Evaluation and Comparison of Existing Cache Designs Implemented as an IRAM

Christoforos E. Kozyrakis and Helen J. Wang

http://www.cs.berkeley.edu/~kozyraki/project/cs252

Intelligent RAM

- Integration of logic and memory in a single DRAM chip:
  – lower memory latency
  – higher memory bandwidth
  – higher energy efficiency, flexible memory granularity, board space savings
- Potential Drawback:
  – logic slowdown in a DRAM process.

Motivation

- Evaluate the expected performance of IRAM as a function of logic, SRAM and DRAM speed in a DRAM process. In which cases does IRAM have a performance advantage?

- Compare the IRAM implementations of a simple and a complex processor/cache architecture. Which one is more appropriate for IRAM?

Performance Factors

Factor	*Optimistic Slowdown*	*Pessimistic Slowdown*
Logic	1	2
SRAM access	1.1	1.3
DRAM access	0.1	0.2

- All factors expressed as slowdown for uniformity.

IRAM Organization

- Processor core.
- On-chip SRAM I/D caches for performance reasons.
- On-chip DRAM.
- No SRAM L2 cache needed because on-chip DRAM equally fast.
- IRAM implementations of Alpha and P6: same organizations in a DRAM chip, with no L2 cache.

Methodology

- Use two existing architectures to evaluate the IRAM performance of similar organizations : the Alpha 21064 and the Pentium Pro.
- Analytical model for execution time calculation of IRAM implementations:

  Time = processor time component * ***logic slowdown*** +
  (L1 miss count * original memory access cost) /
  DRAM speedup

Measurement Method

- Monitor execution using hardware performance counters:
  - capture processor and memory events
  - no interference
  - more accurate measurements
  - faster and easier than simulation
- Data collection based on:
  - interrupt handling on counter overflow (Alpha)
  - reading the actual values of the counters (P6)

Processor Characteristics

	Alpha 21064	*Pentium Pro*
Year	1992	1996 (Q4)
Architecture	Alpha	IA-32
Clock speed	133MHz	200MHz
Pipeline stages (int,fp)	7,10	14,16
Issue rate	2	3
Out of order issue	no	yes
L1 Icache	8KB (DM)	8K (4-way)
L1 Dcache	8KB (DM)	8K (2-way)
L2 Cache	256KB (off-chip)	256KB (off-chip)
Memory latency (L1/L2/DRAM)	22.5ns/52.5ns/266.5ns	15ns/35ns/285ns
Transistors	1.7M	5.5M
Die size	234mm2	306 mm2
Technology	0.75um CMOS	0.35um BiCMOS
SPECint92	74.7	320
SPECfp92	112.5	283

Benchmarks

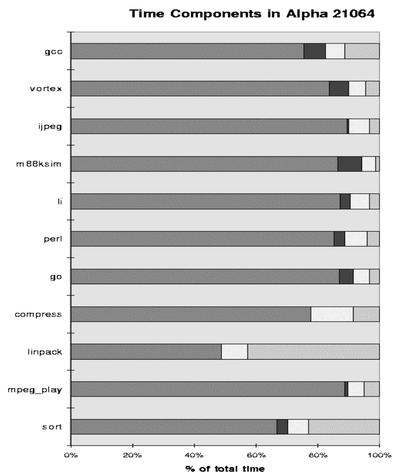
Benchmark	*Characteristics*
gcc, vortex, ijpeg, li, m88ksim, perl, go, compress	From SPEC95 suite (reference input)
linpack	1000x1000 linear equation solver (double precision)
mpeg_encode	Encodes 48 720x480 color frames into a mpeg video file
sort	Disk-to-disk sort of 100-bytes records in a 21 MB database

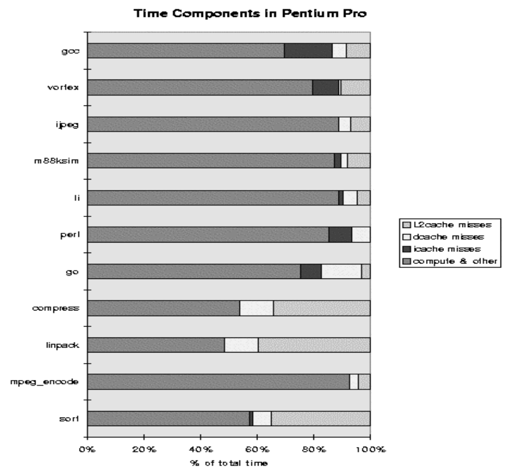
• Benchmarks selected from various fields of applications with different resource requirements.

