COMP4211 Advanced Architectures & Algorithms

Week 11 Seminar

A New Direction for Computer Architecture Research

Lih Wen Koh

19 May 2004

Outline

- Overview of Current Computer Architecture Research
- The Desktop/Server Domain
  - Evaluation of current processors in the desktop/server domain
    - Benchmark performance
    - Software effort
    - Design complexity
- A New Target Domain: Personal Mobile Computing
  - Major requirements of personal mobile computing applications
  - Evaluation of current processors in the personal mobile computing domain
- Vector IRAM by UC Berkeley

Outline

- ***Overview of Current Computer Architecture Research***
- The Desktop/Server Domain
  - Evaluation of current processors in the desktop/server domain
    - Benchmark performance
    - Software effort
    - Design complexity
- A New Target Domain: Personal Mobile Computing
  - Major requirements of personal mobile computing applications
  - Evaluation of current processors in the personal mobile computing domain
- Vector IRAM by UC Berkeley

Overview of Current Computer Architecture Research

- Current computer architecture research have a bias for the past – desktop and server applications
- Next decade's technology domain – personal mobile computing
- Question: What are these future applications?
- Question: What is the set of requirements for this domain?
- Question: Do current microprocessors meet these requirements?

Billion-transistor microprocessors

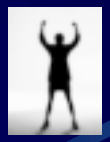
Table 1. Number of memory transistors in the billion-transistor microprocessors.

Architecture	Key idea	No. of memory transistors (millions)
Advanced superscalar	Wide-issue superscalar processor with speculative execution; multilevel on-chip caches	910
Superspeculative	Wide-issue superscalar processor with aggressive data and control speculation; multilevel, on-chip caches	820
Trace	Multiple distinct cores that speculatively execute program traces; multilevel on-chip caches	600
Simultaneous multithreading	Wide superscalar with support for aggressive sharing among multiple threads; multilevel on-chip caches	810
Chip multiprocessor	Symmetric multiprocessor; system shared second-level cache	450
IA-64	VLIW architecture with support for predicated execution and long-instruction bundling	600
Raw	Multiple processing tiles with reconfigurable logic and memory interconnected through a reconfigurable network	670

Billion-transistor microprocessors

- The amount of transistors used for caches and main memory in billion-transistor processors varies from 50-90% of the transistor budget.
- Mostly on caches and main memory → to store redundant, local copies of data normally found else where in the system
- Question: Is this the best utilization of half a billion transistors for future applications?

Outline



- ✓ Overview of Current Computer Architecture Research
- ***The Desktop/Server Domain***
  - Evaluation of current processors in the desktop/server domain
    - Benchmark performance
    - Software effort
    - Design complexity
- A New Target Domain: Personal Mobile Computing
  - Major requirements of personal mobile computing applications
  - Evaluation of current processors in the personal mobile computing domain
- Vector IRAM by UC Berkeley

The Desktop/Server Domain

- Evaluation of billion-transistor processors (grading system: + for strength, 0 for neutrality, - for weakness)

Table 2. Evaluation of billion-transistor processors for the desktop/server domain. "Wide superscalar" includes the advanced superscalar and superspeculative processors.

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

The Desktop/Server Domain

- Desktop
  - *Wide superscalar*, *trace* and *simultaneous multithreading* processors should deliver the highest performance on SPECint04
    - Use out-of-order and advanced prediction techniques to exploit ILP
  - *IA-64* will perform slightly worse because of immature VLIW compilers
  - *CMP* and *Raw*
    - will have inferior performance in integer applications which are not highly parallelizable
    - performance is better in FP applications where parallelism and high memory bandwidth are more important than out-of-order execution

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

The Desktop/Server Domain

- Server
  - *CMP* and *SMT* will provide the best performance due to their ability to use coarse-grained parallelism even with a single chip
  - *Wide superscalar, trace* and *IA-64* will perform worse because out-of-order execution provides only a small benefit to online transaction processing (OLTP) applications
  - *Raw* – difficult to predict the potential success of its software to map the parallelism of databases on reconfigurable logic and software-controlled caches

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

The Desktop/Server Domain

- Software Effort
  - *Wide superscalar*, *trace* and *SMT* processors – can run existing executables
  - *CMP* – can run existing executables; but need to be rewritten in a multithreaded or parallel fashion which is neither easy nor automated.
  - *IA-64* will supposedly run existing executables, but significant performance increases will require enhanced VLIW compilers.
  - *Raw* relies on the most challenging software development for sophisticated routing, mapping and runtime-scheduling tools, compilers and reusable libraries.

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

The Desktop/Server Domain

- Physical Design Complexity
  - Includes effort for design, verification and testing of an IC
  - *Wide superscalar* and *multithreading* processors use complex techniques e.g. aggressive data/control prediction, out-of-order execution, multithreading and non-modular designs (individually designed multiple blocks)
  - *IA-64* – the basic challenge is the design and verification of forwarding logic among the multiple functional units on the chip
  - *CMP, trace* and *Raw* – modular design; but complex out-of-order, cache coherency, multiprocessor communication, register remapping etc.
  - *Raw* – requires design and replication of a single processing tile and network switch; verification is trivial in terms of the circuits, but verification of the mapping software is also required, which is often not trivial.

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

The Desktop/Server Domain

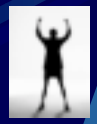
Table 2. Evaluation of billion-transistor processors for the desktop/server domain. "Wide superscalar" includes the advanced superscalar and superspeculative processors.

Characteristic	Wide superscalar	Trace	Simultaneous multithreading	Chip multiprocessor	IA-64	Raw
SPECint04 performance	+	+	+	0	+/0	0
SPECfp04 performance	+	+	+	+	+	0
TPC-F performance	0	0	+	+	0	–
Software effort	+	+	0	0	0	–
Physical-design complexity	–	0	–	0	0	+

- Conclusion:
  - Current billion-transistor processors are optimized for desktop/server computing and promise impressive performance.
  - The main concern is the design complexity of these architectures.

