FP 14.1 Intelligent RAM (IRAM): Chips that remember *and* compute

David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberley Keeton, Christoforos Kozyrakis, Randi Thomas, and Kathy Yelick

Computer Science/EECS University of California, Berkeley CA

The division of the semiconductor industry into microprocessor and memory camps provides many advantages: fabrication lines can be tailored to a device, packages are tailored to the pinout and power of a device, and the number of memory chips in a computer is independent of the number of processors.

The split has disadvantages as well. While microprocessors have been improving performance by 60% per year, DRAM access time has been improving by 7% per year. This processor-memory performance gap limits many applications. For example, one microprocessor spends 75% of its time in the memory hierarchy for data base and matrix computations[1]. These delays occur despite tremendous resources being spent trying the bridge this gap. Table 1 shows that up to 60% of the area and 90% of the transistors of recent microprocessors are dedicated to the growing "Memory Gap Penalty": on-chip memory latency-hiding hardware such as caches.

The DRAM industry also has difficulties. The number of DRAMs for the minimum memory of PCs is shrinking–-from 32 1-Mb DRAMs in 1986 to 2 64-Mb DRAMs today–-because the growth rate of the minimum memory size is half the growth rate of DRAM . A challenge for wide DRAMs is that some customers want parity protection and some do not. Finally, today's cache-oriented microprocessors need lower latency but instead are offered higher bandwidth with higher latency. Hence customers may no longer automatically switch to the larger capacity DRAM because the minimum memory capacity may be too large; the larger capacity DRAM will need to be in a wider configuration that is more expensive per bit than the narrow version of a smaller DRAM; the wider capacity doesn't match the width needed for error checking; or memory latency is higher. Thus the 256 Mb or 1 Gb DRAM may be greeted with indifference.

Hence its time to reconsider unifying logic and memory. Since most of the transistors on this merged chip will be devoted to memory, we call it "Intelligent RAM". IRAM is attractive because the gigabit DRAM chip has enough transistors for both a powerful processor and a memory big enough to contain whole programs and data sets; it contains 1024 memory blocks each 1 Kb wide[2]; it needs more metal layers to accelerate the long lines of 600 mm2 chips[2]; and it may require faster transistors for the high speed interface of synchronous DRAM. Potential advantages of IRAM include lower memory latency(≈0.1X), higher memory bandwidth(≈100X), lower system power, adjustable memory width and size, and less board space. Challenges for IRAM include high chip yield given processors have not been repairable via redundancy; high memory retention rates given processors have usually need higher power than DRAMs; and a fast processor given logic is slower in a DRAM process.

One microprocessor was described in sufficient detail to allow us to estimate performance of an IRAM using a similar organization [1]. Given the breakdown of where time is spent, we estimate the performance of each piece in an IRAM. Table 3 shows the performance factor used to scale the Alpha performance parameters to estimate the speed of an IRAM. Rather than pick a single number for each category, we pick optimistic and pessimistic factors. The latency to IRAM main memory should be 5 to 10 times faster (factor of 0.1 to 0.2) than the 200-300 ns latency of typical computers.

Table 2 shows the optimistic and pessimistic performance for an IRAM organized like an Alpha 21164. The small SPEC92 benchmarks are the poorest performers in IRAM, being 1.2 to 1.8 times slower. The database varies from a little slower to a little faster, while linpack varies from 1.2 to 1.8 times faster. These programs are more representative than SPEC92, which was replaced in part due to limited memory traffic.

An alternative computing style is vector processing which works on linear arrays of numbers. Vector processors do not need caches, but rely instead on low latency memory, often made from SRAM, and high bandwidth using 100s of memory banks. Thus a gigabit IRAM memory system naturally matches the needs of a vector processor.

An IRAM vector microprocessor might look like Figure 1. In a 0.18 micron DRAM process with a 600 mm2 chip, an IRAM could have 16 Add-Multiply units running at 500 MHz and 16, 1024-bit wide memory ports at 50-MHz giving a collective 100 GB/s of memory bandwidth. It could run linpack at 8 GFLOPS, more than five times faster than the fastest Cray vector supercomputer processor (Cray T-90).

The popularity of IRAM is limited by the amount of memory on-chip. If IRAMs succeed, IRAM products should increase as memory size expands: from graphics today (10 Mb) to the game and embedded markets (32 Mb), and to network computers and portable PCs (128-256 Mb). The semiconductor industry may soon see head-to-head competition between its currently segregated logic and memory camps.

