Scaling Processors to 1 Billion Transistors and Beyond: IRAM

Stylianos Perissakis, Christoforos E. Kozyrakis, Thomas Anderson, Krste Asanović, Neal Cardwell, Richard Fromm, Jason Golbus, Benjamin Gribstad, Kimberly Keeton, David Patterson, Randi Thomas, Noah Treuhaft, Katherine Yelick

Computer Science Division
University of California at Berkeley

Abstract

Conventional architectures have been developed with a transistor budget of a few hundred thousand and have evolved to designs of about 10 million transistors, achieving impressive performance. However, we believe that these architectures will not scale efficiently another hundredfold to utilize billion-transistor chips effectively. Here we introduce an alternative way of using the huge amount of real estate available on such a chip: integrating the processor and the main memory on the same die. We call this architecture IRAM, for Intelligent RAM. We claim that a vector microprocessor is an ideal architecture for an IRAM chip. We discuss some of its merits and present a potential IRAM implementation fabricated in a Gigabit DRAM process.

Keywords: billion transistor chips, scalable performance, IRAM, processor-memory integration, system-on-a-chip, vector architecture

Limitations of Conventional Architectures

The importance of an efficient memory system is increasing as fabrication processes scale down, yielding faster processors and larger memories. This trend widens the processor-memory gap. Not long ago, the off-chip main memory was able to supply the CPU with data at an adequate rate. Today, with processor performance increasing at a rate of about 60% per year and memory latency improving by just 7% per year [1], it takes dozens of clock cycles for data to travel between the CPU and main memory. To bridge this gap, designers are investing vast amounts of resources. Expensive static RAMs are commonly used as off-chip caches, and even within microprocessor chips, an increasing fraction of the area budget is devoted to caches. For instance, almost half of the die area in the Alpha 21164 is occupied by caches, used solely for hiding memory latency. All this cache memory is just a redundant copy of information that would not be necessary if main memory had kept up with processor speed. Still, some applications show poor locality, resulting in low performance even with large caches.

Apart from caches, many other latency tolerance techniques have been proposed and gained considerable acceptance in the industry. For instance, most high-performance processors today combine large caches with some form of out-of-order execution and speculation. Yet these are not efficient solutions for the future, as they require a disproportionate increase in chip area and complexity. Consider, for example, the MIPS R5000 and R10000 processors. The first is a simple RISC processor, while the second is a complex, out-of-order, speculative one. The R10000 takes 3.43 times more area than the R5000, but its performance, as measured by its SPECint95 peak rating, is only 1.64 times higher [2].

Beyond the uniprocessor, the possibility exists for the integration of more than one processor on a single die. But this integration will place even greater demands on the memory system. Clearly, a significant amount of memory should be placed in close proximity to the processors – that is, on the same die. But more processors on a die results in less on-chip memory for each of them, for any given die size, thus increasing the number of slow, off-chip memory accesses. In general, increasing the computing resources on a chip without a corresponding increase in the on-chip memory will lead to a system that is not balanced. Functional units will often be starved for data because of the high latency and limited bandwidth to and from off-chip memory.

Power dissipation is quickly becoming another major concern in processor architecture. The increasing importance of portable electronic systems dictates low power processor and system designs. It is plausible that when billion-transistor chips are available, the desktop will not be the main focus of the computer industry. Integrating more system functions on a single chip will become very important because of the related energy as well as board space advantages.

The IRAM Vision

The IRAM approach to the billion transistor microprocessor is to surround the processor with a large amount of DRAM, rather than SRAM, memory. Its is based on the observation that DRAM can accommodate 30 to 50 times more data than the same chip area devoted to SRAM [5]. This on-chip memory can be treated as main memory, and in many cases the entire application will fit in the on-chip storage. Having the entire memory on chip, coupled to the processor through a high bandwidth and low latency interface, allows for processor designs that demand fast memory systems.