Outline

- ✓ Overview of Current Computer Architecture Research



- ✓ The Desktop/Server Domain
  - ✓ Evaluation of current processors in the desktop/server domain
    - ✓ Benchmark performance
    - ✓ Software effort
    - ✓ Design complexity

- ***A New Target Domain: Personal Mobile Computing***
  - Major requirements of personal mobile computing applications
  - Evaluation of current processors in the personal mobile computing domain

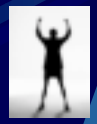
- Vector IRAM by UC Berkeley

A New Target Domain: Personal Mobile Computing

- Trend 1: Multimedia applications: video, speech, animation, music.
- Trend 2: Popularity of portable electronics
  - PDA, digital cameras, cellular phones, video game consoles
- … ***Personal Mobile Computing*** …
- Convergent devices
  - Goal: a single, portable, personal computing and communication device that incorporate necessary functions of a PDA, laptop computer, cellular phone etc.
- Greater demand for computing power, but at the same time, the size, weight and power consumption of these devices must remain constant
- Key features:
  - Most important feature: interface interaction with the user
    - Voice and image I/O
    - Applications like speech and pattern recognition
  - Wireless infrastructure
    - Networking, telephony, GPS information

Outline

- ✓ Overview of Current Computer Architecture Research



- ✓ The Desktop/Server Domain
  - ✓ Evaluation of current processors in the desktop/server domain
    - ✓ Benchmark performance
    - ✓ Software effort
    - ✓ Design complexity

- A New Target Domain: Personal Mobile Computing
  - ***Major requirements of personal mobile computing applications***
  - Evaluation of current processors in the personal mobile computing domain

- Vector IRAM by UC Berkeley

Major microprocessor requirements

- Requirement 1: High performance for multimedia functions
- Requirement 2: Energy and power efficiency
  - Design for portable, battery-operated devices
  - Power budget < 2 Watts
  - Processor design of power target < 1 Watt
  - Power budget of current high-performance microprocessors (tens of Watts) is unacceptable
- Requirement 3: Small size
  - Code size
  - Integrated solutions (external cache and main memory not feasible)
- Requirement 4: Low design complexity
  - Scalability in terms of both performance and physical design

Characteristics of Multimedia Applications

- Real-time response
  - Worst case guaranteed performance sufficient for real-time qualitative perception
  - Instead of maximum peak performance
- Continuous-media data types
  - Continuous stream of input and output
  - Temporal locality in data memory accesses is low! Data caches may well be an obstacle to high performance for continuous-media data types
  - Typically narrow data 8-16 bit for image pixels and sound samples
  - SIMD-type operations desirable

Characteristics of Multimedia Applications

- Fine-grained parallelism
  - Same operation is performed across sequences of data in vector or SIMD fashion
- Coarse-grained parallelism
  - A pipeline of functions process a single stream of data to produce the end results.

Characteristics of Multimedia Applications

- High instruction reference locality
  - Typically small kernels/loops that dominate the processing time
  - High temporal and spatial locality for instructions
  - Example: Convolution equation – for signal filtering

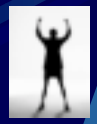
$$y[n] = \sum_{k} x[k].h[n-k]$$

```
for n = 0 to N
        y[n] = 0;
        for k = n to N
                y[n] += x[k] * h[n-k];
        end for;
end for;
```

- High memory bandwidth
  - For applications such as 3D graphics
- High network bandwidth
  - Data (e.g. video) streaming for external sources requires high network and I/O bandwidth.

Outline

- ✓ Overview of Current Computer Architecture Research



- ✓ The Desktop/Server Domain
  - ✓ Evaluation of current processors in the desktop/server domain
    - ✓ Benchmark performance
    - ✓ Software effort
    - ✓ Design complexity

- A New Target Domain: Personal Mobile Computing
  - ✓ Major requirements of personal mobile computing applications
  - ***Evaluation of current processors in the personal mobile computing domain***

- Vector IRAM by UC Berkeley

Processor Evaluation

- Real-time response
  - Out-of-order techniques + caches → unpredictable performance, hence difficult to guarantee real-time response
- Continuous-media data types
  - Question: Does temporal locality in data memory accesses still hold?
  - Claim: Data caches may well be an obstacle to high performance for continuous-media data types
- Parallelism
  - MMX-like multimedia extensions for exploiting fine-grained parallelism
    - but this exposes data alignment issues, restriction on number of vector or SIMD elements operated on by each instruction
  - Coarse-grained parallelism is best on SMT, CMP and Raw architectures.

Processor Evaluation

- Memory bandwidth
  - Cache-based architectures have limited memory bandwidth
  - Could potentially use streaming buffers and cache bypassing to help sequential bandwidth, but this does not address bandwidth requirements of indexed or random accesses
  - Recall: 50-90% of transistor budget is dedicated to caches!
- Code size
  - Code size is a weakness (especially for IA-64) because loop unrolling and software pipelining are heavily relied upon to gain performance
  - Code size is also a problem for Raw architecture as programmers must program the reconfigurable portion of each datapath

Processor Evaluation

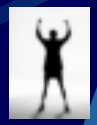
- Energy/power efficiency
  - Redundant computation for out-of-order models
  - Complex issue-logic
  - Forwarding across long wires
  - Power-hungry reconfigurable logic
- Design scalability
  - The main problem is the forwarding of results across large chips or communication among multiple cores/tiles.
  - Simple pipelining of long interconnects is not a sufficient solution
    - exposes the timing of forwarding or communication to the scheduling logic or software
    - Increases complexity

Processor Evaluation

- Conclusion:
  - Current processors fail to meet many of the requirements of the new computing model.
- Question:
  - What design will?

