Vector IRAM

A Media-enhanced Vector Processor with Embedded DRAM

Christoforos Kozyrakis, Joseph Gebis, David Martin, Samuel Williams, Ioannis Mavroidis, Steven Pope, Darren Jones*, and David Patterson

Computer Science Division
University of California at Berkeley
http://iram.cs.berkeley.edu

* MIPS Technologies Inc.
Mountain View, CA
http://www.mips.com

Motivation and Goals

- Processor features for PostPC systems:
  - High, predictable performance for multimedia
  - Low power and energy consumption
  - Tolerance to memory latency
  - Scalability and modularity
  - Low design complexity
  - System integration
  - Mature, HLL-based software model
- Design a prototype processor chip
  - Complete proof of concept
  - Explore detailed architecture and design issues
  - Fast platform for software development

Key Technologies

- Vector processing
  - High performance for media processing
  - Low power/energy for issue and control logic
  - Low design complexity
  - Scalable and modular
  - Well understood compiler technology
- Embedded DRAM
  - High bandwidth for vector processing
  - Low power/energy for memory accesses
  - Scalable and modular
  - System integration

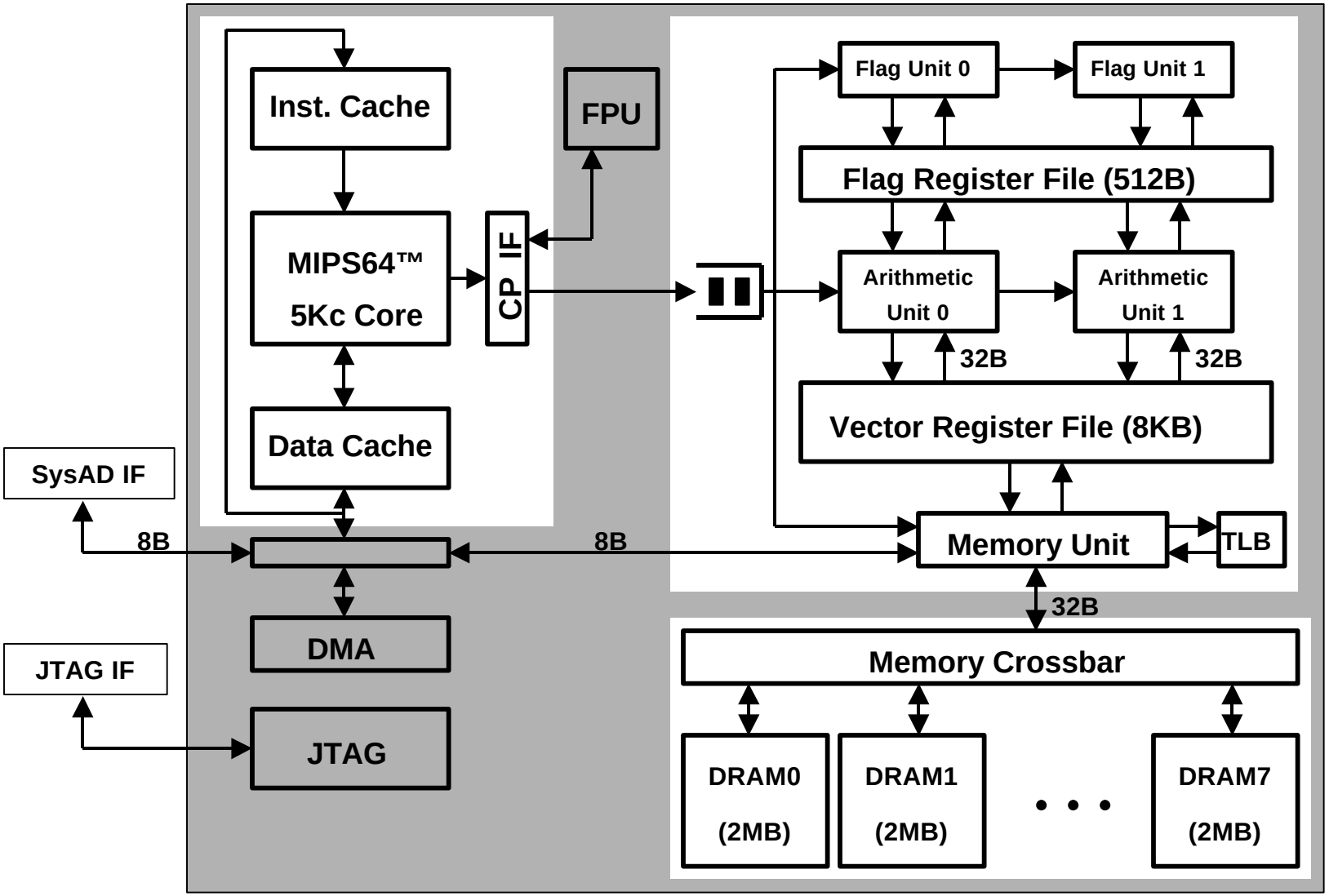
Vector Instruction Set

- Complete load-store vector ISA
  - Uses the MIPS64™ ISA coprocessor 2 opcode space
  - Data types supported: 64b, 32b, 16b (and 8b)
  - Architecture state
    - 32 general-purpose vector registers
    - 32 vector flag registers, 16 scalar registers, control registers