IRAM has several potential advantages [3]. The on-chip memory can support high bandwidth and low latency by using a wide interface and eliminating the delay of pads and buses. Energy consumption in the memory system is decreased several times due to the reduction of off-chip accesses through high-capacitance buses [5]. IRAM's energy efficiency, together with the higher system integration that it offers, can be sufficient for its success in the growing areas of portable and embedded applications.

In addition, IRAM gives the designer the freedom to customize the memory system and the processor-memory interface to the needs of the application at hand: the memory can be divided in as many banks as needed, with custom width and depth and multiple address streams between processor and memory. The timing of the interface can be adjusted to match the demands of the system. For example, it may no longer be necessary to adhere to the common multiplexed address scheme. In low cost systems, where a small amount of memory is preferable but a wide interface is still desirable for performance reasons, today's high density DRAMs force a compromise either in the memory size and cost, or in the interface width and consequently performance. This can be avoided in IRAM systems, where the total memory size can be independent of the width of the interface.

Since the majority of pins in conventional microprocessors are devoted to wide memory interfaces, an IRAM will have a much more streamlined system interface. Fewer pins will result in a smaller package, while serial interfaces like FibreChannel directly attached to the chip can provide ample I/O and network bandwidth without being limited by conventional slow I/O buses. Finally, the on-chip processor can be used as a built-in self-test (BIST) engine to test the memory, thereby reducing the cost of the chip.

IRAM provides an exciting base for research on architectures that can turn its high bandwidth and low latency into application speedups. Yet, for any new architecture to be widely accepted, it has to be able to run a significant body of software. The more revolutionary the software model is, the higher the cost-performance benefit of the architecture must be [1]. Innovations that are binary compatible with existing software, like cache design or branch prediction, can be successful with just a small performance advantage. Architectures that require recompilation of existing programs demand several times improvement in performance. Instruction set innovations, like RISC, fall into this category. Finally, given the rapid rate of processor performance improvement and the long time needed for software development, if software needs to be rewritten, then the payoff must be even larger.

Vector IRAM

For IRAM systems the limit to performance is not the bandwidth that the memory can provide, but the bandwidth that the processor can consume. Conventional microprocessor architectures implemented as IRAMs have little to gain from the on-chip main memory [4]. One architecture that appears to be a natural match to IRAM because of its bandwidth demands is the vector microprocessor [6]. A vector IRAM organization (V-IRAM) is a promising solution for providing scalable performance in the billion-transistor era. In our model, the vector processor contains multiple parallel pipelines that operate concurrently. The vector registers are striped across the pipelines, allowing multiple vector elements to be processed in a clock cycle. Increasing the number of pipelines provides a straightforward way to scale performance, as the capacity of integrated circuits increases.

Although vector architectures are commonly associated with very expensive supercomputers, V-IRAM is a cost-effective system, as it provides a scalar processor with a vector unit and its memory system on a single die. Vector computers today often use SRAM main memory for low latency and exotic packaging technology to provide enough bandwidth to the processor. These costs are avoided by the single-chip vector IRAM.

Vector processors have traditionally been used for scientific calculations, but many other applications could benefit from a low cost vector microprocessor. Emerging applications like multimedia (video, image and audio processing) are inherently vectorizable, as a vector instruction set is the natural way to express concurrent operations on arrays of data, like pixels or audio samples. Many database primitives, like sort, search and hash join, have been vectorized and memory intensive database applications like decision support and data mining could benefit from IRAM systems with a vector processor. It is striking