Outline

- ✓ Overview of Current Computer Architecture Research



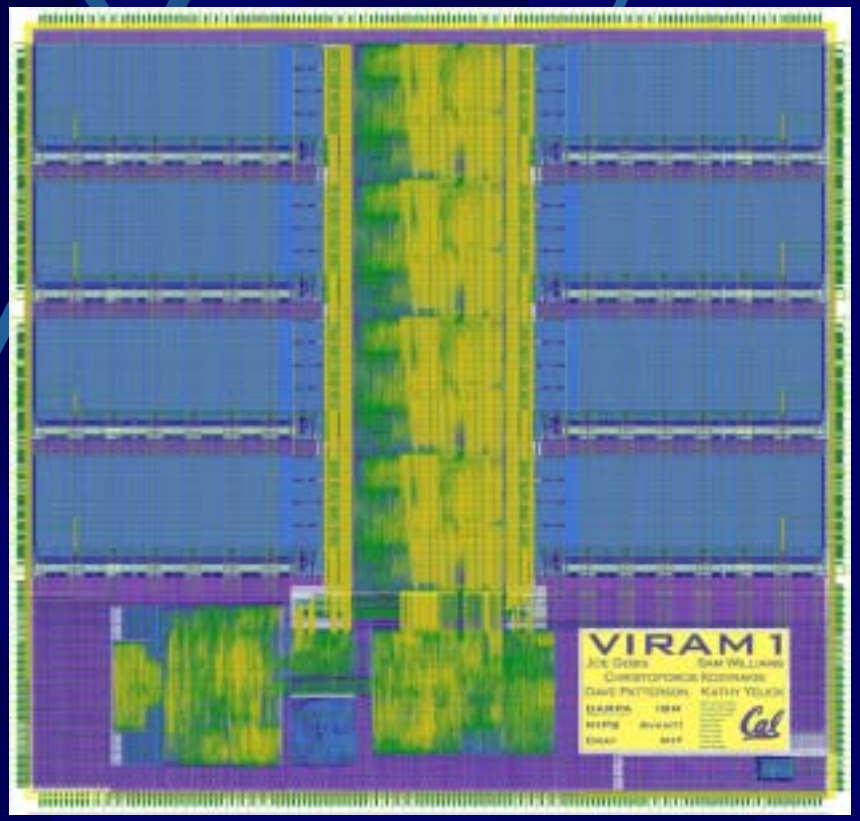
- ✓ The Desktop/Server Domain
  - ✓ Evaluation of current processors in the desktop/server domain
    - ✓ Benchmark performance
    - ✓ Software effort
    - ✓ Design complexity

- A New Target Domain: Personal Mobile Computing
  - ✓ Major requirements of personal mobile computing applications
  - ✓ Evaluation of current processors in the personal mobile computing domain

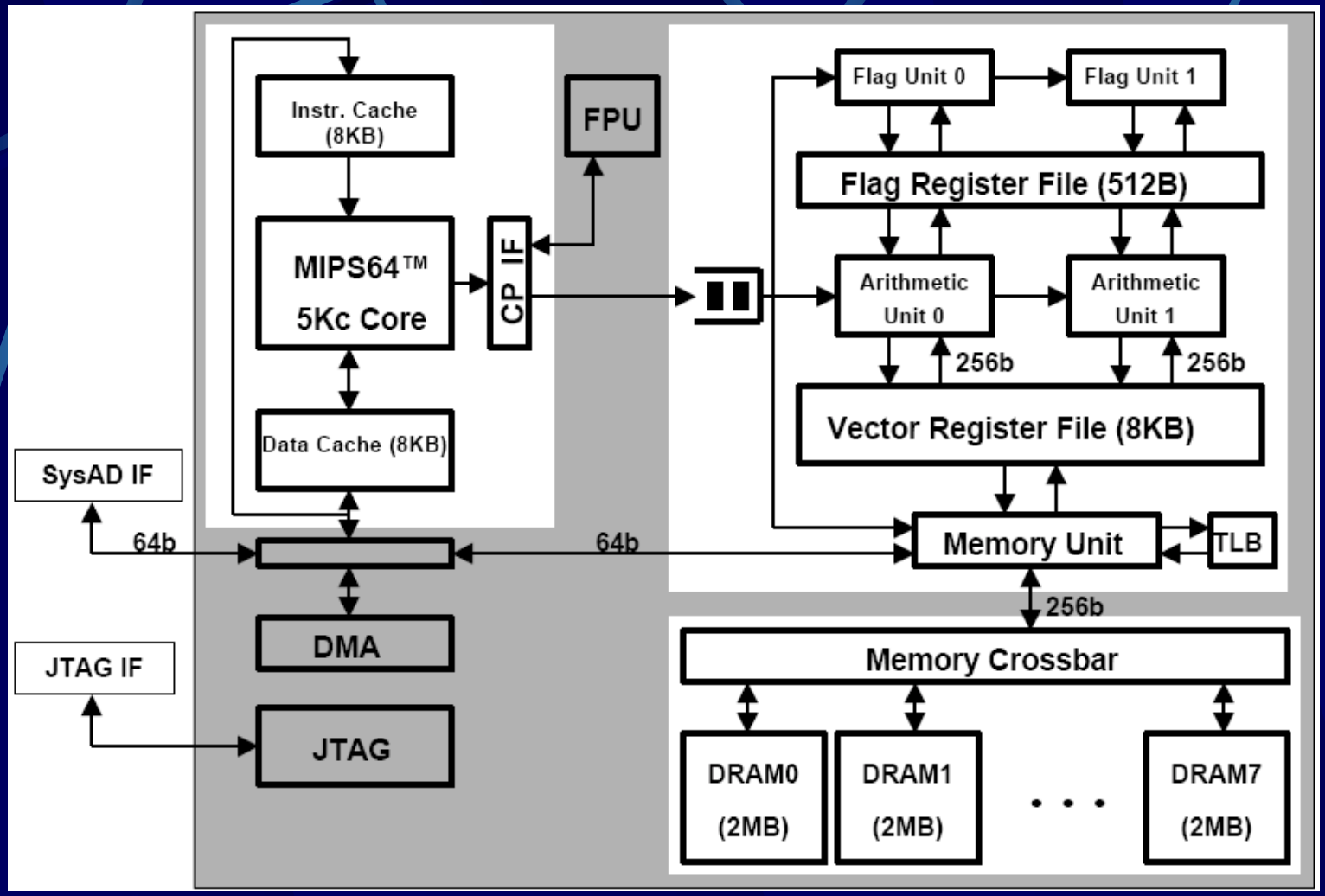
- ***Vector IRAM by UC Berkeley***

Vector IRAM processor

- Targeted at matching the requirements of the personal mobile computing environment
- 2 main ideas:
  - *Vector processing* – addresses demands of multimedia processing
  - *Embedded DRAM* – addresses the energy efficiency, size and weight demands of portable devices



