Appeared in Proc. IEEE 1992 Custom Integrated Circuits Conference, May 1992.

# Computational Ram: A Memory-SIMD Hybrid and its Application to DSP

Duncan G. Elliott, W. Martin Snelgrove and Michael Stumm

**Department of Electrical Engineering, University of Toronto**
**Toronto, Canada M5S lA4 (416)978-3381**

**Computational RAM (C-RAM) is conventional RAM with SIMD processors added to the sense amplifiers . These bit-serial, externally programmed processors add only a small amount of area to the chip and in a 32Mbyte memory have an aggregate performance of 13 billion 32 bit operations per second. The chips are extendible and completely software programmable. In this paper we describe (1) the C-RAM architecture, (2) a working 8Kbit prototype, (3) a full scale C-RAM designed in a 4Mbit DRAM process, and (4) C-RAM applications.**

## Introduction

Massively parallel SIMD computers have been used for computationally intensive tasks for a some time. These SIMD machines include the AMT DAP [l], Goodyear's MPP [2], Thinking Machine's Connection Machine [3], and MasPar's MP-1 [4]. These machines (presumably for ease of design and manufacture) partition the processing elements (PEs) and memory into separate chips. With this division, the number of PEs which can be integrated on a chip is predominantly limited by the number of pins which can be used to connect to off-chip memory. GAPP [5], Pixel Planes [6], VIP [7], and SVP [8] [9 address the problem by placing all local memory on chip with thePEs.

Video frame buffers and computer main memories are primarily DRAM because of its high density and low cost. During each memory cycle, the contents of an entire row are accessed and sensed, then data may be read or written, and finally all data is written back (the refresh, latched sensing assumed). A tremendous amount of aggregate memory bandwidth is available within the memory chips at the sense amps. Consider a workstation with 32 Mbytes of 1Mx lbit DRAM, a 120ns memory cycle, 32 bit busses, and a 33 MHz CPU (with an optimistic 100% cache hit rate). At the sense amps, 1.1 terabytes/s are available (figure l), but the pins on the memory chips only permit a bandwidth of 270 megabytes/s. The bandwidth is further reduced at the system bus. Finally, a cache is needed to increase the effective memory bandwidth for the processor; but even with this improvement there are 4 orders of magnitude less bandwidth than at the starting point - the sense amps.

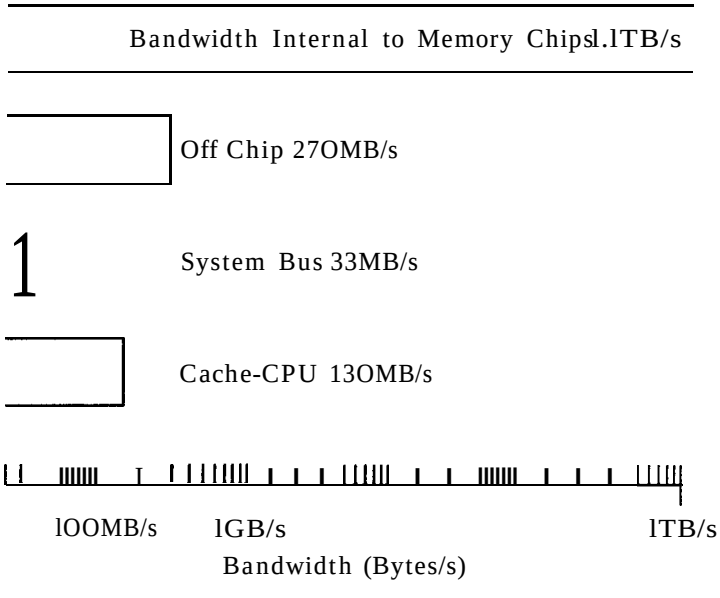


Figure 1: Memory Bandwidth Throughout System

The goal of our work has been to exploit that internal memory bandwidth by building in massively parallel computing without hindering the memory function or contributing significantly to the chip area. Tradeoffs between the requirements of the memory and the requirements of the PEs are almost always decided in favour of the memory since it occupies the vast majority of the area.