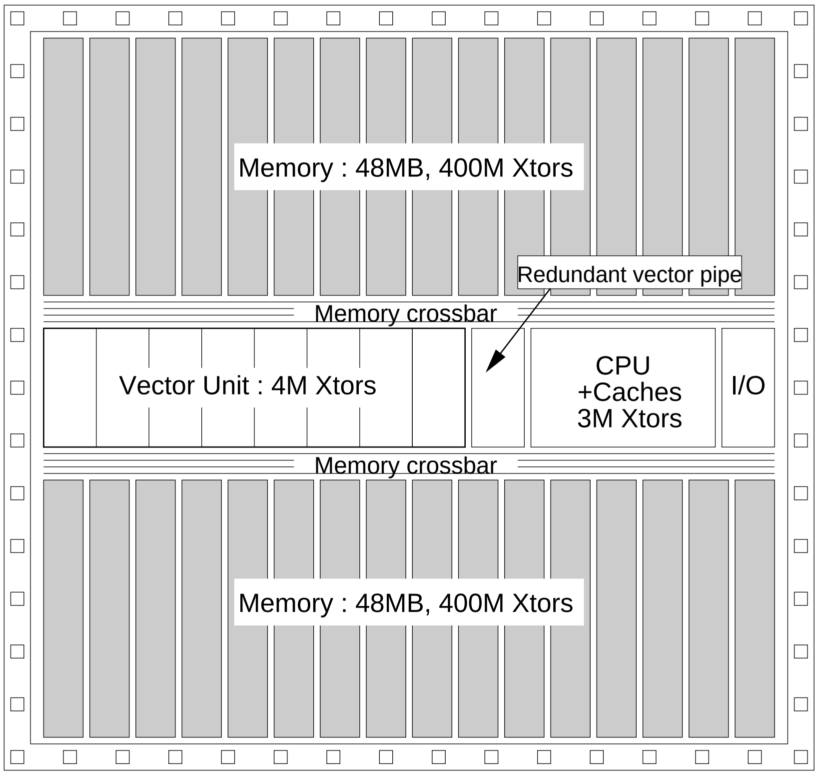

Figure 1: Potential V-IRAM floorplan.

that even integer applications that are not commonly considered to be vectorizable can often achieve significant speedup through vectorization of their inner loops. For example, the SPECint95 benchmark `m88ksim` and data decompression achieve speedups of 42% and 36% respectively through vectorization [6]. On PGP encryption, a vector microprocessor has been shown to significantly outperform an aggressive superscalar processor while occupying less than one tenth of the die area [7]. In addition to the vector processor, IRAM includes a fast scalar processor with SRAM primary caches, so codes that have nothing to gain from a vector processor will still benefit from the fast main memory system.

Vector programming provides a simple way to exploit fine grain data parallelism. A large amount of research and development has been invested to date on vectorizing compilers and programmer annotations to aid in vectorization. Such compilers have been in use by the community for years. In contrast, compilers for VLIW, multithreaded, and MIMD multiprocessors are much more experimental and typically require much more programmer intervention.

Because of the simplicity of their circuits, vector processors can operate at higher clock speeds than other architectural alternatives. Simpler logic, higher code density and the ability to selectively activate the vector and scalar units when necessary, provide higher energy efficiency as well. Energy efficiency has increased importance in the IRAM context, where it is necessary to keep the die temperature relatively low, in order to keep the data retention time at an acceptable level. Finally, a vector unit with a wide interface to memory can operate as a parallel BIST engine, further reducing the testing time of the DRAM and the associated cost.

Thus, an IRAM vector microprocessor is a general-purpose, high-performance, cost-effective, scalable architecture for future systems.

The capacity of 1 billion transistors provides enough hardware resources for the needed processing units and memory on a single chip. A possible floorplan of a Gigabit generation V-IRAM is shown in Figure 1. A minimum feature size of 0.13μm and a 400mm^2 die will be typical for first generation production chips. Here we assume a full size DRAM die with a quarter of the area dedicated to logic instead of memory. The vector unit consists of 2 load, 1 store and 2 arithmetic units, each with 8 64-bit pipelines, running at 1GHz. Given that clock rates of 600MHz have already been achieved for superscalar microprocessors, a 1GHz clock rate is a realistic, if not conservative, projection. Hence, the peak performance of this IRAM implementation is 16GFLOPS (at 64bits/operation), or 128GOPS, when each pipeline is split to multiple 8-bit pipelines for multimedia operations. The on-chip memory system has a total capacity of 96MBytes and is organized as 32 sections, each comprised of 16 1.5Mbit banks, and an appropriate crossbar switch. Assuming a pipelined SDRAM-like interface with 20nsec latency and a 4nsec cycle, the memory system can meet the bandwidth demands of the vector unit, at 192GBytes/sec. V-IRAM also includes a dual-issue processor with first-level instruction and data caches.