VIRAM Prototype Architecture



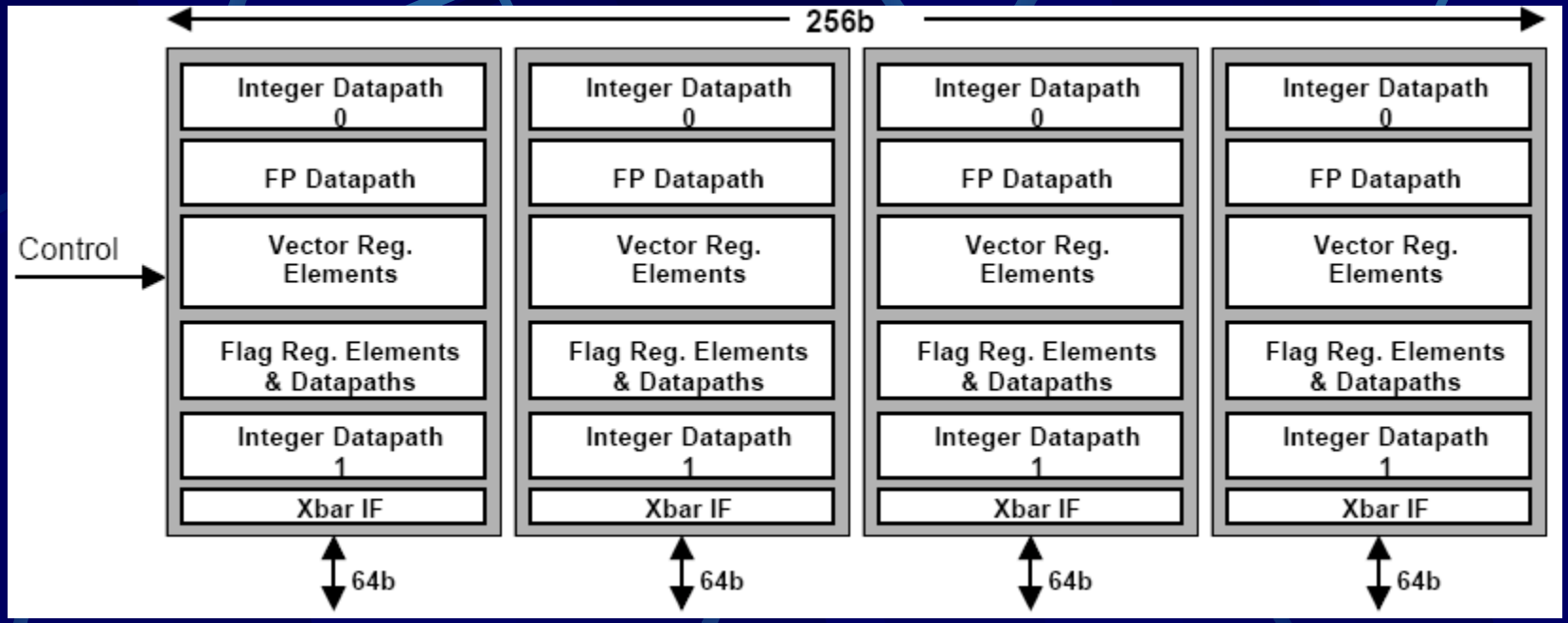
VIRAM Prototype Architecture

- Uses in-order, scalar processor with L1 caches, tightly integrated with a vector execution unit (with 8 lanes)
- 16MB of embedded DRAM as main memory
  - connected to the scalar and vector unit through a crossbar
  - Organized in 8 independent banks, each with a 256-bit synchronous interface

  → provides sufficient sequential and random bandwidth even for demanding applications

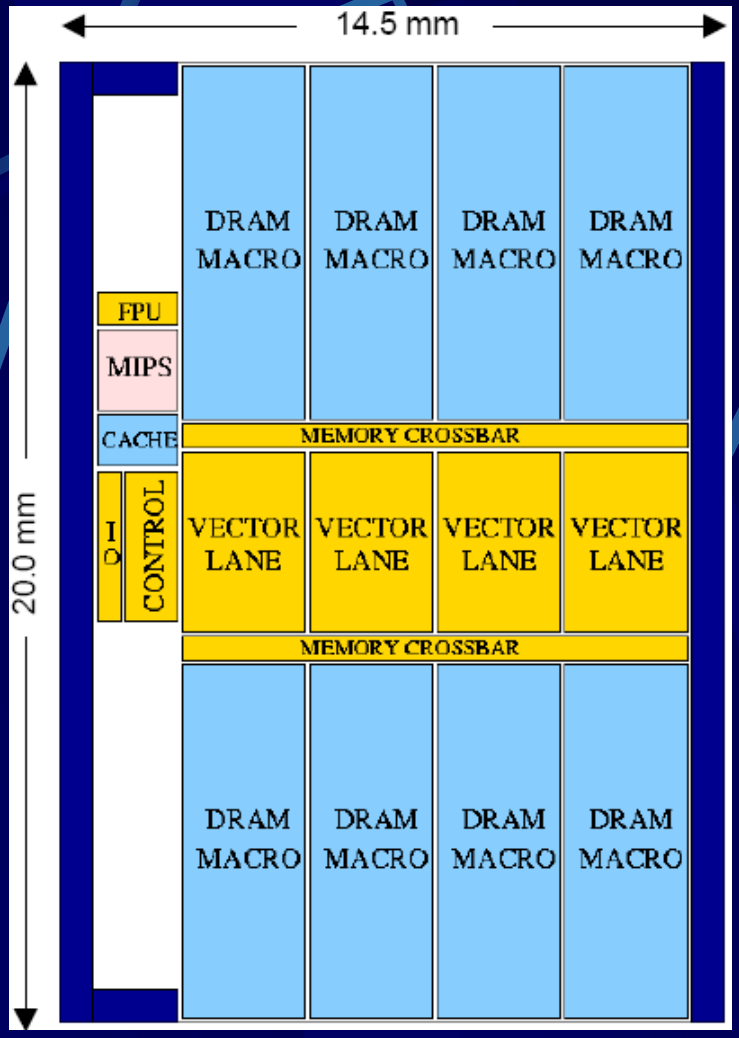
  → reduces the penalty of high energy consumption by avoiding the memory bus bottlenecks of conventional multi-chip systems
- DMA engine for off-chip access