A commercial version of C-RAM could replace commodity memory in a variety of applications, thus adding array processing capability to the hardware. When C•RAM is used as main memory or frame buffer memory for a conventional Von Neumann 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.

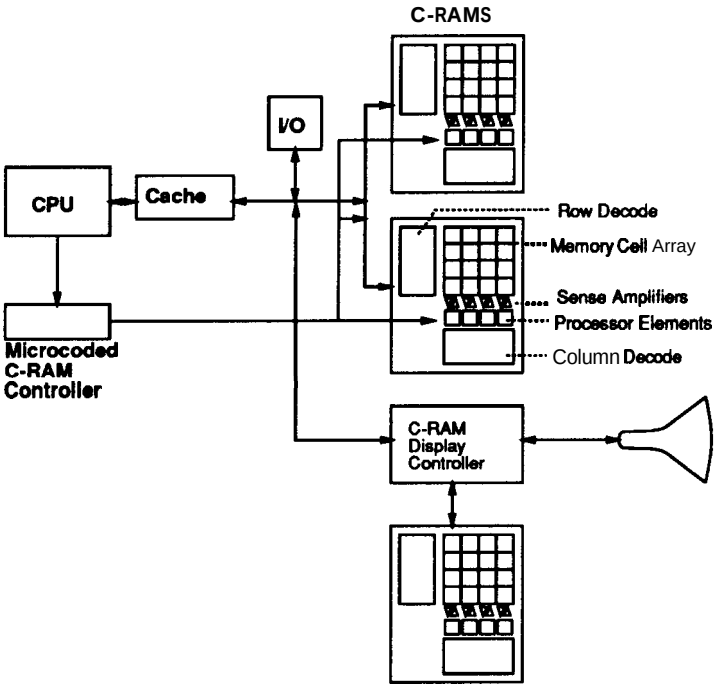


Figure 2: C-RAM Computer Architecture

C-RAM can also be used without the host processor as an embedded video or signal processor.

## Architecture

The development of the PE architecture of C-RAM was guided most strongly by the requirement for small area and a narrow pitch. To that end a bit-serial approach was taken. If enough parallelism exists in an application, it makes little difference whether n 32-bit processors or 32n l-bit processors tend to it. Furthermore, the bit-serial processors use less area per bit and can efficiently handle variable precision calculations.

The ALU was selected for speed from a range of designs which have a narrow implementation. The chosen ALU (figure 3.), is an arbitrary function of three inputs (memory and two registers: X, Y) implemented as a multiplexor. The global instruction is an 8-bit 'truth table' fed from off chip (multiplexed through the address pins).

Grid communication among processors would be a desirable luxury, but 2048 PEs arranged in a minimum perimeter (32 x 64) grid would require 192 connections off chip defeating the cost goal. Instead, a one dimensional shift register is sufficient for typical tasks as shown later. The shift register between adjacent PEs is implemented by putting a second input on both data registers. Using one register for each direction allows their speeds to be matched and permits simultaneous left and right