Performance Evaluation

- Program execution time analysis.
- Speedup of IRAM over conventional implementations.
- Performance comparison of IRAM Alpha and IRAM P6.
- Performance comparison of IRAM Alpha and original P6.
- Effect of off-chip accesses in IRAM.
- Effect of IRAM on specific program components.

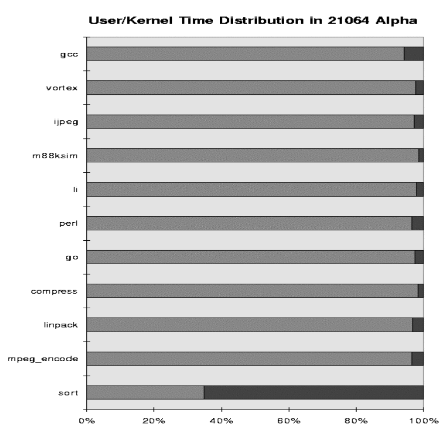
Time Components

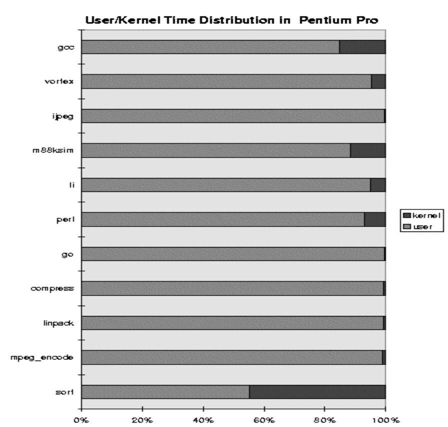




• Sort and Linpack are memory intensive, others are CPU intensive

Kernel/User time distribution

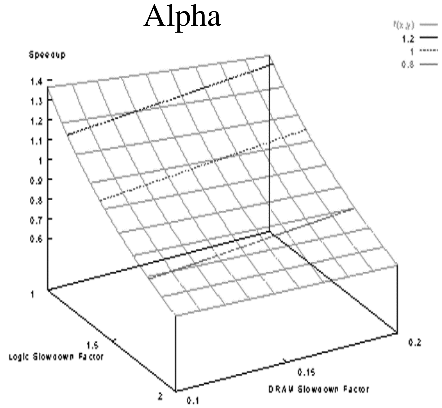




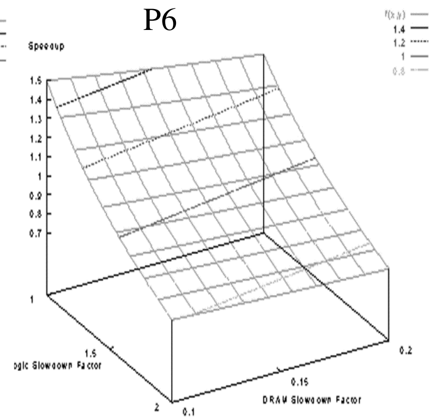
• Only sort spends more than 10% in kernel mode.

Sort: IRAM Vs. Conventional

Alpha



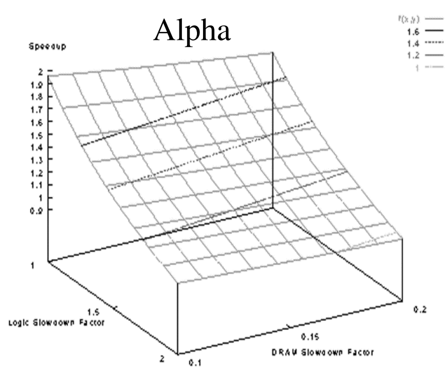
P6



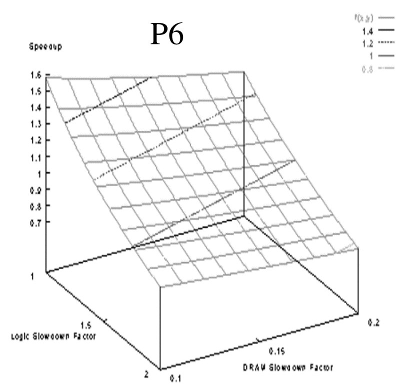
- Speedup > 1 when the logic slowdown factor < 1.5.
- Maximum Speedup: 1.32 (Alpha), 1.5 (P6).