Modular Vector Unit Design



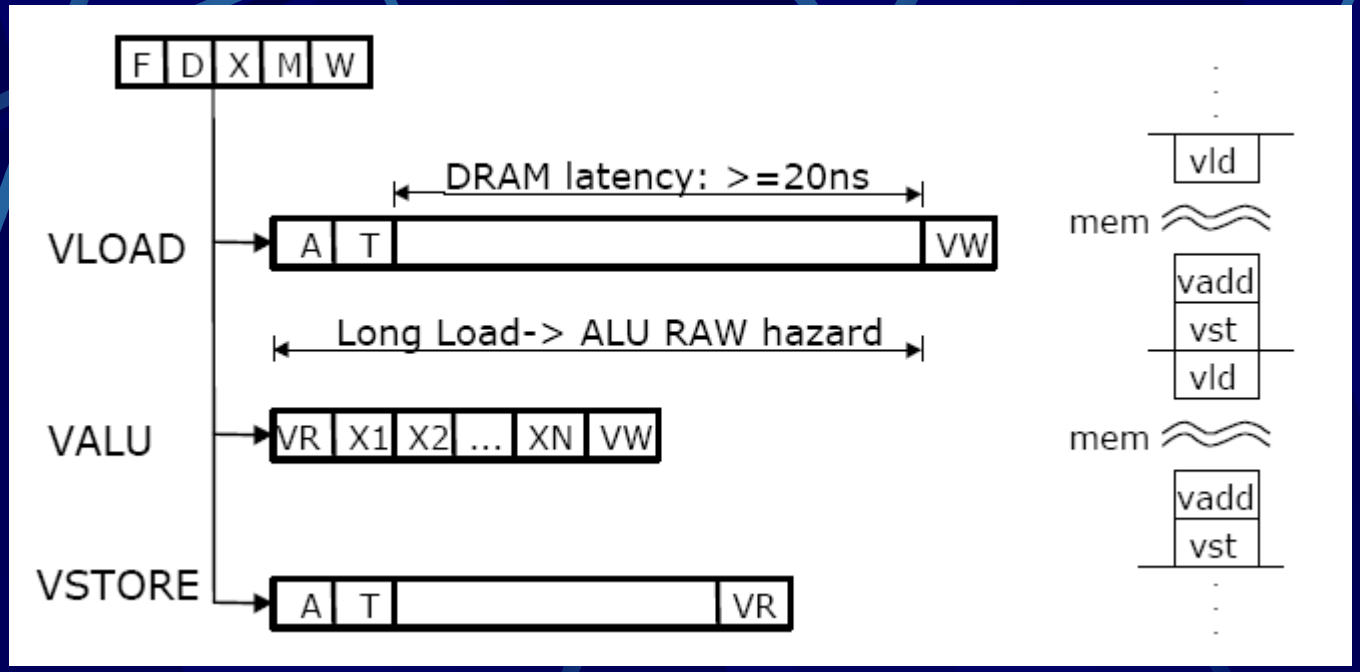
- Vector unit is managed as a co-processor
- Single-issue, in-order pipeline for predictable performance
- Efficient for short vectors
  - Pipelined instruction start-up
  - Full support for instruction chaining
- 256-bit datapath can be configured as 4 64-bit operations, 8 32-bit operations or 16 16-bit operations (SIMD)

Embedded DRAM in VIRAM



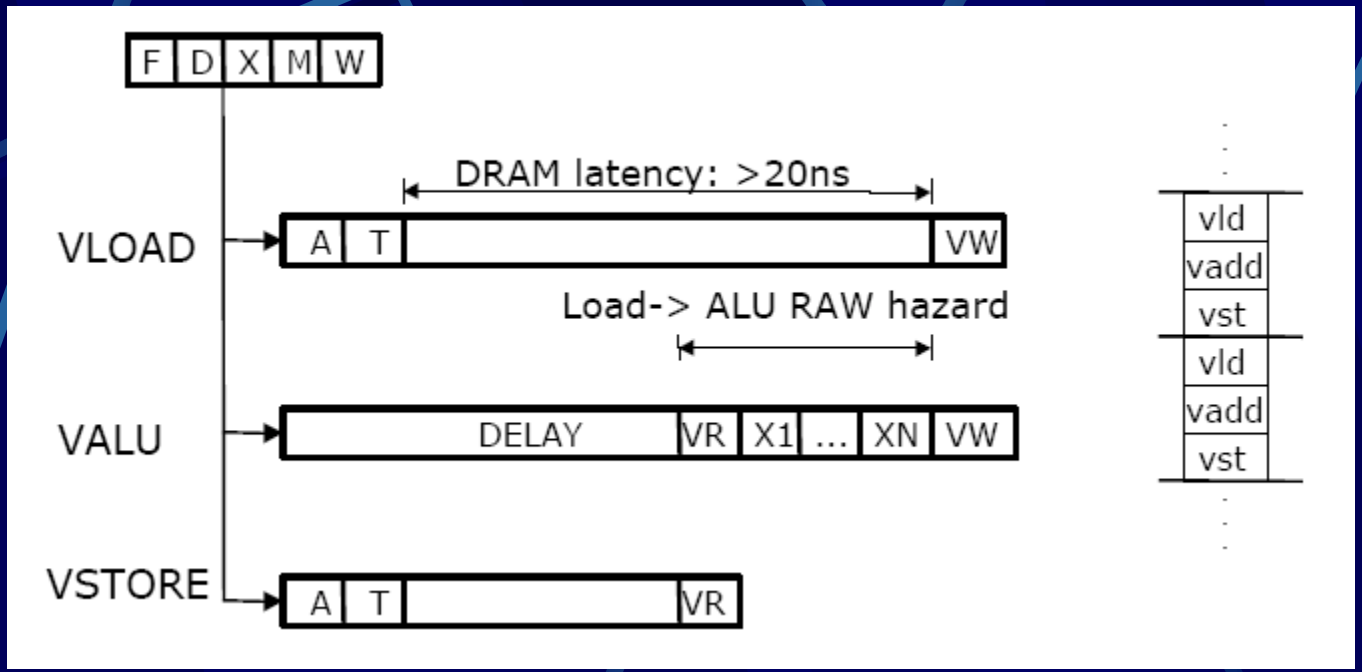
- DRAM – Dynamic RAM – information must be periodically “refreshed” to mimic the behaviour of static storage
- On-chip DRAM connected to vector execution lanes via memory crossbars
- c.f. most SRAM cache-based machines – SRAM is more expensive, less dense
- In conventional architectures:
  - most of the instructions and data are fetched from two lower levels of the memory hierarchy – the L1 and L2 caches which use small SRAM-based memory structures.
  - Most of the reads from the DRAM are not directly from the CPU, but are (burst) reads initiated to bring data and instructions into these caches.
- Each DRAM macro is 1.5MB in size
- DRAM latency is included in the vector execution pipeline

