A New Direction for Computer Architecture Research

Christoforos E. Kozyrakis David A. Patterson

Computer Science Division
University of California at Berkeley
Berkeley, CA 94720

Abstract

In this paper we suggest a different computing environment as a worthy new direction for computer architecture research: personal mobile computing, where portable devices are used for visual computing and personal communications tasks. Such a device supports in an integrated fashion all the functions provided today by a portable computer, a cellular phone, a digital camera and a video game. The requirements placed on the processor in this environment are energy efficiency, high performance for multimedia and DSP functions, and area efficient, scalable designs.

We examine the architectures that were recently proposed for billion transistor microprocessors. While they are very promising for the stationary desktop and server workloads, we discover that most of them are unable to meet the challenges of the new environment and provide the necessary enhancements for multimedia applications running on portable devices.

We conclude with Vector IRAM, an initial example of a microprocessor architecture and implementation that matches the new environment.

1 Introduction

Advances in integrated circuits technology will soon provide the capability to integrate one billion transistors in a single chip [1]. This exciting opportunity presents computer architects and designers with the challenging problem of proposing microprocessor organizations able to utilize this huge transistor budget efficiently and meet the requirements of future applications. To address this challenge, IEEE Computer magazine hosted a special issue on "Billion Transistor Architectures" [2] in September 1997. The first three articles of the issue discussed problems and trends that will affect future processor design, while seven articles from academic research groups proposed microprocessor architectures and implementations for billion transistor chips. These proposals covered a wide architecture space, ranging from out-of-order designs to reconfigurable systems. In addition to the academic proposals, Intel and Hewlett-Packard presented the basic characteristics of their next generation IA-64 architecture [3], which is expected to dominate the high-performance processor market within a few years.

It is no surprise that the focus of these proposals is the computing domain that has shaped processor architecture for the past decade: the uniprocessor desktop running technical and scientific applications, and the multiprocessor server used for transaction processing and file-system workloads. We start with a review of these proposals and a qualitative evaluation of them for the concerns of this classic computing environment.

In the second part of the paper we introduce a new computing domain that we expect to play a significant role in driving technology in the next millennium: personal mobile computing. In this paradigm, the basic personal computing and communication devices will be portable and battery operated, will support multimedia functions like speech recognition and video, and will be sporadically interconnected through a wireless infrastructure. A different set of requirements for the microprocessor, like real-time response, DSP support and energy efficiency, arise in such an environment. We examine the proposed organizations with respect to this environment and discover that limited support for its requirements is present in most of them.

Finally we present Vector IRAM, a first effort for

The authors can be contacted through email at {kozyraki,patterson}@cs.berkeley.edu.

Architecture	Source	Key Idea	Transistors used for Memory
Advanced Superscalar	[4]	wide-issue superscalar processor with speculative execution and multilevel on-chip caches	910M
Superspeculative Architecture	[5]	wide-issue superscalar processor with aggressive data and control speculation and multilevel on-chip caches	820M
Trace Processor	[6]	multiple distinct cores, that speculatively execute program traces, with multilevel on-chip caches	600M [1]
Simultaneous Multithreaded (SMT)	[7]	wide superscalar with support for aggressive sharing among multiple threads and multilevel on-chip caches	810M
Chip Multiprocessor (CMP)	[8]	symmetric multiprocessor system with shared second level cache	450M [1]
IA-64	[3]	VLIW architecture with support for predicated execution and long instruction bundling	600M [1]
RAW	[9]	multiple processing tiles with reconfigurable logic and memory, interconnected through a reconfigurable network	670M

Table 1: The billion transistor microprocessors and the number of transistors used for memory cells for each one[1]. We assume a billion transistor implementation for the Trace and IA-64 architecture.

a microprocessor architecture and design that matches the requirements of the new environment. Vector IRAM combines a vector processing architecture with merged logic-DRAM technology in order to provide a scalable, cost efficient design for portable multimedia devices.

This paper reflects the opinion and expectations of its authors. We believe that in order to design successful processor architectures for the future, we first need to explore the future applications of computing and then try to match their requirements in a scalable, cost-efficient way. The goal of this paper is to point out the potential change in applications and motivate architecture research in this direction.