Linpack: IRAM Vs. Conventional

Alpha



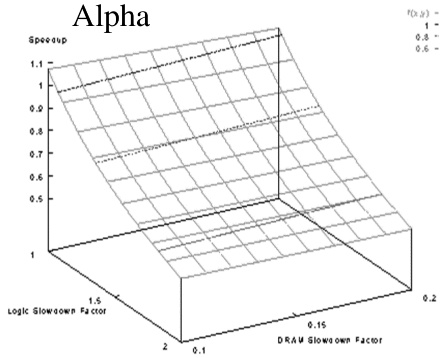
P6



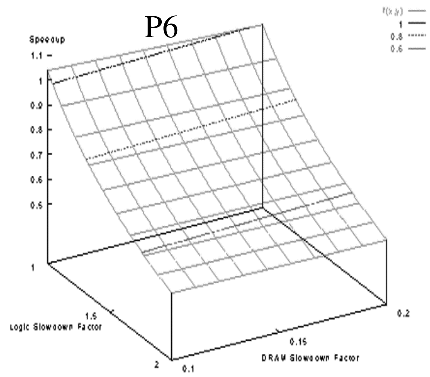
- Alpha: speedup for all factor values, maximum 1.95.
- P6: speedup for logic slowdown < 1.4, maximum 1.57.

Mpeg_encode: IRAM Vs. Conventional

Alpha



P6

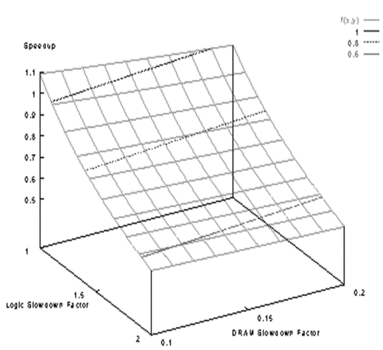


• Performance comparable only for almost no logic slowdown.

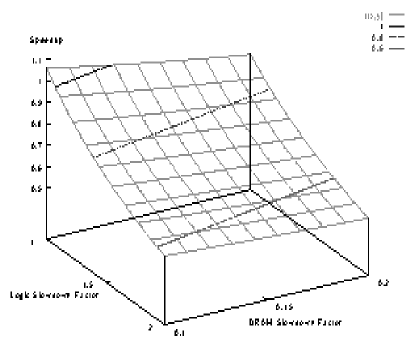
• Maximum performance degradation is 0.5!

• Reason: CPU intensive application.

SPECint95: IRAM Vs. Conventional



Alpha



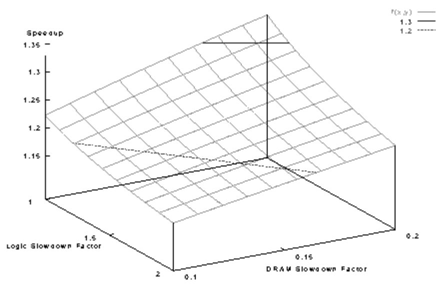
P6

• Averaged speedup for the SPECint95 benchmarks.

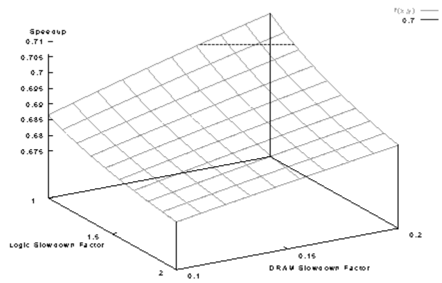
• Performance similar to mpeg_encode.

IRAM Alpha Vs. IRAM P6

Linpack



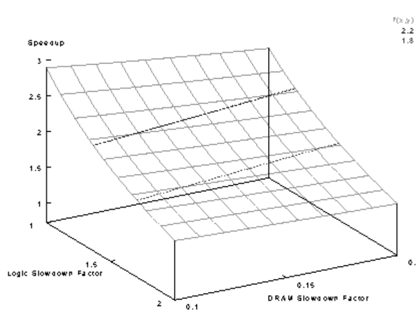
Sort



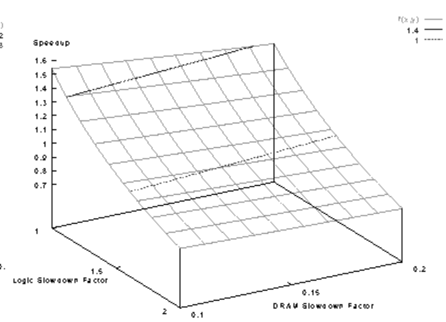
• Linpack : Alpha is 1.15 to 1.35 times faster.
• Sort: P6 is 1.38 times faster.

