Martin

C-RAM: Memory with a Fast SIMD Processor

Duncan G. Elliott and W. Martin Snelgrove

May 1, 1990

Abstract

Computational RAM (C-RAM) is memory with SIMD processors added to the sense amplifiers. These processors add a small amount of area to the chip and have an aggregate performance in the billions of operations per second in a workstation setting.

1 Introduction

The architecture which we call Computational RAM (C-RAM is a hybrid of RAM and all SIMD processor. A single bit processor element (PE) is placed next to each sense amplifier within the RAM chips. The processor elements are cheap to integrate into the memory, exploit, the tremendous memory bandwidth internal to the chip and, if used as computer main memory can communicate with a conventional processor via "shared memory".

Single instruction-stream, multiple data-stream (SIMD) computers have been around for 2 decades. Some of the better known commercial SIMD architectures which could be described as massively parallel are ICL's Distributed Array Processor (DAP) [1], Goodyear's MPP[4], NCR's Geometric Array Processor (GAPP)[3], Thinking machines' Connection Machine series [2], and more recently MasPar's MP-1[5]. All amortize the cost of instruction fetch and decode hardware over a large number of processing elements.

There are several reasons why massively parallel SIMD machines have not become commonplace; the first is economic. Most SIMD machines have been designed for low quantity production and are consequently quite expensive. Second, memory is not typically integrated into the same chips as the processors so the maximum number of processors per chip is limited by the number of pins in an IC package. Third, the host-to-SIMD interface often poses a bottleneck when results

have to be shipped between the sequential and parallel processors. Fourth, SIMD machines are more difficult to program than conventional machines.

2 Motivation

C-RAM was designed with the goal of avoiding the common pitfalls of SIMD We address the issues of cost, integration, memory bandwidth, and host bandwidth. C-RAM is, unfortunately, no easier to program.