IRAM Challenges

For IRAM to succeed in becoming a mainstream architecture, a number of critical issues have to be resolved. Early applications of integrated memory and logic, mainly in the area of graphics, have demonstrated that process and circuit issues like noise, yield and data retention can be overcome. The slower transistors of current DRAM processes are a problem, but developments in the DRAM industry promise that future DRAM transistors will improve and approach the performance of transistors in logic processes [8].

A more serious consideration is the bounded amount of DRAM that can fit on a single IRAM. At the Gigabit generation, 96MBytes may be sufficient for PCs and portable computers, but not for high-end workstations and servers. A potential solution is to back up IRAM with commodity external DRAM. In this case, the off-chip memory could be managed as secondary storage with pages swapped between on-chip and off-chip memory. Alternatively, multiple IRAMs could be interconnected with a high speed network to form a parallel computer. Ways to achieve this have already been proposed in the literature [9] [10] [11]. However, historical trends indicate that the end-user demand for memory scales at a much lower rate than the available capacity per chip. So, over time a single IRAM will be sufficient for increasingly larger systems, from portable and low-end PCs to workstations and servers.

Finally, a vector architecture is not the only option for IRAM. With the dramatic improvements in memory bandwidth and latency that IRAM offers, many other ideas may become practical. Our research goals include further evaluation of alternative architectures and investigation of other ways of translating the phenomenal bandwidth and low latency of IRAM into application performance.

Acknowledgments

This research is supported by DARPA (DABT63-C-0056), the California State MICRO program and by research grants from Intel and Sun Microsystems.

References

[1] J.L.Hennessy, D.A.Patterson, *Computer Architecture: A Quantitative Approach*, second edition, Morgan Kaufmann, San Mateo, California, 1996.

[2] SPEC Disclosures for Siemens Nixdorf RM300/C50 and SGI Origin2000 R10K, `http://www.specbench.org/osg/cpu95/results`

[3] D.Patterson, T.Anderson, N.Cardwell, R.Fromm, K.Keeton, C.Kozyrakis, R.Thomas, K.Yelick, "A Case for Intelligent RAM: IRAM", IEEE Micro, vol.17, no.2, April 1997.

[4] N.Bowman, N.Cardwell, C.E.Kozyrakis, C.Romer, H.Wang, "Evaluation of Existing Architectures in IRAM Systems", *Workshop on Mixing Logic and DRAM, ISCA-97*, Denver, CO, 1 June 1997.

[5] R.Fromm, S.Perissakis, N.Cardwell, C.E.Kozyrakis, B.McGaughy, D.Patterson, T.Anderson, K.Yelick, "The Energy Efficiency of IRAM Architectures", *24th Annual International Symposium on Computer Architecture*, Denver, CO, 2-4 June 1997.

[6] K.Asanović, *Vector Microprocessors*, Ph.D. thesis, University of California, Berkeley, CA, 1997.

[7] C.Lee, *Workshop on Mixing Logic and DRAM, ISCA-97*, Denver, CO, 1 June 1997.

[8] Panel Discussion: "DRAM + Logic Integration: Which Architecture and Fabrication Process ?" *IEEE International Solid State Circuits Conference*, San Francisco, CA, February 1996.

[9] K.Keeton, R.Arpaci-Dusseau, D.A.Patterson, "IRAM and SmartSIMM: Overcoming the I/O Bus Bottleneck", *Workshop on Mixing Logic and DRAM, ISCA-97*, Denver, CO, 1 June 1997.

[10] D.C.Burger, S.Kaxiras, J.R.Goodman, "DataScalar Architectures", *24th Annual International Symposium on Computer Architecture*, Denver, CO, 2-4 June 1997.

[11] S.Kaxiras, R.Sugumar, J.Scharzmeier, "Distributed Vector Architecture: Beyond a Single Vector IRAM", *Workshop on Mixing Logic and DRAM, ISCA-97*, Denver, CO, 1 June 1997.