Intelligent RAM: a Radical Solution?

João Paulo Portela Araújo

Departamento de Informática, Universidade do Minho
4710 – 057 Braga, Portugal
cei5076@di.uminho.pt

Abstract. The goal of an "intelligent" RAM is to design a cost-effective computer by including processing capabilities into a memory device, as an alternative to the current processor chips that contain part of the memory hierarchy on-chip. This merge leads to a reduction in memory latency, an increase in memory bandwidth, and an improvement in energy efficiency, and it also supports a more flexible selection of memory size and organization. This communication reviews current approaches and their most important features, exploring their opportunities and challenges.

1 Introduction

The rate of improvement in microprocessor speed exceeds the rate of improvement in DRAM memory speed[1]. There are a number of reasons to account for this growing disparity. The division of the semiconductor industry into microprocessor and memory camps is the prime reason, though it has its own advantages. Microprocessor fabrication lines offer fast transistors to make fast logic and many metal layers to accelerate communication and simplify power dissipation, while DRAM fabrication lines offer many polysilicon layers to achieve both small DRAM cells and low leakage current to reduce DRAM refresh rate. Separate chips also mean separate packages, allowing microprocessors to use expensive packages that dissipate high power (5 to 50 watt) and provide hundreds of pins to make wider connection to external memory, while allowing DRAMs to use inexpensive packages which dissipate low power (1 watt) and use only a few dozen pins. Separate packages in turn mean computer designers can scale the number of memory chips independent of the number of processors.

As mentioned above, microprocessor and DRAM technologies are headed in different directions: the former is increasing in speed while the latter is increasing in capacity. This technological difference has led to what is known as the processor-memory performance gap. In this communication some techniques to improve effective memory bandwidth are presented, that guide us into a direction: intelligent memories.

The integration of processor and memory in a single DRAM chip has been proposed in order to overcome this problem. Such an architecture will provide high memory bandwidth and low memory latency. Four models are discussed in this communication: Active Pages, CRAM, PPRAM and IRAM. For their enormous potentialities, IRAM is portrayed with larger detail in section 5, where it is discussed the theme of the extraction of parallelism from the code and existent adaptation forms in the vector processor for code parallelizable or not-parallelizable. Also in this section is analysed the Berkeley's project, where are the challenges for these architectures, and where the problems the companies will have to face are identified. It also addresses the lines so that they succeed not only in the embedded systems but also in the general-purpose processors.

2 Bridging the Processor-Memory Performance Gap

Having the industry split into two camps - memory manufacturers and processor manufac-

turers, also has its inherent disadvantages. While microprocessor performance has been improving at a rate of 60% per year, the access time to DRAM has been improving at less than 10% per year [2].

Thus though each is improving exponentially, the exponent for the microprocessor is substantially larger than that for the DRAMs. The difference between diverging exponentials also grows exponentially, so although the disparity between processor and memory speed is already an issue, downstream someplace it will be a much bigger one.

Because of the growing memory access latencies, any request that misses in the caches may eventually take hundred of cycles to satisfy. Thus system speed will now be dominated by memory performance. Quantitatively the problem involved is reducing the average memory access time. This involves primarily improving each of the three factors of hit time, miss penalty and miss rate [1]. The two factors of memory latency and memory bandwidth are closely related. Improvement in memory bandwidth is critically necessary to support aggressive memory latency techniques. The range of techniques to improve effective memory bandwidth is described below.

2.1 Wider and Faster Connection to Memory

With it is of public knowledge, memory bandwidth is the amount of data bits transferred per second. Therefore traditional approaches to improving the memory bandwidth include speeding up the memory clock, increasing the bus width, or both. For conventional DRAMs, these approaches are reaching their practical limits. Of late, novel memory system interfaces like Rambus (RDRAM) and Synchronous link (SLDRAM) have emerged which promise bandwidth on the order of multi-gigabytes/second.

2.2 Larger On-Chip Caches

Larger caches improve effective bandwidth by sending fewer requests (misses) across the interconnect. Present day on-chip caches are reaching the megabyte range. Although caches this large will be able to hold the working sets for many applications, there will always be programs whose access patterns aren't amenable to caching. There will also be programs with working sets that are too large to fit in these caches.

