A PetaOp/s is Currently Feasible by Computing in RAM

Duncan Elliott, Martin Snelgrove, Christian Cojocaru* and Michael Stumm*
Department of Electrical and Computer Engineering, University of Toronto, Canada M5S 1A4
*Department of Electronics, Carleton University, Ottawa, Canada K1S 5B6
dunc@eecg.toronto.edu

Abstract

Technology considerations dictate that a petaOPS computer implemented with currently available technology would do most of its computing with simple processors integrated into memory and thereby exploit the high internal memory bandwidth. Such a system is proposed.

Technology and Memory Assumed

We focus on what would be required, *given technology now available*, to make a petaOPS computer system. In particular we address memory bandwidth, packaging, and power consumption.

A petaOPS computer system would necessarily be large, and would have to be implemented with the most reliable and highly integrated technology available. Now, and for the next few years, that is silicon CMOS memory technology running at room temperature. Of memory technologies, DRAM has higher density, lower cost per bit and moderate speed with cycle times under 100ns. SRAM, on the other hand, is more expensive, but is available with cycle times under 10ns. Ultimately, the memory size/speed requirements of the set of applications to be run will favor SRAM or DRAM.

We assume the presence of significant amounts of memory and memory traffic in a petaOPS architecture. We discount specialized architectures like "datapath-only" systolic arrays (being free of RAM) on the assumption that they would not be broadly enough applicable to make the system economic.

The memory bandwidth requirement for a petaOPS machine (floating point or integer) demands that the bottle-neck of the memory chip pins be circumvented. For this example, we use a 32 bit functional-unit word and 16Mb, 10ns-cycle SRAM chips with 16 data pins. If only one memory access is made in every 8 operations, 2.5 million memory chips are required to supply the 0.5 petabytes/s memory bandwidth. We optimistically assume that registers and caches local to the functional unit chip make up the other sources and destinations. Off-chip caches don't solve the bottleneck of bringing data through memory pins. If instead, the functional-units are connected directly to the memory columns and each operation requires 3 memory accesses, as few as 15 thousand chips are required to deliver the necessary 12PB/s; provided that: memory columns are 256 bits long, the IC geometries will permit connection to the columns or sense amplifiers, and that all columns can be made active during one cycle (at the expense of extra power pins). Even when an order of magnitude is lost due to compromises in the above assumptions, the bandwidth gains of putting processing in the memory are still tremendous. With the higher densities of DRAM, the gains are greater. Assuming a 256Mb, 100ns-cycle DRAM with 16 data pins, the afore-mentioned bandwidths require 25 million chips for off-chip access and 9200 chips for integrated processors.

Even the busing between memory and off-chip functional units is overwhelming. Half a petabyte/s transferred at a sustained clock rate of 500MHz still requires 8 million bus wires.

The memory bandwidth requirements of a petaOPS machine make a strong argument for integrating the processing power into the memory. These are approximate lower bounds on the number of memory chips necessary for the required memory bandwidth. We will see that differences in memory geometry and speed, as well as the efficiency of the processor affect the amount of memory needed for a petaOPS.

Power Limitations

For an overall power consumption of 10kW, the energy consumed per operation in a petaOPS system would have to be 10^{-11}J. A power consumption in the tens of kilowatts is a comfortable limit for an air-cooled machine.

CMOS consumes energy CV_{DD}^2 during one cycle of charging and discharging a circuit node between ground and a power supply V_{DD}. The minimum practical value of V_{DD} for room-temperature operation appears to be about 1V [1,2], since going below $V_{DD} = 4V_t$ doesn't improve energy per operation and where MOS threshold voltage V_t has to be greater than about 0.2V to control leakage currents that are determined by the room-temperature value $kT/q \approx 26\text{mV}$. Assuming that an average operation involves reading and changing roughly 100 bits and that circuit nodes are precharged to $V_{DD}/2$, we require

$$10^{-11}\text{J} > 100\text{bits}\, C\,(0.5\text{V})^2 \qquad (1)$$

which limits the capacitance to an average of 400fF.

While adiabatic computing [3] attempts to reduce CV^2f power, it still has switching overheads of the same general form but with a diode voltage (around

Presented in the *PetaFLOPS Frontier Workshop* held at the *IEEE Frontiers Of Massively Parallel Computation Symposium*, McLean, VA, February 1995

$0.7V$) replacing the V_{DD} term. At present, adiabatic computing appears to consume roughly the same power as $1V$ CMOS, and hence it does not appear to offer a solution with present technology.

A $400fF$ capacitance is barely greater than the bit-line pair capacitance of a typical modern DRAM [4] which is charged for every memory access. A cache doesn't help power consumption. Even though a high hit rate reduces the number of accesses to main memory, the cache RAM charges similar capacitances on each access.

A conventional memory architecture also wastes almost all bit reads because only a small fraction of the bits read by the sense amplifiers on a given cycle are actually used. This is obviously unacceptable, since we already have a tight power-budget when assuming that bits are used with perfect efficiency. Driving signals off-chip also comes with the expense of charging many pF per wire.