  - 91 instructions, 661 opcodes
    - Arithmetic (integer/FP), logical, DSP, vector processing
    - Strided and indexed loads and stores
- Not specified by the ISA
  - Vector register length
  - Functional unit datapath width
  - Alignment restrictions for vectors in memory

Vector ISA Enhancements

- DSP support
  - Fixed-point shift, multiply and multiply-add, saturated arithmetic, rounding modes
  - Simple instructions for intra-register permutations for reductions and butterfly operations
    - High performance for dot-products and FFT without the complexity of a random permutation
- Compiler and OS support
  - Conditional execution of vector operations
  - Support for software speculation of load operations
  - MMU-based virtual memory
  - Valid and dirty bits for vector registers
  - Restartable arithmetic exceptions

VIRAM Prototype Architecture



Architecture Details (1)

- MIPS64™ 5Kc core (200 MHz)
  - Single-issue core with 6 stage pipeline
  - 8 KByte, direct-map instruction and data caches
  - Single-precision scalar FPU
- Vector unit (200 MHz)
  - 8 KByte register file (32 64b elements per register)
  - 4 functional units:
    - 2 arithmetic (1 FP), 2 flag processing
    - 256b datapaths per functional unit
  - Memory unit
    - 4 address generators for strided/indexed accesses
    - 2-level TLB structure: 4-ported, 4-entry microTLB and single-ported, 32-entry main TLB
    - Pipelined to sustain up to 64 pending memory accesses

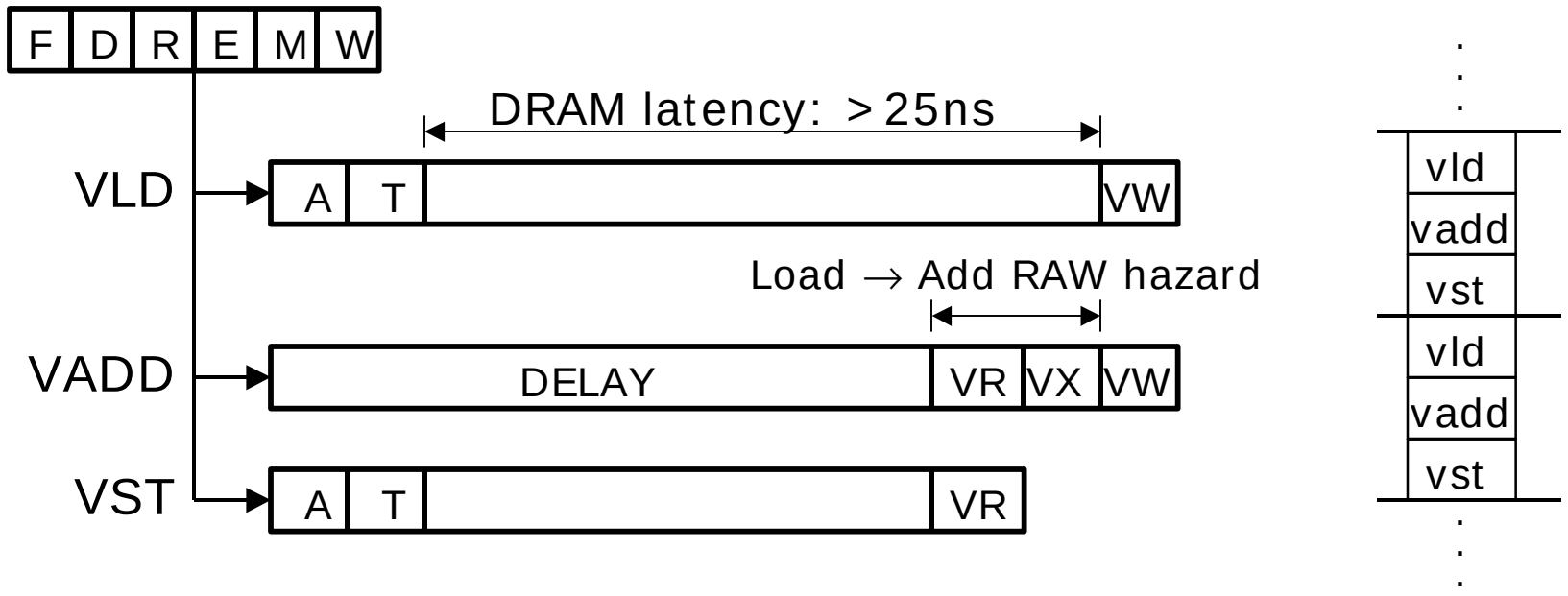
Architecture Details (2)