Non-Delayed Pipeline



- Random access latency could lead to stalls due to long load

Delayed Vector Pipeline



- Solution: include random access latency in vector unit pipeline
- Delay arithmetic operations and stores to shorten RAW hazards

Vector Instruction Set

- Complete load-store vector instruction set
  - extends the MIPS64™ ISA with vector instructions
  - Data types supported: 64, 32, 16 and 8 bit
  - 32 general purpose vector registers, 32 vector flag registers, 16 scalar registers
  - 91 instructions: arithmetic, logical, vector processing, sequential/strided/indexed loads and stores
- ISA does not include:
  - Maximum vector register length
  - Functional unit datapath width
- DSP support
  - Fixed-point arithmetic, saturating arithmetic
  - Intra-register permutations for butterfly operations

Vector Instruction Set

- Compiler and OS support
  - Conditional execution of vector operations
  - Support for software speculation of load operations
  - MMU-based virtual memory
  - Restartable arithmetic exceptions
  - Valid and dirty bits for vector registers

Vector IRAM for Desktop/Server Applications?

- Desktop domain:
  - - Do not expect vector processing to benefit integer applications.
  - + Floating point applications are highly vectorizable.
  - + All applications should benefit from low memory latency and high memory bandwidth of vector IRAM.
- Server domain:
  - - Expect to perform poorly due to limited on-chip memory.
  - + Should perform better on decision support instead of online transaction processing.

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Vector IRAM for Desktop/Server Applications?

- Software effort
  - + Vectorizing compilers have been developed and used in commercial environments for decades
  - - But additional work is required to tune compilers for multimedia workloads and make DSP features and data types accessible through high-level languages
- Design complexity
  - + Vector IRAM is highly-modular

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Vector IRAM for Personal Mobile Computing?

- Real-time response
  - + in-order, does not rely on data caches → highly predictable
- Continuous data types
  - Vector model is superior to MMX-like, SIMD extensions:
    - Provides explicit control of the number of elements each instruction operates on
    - Allows scaling of the number of elements each instruction operates on without changing the ISA
    - Does not expose data packing and alignment to software

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Vector IRAM for Personal Mobile Computing?

- Fine-grained parallelism
  - Vector processing
- Coarse-grained parallelism
  - High-speed multiply-accumulate achieved through instruction chaining
  - Allow programming in high-level language, unlike most DSP architectures.
- Code size
  - Compactness possible because a single vector instruction specify whole loops
  - Code size is smaller than VLIW; comparable to x86 CISC code
- Memory bandwidth
  - Available from on-chip hierarchical DRAM

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Performance Evaluation of VIRAM

