as ILOC passes
  - Run SimpleScalar and Wattch to collect run-time and energy stats

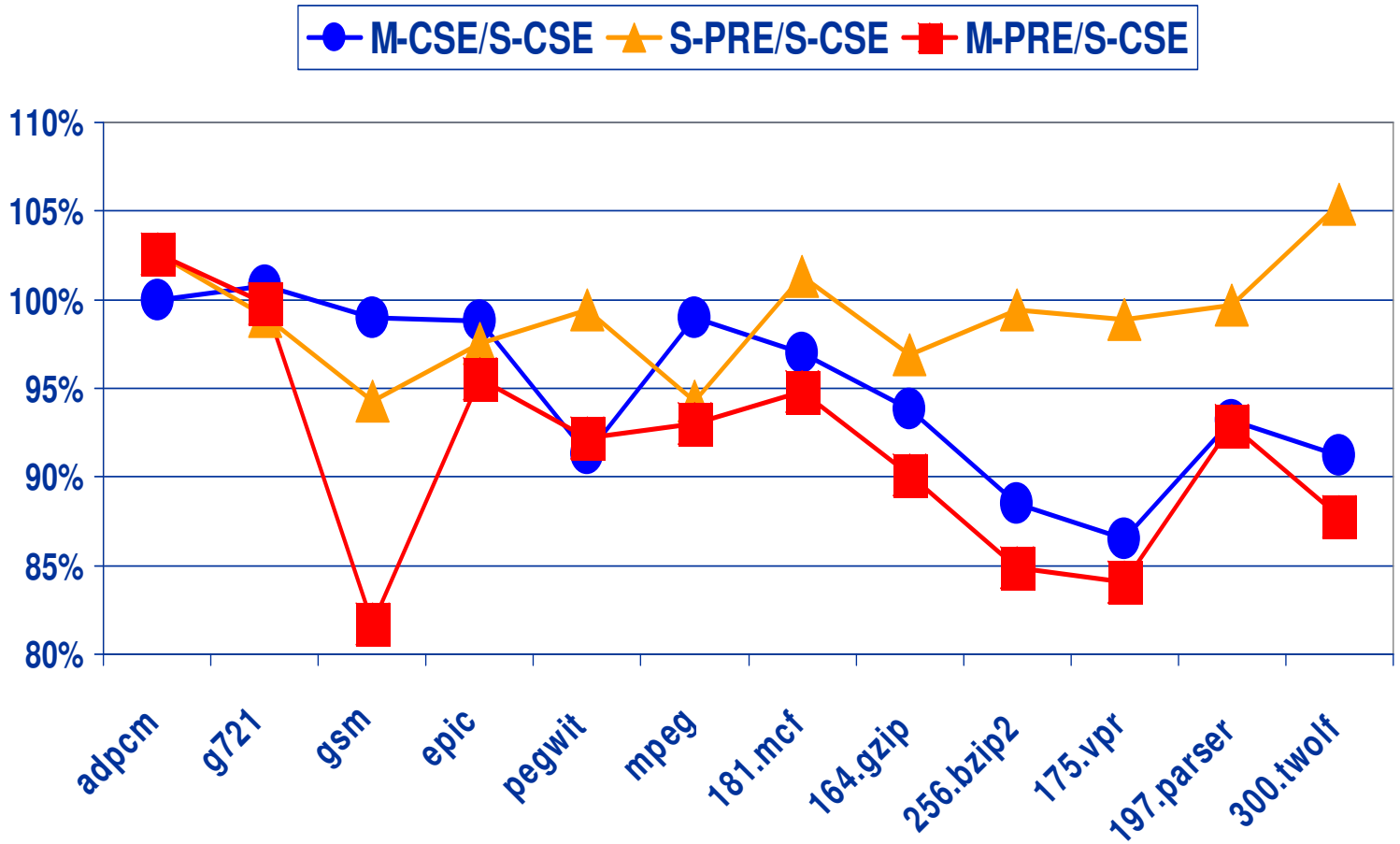
Result: Dynamic Loads