- Main memory system
  - No SRAM cache for the vector unit
  - 8 2-MByte DRAM macros
    - Single bank per macro, 2Kb page size
    - 256b synchronous, non-multiplexed I/O interface
    - 25ns random access time, 7.5ns page access time
  - Crossbar interconnect
    - 12.8 GBytes/s peak bandwidth per direction (load/store)
    - Up to 5 independent addresses transmitted per cycle
- Off-chip interface
  - 64b SysAD bus to external chip-set (100 MHz)
  - 2 channel DMA engine

Vector Unit Pipeline

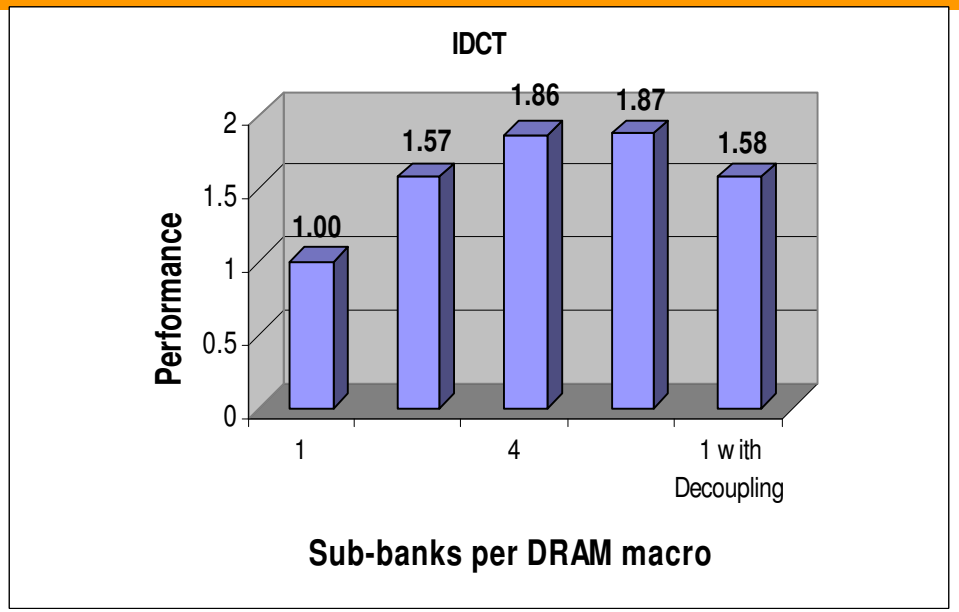
- Single-issue, in-order pipeline
- Efficient for short vectors
  - Pipelined instruction start-up
  - Full support for instruction chaining, the vector equivalent of result forwarding
- Hides long DRAM access latency
  - Random access latency could lead to stalls due to long load→use RAW hazards
  - Simple solution: “delayed” vector pipeline

