Evaluation of Existing Architectures in IRAM Systems

Christoforos Kozyrakis, Ngeci Bowman,
Neal Cardwell, Cynthia Romer and Helen Wang

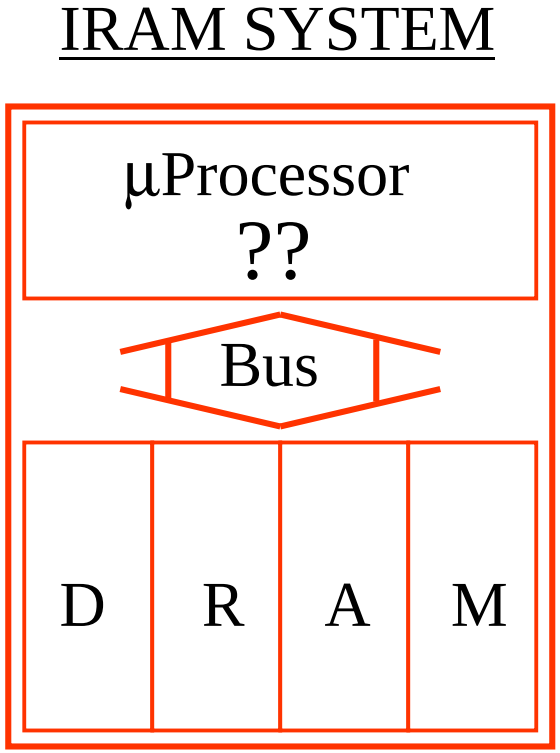
Computer Science Division
University of California at Berkeley

http://iram.cs.berkeley.edu

Workshop on "Mixing Logic and DRAM", ISCA'97

Motivation

- Intelligent RAM promises:
  - high memory bandwidth (100x)
  - low memory latency (0.1x)
  - high energy efficiency (4x)
  - higher system integration
- Which microprocessor architecture can turn these advantages into significant application performance benefits?



Evolutionary IRAM Approach

- Use an existing processor architecture:

  simple RISC micro, superscalar or out-of-order execution organization

- Advantages:
  - Good knowledge of how to design and implement them
  - Performance trade-offs are well understood
  - "Out of the box" solutions both for system software and applications - software compatibility
  - Higher performance by tuning programs and compilers to new memory hierarchy characteristics
- This work: evaluate potential performance benefits of this approach

Experimental Methodology

- Identified main IRAM architectural features
- Two complementary evaluation methods:
  - Through measurements and extrapolations
  - Through detailed system simulations
- IRAM systems with existing architectures can provide only moderate performance benefits even under optimistic assumptions

IRAM Architectural Considerations

- IRAM systems using existing DRAM technology:
  - 256Mbit DRAM 0.25μm CMOS process
  - 1/4 of die area for microprocessor
  - Up to 24MBytes of on-chip DRAM
- Memory access latency can be as low as 21ns
- Logic speed potentially up to 1.5 times slower compared to conventional processors for initial implementations
- No level 2 cache necessary since on-chip DRAM can be almost as fast
- Memory bus as wide as cache line