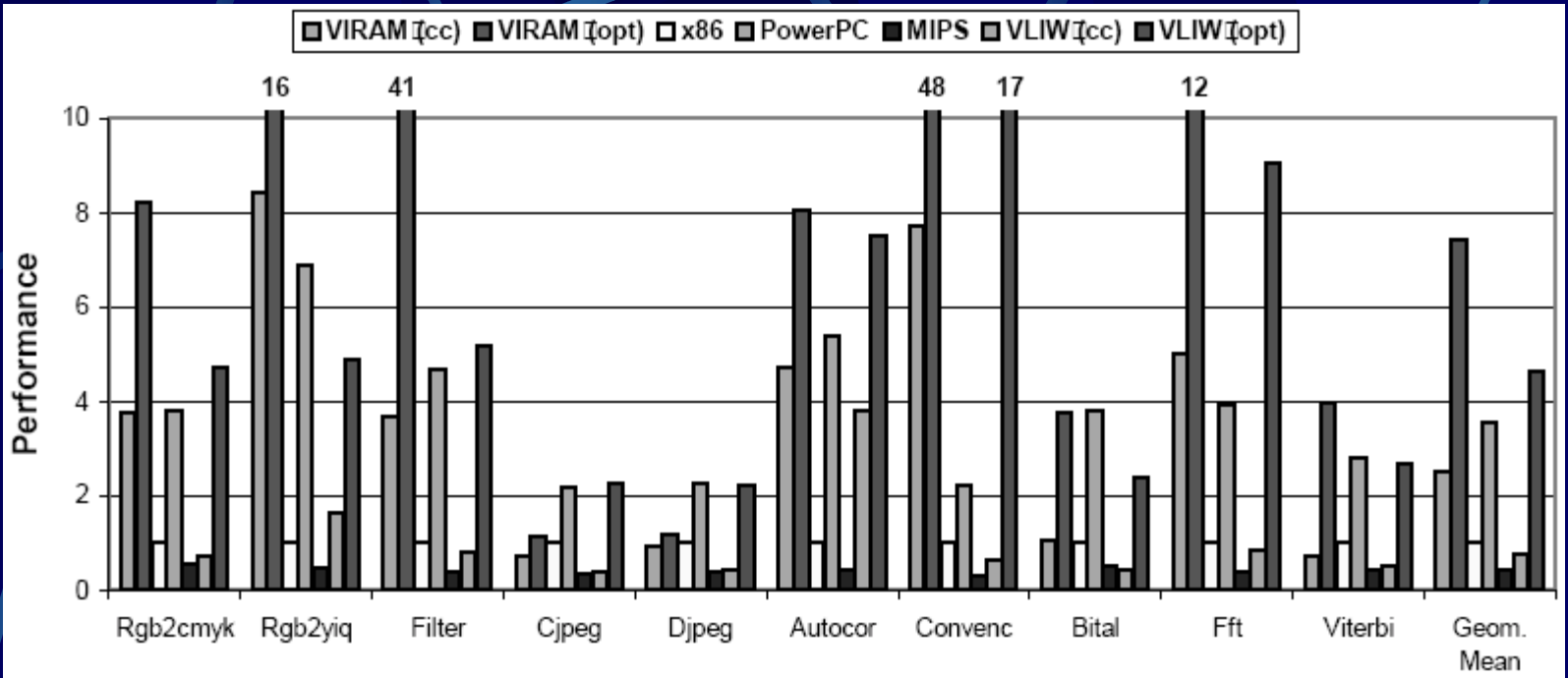


Figure 5. The performance of the embedded processors for the EEMBC benchmarks.

	Architecture	Processor	Issue Width	Execution Style	Cache Size L1I	Cache Size L1D	Cache Size L2	Clock Freq.	Power
VIRAM	Vector	VIRAM	1	in order	8K	–	–	200 MHz	2.0 W
x86	CISC	K6-IIIE+	3	out of order	32K	32K	256K	550 MHz	21.6 W
PowerPC	RISC	MPC7455	4	out of order	32K	32K	256K	1000 MHz	21.3 W
MIPS	RISC	VR5000	2	in order	32K	32K	–	250 MHz	5.0 W
Trimedia	VLIW+SIMD	TM1300	5	in order	32K	16K	–	166 MHz	2.7 W
VelociTI	VLIW+DSP	TMS320C6203	8	in order	96K	512K	–	300 MHz	1.7 W

Table 3. The characteristics of the embedded architectures and processors we compare in this paper.

Performance Evaluation of VIRAM

- Performance is reported in iterations per cycle; and is normalized by the x86 processor.
- With unoptimized code, VIRAM outperforms the x86, MIPS and VLIW processors running unoptimized code; 30% and 45% slower than the 1GHz PowerPC and VLIW processors running optimized code
- With optimized/scheduled code, VIRAM is 1.6 to 18.5 times faster than all others.
- Note:
  - VIRAM is the only single-issue design in the processor set
  - VIRAM is the only one not using SRAM caches
  - VIRAM's clock frequency is the second slowest.

Vector IRAM for Personal Mobile Computing?

	VIRAM (Vector)		x86 (CISC)	PowerPC (RISC)	MIPS (RISC)	Trimedia (VLIW+SIMD)	
	(cc)	*(opt)*				*(cc)*	*(opt)*
Rgb2cmyk	672 (0.9)	272 (0.4)	720 (1.0)	484 (0.7)	1,782 (2.5)	2,560 (3.6)	6,144 (8.5)
Rgb2yiq	528 (0.6)	416 (0.5)	896 (1.0)	492 (0.5)	1,577 (1.8)	4,352 (4.8)	34,560 (38.6)
Filter	1,328 (1.4)	708 (0.7)	944 (1.0)	1,188 (1.3)	1,997 (2.1)	4,672 (4.9)	3,584 (3.8)
Cjpeg	60,256 (2.0)	58,880 (1.9)	30,010 (1.0)	48,440 (1.6)	58,655 (1.9)	114,944 (3.8)	180,032 (6.0)
Djpeg	70,304 (2.0)	68,448 (1.9)	35,962 (1.0)	47,672 (1.3)	58,173 (1.6)	117,440 (3.3)	163,008 (4.5)
Average	**(1.4)**	**(1.1)**	**(1.0)**	**(1.1)**	**(2.0)**	**(4.1)**	**(12.3)**

	VIRAM (Vector)		x86 (CISC)	PowerPC (RISC)	MIPS (RISC)	VelociTI (VLIW+DSP)	
	(cc)	*(opt)*				*(cc)*	*(opt)*
Autocor	1,072 (1.9)	328 (0.6)	544 (1.0)	1,276 (2.3)	1,137 (2.1)	992 (1.8)	1,472 (2.7)
Convenc	704 (0.9)	352 (0.4)	784 (1.0)	1,788 (2.3)	1,618 (2.1)	1,120 (1.4)	2,016 (2.6)
Bital	1,024 (1.5)	592 (0.9)	672 (1.0)	1,820 (2.7)	1,495 (2.2)	2,304 (3.4)	1,376 (2.0)
Fft	3,312 (0.2)	720 (0.1)	15,670 (1.0)	5,868 (0.4)	5,468 (0.3)	2,944 (0.2)	3,552 (0.2)
Viterbi	2,592 (1.9)	1,152 (0.9)	1,344 (1.0)	6,648 (4.9)	3,799 (2.8)	1,920 (1.4)	2,560 (1.9)
Average	**(1.3)**	**(0.6)**	**(1.0)**	**(2.5)**	**(1.9)**	**(1.6)**	**(1.9)**

Table 4. The code size comparison between vector, RISC, CISC, and VLIW architectures. The numbers in parenthesis represent the ratio to the x86 code size.

Vector IRAM for Personal Mobile Computing?

- Energy/power efficiency
  - Vector instruction specifies a large number of independent operations → no energy wasted for fetching and decoding instruction; checking dependencies and making predictions
  - Execution model is in-order → limited forwarding is needed, simple control logic and thus power efficient
  - Typical power consumption:
    - MIPS core: 0.5W
    - Vector unit: 1.0 W
    - DRAM: 0.2 W
    - Misc: 0.3 W

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Vector IRAM for Personal Mobile Computing?

- Design scalability
  - The processor-memory crossbar is the only place where vector IRAM uses long wires
  - Deep pipelining is a viable solution without any h/w or s/w complications

Table 4. Evaluation of vector IRAM for two computing domains.