IRAM Alpha Vs. P6

Linpack:



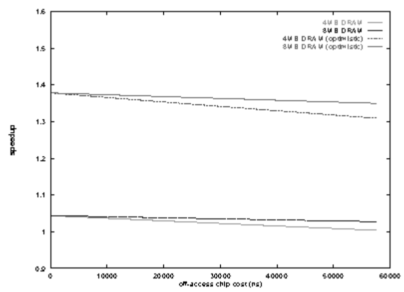
Sort:



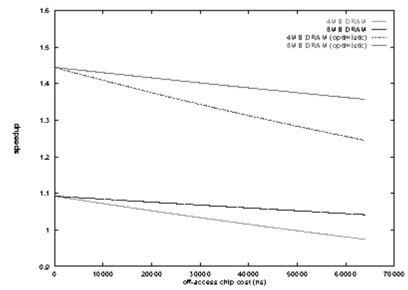
• Linpack: IRAM Alpha is up to 2.85 times faster than P6.
• Sort: IRAM Alpha is up to 1.53 times faster.

Effect of Off-chip accesses

Alpha



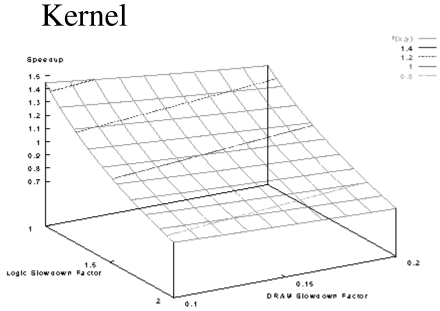
P6



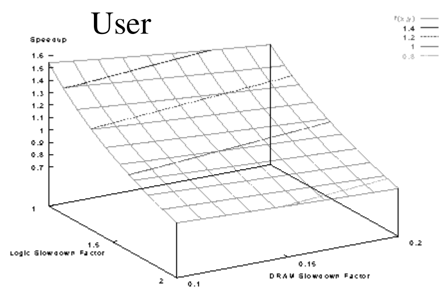
- Off-chip access causes the transfer of a whole page into IRAM.
- P6: speedup reduced up to 10% for 4MB implementations.

Effects on Specific Program Components

Kernel

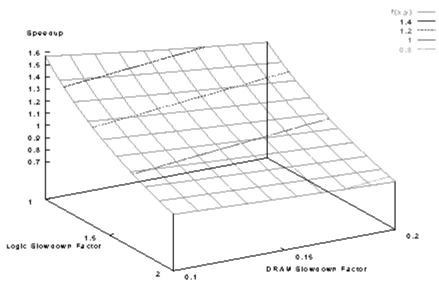


User

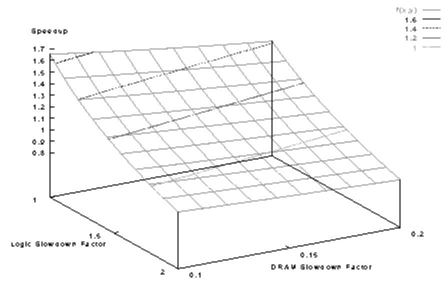


- Sort: Both modes behave identical; equal speedup is achieved.

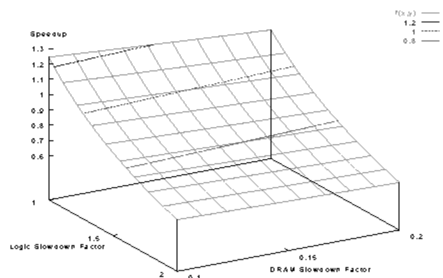
Effect on Specific Program Components II



read phase



write phase



sort phase

Conclusions

- Memory intensive applications benefit significantly from IRAM when logic slowdown is less than 1.5. Maximum speedup between 1.3 and 1.9.
- For CPU intensive applications, IRAM performs equally well only under no logic slowdown.
- Logic slowdown is the critical parameter.

Conclusions

- A simple processor architecture is more appropriate for IRAM than a complex one.
- An IRAM implementation of a simple architecture performs better than the conventional complex one for memory intensive applications.
- Off-chip access cost should be an issue in future IRAM implementations.