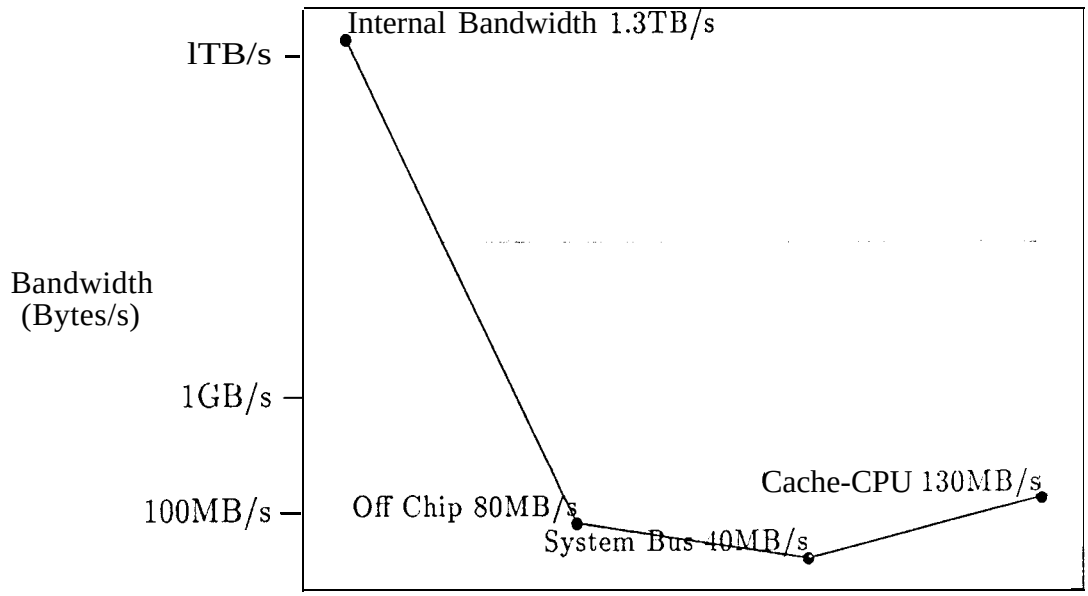

Figure 1: Memory Bandwidth Throughout System

A tremendous amount of aggregate memory bandwidth is available within memory chips at the sense amp interface. For example, take a workstation with 32 Mbytes of 4Mbit DRAM, a 100ns memory cycle, 32 bit busses, and a 33 MHz CPU. At the sense amps, 1.3 terabytes/s is available but the pins on the memory chips only permit SO megabytes/s. The bandwidth is further reduced at the system bus. Finally, a cache is needed to bring the effective memory bandwidth up to the processor's requirements, but this still leaves us with only $\frac{1}{10000}$ of the bandwidth we started out with.

This internal memory bandwidth is difficult to use because the datapath is extremely wide. In C-RAM we associate one processor element with each sense amp. This integration of processor

and memory permits higher density designs. The number of PEs which can be placed on a chip with most other SIMD architectures is limited by the number of pins which can be used to connect to external memory.

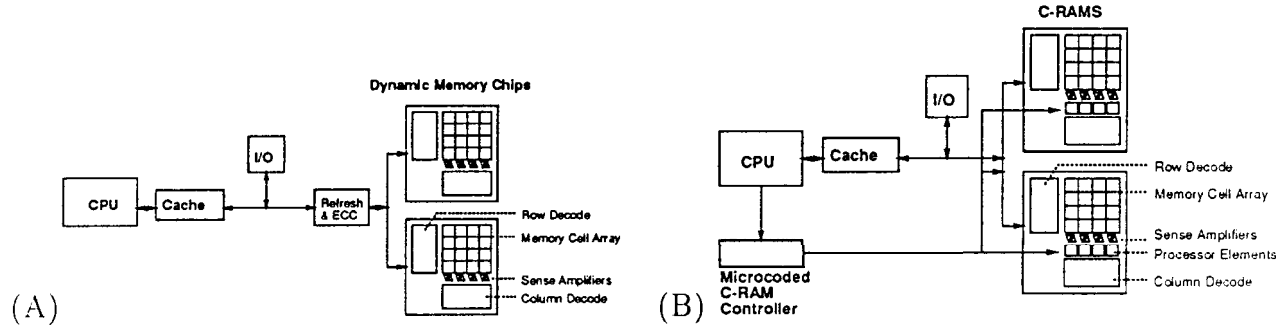

Figure 2: (A)Conventional and (B)C-RAM Computer Architectures

When C-RAM is used as main memory for a conventional processor (figure 2, the communication between SIMD and host is simply shared memory. Data and results do not have to be copied between the two processors. C-RAM can be read and written at speeds similar to conventional memories Further, the host can steal memory cycles from C-RAM (eg. for filling a cache line) without altering the state of the PEs.

3 Design

A small proof-of-concept S Kbit C-RAM has been designed in 1.2μm BNR-CMOS4S and is being fabricated by Northern Telecom through the Canadian Microelectronics Corporation.

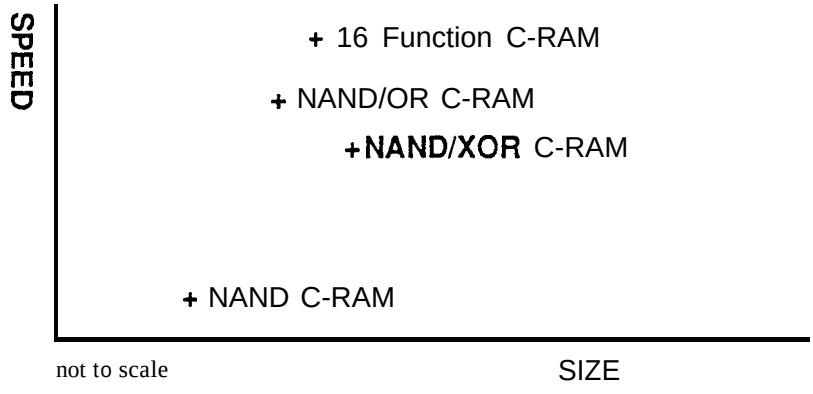

Figure 3: Arithmetic Performance vs. Size

The PEs have single bit wide data paths. Arithmetic is performed in a bit serial fashion