Acknowledgments:

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

References:

[1] Cvetanovic, Z; Bhandarkar, D., "Performance Characterization of the Alpha 21164," Proc. High Performance Computer Architecture, San Jose, CA p. 270-80, Feb.,1996.

[2] Yoo, H.J., et al., "A 32-Bank 1 Gb DRAM with 1 GB/s Bandwidth," ISSCC Digest of Technical Papers, pp. 378-379,Feb., 1996.

Year	Microprocessor	On-Chip Cache Size	Memory Gap Penalty: % Die Area (not counting pad ring)	Memory Gap Penalty: % Transistors
1992	1st generation 64b RISC	I: 8 KB, D: 8 KB	21.4%	59.5%
1994	2nd generation 64b RISC	I: 8 KB, D: 8 KB, L2: 96 KB	37.4%	77.4%
1996	Low power, embedded RISC	I: 16 KB, D: 16 KB	60.8%	94.5%
1989	4th generation x86	8 KB	19.9%	50%
1993	5th generation x86	I: 8 KB, D: 8 KB	31.9%	32%
1995	6th generation x86 (2 chips, processor and L2 cache)	I: 8 KB, D: 8 KB, L2: 512 KB	P: 22.5% +L2: 100% (Total: 64.2%)	P: 18.2% +L2: 100% (Total: 87.5%)

Table 1: Memory Gap Penalty for conventional microprocessors (I = instruction, D = data, L2 = level 2).

Category	SPECint92		SPECfp92		Database		Sparse Linpack	
	Opt.	Pes.	Opt.	Pes.	Opt.	Pes.	Opt.	Pes.
Fraction of time in processor	1.02	1.57	0.89	1.36	0.30	0.46	0.35	0.54
Fraction of time in I cache misses	0.04	0.05	0.01	0.01	0.18	0.21	0.00	0.00
Fraction of time in D cache misses	0.14	0.17	0.26	0.30	0.15	0.18	0.08	0.10
Fraction of time in L2 cache misses	0.05	0.05	0.06	0.06	0.20	0.20	0.07	0.07
Fraction of time in L3 cache misses	0.00	0.00	0.00	0.00	0.03	0.05	0.06	0.12
Total = ratio of time vs. µprocessor [1] (>1 means IRAM slower)	1.25	1.83	1.21	1.74	0.85	1.10	0.56	0.82

Table 2: Estimated performance of conventional IRAM for four programs (int = integer, fp = floating point).

Component of micro-processor execution time	Optimistic Scale Factor	Pessimistic Scale Factor
Logic	1.3	2.0
SRAM	1.1	1.3
DRAM	0.1	0.2

Table 3: Scale factors for estimating IRAM performance (larger is slower).

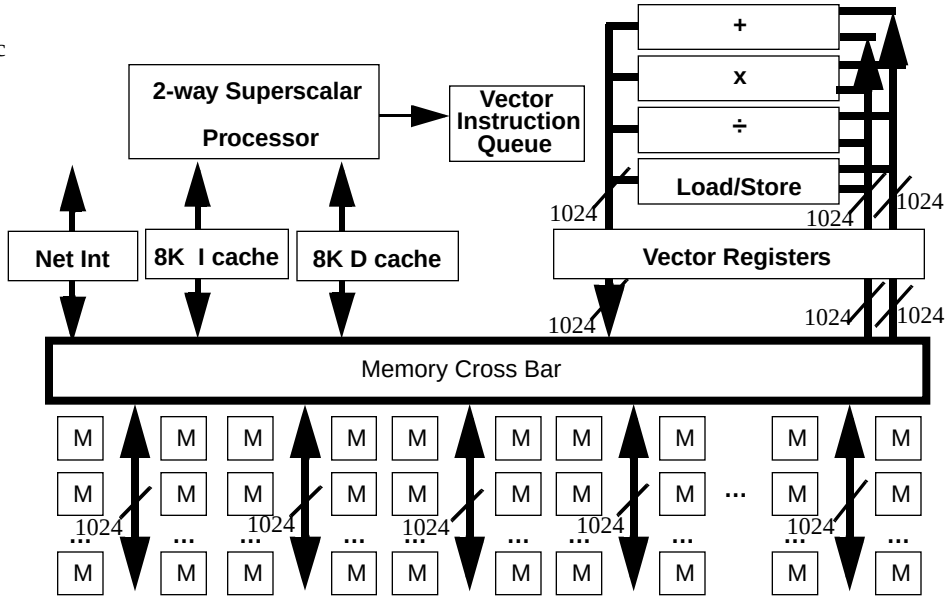

Figure 1. Organization of a vector IRAM in a 0.18 micron DRAM process.