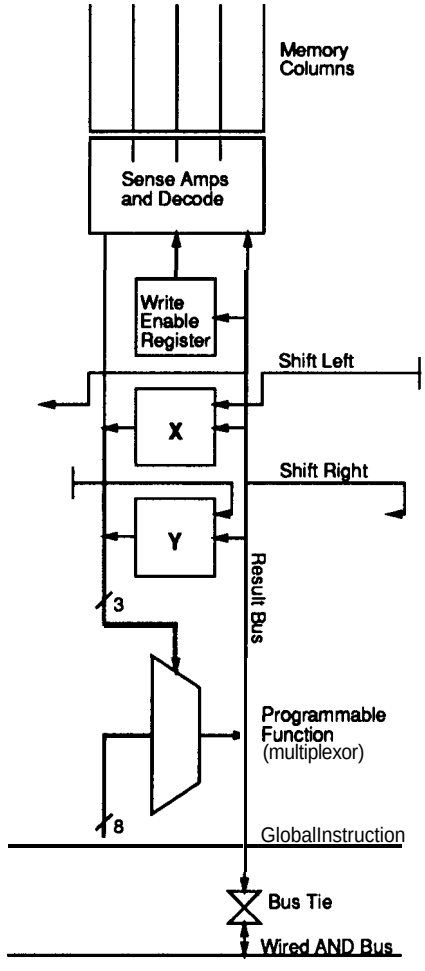


Figure 3: Processor Element Implemented

shift for such operations as dilation and erosion.

Conditional execution of code is performed by putting the result of an expression in the ***WriteEnableRegister,*** effectively halting those PEs which don't satisfy the condition. The bus tie, when enabled, yields the Wired-AND of all PEs' result busses, as well as the C-RAM controller's output. This bus can be used for broadcasting data, interprocessor communication, and alerting the controller to such events as arithmetic exceptions or solution reached (eg. convergence).

## Prototype

A small proof-of-concept 8 Kbit C-RAM (without the shift register) has been designed and fabricated in $1.2\mu m$ CMOS. With the exception of an error in one of the 64 column decoders, the chips are functional. The read-operate-write cycle time is 114ns. A faster cycle was expected but a charge-sharing problem was discovered. A die photo of the 64 PE by 128 bit CRAM is shown in figure 4. Each processor element along the bottom of