2 Overview of the Billion Transistor Processors

Table 1 summarizes the basic features of the billion transistor implementations for the proposed architectures as presented in the corresponding references. For the case of the Trace Processor and IA-64, descriptions of billion transistor implementations have not been presented, hence certain features are speculated.

The first two architectures (Advanced Superscalar and Superspeculative Architecture) have very similar characteristics. The basic idea is a wide superscalar organization with multiple execution units or functional cores, that uses multi-level caching and aggressive prediction of data, control and even sequences of instructions (traces) to utilize all the available instruction level parallelism (ILP). Due their similarity, we group them together and call them "Wide Superscalar" processors in the rest of this paper.

The Trace processor consists of multiple superscalar processing cores, each one executing a trace issued by a shared instruction issue unit. It also employs trace and data prediction and shared caches.

The Simultaneous Multithreaded (SMT) processor uses multithreading at the granularity of issue slot to maximize the utilization of a wide-issue out-of-order superscalar processor at the cost of additional complexity in the issue and control logic.

The Chip Multiprocessor (CMP) uses the transistor budget by placing a symmetric multiprocessor on a single die. There will be eight uniprocessors on the chip, all similar to current out-of-order processors, which will have separate first level caches but will share a

[1]These numbers include transistors for main memory, caches and tags. They are calculated based on information from the referenced papers. Note that CMP uses considerably less than one billion transistors, so 450M transistors is much more than half the budget. The numbers for the Trace processor and IA-64 were based on lower-limit expectations and the fact that their predecessors spent at least half their transistor budget on caches.

large second level cache and the main memory interface.

The IA-64 can be considered as the commercial reincarnation of the VLIW architecture, renamed "Explicitly Parallel Instruction Computer". Its major innovations announced so far are support for bundling multiple long instructions and the instruction dependence information attached to each one of them, which attack the problem of scaling and code density of older VLIW machines. It also includes hardware checks for hazards and interlocks so that binary compatibility can be maintained across generations of chips. Finally, it supports predicated execution through general-purpose predication registers to reduce control hazards.

The RAW machine is probably the most revolutionary architecture proposed, supporting the case of reconfigurable logic for general-purpose computing. The processor consists of 128 tiles, each with a processing core, small first level caches backed by a larger amount of dynamic memory (128 KBytes) used as main memory, and a reconfigurable functional unit. The tiles are interconnected with a reconfigurable network in an matrix fashion. The emphasis is placed on the software infrastructure, compiler and dynamic-event support, which handles the partitioning and mapping of programs on the tiles, as well as the configuration selection, data routing and scheduling.

Table 1 also reports the number of transistors used for caches and main memory in each billion transistor processors. This varies from almost half the budget to 90% of it. It is interesting to notice that all but one do not use that budget as part of the main system memory: 50% to 90% of their transistor budget is spent to build caches in order to tolerate the high latency and low bandwidth problem of external memory.

In other words, the conventional vision of computers of the future is to spend most of the billion transistor budget on redundant, local copies of data normally found elsewhere in the system. Is such redundancy really our best idea for the use of 500,000,000 transistors[2] for applications of the future?

[2]While die area is not a linear function of the transistor number (memory transistors can be placed much more densely than logic transistors and redundancy enables repair of failed rows or columns), die cost is non-linear function of die area [10]. Thus, these 500M transistors are very expensive.

3 The Desktop/Server Computing Domain

Current processors and computer systems are being optimized for the desktop and server domain, with SPEC'95 and TPC-C/D being the most popular benchmarks. This computing domain will likely be significant when the billion transistor chips will be available and similar benchmark suites will be in use. We playfully call them "SPEC'04" for technical/scientific applications and "TPC-F" for on-line transaction processing (OLTP) workloads.

Table 2 presents our prediction of the performance of these processors for this domain using a grading system of "+" for strength, "○" for neutrality, and "−" for weakness.

For the desktop environment, the Wide Superscalar, Trace and Simultaneous Multithreading processors are expected to deliver the highest performance on integer SPEC'04, since out-of-order and advanced prediction techniques can utilize most of the available ILP of a single sequential program. IA-64 will perform slightly worse because VLIW compilers are not mature enough to outperform the most advanced hardware ILP techniques, which exploit run-time information. CMP and RAW will have inferior performance since desktop applications have not been shown to be highly parallelizable. CMP will still benefit from the out-of-order features of its cores. For floating point applications on the other hand, parallelism and high memory bandwidth are more important than out-of-order execution, hence SMT and CMP will have some additional advantage.