Domain	Characteristics	Rating
Desktop/server computing	SPECint04	–
	SPECfp04	+
	TPC-F	0
	Software effort	0
	Physical-design complexity	0
Personal mobile computing	Real-time response	+
	Continuous data types	+
	Fine-grained parallelism	+
	Coarse-grained parallelism	0
	Code size	+
	Memory bandwidth	+
	Energy/power efficiency	+
	Design scalability	0

Vector IRAM for Personal Mobile Computing?

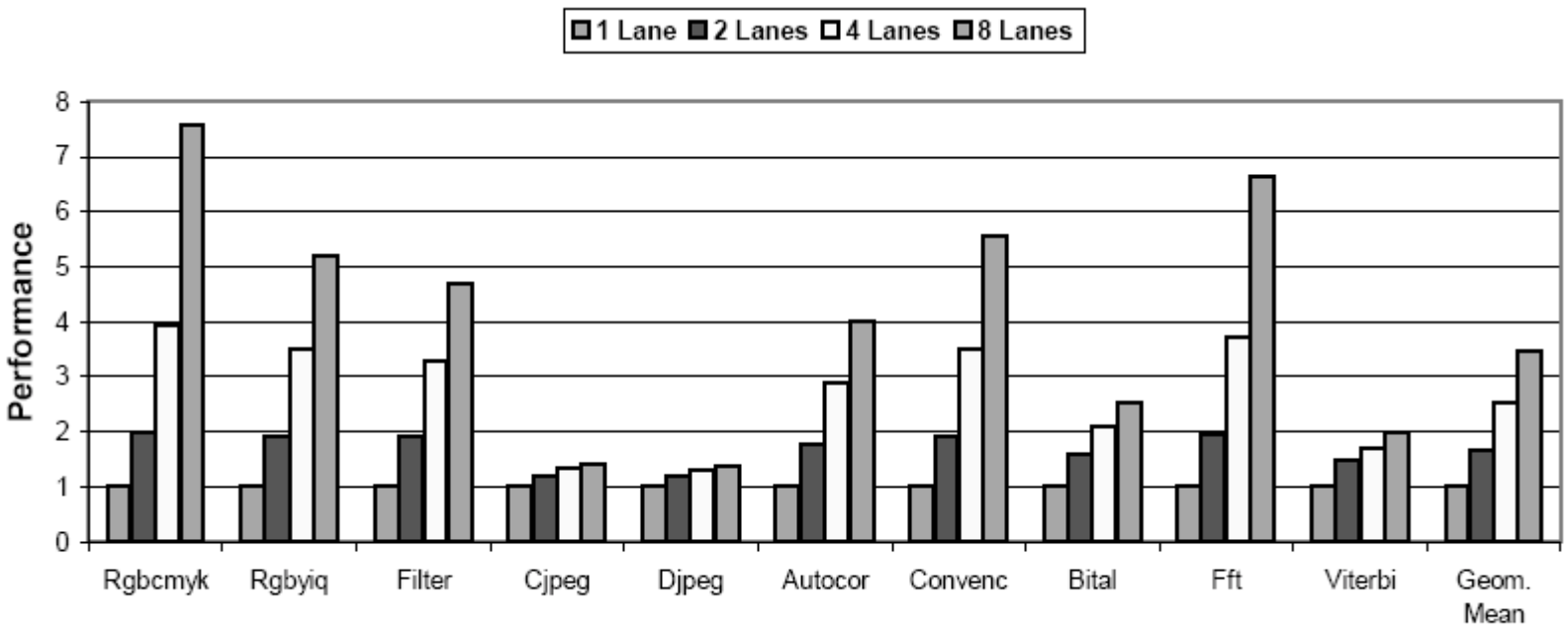


Figure 7. The performance of the VIRAM processor as a function of the number of vector lanes.

- Performance scales well with the number of vector lanes.
- Compared to the single-lane case, two, four and eight lanes lead to ~ 1.5x, 2.5x, 3.5x performance improvement respectively.

Conclusion

- Modern architectures are designed and optimized for desktop and server applications.
- Newly emerging domain – Personal Mobile Computing – poses a different set of architectural requirements.
- We have seen that modern architectures do not meet many of the requirements of applications in the personal mobile computing domain.
- VIRAM – an effort by UC Berkeley to develop a new architecture targeted at applications in the personal mobile computing domain.
- Early results show a promising improvement in performance without compromising the requirements of low power.

References

- *A New Direction for Computer Architecture Research*
  - Christoforos E. Kozyrakis, David A. Patterson, UC Berkeley, Computer Magazine, IEEE Nov 1998.
- *Vector IRAM: A Microprocessor Architecture for Media Processing*
  - Christoforos E. Kozyrakis, UC Berkeley, CS252 Graduate Computer Architecture, 2000.
- *Vector IRAM: A Media-Oriented Vector Processor with Embedded DRAM*
  - C. Kozyrakis, J. Gebis, D. Martin, S. Williams, I. Mavroidis, S. Pope, D. Jones, D. Patterson, K. Yelick. 12th Hot Chips Conference, Palo Alto, CA, August 2000
- *Exploiting On-Chip Memory Bandwidth in the VIRAM Compiler*
  - D. Judd, K. Yelick, C. Kozyraki, D. Martin, and D. Patterson, Second Workshop on Intelligent Memory Systems, Cambridge, November 2000
- *Vector v.s. Superscalar and VLIW Architectures for Embedded Multimedia Benchmarks*
  - C. Kozyrakis, D. Patterson. 35th International Symposium on Microarchitecture, Instabul, Turkey, November 2002
- *Memory-Intensive Benchmarks: IRAM vs. Cache-Based Machines*
  - Brian R. Gaeke, Parry Husbands, Xiaoye S. Li, Leonid Oliker, Katherine A. Yelick, and Rupak Biswas. Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS). Ft. Lauderdale, FL. April, 2002
- *Logic and Computer Design Fundamentals*
  - M. Morris Mano, Charles R. Kime, 3rd edition 2004, Prentice Hall, ISBN 0-13-1911651