Delayed Vector Pipeline



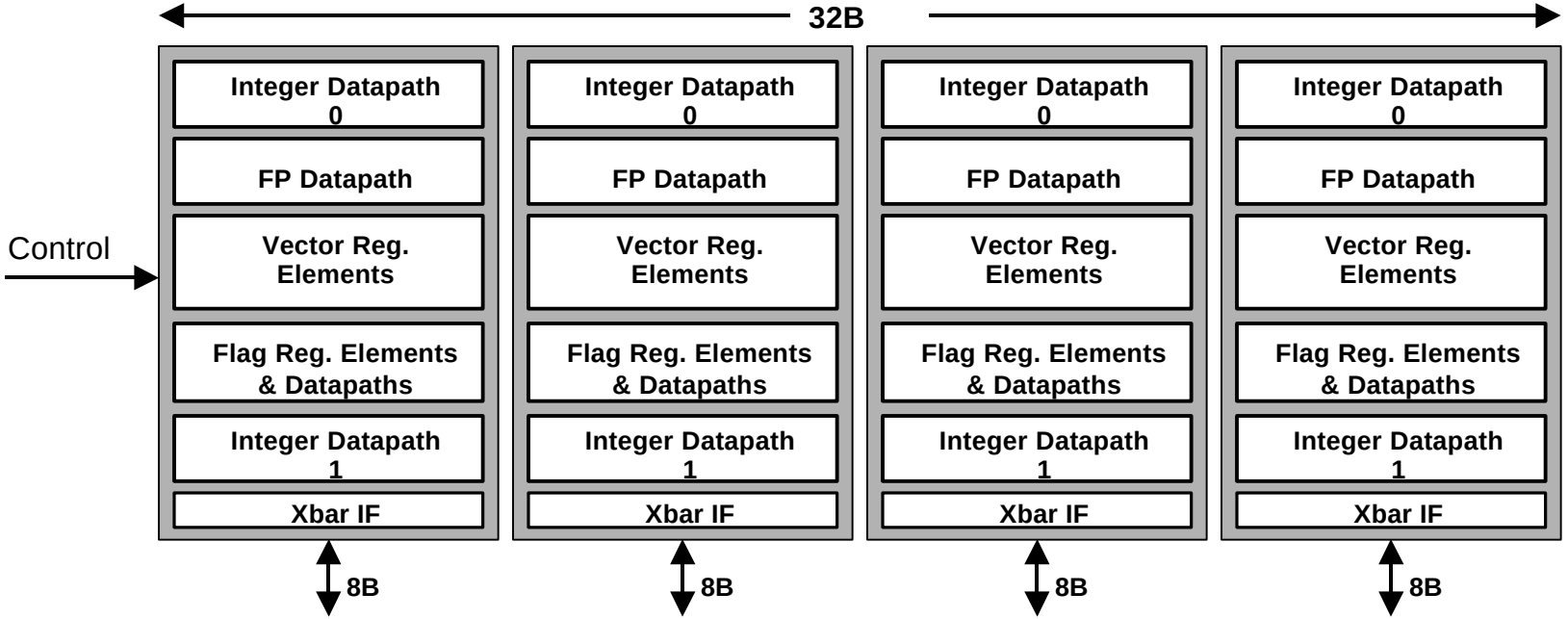
- Random access latency included in the vector unit pipeline
- Arithmetic operations and stores are delayed to shorten RAW hazards
- Long hazards eliminated for the common loop cases
- Vector pipeline length: 15 stages