For the server domain, CMP and SMT will provide the best performance, due to their ability to utilize coarse-grain parallelism even with a single chip. Wide Superscalar, Trace processor or IA-64 systems will perform worse, since current evidence is that out-of-order execution provides little benefit to database-like applications [11]. With the RAW architecture it is difficult to predict any potential success of its software to map the parallelism of databases on reconfigurable logic and software controlled caches.

For any new architecture to be widely accepted, it has to be able to run a significant body of software [10]. Thus, the effort needed to port existing software or develop new software is very important. The Wide Superscalar and Trace processors have the edge, since they can run existing executables. The same holds for SMT and CMP but, in this case, high performance can be de-

	Wide Superscalar	Trace Processor	Simultaneous Multithreading	Chip Multiprocessor	IA-64	RAW
SPEC'04 Int (Desktop)	+	+	+	○	+	○
SPEC'04 FP (Desktop)	+	+	+	+	+	○
TPC-F (Server)	○	○	+	+	○	-
Software Effort	+	+	○	○	○	-
Physical Design Complexity	-	○	-	○	○	+

Table 2: The evaluation of the billion transistor processors for the desktop/server domain. Wide Superscalar processors includes the Advanced Superscalar and Superspeculative processors.

livered if the applications are written in a multithreaded or parallel fashion. As the past decade has taught us, parallel programming for high performance is neither easy nor automated. For IA-64 a significant amount of work is required to enhance VLIW compilers. The RAW machine relies on the most challenging software development. Apart from the requirements of sophisticated routing, mapping and run-time scheduling tools, there is a need for development of compilers or libraries to make such an design usable.

A last issue is that of physical design complexity which includes the effort for design, verification and testing. Currently, the whole development of an advanced microprocessor takes almost 4 years and a few hundred engineers [2][12][13]. Functional and electrical verification and testing complexity has been steadily growing [14][15] and accounts for the majority of the processor development effort. The Wide Superscalar and Multithreading processors exacerbate both problems by using complex techniques like aggressive data/control prediction, out-of-order execution and multithreading, and by having non modular designs (multiple blocks individually designed). The Chip Multiprocessor carries on the complexity of current out-of-order designs with support for cache coherency and multiprocessor communication. With the IA-64 architecture, the basic challenge is the design and verification of the forwarding logic between the multiple functional units on the chip. The Trace processor and RAW machine are more modular designs. The trace processor employs replication of processing elements to reduce complexity. Still, trace prediction and issue, which involves intra-trace dependence check and register remapping, as well as intra-element forwarding includes a significant portion of the complexity of a wide superscalar design. For the RAW processor, only a single tile and network switch need to be designed and replicated. Verification of a reconfigurable organization is trivial in terms of the circuits, but verification of the mapping software is also required.

The conclusion from Table 2 is that the proposed billion transistor processors have been optimized for such a computing environment and most of them promise impressive performance. The only concern for the future is the design complexity of these organizations.

4 A New Target for Future Computers: Personal Mobile Computing

In the last few years, we have experienced a significant change in technology drivers. While high-end systems alone used to direct the evolution of computing, current technology is mostly driven by the low-end systems due to their large volume. Within this environment, two important trends have evolved that could change the shape of computing.

The first new trend is that of multimedia applications. The recent improvements in circuits technology and innovations in software development have enabled the use of real-time media data-types like video, speech, animation and music. These dynamic data-types greatly improve the usability, quality, productivity and enjoyment of personal computers [16]. Functions like 3D graphics, video and visual imaging are already included in the most popular applications and it is common knowledge that their influence on computing will only increase:

- "90% of desktop cycles will be spent on 'media' applications by 2000" [17]
- "multimedia workloads will continue to increase in importance" [2]

Figure 1: Personal mobile devices of the future will integrate the functions of current portable devices like PDAs, video games, digital cameras and cellular phones.