2.3 Dynamic Access Ordering [3]

This method maximizes memory performance for streaming computations (e.g., signal/image processing, multimedia compression and decompression). This approach is based on access ordering, or changing the order of memory requests to improve the rate at which those requests are serviced by a memory system with non-uniform access times.

2.4 Logic/DRAM Integration

The processor-memory bandwidth gap is becoming an increasing impediment to performance, which might shift the focus to integrating the processor on the same die (or in the same package) as the main memory. This eliminates the need for expensive, high bandwidth inter-chip interconnects. However, as mentioned earlier, the difference between manufacturing processors (logic) and memory gives rise to a lot of challenges to make this technology feasible. The on-chip architecture must be efficiently arranged for improved flexibility in connected DRAM arrays and logic circuits.

The next section gives architectural descriptions of a few intelligent memory models, each one have obviously its working prototype.

3. Intelligent Memories

The growing processor-memory performance gap has spawned a number of intelligent memory projects that attempt to improve memory performance. Those that are discussed in this paper are Active Pages, CRAM, PPRAM and IRAM; each model has its advantages and shortcomings.

3.1 Active Pages [4]

The Active Pages model divides up all the memory in a DRAM chip into equally sized pages and assigns a logic block to each page. This logic block can be built in a number of ways, e.g. as a simple processor, as a piece of reconfigurable fabric, etc. Therefore, more complex operations such as float point arithmetic have to be done at the CPU. Since a typical application contains both simple and complex operations, task breakdown, or "task partitioning," is necessary and is done at the software level using a number of Active Pages specific functions.

Under the Active Pages model, the CPU becomes a dispatcher that rapidly assigns tasks to all available Active Pages in the system. These pages then do computations in parallel and report the results back to the CPU if necessary. Since only complex operations are sent to the CPU, the data bus is no longer a bottleneck. As a result, the CPU wastes less time in idle cycles.

3.2 CRAM [5]

In order to take advantage of DRAM's high internal bandwidth, CRAM researchers assign every sense amplifier to a processing element (PE). At a high level, this PE consists of a simple Single Instruction Multiple Data (SIMD) processor and a set of buses for data transfer. The processor has two 1-bit registers, X and Y, and an arbitrary function Arithmetic Logic Unit (ALU). This ALU accepts two inputs and places the 1-bit output onto the result bus. One of the two ALU input ports, call it `A` for ease of reference, has three bits, coming from register X, Y, and the sense amplifier. The other input port, `B`, contains an 8-bit instruction from the global bus. Essentially, the ALU is a multiplexor with `A` being the 3-bit selector that controls which one of the 8-bits from `B` gets placed onto the result bus.

3.3 PPRAM [6]

A PPRAM system is made up of multiple PPRAM nodes and these nodes have 3 components: a logic block, a memory block, and a communication block. The logic block can be anything from a general purpose processor to an I/O controller. Each logic block is accompanied by a memory block made up of conventional memory, DRAM, SRAM, etc.

Memory is distributed for each processor as local memory (i.e., distributed-memory multiprocessor) to exploit its inherent high memory bandwidth. Memory itself consists of 2-dimensional array of memory cells, and a whole row of the memory array (e.g., 1024 bits) can be read and written at once. Each processor provides more than one row buffer which contains the contents of a row of the memory array. Some of these row buffers works as a sort of cache memory, so that each buffer corresponds to a cache line.

3.4 IRAM [2]

A less aggressive change to the traditional system architecture than PPRAM, UC Berke-

ley's Intelligent RAM model merges the processor, cache, and main memory into one chip.

A high speed bus connects the CPU to the level 1 cache. The on-chip main memory offers a performance comparable to that for a level 2 cache. Between the level 1 cache and memory(DRAM) is the Memory Interface Unit, which is what IRAM uses to tap into the enormous DRAM internal bandwidth. Given the importance of this model, it will be explored with more detail in the next section.

3.5 Summary

The Active Pages model assigns one logic block to each page of memory. Therefore, more memory means more processing power. This explains why RADRAM tends to get better speedups on larger data size.