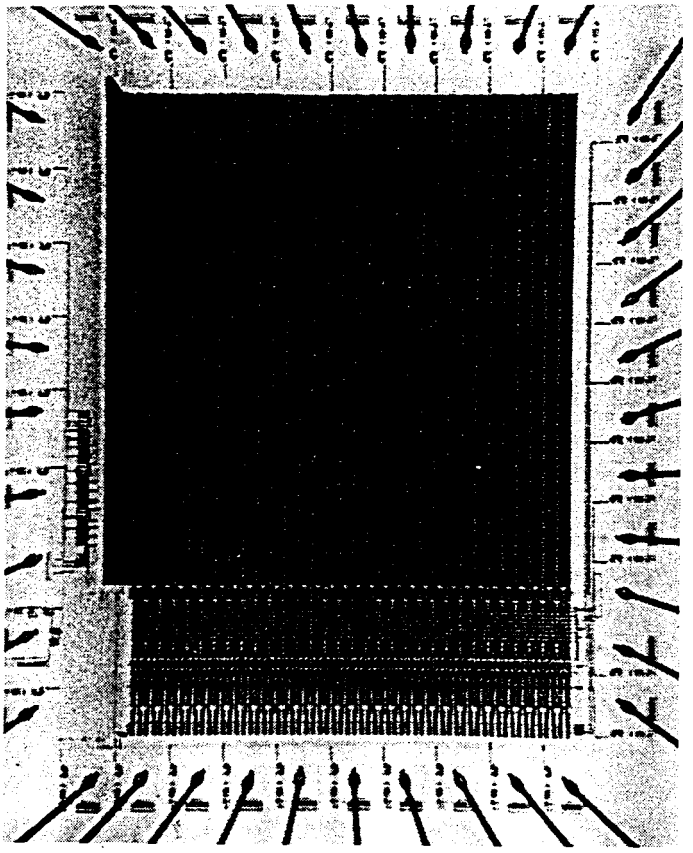

Figure 4: 64 Processor C-RAM Die Photo

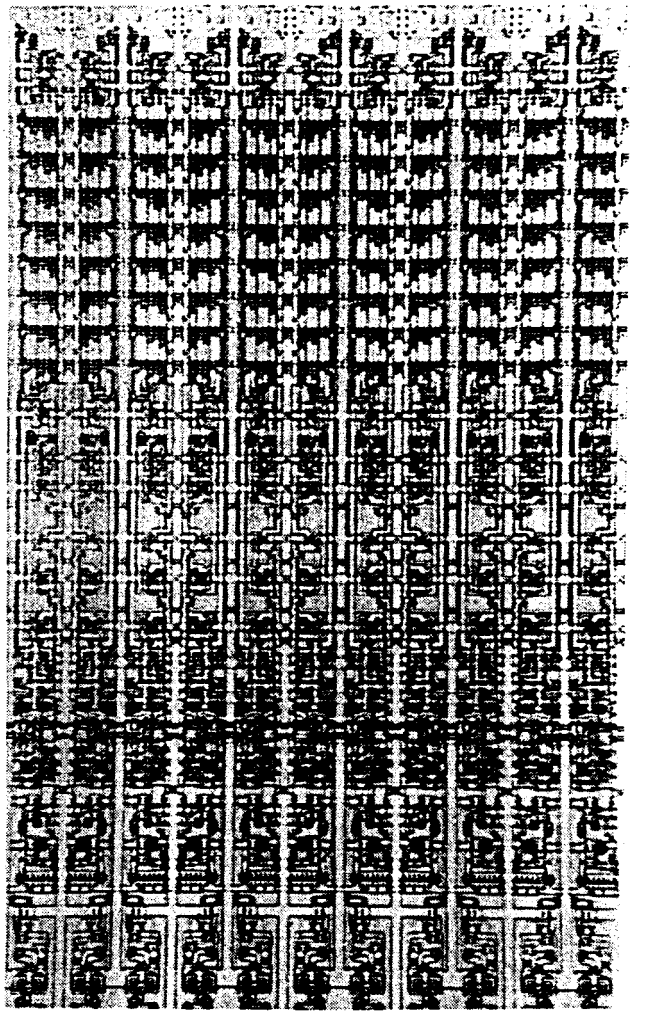

Figure 5: Detail of Processor Elements (11 shown)

the memory array fits in the pitch of and is connected to a sense amplifier. Figure 5 shows 11 PEs in detail. The PEs take only 9% of the total area in this ASIC process (the array of 6-transistor SRAM cells dominates the chip aria). In a commodity memory process, the memory cells are smaller, leaving the PEs taking a greater portion of chip area.

The ALU was implemented as a dynamic-logic multiplexor with output to the pre-charged *ResultBus*. The registers are 5-transistor static CMOS latches. The PE has a three phase cycle: precharge, ALU, and write (registers and memory).

An earlier prototype used a pass transistor as a bus-tie, but the ALU lacked drive and bus operations were slow. The present bus-tie design, upon receiving a 0 on either side, actively drives the other side low. The bus tie is disabied during the combined *ResultBus* and *WiredANDBus* precharge.

The row and column address. and instruction signals are not multiplexed onto the same pins in this proof of concept design the way they are on the DRAM version so the pin count is high (40 pins). Quaternary-predecoded column addresses are supplied from off chip (doubling the number of column address pins) so that subsets of the PEs can be selected. Since this feature was seldom used (especially by high-level-language programs) and can be emulated in software, the feature has been dropped. A previous design had a write-multiple-rows feature on the memory as well. but again, the feature did not seem cost-effective.

## High Density Design

The design of a C-RAM based on commodity submicron 4Mbit DRAM technology and a proven memory design is underway. DRAM memory cells are smaller and consequently, 4 sense amplifiers fit in the pitch of one PE (less than $20\mu m$). The cycle time for read, two ALU operations and write-back is 15Ons. The speed of DRAM operations are not significantly impacted. With fewer PEs per given amount of memory the DRAM version is lower performance than the SRAM prototype, but its density makes it a more viable chip.

dard DRAM during a ***RAS-CAS*** cycle, but takes in a row address and an instruction during a ***RAS-OPS*** (operate strobe) cycle. Multiple ***OPS*** cycles can occur while ***RAS*** is held low. The chip is organized as lMx4bit. To reduce power consumption, each PE is connected to two memory arrays of which only one is powered (precharge, row decode, sense) during a ***RAS*** cycle. The 16 arrays of 256Kbits are organized as 1024 sense amps x ***256*** word lines.