- "many users would like outstanding 3D graphics and multimedia" [12]
- "image, handwriting, and speech recognition will be other major challenges" [15]

At the same time, portable computing and communication devices have gained large popularity. Inexpensive "gadgets", small enough to fit in a pocket, like personal digital assistants (PDA), palmtop computers, webphones and digital cameras were added to the list of portable devices like notebook computers, cellular phones, pagers and video games [18]. The functions supported by such devices are constantly expanded and multiple devices are converging into a single one. This leads to a natural increase in their demand for computing power, but at the same time their size, weight and power consumption have to remain constant. For example, a typical PDA is 5 to 8 inches by 3.2 inches big, weighs six to twelve ounces, has 2 to 8 MBytes of memory (ROM/RAM) and is expected to run on the same set of batteries for a period of a few days to a few weeks [18]. One should also notice the large software, operating system and networking infrastructure developed for such devices (wireless modems, infra-red communications etc): Windows CE and the PalmPilot development environment are prime examples [18].

Our expectation is that these two trends together will lead to a new application domain and market in the near future. In this environment, there will be a single personal computation and communication device, small enough to carry around all the time. This device will include the functions of a pager, a cellular phone, a laptop computer, a PDA, a digital camera and a video game combined [19][20] (Figure 1) . The most important feature of such a device will be the interface and interaction with the user: voice and image input and output (speech and voice recognition) will be key functions used to type notes, scan documents and check the surrounding for specific objects [20]. A wireless infrastructure for sporadic connectivity will be used for services like networking (www and email), telephony and global positioning system (GPS), while the device will be fully functional even in the absence of network connectivity.

Potentially this device will be all that a person may need to perform tasks ranging from keeping notes to making an on-line presentation, and from browsing the web to programming a VCR. The numerous uses of such devices and the potential large volume [20] lead us to expect that this computing domain will soon become at least as significant as desktop computing is today.

The microprocessor needed for these computing devices is actually a merged general-purpose processor and digital-signal processor (DSP), at the power budget of the latter. There are four major requirements: high performance for multimedia functions, energy/power efficiency, small size and low design complexity.

The basic characteristics of media-centric applications that a processor needs to support or utilize in order to provide high-performance were specified in [16] in the same issue of IEEE Computer:

- *real-time response*: instead of maximum peak performance, sufficient worst case guaranteed performance is needed for real-time qualitative perception for applications like video.
- *continuous-media data types*: media functions are typically processing a continuous stream of input that is discarded once it is too old, and continuously send results to a display or speaker. Hence, temporal locality in data memory accesses, the assumption behind 15 years of innovation in conventional memory systems, no longer holds. Remarkably, data caches may well be an obstacle to high performance for continuous-media data types. This data is also narrow, as pixel images and sound samples are 8 to 16 bits wide, rather than the 32-bit or 64-bit data of desktop machines. The ability to perform multiple operations on such types on a single wide datapath is desirable.

	Wide Superscalar	Trace Processor	Simultaneous Multithreading	Chip Multi Processor	IA-64	RAW
Real-time response	−	−	○	○	○	○
	unpredictability of out-of-order, branch prediction and/or caching techniques					
Continuous Data-types	○	○	○	○	○	○
	caches do not efficiently support data streams with little locality					
Fine-grained Parallelism	○	○	○	○	○	+
	MMX-like extensions less efficient than full vector support					reconfigurable logic unit
Coarse-grained Parallelism	○	○	+	+	○	+
Code size	○	○	○	○	−	○
	potential use of loop unrolling and software pipelining for higher ILP				VLIW instr.	hardware configuration
Memory Bandwidth	○	○	○	○	○	○
	cache-based designs					
Energy/ power Efficiency	−	−	−	○	○	−
	power penalty for out-of-order schemes, complex issue logic, forwarding and reconfigurable logic					
Physical Design Complexity	−	○	−	○	○	+
Design Scalability	−	○	−	○	○	○
	long wires for forwarding data or for reconfigurable interconnect					

Table 3: The evaluation of the billion transistor processors for the personal mobile computing domain.

- *fine-grained parallelism*: in functions like image, voice and signal processing, the same operation is performed across sequences of data in a vector or SIMD fashion.

- *coarse-grained parallelism*: in many media applications a single stream of data is processed by a pipeline of functions to produce the end result.

- *high instruction-reference locality*: media functions usually have small kernels or loops that dominate the processing time and demonstrate high temporal and spatial locality for instructions.

- *high memory bandwidth*: applications like 3D graphics require huge memory bandwidth for large data sets that have limited locality.