Similar to Active Pages, CRAM makes memory smarter by adding processing elements to it. This model taps the sense amplifier to exploit the large internal memory bandwidth. Initial results have demonstrated tremendous speedups. However, these nearly impossible speedups are only achievable on highly parallelizable applications.

While Active Pages and CRAM stay somewhat with the traditional architecture, PPRAM calls for a drastic shift in the system paradigm; instead of having CPU, cache, and memory, a PPRAM system consists of a large number of different PPRAM chips communicating through specialized interfaces. Although PPRAM does not have the over 41,000X speedup of the CRAM model, the results have shown that it is at least as viable, if not more feasible, an alternative for future systems as any other approach.

Compared to the PPRAM model, the IRAM model requires a less drastic change to the traditional system architecture; IRAM merges memory and processor into one chip and uses a special memory interface unit to maximize memory to processor bandwidth. As expected, this maximization of bandwidth has helped IRAM's performance on memory-intensive applications.

4 IRAM

4.1 IRAM Approach

The Intelligent RAM (IRAM) approach is to use the on-chip real-estate for dynamic RAM(DRAM) memory instead of SRAM caches. It is based on the fact that DRAM can accommodate 30 to 50 times more data than the same chip area devoted to caches. This on-chip memory can be treated as main memory instead of a redundant copy, and in many cases the entire application will fit into the on-chip storage. Having the entire memory on the chip, coupled to the processor through a high bandwidth and low-latency interface, allows for processor designs that demand fast memory systems.

Advantages. IRAM systems [10] have several potential advantages. 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. Since the majority of pins in conventional microprocessors are devoted to wide memory interfaces, an IRAM can have a much more streamlined interface. Fewer pins result in a smaller package, and serial interfaces that are directly attached to the chip can provide ample I/O bandwidth without being limited by conventional slow I/O buses.

The IRAM approach can be combined with most processor organizations because of the inherent cost advantages of system-level integration. Alas, the first impulse of many com-

puter architects when offered a new technology is simply to build larger caches for conventional architectures. Such designs gain little performance from the on-chip main memory because they were developed with the implicit assumption of a slow memory system that is rarely accessed. Using the IRAM approach creates the potential for superior performance on architectures that can effectively exploit the higher memory bandwidth and lower memory latency.

For any other architecture to be widely accepted, however, it has to be able to run a significant body of software. As the software model becomes more revolutionary, the cost-performance benefit of the architecture must increase for wide acceptance. Given the rapid rate of processor performance improvement and the long time needed for software development, the amount of available code and the simplicity of the programming model are extremely important.

4.2 Vector IRAM Architecture [9]

An architecture that appears to be a natural match to IRAM because of its bandwidth demands and its well understood programming model is the vector processor. In the Berkeley IRAM model, the vector processor consists of a vector execution unit combined with a fast in-order scalar core. The combination of a vector unit with a scalar processor creates a general purpose architecture that can deliver high performance without the issue complexity of superscalar designs or the compiler complexity of VLIW.

Although vector architectures are commonly associated with expensive supercomputers, V-IRAM is a cost-effective system, providing a scalar processor with a vector unit and the memory system on a single die. Vector computers often use SRAM main memory for low latency and use exotic packaging to provide enough bandwidth to the processor, but a single-chip vector VRAM avoids these costs. The vector unit contains multiple parallel pipelines operating concurrently and vector registers 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, without requiring changes to the instruction issue logic or recompilation.

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: a vector instruction set is the natural wav to express concurrent operations of arrays of data, like pixels or audio samples. For example, the Intel MMX extension can be considered a modest vector unit. Many data base 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.

Even integer applications that are nor commonly considered to be vectorizable can often achieve significant speedup through vectorization of their inner loops. In pretty good privacy (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. In addition to the vector processor, vector IRAM includes a fast scalar processor with small SRAM primary caches, so even non-vectorizable codes will benefit from the fast memory system.

Vector programming provides a simple wav lo exploit fine-grain data parallelism. Instruction and data dependencies can be efficiently expressed and passed to the hardware. A large amount of research has been invested in vectorizing compilers and in programmer annotations to aid in vectorization, which have been in the 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.