Method I: Measurements and Extrapolation

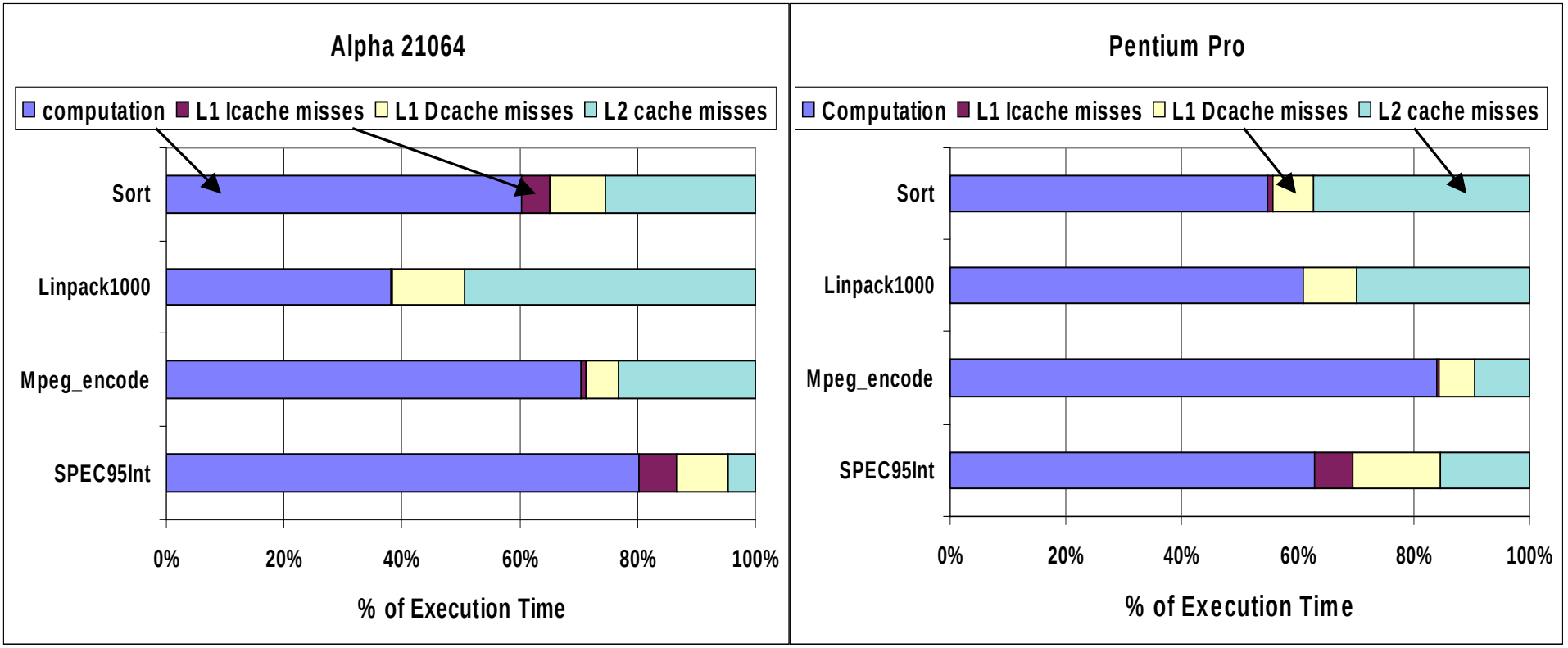
- Execution time analysis of a simple (Alpha 21064) and a complex architecture (Pentium Pro) to predict performance of similar IRAM implementations
- Used hardware counters for execution time measurements
- Benchmarks: SPEC95Int, Mpeg_encode, Linpack1000, Sort.
- IRAM implementations: same architectures with 24MBytes of on-chip DRAM but no L2 caches; all benchmarks fit completely in on-chip memory.
- IRAM execution time model:

$$ET = \frac{computation_time}{clock_speedup} + \frac{L1miss_count*memory_access_time}{memory_access_speedup}$$