- *high network bandwidth*: streaming data like video or images from external sources requires high network and I/O bandwidth.

With a budget of less than two Watts for the whole device, the processor has to be designed with a power target less than one Watt, while still being able to provide high-performance for functions like speech recognition. Power budgets close to those of current high-performance microprocessors (tens of Watts) are unacceptable.

After energy efficiency and multimedia support, the third main requirement for personal mobile computers is *small size* and *weight*. The desktop assumption of several chips for external cache and many more for main memory is infeasible for PDAs, and integrated solutions that reduce chip count are highly desirable. A related matter is code size, as PDAs will have limited memory to keep down costs and size, so the size of program representations is important.

A final concern is *design complexity*, like in the desktop domain, and *scalability*. An architecture should scale efficiently not only in terms of performance but also in terms of physical design. Long interconnects for on-chip communication are expected to be a limiting factor for future processors as a small region of the chip (around 15%) will be accessible in a single clock cycle [21] and therefore should be avoided.

5 Processor Evaluation for Mobile Multimedia Applications

Table 3 summarizes our evaluation of the billion transistor architectures with respect to personal mobile computing.

The support for multimedia applications is limited in most architectures. Out-of-order techniques and caches make the delivered performance quite unpredictable for guaranteed real-time response, while hardware controlled caches also complicate support for continuous-media data-types. Fine-grained parallelism is exploited by using MMX-like or reconfigurable execution units. Still, MMX-like extensions expose data alignment issues to the software and restrict the number of vector or SIMD elements operations per instruction, limiting this way their usability and scalability. Coarse-grained parallelism, on the other hand, is best on the Simultaneous Multithreading, Chip Multiprocessor and RAW architectures.

Instruction reference locality has traditionally been exploited through large instruction caches. Yet, designers of portable system would prefer reductions in code size as suggested by the 16-bit instruction versions of MIPS and ARM [22]. Code size is a weakness for IA-64 and any other architecture that relies heavily on loop unrolling for performance, as it will surely be larger than that of 32-bit RISC machines. RAW may also have code size problems, as one must "program" the reconfigurable portion of each datapath. The code size penalty of the other designs will likely depend on how much they exploit loop unrolling and in-line procedures to expose enough parallelism for high performance.

Memory bandwidth is another limited resource for cache-based architectures, especially in the presence of multiple data sequences, with little locality, being streamed through the system. The potential use of streaming buffers and cache bypassing would help for sequential bandwidth but would still not address that of scattered or random accesses. In addition, it would be embarrassing to rely on cache bypassing when 50% to 90% of the transistors are dedicated to caches!

The energy/power efficiency issue, despite its importance both for portable and desktop domains [23], is not addressed in most designs. Redundant computation for out-of-order models, complex issue and dependence analysis logic, fetching a large number of instructions for a single loop, forwarding across long wires and use of the typically power hungry reconfigurable logic increase the energy consumption of a single task and the power of the processor.

As for physical design scalability, forwarding results across large chips or communication among multiple core or tiles is the main problem of most designs. Such communication already requires multiple cycles in high-performance out-of-order designs. Simple pipelining of long interconnects is not a sufficient solution as it exposes the timing of forwarding or communication to the scheduling logic or software and increases complexity.

The conclusion from Table 3 is that the proposed processors fail to meet many of the requirements of the new computing model. This indicates the need for modifications of the architectures and designs or the proposal of different approaches.

6 Vector IRAM

Vector IRAM (VIRAM) [24], the architecture proposed by the research group of the authors, is a first effort for a processor architecture and design that matches the requirements of the mobile personal environment. VIRAM is based on two main ideas, vector processing and the integration of logic and DRAM on a single chip. The former addresses many of the demands of multimedia processing, and the latter addresses the energy efficiency, size, and weight demands of PDAs. We do not believe that VIRAM is the last word on computer architecture research for mobile multimedia applications, but we hope it proves to be an promising first step.