Although compiler researchers have looked at vector architectures, surprisingly the computer architecture research community has largely ignored vector architectures while advancing superscalar, VLIW, and multithreaded designs. Hence, innovation at the architecture compiler interface may allow even more programs to vectorize, making vector IRAM even more attractive.

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 also provide higher energy efficiency. Energy efficiency has increased importance in the IRAM context, where it is necessary to keep the die temperature relatively low to keep the data retention rate at an acceptable level. Since empirical data suggest that it has to be doubled for every 10 degree increase in die temperature.

Finally, a vector unit with a wide interface to memory can operate as a parallel built-in self-test engine for the memory array, significantly reducing the DRAM testing time and the associated cost.

As the MIPS R5000 demonstrates, a single scalar processor can have a reasonable performance. More over no-chip memory could reduce processor memory latency by factors of 5 to 10 and increase memory bandwidth by factors of 50 to 200. When such a processor is combined with a vector unit and low latency, high band width DRAM memory, it becomes a general purpose, high performance, cost-effective, and scalable system.

4.3 Berkeley V-IRAM System[14]

In a possible “gigabit” generation of V-IRAM [9], a minimum feature size of 0.13 mm and a die of 400 mm^2 will be typical for first-generation production chips-assuming a full-size DRAM die with a quarter of the area dedicated to logic.

The vector unit consists of two load, one store, and two arithmetic units, each with eight 64-bit pipelines running at 1 GHz. Hence, the peak performance of the V-IRAM implementation is 16 GFLOPS (at 64 bits per operation) or 128 GOPS, when each pipeline is split into multiple 8 bit pipelines for multimedia operations.

The on chip memory system has a total capacity of 96 Mbytes and is organized as 32 sections, each with 16 1.5 Mbit banks and an appropriate crossbar switch.

More recently the same authors project, in partnership with to manufacturers of DRAM alterations for the inclusion of 2-way superscalar processors, floating-point co-processor, and the network interface and DMA engine [7].

4.4 Challenges to IRAM

A more serious architectural consideration is the bounded amount of DRAM that can fit on a single IRAM chip. At the gigabit generation, The amount of acceptable memory for the portable computers, won't be with certainty equally acceptable for high end workstations. A potential solution is to back up a single IRAM chip with commodity external DRAM, using the off chip memory 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. Fortunately, historical trends indicate that the end user demand for memory will scale at a lower rate than the available capacity per chip. So, over time a single IRAM chip will be sufficient for increasingly larger systems, from portable and low end PCs to workstations and even servers.

If IRAM technology proves successful, there may be even more dramatic integration of processor and memory, distributing portions of processors closer to the individual memory

banks. The challenge for such innovation is software migration. Fortunately, the synergy between IRAM and vector processing suggests a high performance architecture with mature compiler technology, offering a simple and low cost solution whit software migration for general propose systems.

5 Conclusion

Of the four presented models of Intelligent RAM, all of them have demonstrated performance gain over a range of applications, indicating that merging logic into Dram Chips in it is the way to proceed. In particular, Intelligent RAM (IRAM) merges processing and memory in a single chip to lower memory latency, increase memory bandwidth, and improve energy efficiency as well as to allow more flexible selection of memory size and organization.

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, even integer applications that are nor commonly considered to be vectorizable. In addition, to the vector processor includes a fast scalar processor with small SRAM primary caches, so even non-vectorizable codes will benefit from the fast memory system. Instruction and data dependencies can be efficiently expressed and passed to the hardware.

One distinction that IRAM presents is that since IRAM can use these on-chip memory as the main memory, instead of caches which really keeps redundant copy of the data that can be found elsewhere in the system. A potential problem with using the on-chip memory as the main memory is upgradability, since it is not possible to add more one-chip DRAM one it was been fabricated. A solution is to back up the on-chip DRAM with commodity off-chip DRAM, and swap pages between them as needed. This is almost like using the on-chip DRAM as caches. The difference here is that the on-chip memory do not contain duplicate copies of the data that is on the off-chip DRAM, although it exists the difficulty of keep track of location of data that may move between on-chip and off-chip data.