Several PE designs were considered. Their names are derived from the functions which the ALU performs. These are compared in figure 3 by an estimate of their area and their performance for integer addition. All operate on the contents of memory and a single bit register. A minimal PE requires a function which is logically complete such as NAND or NOR. The “NAND C-RAM" is small and complete but slow. The exclusive-or operation consumes most of the cycles. Adding an XOR to make “NAND/XOR C-RAM" dramatically speeds up arithmetic. “NAND/OR C-RAM" adds an extra function while reducing the number of transistors. When both NAND and OR operations are selected, the wire-ANDed result is an XOR of the operancls. Finally, the PE built was “16 Function C-RAM" which implements an arbitrary function of I inputs using a multiplexor as shown in figure 4 (2^{2^2} functions are available).

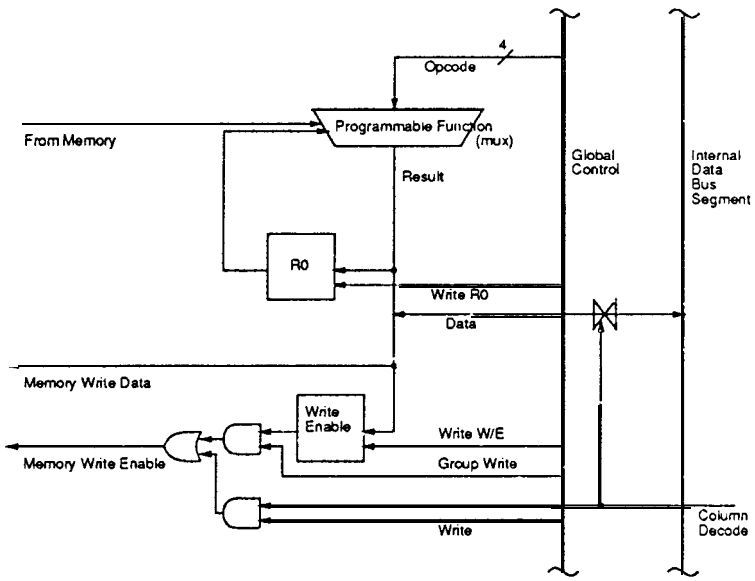

Figure 4: Processor Element Implemented

A checkplot of the top level metal of C-RAM is shown in figure 5. The organization is **1% rows** by 64 columns. A 6 transistor static RAM cell is used. Each sense amplifier is connected to a processor element along the bottom of the memory array. The PEs only take 5% of the total area. The memory cells could be expected to be smaller in an IC process designed for memory. and the PEs would take up a greater portion of the chip area.