The VIRAM processor described in the IEEE special issue consists of an in-order dual-issue superscalar processor with first level caches, tightly integrated with a vector execution unit with multiple pipelines (8). Each pipeline can support parallel operations on multiple media types, DSP functions like multiply- accumulate and saturated logic. The memory system consists of 96 MBytes of DRAM used as main memory. It is organized in a hierarchical fashion with 16 banks and 8 sub-banks per bank, connected to the scalar and vector unit through a crossbar. This provides sufficient sequential and random bandwidth even for demanding applications. External I/O is brought directly to the on-chip memory through high-speed serial lines operating at the range of Gbit/s instead of parallel buses. From a programming point of view, VIRAM can be seen as a vector or SIMD microprocessor.

Desktop/Server Computing		Personal Mobile Computing	
SPEC'04 Int (Desktop)	−	**Real-time Response**	+
SPEC'04 FP (Desktop)	+	**Continuous Data-types**	+
TPC-F (Server)	○	**Fine-grained Parallelism**	+
Software Effort	○	**Coarse-grained Parallelism**	○
Physical Design Complexity	○	**Code Size**	+
		Memory Bandwidth	+
		Energy Efficiency	+
		Design Scalability	○

Table 4: The evaluation of VIRAM for the two computing environments. The grades presented are the medians of those assigned by reviewers.

Table 4 presents the grades for VIRAM for the two computing environments. We present the median grades given by reviewers of this paper, including the architects of some of the other billion transistor architectures.

Obviously, VIRAM is not competitive within the desktop/server domain; indeed, this weakness for conventional computing is probably the main reason some are skeptical of the importance of merged logic-DRAM technology [25]. For the case of integer SPEC'04 no benefit can be expected from vector processing for integer applications. Floating point intensive applications, on the other hand, have been shown to be highly vectorizable. All applications will still benefit from the low memory latency and high memory bandwidth. For the server domain, VIRAM is expected to perform poorly due to limited on-chip memory[3]. A potentially different evaluation for the server domain could arise if we examine decision support (DSS) instead of OLTP workloads. In this case, small code loops with highly data parallel operations dominate execution time [26], so architectures like VIRAM and RAW should perform significantly better than for OLTP workloads.

In terms of software effort, vectorizing compilers have been developed and used in commercial environments for years now. Additional work is required to tune such compilers for multimedia workloads.

As for design complexity, VIRAM is a highly modular design. The necessary building blocks are the in-order scalar core, the vector pipeline, which is replicated 8 times, and the basic memory array tile. Due to the lack of dependencies and forwarding in the vector model and the in-order paradigm, the verification effort is expected to be low.

The open question in this case is the complications of merging high-speed logic with DRAM to cost, yield and testing. Many DRAM companies are investing in merged logic-DRAM fabrication lines and many companies are exploring products in this area. Also, our project is submitting a test chip this summer with several key circuits of VIRAM in a merged logic-DRAM process. We expect the answer to this open question to be clearer in the next year. Unlike the other proposals, the challenge for VIRAM is the implementation technology and not the microarchitectural design.

As mentioned above, VIRAM is a good match to the personal mobile computing model. The design is in-order and does not rely on caches, making the delivered performance highly predictable. The vector model is superior to MMX-like solutions, as it provides explicit support of the length of SIMD instructions, and it does not expose data packing and alignment to software and is scalable. Since most media processing functions are based on algorithms working on vectors of pixels or samples, its not surprising that highest performance can be delivered by a vector unit. Code size is small compared to other architectures as whole loops can specified in a single vector instruction. Memory bandwidth, both sequential and random is available from the on-chip hierarchical DRAM.

VIRAM is expected to have high energy efficiency as well. In the vector model there are no dependencies, so the limited forwarding within each pipeline is needed for chaining, and vector machines do not require chaining to occur within a single clock cycle. Performance comes from multiple vector pipelines work-

[3]While the use of VIRAM as the main CPU is not attractive for servers, a more radical approach to servers of the future places a VIRAM in each SIMM module [27] or each disk [28] and have them communicate over high speed serial lines via crossbar switches.

ing in parallel on the same vector operation as well as from high-frequency operation, allowing the same performance at lower clock rate and thus lower voltage as long as the functional units are expanded. As energy goes up with the square of the voltage in CMOS logic, such tradeoffs can dramatically improve energy efficiency. In addition, the execution model is strictly in order. Hence, the logic can be kept simple and power efficient. DRAM has been traditionally optimized for low-power and the hierarchical structure provides the ability to activate just the sub-banks containing the necessary data.