Comparative studies between high performances embedded architectures[8] demonstrate that vector processor can exploit the high level of vector data parallelism inherent in SVD. Even if the used architectures present similar results, the vector processor overtakes the others, as the dates size grows.

Although IRAM ends up helping more with bandwidth than with latency, this is exactly what is need for a growing class of applications. Almost anything that involves graphics[12], such as compression or 3-D animation, involves manipulating long streams of bits (PDA, Game boy, cameras, cell phone, pager, GPS, Wireless data, speech/vision recognition, etc). Nonetheless, IRAM faces significant roadblocks. Perhaps the most important is DRAM manufacturers historic reluctance to mess with a winning formula. After all, chipmakers have gotten rich stamping out dumb memory for years why change now? Perhaps what's most exciting about IRAM is that it levels the playing field. As Michaei Slater, publisher of Microprocessor Report, points out, established players like Intel are unlikely to begin building IRAM-type devices since they don't know anything about DRAMs. So IRAM may finish off the job that Mr. Patterson started with RISC: destroying the Intel and Motorola duopoly.

Commercial feasibility is something that needs to be addressed in two directions. Fist that DRAM and processor are produced from different fabrication lines; one focuses on density and the other focuses on speed. Put logic and DRAM on the same chip, invariably this means merging two technologically different fabrication lines into one. Second changes are required to the current system architecture. This means that a large amount of

the existing hardware and software will be abandoned. The great challenger it will be to get that the companies accept this philosophy as base for the development in mass not only of embedded systems[13] but for the general propose computers.

I would like to finish this communication recalled the words always presents of Mr. Andrew S. Grove . *"…a strategic inflection point is a time in the life of a business when its fundamentals are about to change. …Let us not mince words: A strategic inflection point can be deadly when unattended to. Companies that begin a decline as a result of its changes rarely recover their previous greatness."*

References

[1] John L. Hennessy and David Patterson: Computer Architecture and Quantitative Approach, Morgan Kaufman, CA, (1996)

[2] Patterson, D., Anderson T., Cardwell N. Fromm R., et al: A Case for Intelligent DRAM: IRAM, IEEE Micro, (April 1997)

[3] McKee, S. A.: Dynamic Access Ordering: Bounds on Memory, University of Virginia, Technical Report CS-94-14, (April 1994)

[4] Oskin M., Chong F., Sherwood T.: Active Pages: A Computation Model for Intelligent Memory, International Symposium on Computer Architecture, Barcelona, (1998)

[5] Elliott D.: Computational Ram: A Memory-SIMD Hybrid and its Application to DSP, The Proceedings of the Custom Integrated circuits Conference, Boston, MA, (3 May 1992)

[6] Murakami,K., Inoue,K., and Miyajima,H.: Parallel Processing RAM (PPRAM) (in English), Japan-Germany Forum on Information Technology, (November 1997)

[7] David Patterson and Katherine Yelick: Bridging the Processor-Memory gap, University of California, Berkeley, CA 94720-1776, (1999)

[8] Jeffrey Herman, John Loo and Xiaoyi Tang: A Comparison of the VIRAM-1 and VLIW Architectures for use on Singular Value Decomposition, University of California, Berkeley, CA 94720-1776 (2001)

[9] Ben Gribstad., et al: Interconnections Issues between Memory and Logic in a Iram System, University of California, Berkeley, CA 94720-1776 (1997)

[10] Patterson, D., et al: Intelligent RAM(IRAM): Chips that compute and remember, International Solid State Circuits Conferences, (February 1997)

[11] David A. Paterson: New Directions in Computer Architecture, EECS, University of California, Berkeley, CA 94720 (1776)

[12] Fromm, R.: Vector IRAM Memory Performance for Image Access Patterns, Technical Report, University of California at Berkeley, (January 2000)

[13] Kozyrakis, C: A Media Enhanced Vector Architecture for Embedded Memory Systems, Technical report UCB/CSD-99-1059, Computer Science Division, University of California at Berkeley, (July 1999)

[14] Patterson, D., et al: Intelligent RAM(IRAM) :The industrial Settings, Applications, nd Architectures, ICCD 1997 International Conference on Computer Design, Austin, TX, USA, 10-12 (October 1997)