Handling Memory Conflicts



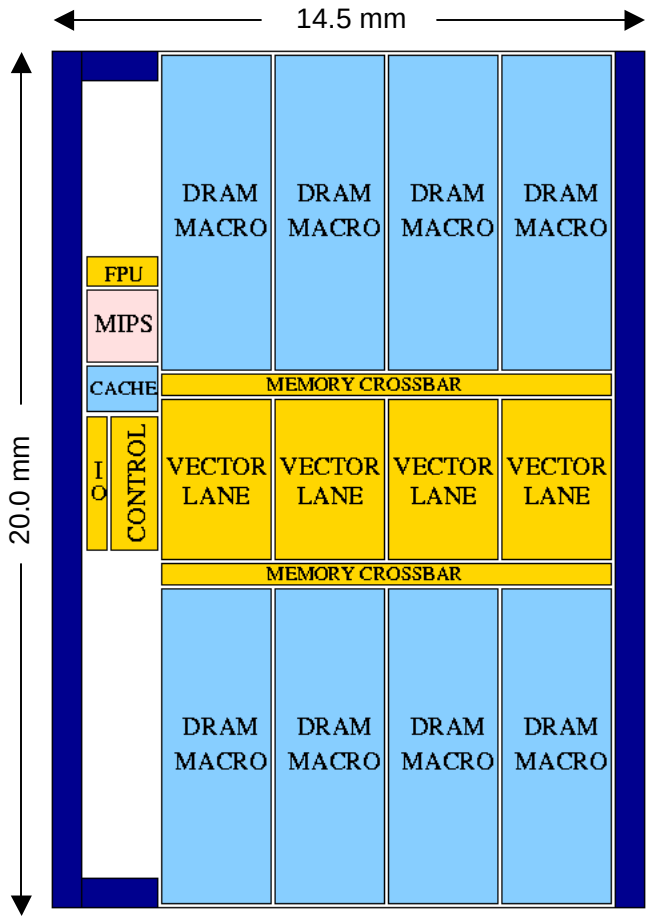
- Single sub-bank DRAM macro can lead to memory conflicts for non-sequential access patterns
- Solution 1: address hashing
  - Selects between 3 address interleaving modes for each virtual page
- Solution 2: address decoupling buffer (128 slots)
  - Allows scheduling of long indexed accesses without stalling the arithmetic operations executing in parallel

Modular Vector Unit Design



- Single 64b "lane" design replicated 4 times
  - Reduces design and testing time
  - Provides a simple scaling model (up or down) without major control or datapath redesign
- Most instructions require only intra-lane interconnect
  - Tolerance to interconnect delay scaling

Floorplan



- Technology: IBM SA-27E
  - 0.18µm CMOS
  - 6 metal layers (copper)
- 290 mm^2 die area
  - 225 mm^2 for memory/logic
  - DRAM: 161 mm^2
  - Vector lanes: 51 mm^2
- Transistor count: ~150M
- Power supply
  - 1.2V for logic, 1.8V for DRAM
- Peak vector performance
  - 3.2/6.4 /12.8 Gops w. multiply-add (64b/32b/16b operations)
  - 1.6/3.2/6.4 Gops wo. multiply-add
  - 1.6 Gflops (single-precision)

Power Consumption

- Power saving techniques
  - Low power supply for logic
    - Possible because of the low clock rate
    - Wide vector datapaths provide high performance
  - Extensive clock gating and datapath disabling
    - Utilizing the explicit parallelism information of vector instructions and conditional execution
  - Simple, single-issue, in-order pipeline
- Power consumption: 2.0 W
  - MIPS core: 0.5 W
  - Vector unit: 1.0 W
  - DRAM: 0.2 W
  - Misc.: 0.3 W

Software Tools

- VIRAM compiler
  - Vectorizing compiler with C/C++/Fortran front-ends
  - Based on the Cray's PDGCS production environment for supercomputers (J90, T3E, SV1, SV2)
  - Extensive vectorization and optimization capabilities including outer loop vectorization
  - No need to use special libraries or variable types for vectorization
- Other software tools
  - Assembler, disassembler, debugger
  - ISA simulator and performance model