As for physical design scalability, the processor-memory crossbar is the only place were long wires are used. Still, the vector model can tolerate latency if sufficient fine-grain parallelism is available, so deep pipelining is a viable solution without any hardware or software complications in this environment.

7 Conclusions

For almost two decades architecture research has been focussed on desktop or server machines. As a result of that attention, today's microprocessors are 1000 times faster. Nevertheless, we are designing processors of the future with a heavy bias for the past. For example, the programs in the SPEC'95 suite were originally written many years ago, yet these were the main drivers for most papers in the special issue on billion transistor processors for 2010. A major point of this article is that we believe it is time for some of us in this very successful community to investigate architectures with a heavy bias for the future.

The historic concentration of processor research on stationary computing environments has been matched by a consolidation of the processor industry. Within a few years, this class of machines will likely be based on microprocessors using a single architecture from a single company. Perhaps it is time for some of us to declare victory, and explore future computer applications as well as future architectures.

In the last few years, the major use of computing devices has shifted to non-engineering areas. Personal computing is already the mainstream market, portable devices for computation, communication and entertainment have become popular, and multimedia functions drive the application market. We expect that the combination of these will lead to the personal mobile computing domain, where portability, energy efficiency and efficient interfaces through the use of media types (voice and images) will be the key features.

One advantage of this new target for the architecture community is its unquestionable need for improvements in terms of "MIPS/Watt", for either more demanding applications like speech input or much longer battery life are desired for PDAs. Its less clear that desktop computers really need orders of magnitude more performance to run "MS-Office 2010".

The question we asked is whether the proposed new architectures can meet the challenges of this new computing domain. Unfortunately, the answer is negative for most of them, at least in the form they were presented. Limited and mostly "ad-hoc" support for multimedia or DSP functions is provided, power is not a serious issue and unlimited complexity of design and verification is justified by even slightly higher peak performance.

Providing the necessary support for personal mobile computing requires a significant shift in the way we design processors. The key requirements that processor designers will have to address will be energy efficiency to allow battery operated devices, focus on worst case performance instead of peak for real-time applications, multimedia and DSP support to enable visual computing, and simple scalable designs with reduced development and verification cycles. New benchmarks suites, representative of the new types of workloads and requirements are also necessary.

We believe that personal mobile computing offers a vision of the future with a much richer and more exciting set of architecture research challenges than extrapolations of the current desktop architectures and benchmarks. VIRAM is a first approach in this direction.

Put another way, which problem would you rather work on: improving performance of PCs running FPPPP or making speech input practical for PDAs?

8 Acknowledgments

The ideas and opinions presented in this paper are the result of discussions within the IRAM group in U.C. Berkeley.

In addition, we want to thank the following people for their useful feedback, comments and criticism on earlier drafts, as well as the grades for VIRAM: Anant Agarwal, Jean-Loup Baer, Gordon Bell, Pradeep

Dubey, Lance Hammond, Wang Wen-Hann, John Hennessy, Mark Hill, John Kubiatowicz, Corinna Lee, Henry Levy, Doug Matzke, Kunle Olukotun, Jim Smith and Gurindar Sohi.

This research is supported by DARPA (DABT63-C-0056), the California State MICRO program, NSF (CDA-9401156) and by research grants from LG Semicon, Hitachi, Intel, Microsoft, SGI/Cray, Sun Microsystems and Texas Instruments.

References

[1] Semiconductor Industry Association. *The National Technology Roadmap for Semiconductors*. SEMATECH Inc., 1997.

[2] D. Burger and D. Goodman. Billion-Transistor Architectures - Guest Editors' Introduction. *IEEE Computer*, 30(9):46–48, September 1997.

[3] J. Crawford and J. Huck. Motivations and Design Approach for the IA-64 64-Bit Instruction Set Architecture. In *the Proceedings of the Microprocessor Forum*, October 1997.

[4] Y.N. Patt, S.J. Patel, M. Evers, D.H. Friendly, and J. Stark. One Billion Transistors, One Uniprocessor, One Chip. *IEEE Computer*, 30(9):51–57, September 1997.

[5] M. Lipasti and L.P. Shen. Superspeculative Microarchitecture for Beyond AD 2000. *IEEE Computer*, 30(9):59–66, September 1997.

[6] J. Smith and S. Vajapeyam. Trace Processors: Moving to Fourth Generation Microarchitectures. *IEEE Computer*, 30(9):68–74, September 1997.