Method I: Processors Characteristics

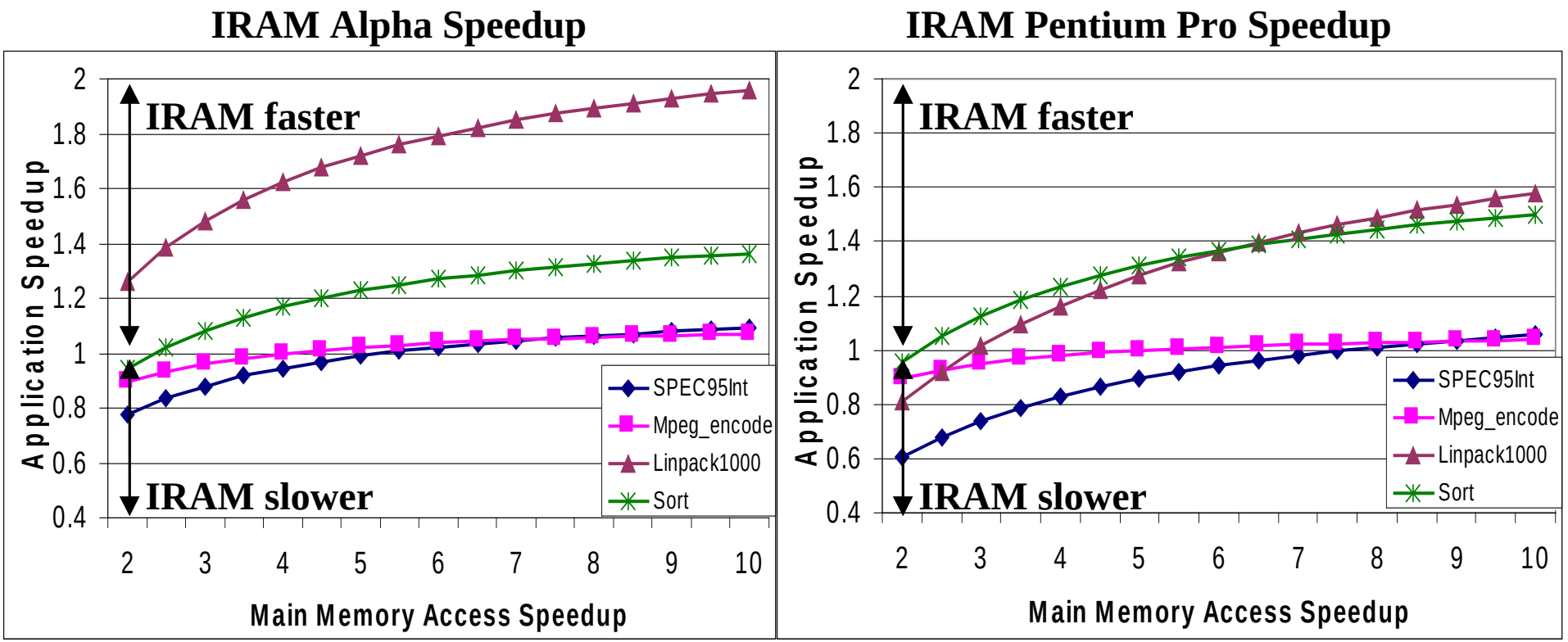
	Alpha 21064	**Pentium Pro**
Pipeline	in-order	out-of-order
CPU Frequency	133 MHz	200MHz
Issue Rate	2-way	3-way
L1 Configuration	8KB I + 8KB D	8KB I + 8KB D
L1 Associativity	Direct map	4-way
L1 Access Time	22.5ns	15ns
L2 Configuration	512KB	256KB
L2 Associativity	Direct map	4-way
L2 Type	Off-chip SRAM	Off-chip SRAM
L2 Access Time	37.5ns	20ns
Memory	64MB EDO DRAM	64MB EDO DRAM
Total Latency	180ns	220ns

Execution Time Analysis of Conventional Systems



- Linpack1000 and Sort spend up to 50% of execution time in main memory
- SPEC95Int and Mpeg_encode are CPU bound

Method I: Results



- Equal clock speeds assumed for conventional and IRAM systems
- Maximum IRAM speedup compared to conventional:
  - Less than 2 for memory bound applications
  - Less than 1.1 for CPU bound applications