With this design, 32Mbytes of C-RAM (8 SIMMs or approximately the footprint of a microprocessor and cache chips) could perform 13 billion 32 bit additions per second (2 15Ons cycles per l-bit full add, 128K PEs) or 8 billion signed multiply-accumulates per second (8 bit multiplication by a constant, 16 bit accumulate). Even if only a fraction of this performance can be used it will shift the preferred location of computing from a central processor to the memory.

## Applications

Taking the DCT (Discrete Cosine Transform) of a monochrome image buffered in CoRAM is a more realistic application than the vector addition above. We assign an 8 x 8 block of pixels to each PE. A 1M pixel image requires 8 C-RAMs UsingChan's algorithm[lO] and 15 bit resolution arithmetic, the transform takes 5.7ms. This operation occupies the PEs for about a third of the time of a 6OHz non-interlaced frame; leaving time to perform other operations such as (de)compression. Image size has no effect on execution time; larger images use more memory and therefore are processed by more PEs. Given the 144 multiplications and 464 additions, 1.7 billion operations are performed per second. Obviously, not all problems have this degree of parallelism.

A 2-dimensional convolution on a 1024x 1024 pixel image with a 3x3 filter is a typical problem requiring grid communication. Four C-RAMs using 8-bit multiplication / 16-bit accumulate take 8.6ms of which 11% is spent communicating. C-RAM offers performance on these applications comparable to algorithm-specific hardware, but with the flexibility of software.

## Conclusions

CoRAM can be used to build a high performance smart frame buffer which could be potentially useful for video standards conversion, image compression, or HDTV processing. A C++ compiler which supports "parallel variables" of arbitrary sizes is being developed. With memory being the traditional leader in IC technology, CoRAM is positioned to increase in speed with increases in memory density -the more memory the more power. A teraFLOPS machine can be had for 64Gbytes of CoRAM ' while a (boolean) tera operation/s requires a mere 37Mbytes.

## Acknowledgments

The authors are grateful for the technical exchange and support from MOSAID Technologies; the fabrication provided by Northern Telecom through the Canadian Microelectronics Corporation; user feedback from Oswin Hall; and support from the Natural Sciences and Engineering Research Council of Canada and MICRONET.

Address email to dunc@eecg. toronto . edu

## References

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

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

[3] W. Daniel Hillis. ***The Connection Machine.*** The MIT Press, 1985.

[4] MasPar. MP-1 Family Data-Parallel Computers. Technical Report PLOO6.0190, Sunnyvale, CA, 1990.

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

[6] John Poulton, Henry Fuchs, John Austin, John Eyles, and Trey Greer. Building a 512x512 Pixel-Planes System. ***Advanced Research in VLSI - Proceedings of the 1987 Stanford Conference,*** pages ***57-71,*** 1987.

[7] K. Chen and C. Svenson. A 512-Processor Array Chip for Video/Image Processing. ***Proc. of From Pixels to Features II,*** pages 349-361, August 1990.

[8] Hiroshi Miyaguchi, Hujime Krasawa, and Xhinichi Watanabe. Digital TV with Serial Video Processor. ***IEEE Transactions on Consumer Electronics, 36(3):318-326,*** August 1990.

[9] Jim Childers, Peter Reinecke, and Hiroshi Miyaguchi. SVP: A Serial Video Processor. ***IEEE 1990 Custom Integrated Circuits Conference,*** pages 17.3.1-17.3.4, May 1990.

[10] S. C. Chan and K. L. Ho. A New Two-Dimensional Fast Cosine Transform. ***IEEE Transactions on Signal Processing,*** 39(2):481-485, February 1991.