[7] S.J. Eggers, J.S. Emer, H.M. Leby, J.L. Lo, R.L. Stamm, and D.M. Tullsen. Simultaneous Multithreading: a Platform for Next-Generation Processors. *IEEE MICRO*, 17(5):12–19, October 1997.

[8] L. Hammond, B.A. Nayfeh, and K. Olukotun. A Single-Chip Multiprocessor. *IEEE Computer*, 30(9):79–85, September 1997.

[9] E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar, W. Lee, V. Lee, J. Kim, M. Frank, P. Finch, R. Barua, J. Babb, S. Amarasinghe, and A. Agarwal. Baring It All to Software: Raw Machines. *IEEE Computer*, 30(9):86–93, September 1997.

[10] J Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach, second edition*. Morgan Kaufmann, 1996.

[11] K. Keeton, D.A. Patterson, Y.Q. He, and Baker W.E. Performance Characterization of the Quad Pentium Pro SMP Using OLTP Workloads. In *the Proceedings of the 1998 International Symposium on Computer Architecture (to appear)*, June 1998.

[12] G. Grohoski. Challenges and Trends in Processor Design: Reining in Complexity. *IEEE Computer*, 31(1):41–42, January 1998.

[13] P. Rubinfeld. Challenges and Trends in Processor Design: Managing Problems in High Speed. *IEEE Computer*, 31(1):47–48, January 1998.

[14] R. Colwell. Challenges and Trends in Processor Design: Maintaining a Leading Position. *IEEE Computer*, 31(1):45–47, January 1998.

[15] E. Killian. Challenges and Trends in Processor Design: Challenges, Not Roadblocks. *IEEE Computer*, 31(1):44–45, January 1998.

[16] K. Diefendorff and P. Dubey. How Multimedia Workloads Will Change Processor Design. *IEEE Computer*, 30(9):43–45, September 1997.

[17] W. Dally. Tomorrow's Computing Engines. Keynote Speech, Fourth International Symposium on High-Performance Computer Architecture, February 1998.

[18] T. Lewis. Information Appliances: Gadget Netopia. *IEEE Computer*, 31(1):59–68, January 1998.

[19] V. Cerf. The Next 50 Years of Networking. In *the ACM97 Conference Proceedings*, March 1997.

[20] G. Bell and J. Gray. *Beyond Calculation, The Next 50 Years of Computing*, chapter The Revolution Yet to Happen. Springer-Verlag, February 1997.

[21] D. Matzke. Will Physical Scalability Sabotage Performance Gains? *IEEE Computer*, 30(9):37–39, September 1997.

[22] L. Goudge and S. Segars. Thumb: reducing the cost of 32-bit RISC performance in portable and consumer applications. In *the Digest of Papers, COMPCON '96*, February 1996.

[23] T. Mudge. Strategic Directions in Computer Architecture. *ACM Computing Surveys*, 28(4):671–678, December 1996.

[24] C.E. Kozyrakis, S. Perissakis, D. Patterson, T. Anderson, K. Asanovic, N. Cardwell, R. Fromm, J. Golbus, B. Gribstad, K. Keeton, R. Thomas, N. Treuhaft, and K. Yelick. Scalable Processors in the Billion-Transistor Era: IRAM. *IEEE Computer*, 30(9):75–78, September 1997.

[25] D. Lammers. Holy grail of embedded dram challenged. EE Times, 1997.

[26] P. Trancoso, J. Larriba-Pey, Z. Zhang, and J. Torrellas. The Memory Performance of DSS Commercial Workloads in Shared-Memory Multiprocessors. In *the Proceeding of the Third International Symposium on High-Performance Computer Architecture*, January 1997.

[27] K. Keeton, R. Arpaci-Dusseau, and D.A. Patterson. IRAM and SmartSIMM: Overcoming the I/O Bus Bottleneck. In *the Workshop on "Mixing Logic and DRAM: Chips that Compute and Remember", the 24th Annual International Symposium on Computer Architecture*, June 1997.

[28] K. Keeton, D.A. Patterson, and J.M. Hellerstein. The Intelligent Disk (IDISK): A Revolutionary Approach to Database Computing Infrastructure. submitted for publication, March 1998.