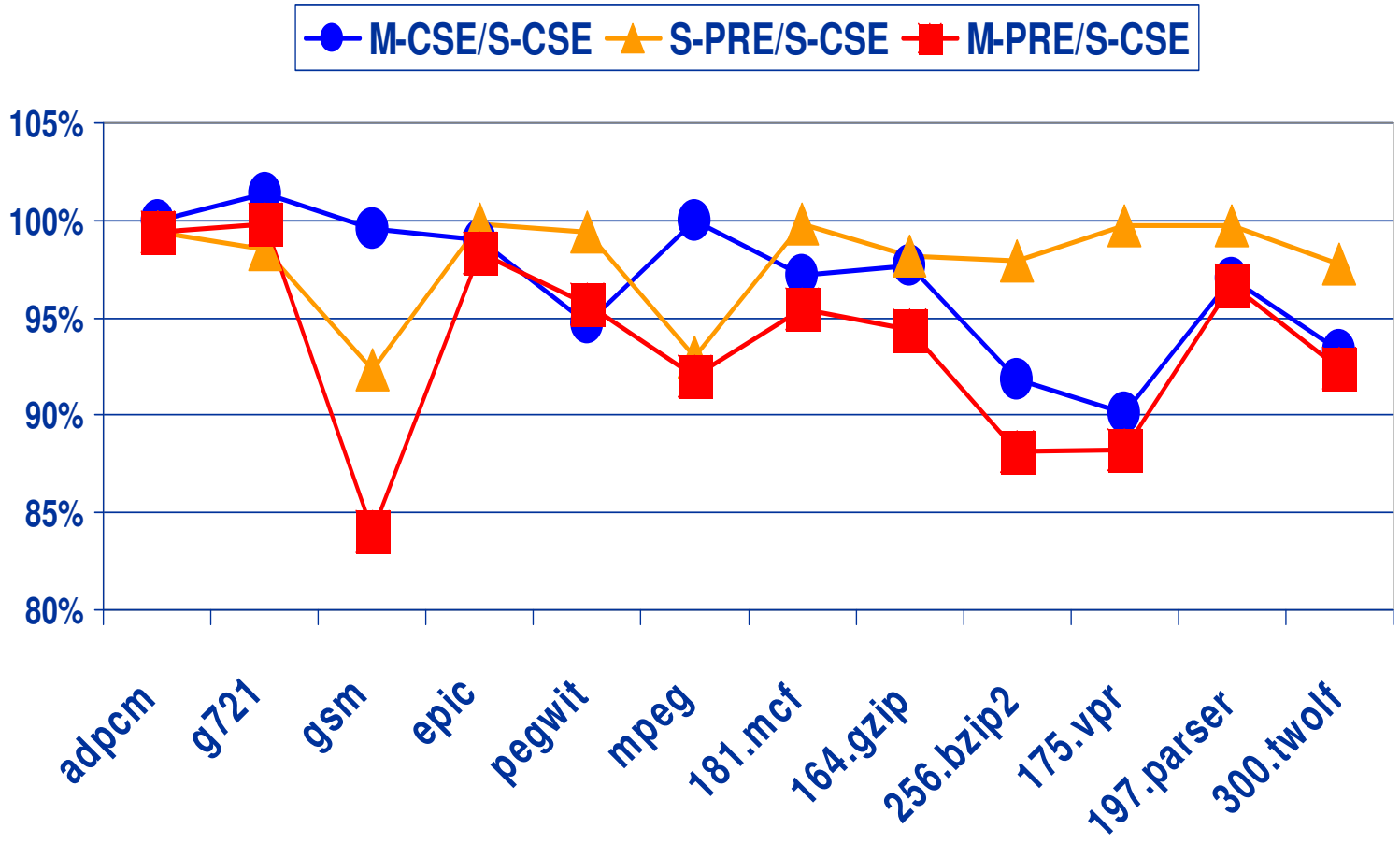
Result: Execution Cycles



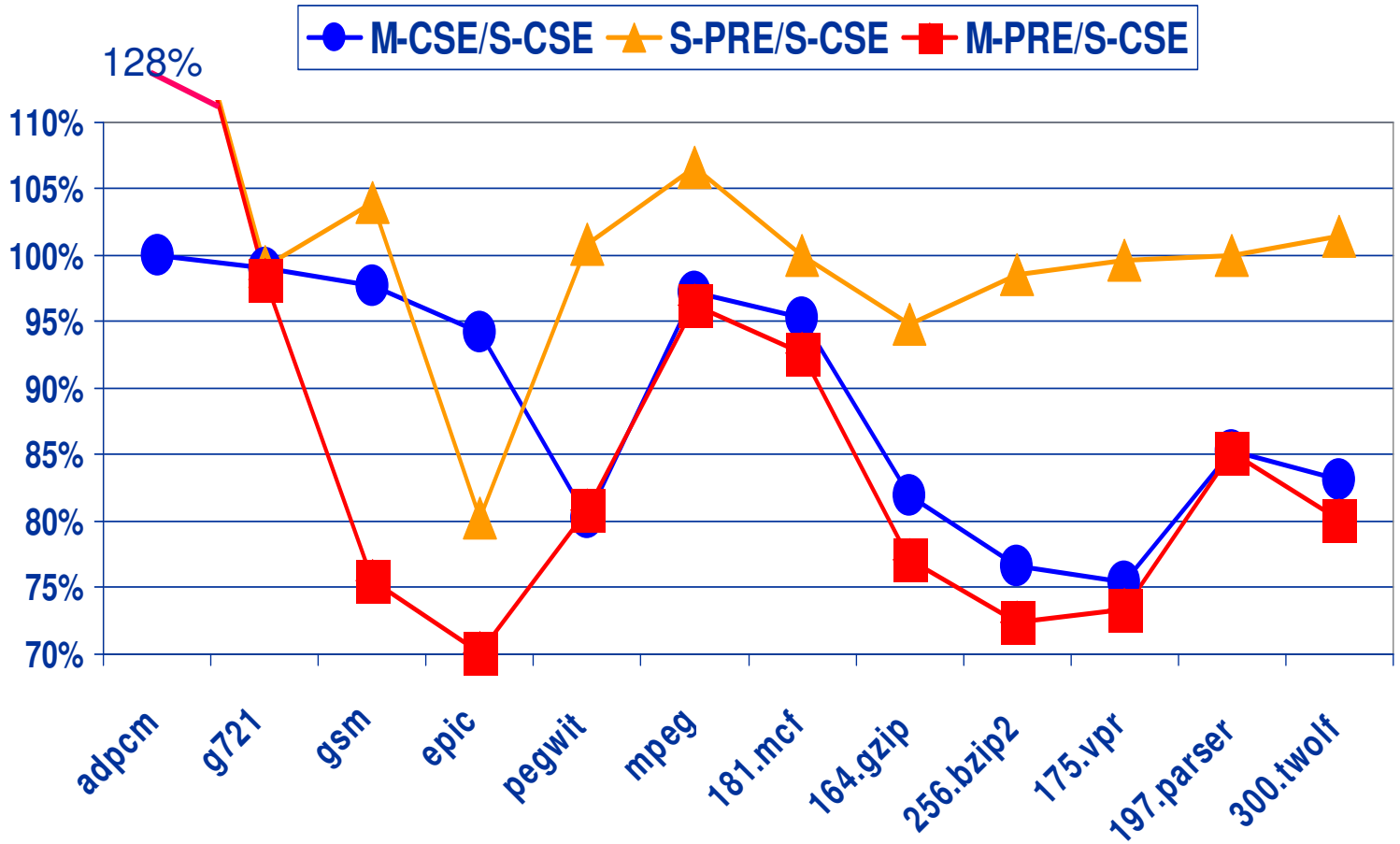
Result: Clock Energy



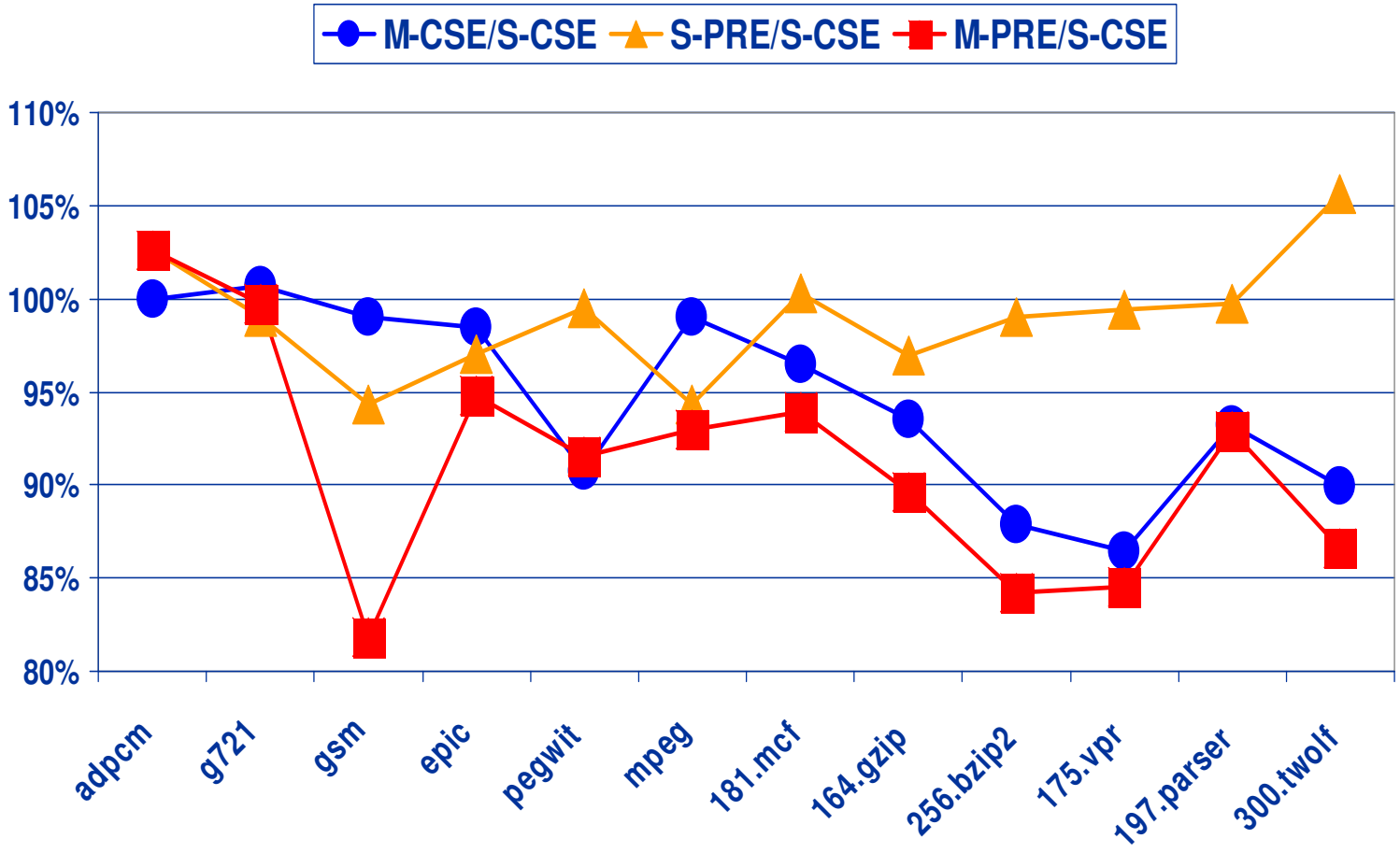