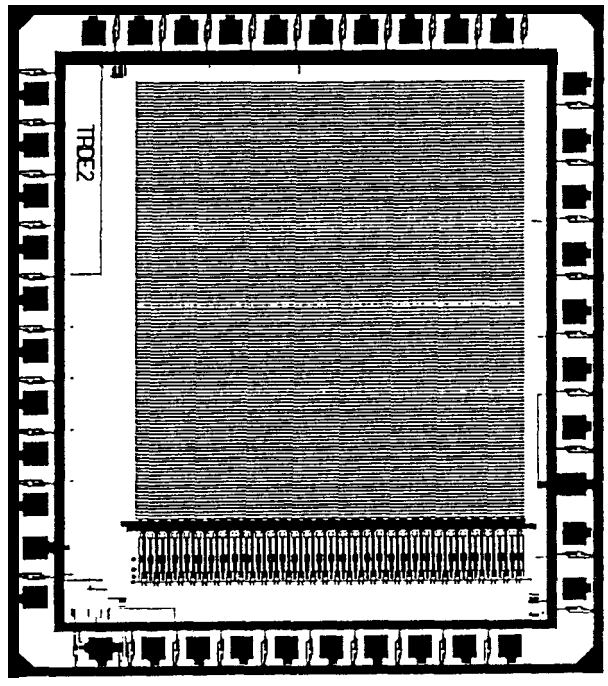

Figure 5: C•RAM Layout

4 Results

From simulations, a workstation containing 32 megabytes of the C-RAM chips as dcsigncd could perform l50 billion 32-bit integer operations per second (2 million processors / 11us per 32 bit add). WE estimate that a less expensive DRAM version would have one quarter of the performance of our SRAM version. Problems which could utilize even a few percent of the power of C-RAM would achieve performance similar to that of supercomputers.

We are looking at applications in a number of areas. Traditional numerical problems such as weather prediction, computational fluid dynamics and other finite element analysis problems and good candidates since these have uniform cells which can each be operated on by a PE. Similarily graphics rendering and image processing can be parallelized by assigning one or more pixels to each PE.

Figure 6 gives the timing for a read-operate-write cycle (taken from HSPICE). First the bitlines and result busses are precharged. The row address is applied? specifying which bit in each PE's "local" memory is addressed; and the appropriate wordline is selected. The opcode is specified (opcode bit 0 shown) and the result bus becomes valid. Finally, the result is written back to registers and memory.

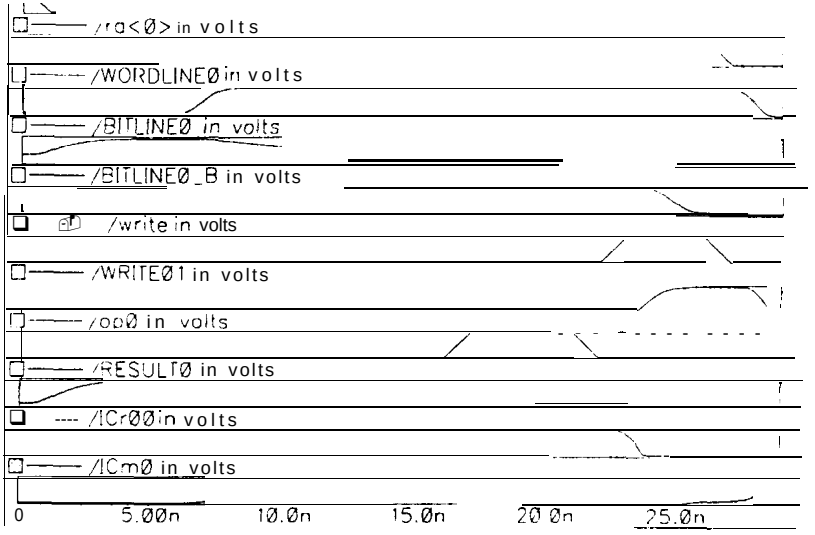

Figure 6: C•RAM Timing

5 Conclusions

Processors can be placed in the main memory of a conventional computer yielding a massively parallel SIMD computer for a small additional cost,. A first experimental design of C-RAM using static memory has been sent for fabrication. We 'hope to further demonstrate the cost effectiveness of this approach to computing by designing a DRAM version in the near future.

References

[1] *Terry Fountain. Processor Arrays Architecture and Applications.* Academic Press. 1987

[2] W. Daniel Hillis *The Connection Machine.* The MIT Press; 1985.

[3] Geometric Arithmetic Parallel Processor. Technical Report NCR45CG72. NCR Corporation Dayton, Ohio, 1984.

[4] David H. Schaefer. History of the MPP. In J. L. Potter, editor, *The Massively Parallel Processor,* pages 1-5. The MIT Press, 1985.

[5] Tom Williams. Massively Parallel Processor Array Spits out 30,000 MIPS. *Computer Design* pages 45-46, October 1989