Performance: Efficiency

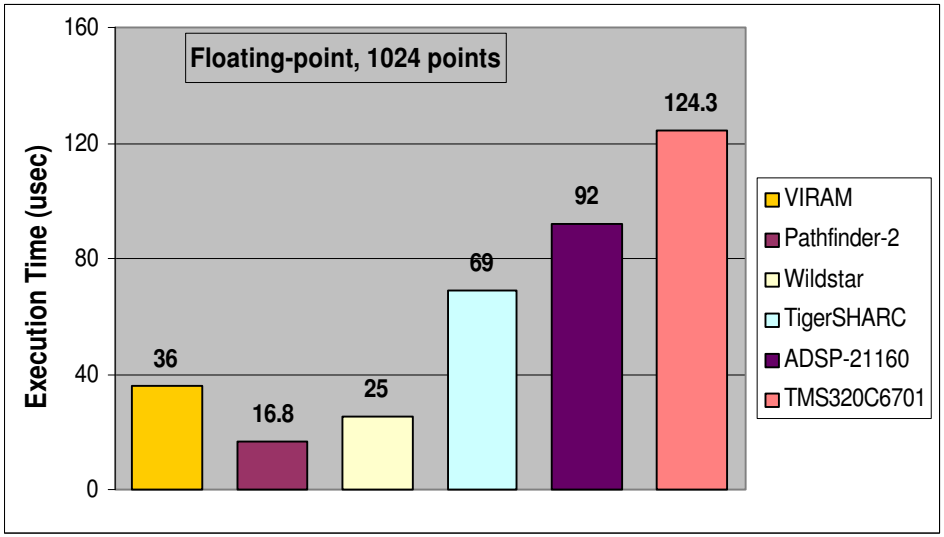
	Peak	Sustained	% of Peak
Image Composition	6.4 GOPS	**6.40 GOPS**	100%
iDCT	6.4 GOPS	**3.10 GOPS**	48.4%
Color Conversion	3.2 GOPS	**3.07 GOPS**	96.0%
Image Convolution	3.2 GOPS	**3.16 GOPS**	98.7%
Integer VM Multiply	3.2 GOPS	**3.00 GOPS**	93.7%
FP VM Multiply	1.6 GFLOPS	**1.59 GFLOPS**	99.6%
Average			89.4%

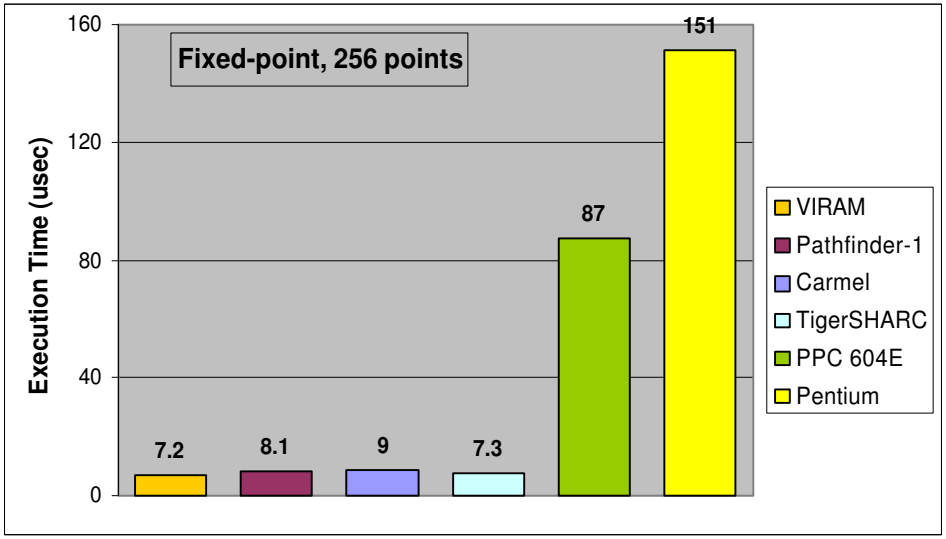
Performance: Comparison

	VIRAM	MMX	VIS
Image Composition	0.13	-	2.2 (17.0x)
iDCT	0.75	3.75 (5.0x)	-
Color Conversion	0.78	8.00 (10.2x)	-
Image Convolution	5.49	5.49 (4.5x)	6.19 (5.1x)
QCIF (176x144)	7.1M	33M (4.6x)	-
CIF (352x288)	28M	140M (5.0x)	-

- QCIF and CIF numbers are in clock cycles per frame
- All other numbers are in clock cycles per pixel

Performance: FFT





Conclusions

- Vector IRAM
  - An integrated architecture for media processing
  - Based on vector processing and embedded DRAM
  - Simple, efficient, and scalable
- Prototype
  - 16 MBytes DRAM, 256b vector unit
  - 150M transistors, 290 mm^2
  - 1.6 Gops/W at 200 MHz
- Prototype status
  - RTL model completed
  - Back-end design and verification in progress
  - Design tape-out in late fall 2000

Acknowledgments

- IBM Microelectronics
- MIPS Technologies Inc

- Katherine Yelick, Randi Thomas, Thinh Nguyen, James Beck, Dave Judd, Krste Asanovic, and Richard Fromm

- This research is sponsored by DARPA, NSF, and the California State MICRO program