Result: I-Cache Energy



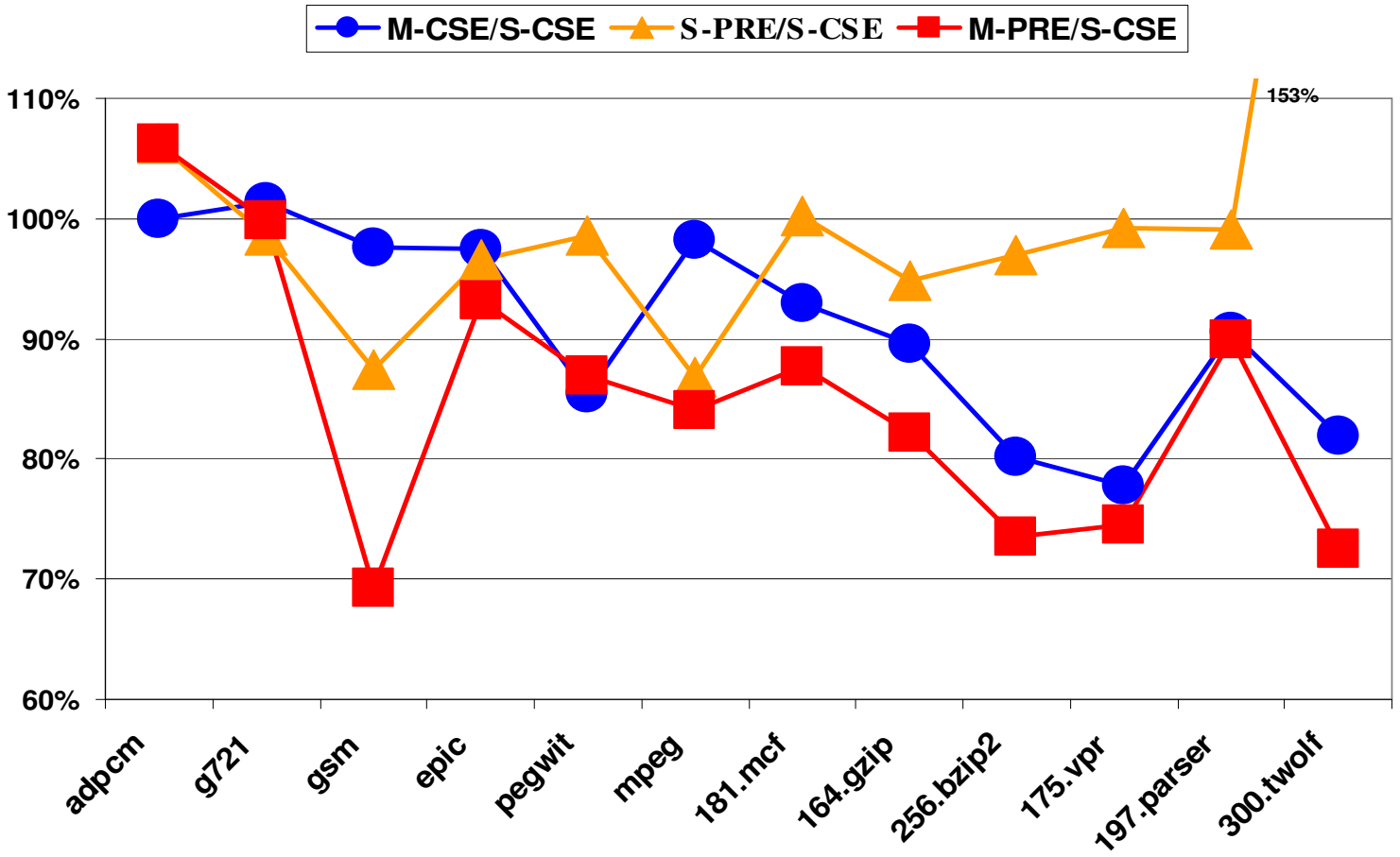
Result: D-Cache Energy



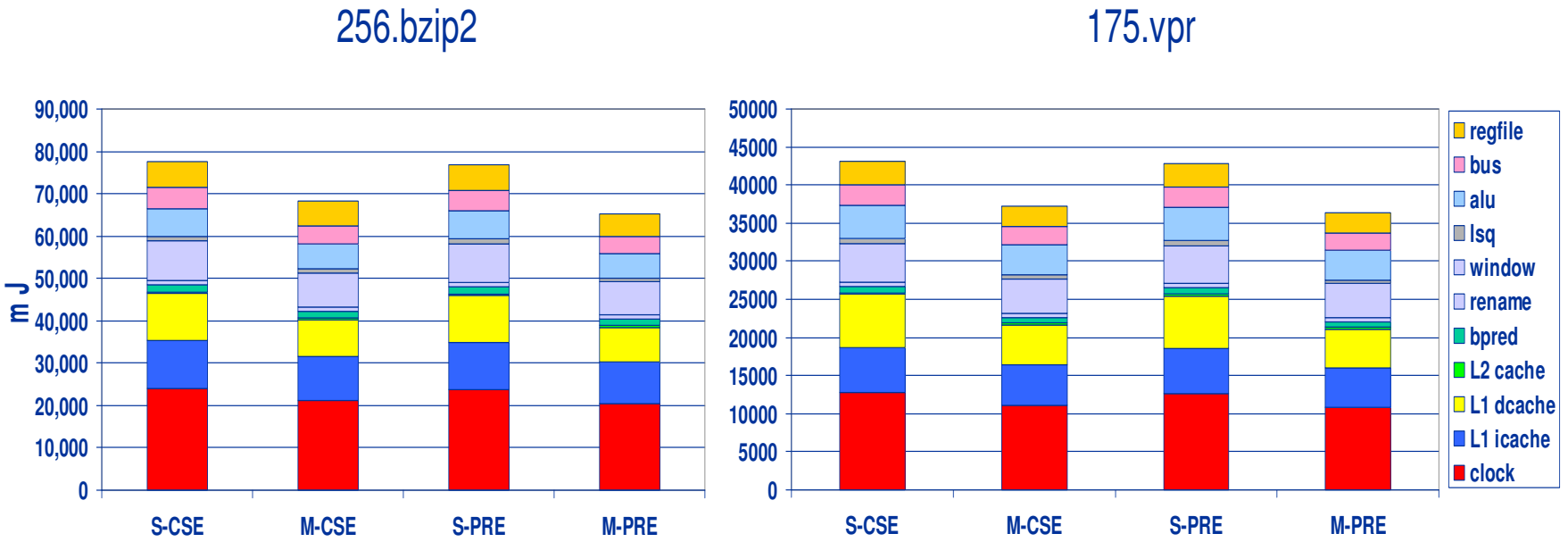
Result: Total Energy



Result: Energy-Delay Product



Result: Application Energy Breakdown



	Clock	I-Cache	D-Cache	Total
256.bzip2	12%, 15%	8%, 10%	23%, 24%	12%, 15%
175.vpr	13%, 15%	10%, 12%	25%, 26%	14%, 15%

Conclusions

- Application energy profiling
  - Top energy consuming components: clocking network and caches
- Memory redundancy elimination to improve energy efficiency
  - Reduce energy in clock, I-Cache, D-Cache
  - Results: up to 15% reduction in energy, 24% in energy-delay product on test apps