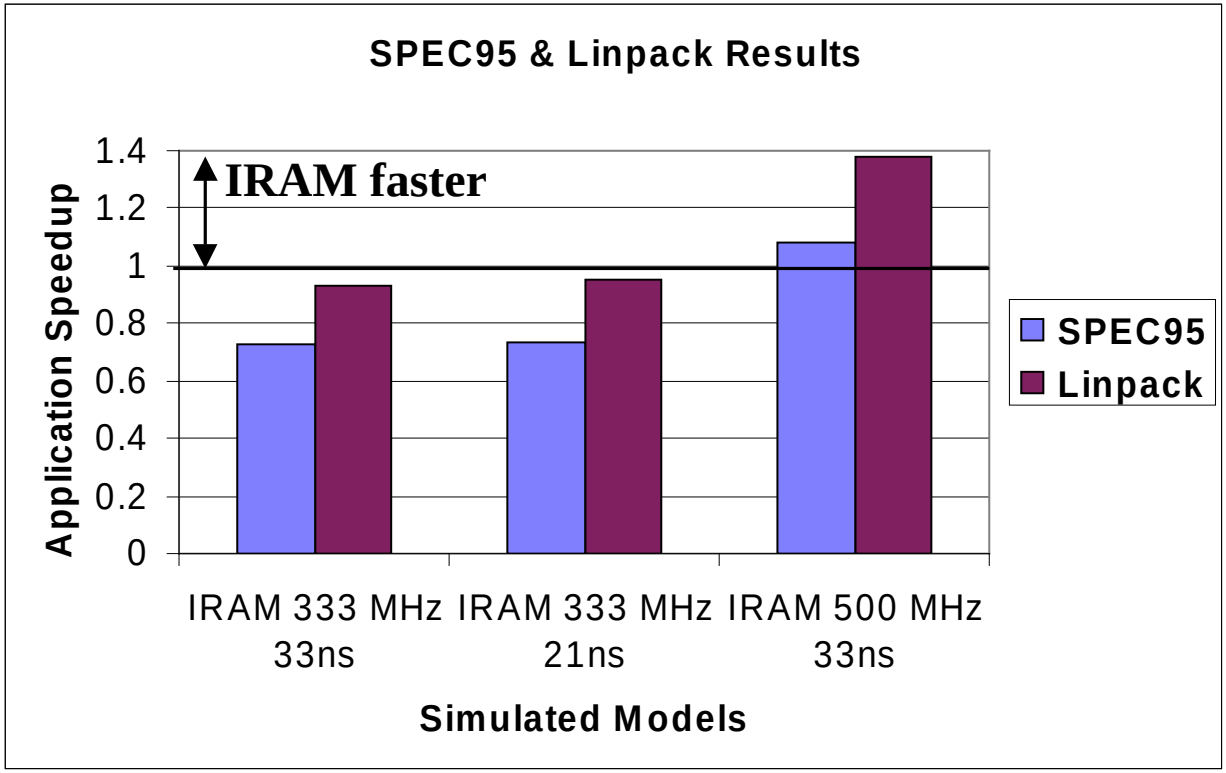
Method II: Detailed System Simulations

- Used SimOS to simulate simple MIPS R4000-based IRAM and conventional architectures
- Equal die size comparison:
  - Area for on-chip DRAM in IRAM systems same as area for level 2 cache in conventional system
- Wide memory bus for IRAM systems
- Main simulation parameters:
  - On-chip DRAM access latency
  - Logic speed (CPU frequency)
- Benchmarks: SPEC95Int (compress, li, ijpeg, perl, gcc), SPEC95Fp (tomcatv, su2cor, wave5), Linpack1000

Simulated Models

	IRAM	Conventional
Pipeline	Simple in-order	Simple in-order
CPU Frequency	333 or 500 MHz	500MHz
Technology	0.25µm DRAM	0.25µm logic
L1 Configuration	64KB I + 64KB D	64KB I + 64KB D
L1 Associativity	2-way	2-way
L1 Block Size	128B	64B I / 32B D
L1 Type	On-chip SRAM	On-chip SRAM
L1 Access Time	1 CPU cycle	1 CPU cycle
L2 Configuration	-	2MB unified
L2 Associativity	-	2-way
L2 Block Size	-	128B
L2 Type	-	On-chip SRAM
L2 Access Time	-	12 CPU cycles
Memory Configuration	24MB DRAM on-chip	24MB 166MHz SDRAM off-chip
Memory Bus Width	1024 bits	128 bits
Total Latency	21 or 33ns	116ns

Method II: Results



- Maximum speedup of 1.4 for equal clock speeds
- Slower than conventional for most other cases

Conclusions

- IRAM systems with existing architectures provide only moderate performance benefits
- High bandwidth/low latency used to speed up memory accesses but not computation
- Reason: existing architectures developed under the assumption of a low bandwidth memory system
- Still attractive for portable/embedded domain
  - 2 to 4 times more energy efficient [R. Fromm, ISCA, Wed. 11.30am]
  - higher system integration

Towards a Revolutionary Approach

- To provide significant performance benefits IRAM systems need microprocessor architectures that turn memory bandwidth into application performance

- Candidates:
  - Vector microprocessor
  - Multithreading architectures
  - Multiprocessor on a chip
  - VLIW organizations
  - Some new idea?

Acknowledgements

- Thanks for advice and/or support:

  DARPA, NSF, California State Micro Program, Intel, Sun Microsystems, SGI/Cray and Neomagic