We therefore claim that bits should usually be processed on-chip with the memory, and in fact very close to the sense amplifiers of the memory chip. It follows that the processors used must be compact and simple, or their sheer size will consume energy in routing signals.

Radical changes to memory architecture, such as breaking up memory arrays and introducing extra row decoders to perform independent addressing will reduce memory density and increase the power cost of communications. The long internal memory words are too long for a uniprocessor; and difficult for a MIMD multiprocessor to utilize, unless the application can benefit from processors autonomously executing their own instruction streams but performing loads and stores to the same address in local memory at the same time. The shared memory address stream suggests the use of a shared instruction stream as well. For these reasons, we use SIMD processing elements (PEs) in the memory.

Attempting to speed up the cycle time is also wrong, because the best energy/operation is obtained at low V_{DD} and hence relatively slow switching. Faster cycle times would also make power supply transients worse, and we already propose to activate many more sense amplifiers at once than is typically done.

Computing in RAM

We have shown the feasibility of placing one-bit SIMD PEs in memory adjacent to the sense amplifiers in both SRAM and DRAM. Our first proof-of-concept design of "Computational RAM" (C•RAM) [5] was fabricated in a 1.2μm CMOS process and had 64 PEs with 128 bits of SRAM per PE. This minimalistic PE, shown in figure 1, requires 77 transistors and fits in the width of a memory column. Together the PEs occupy 9% of the chip area. The second generation of C•RAM [6] is being fabricated in a 0.8μm BiCMOS process and has 512 PEs by 480 bits per PE, as well as some new features. A DRAM version of C•RAM has been designed (but not fabricated) with a PE for every 2Kb of memory.

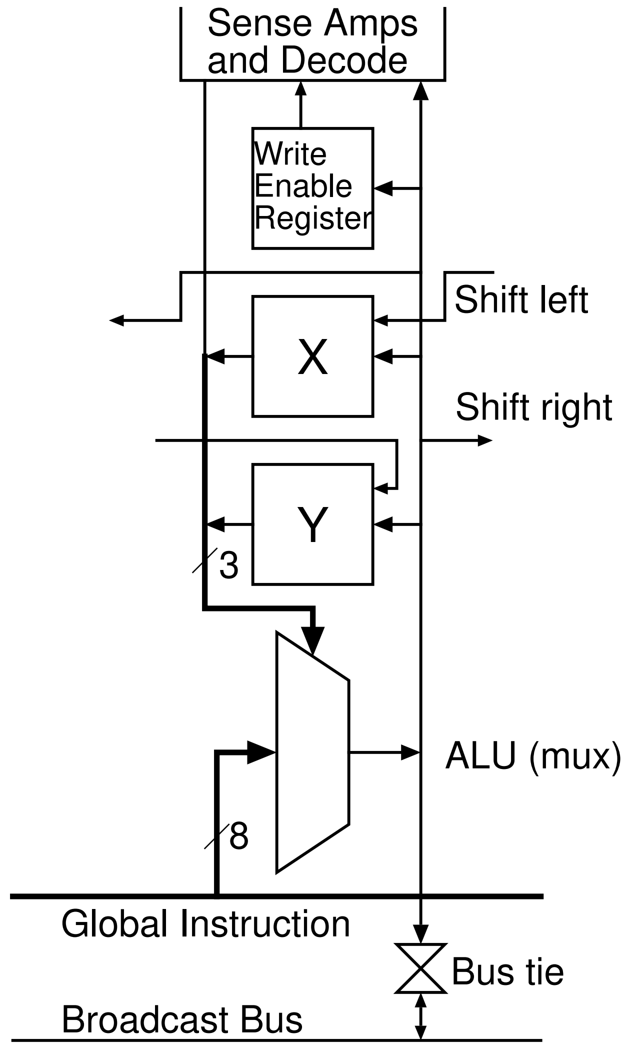

Fig. 1: C•RAM Processing Element

Other architectures include a 3.8GIPS chip from NEC [7] with 64 eight-bit processors attached to 2Mb of SRAM; an SRAM based 64-processor chip with 1-bit PEs from the Supercomputer Research Center [8]; and the DRAM based "Serial Video Processor" with 1024-PEs TI [9] using one-bit processors. These chips demonstrate that computing can be done very close to the memory, and that this can be done with light-weight SIMD PEs, simple buses, and one-dimensional nearest-neighbour communication to keep size (and hence power) under control.

These simple one-bit processors can deliver between $0.1MIPS$ and $1MIPS$ each for 32-bit integer operations. A terabyte of SRAM based C•RAM could deliver a petaOPS (more memory than our lower bound). Assuming one PE per 512 bits with a 20ns read/ALU/write cycle, 32 bit additions can be performed at a rate of 6.7 petaOPS. An 8-bit multiply with 16-bit accumulate achieves 2.6 petaOPS. Yet, single precision floating point multiplies are only performed at a rate of 0.4 petaFLOPS.

In the above examples of performance, C•RAM falls short of producing 1 bit of result for every memory write operation. Since C•RAM is register-poor, intermediate results are stored in memory. In the case of multiplication, partial sums are ferried to and from memory. If multiplies are really needed at the full rate, our processor would have to be redesigned with more registers (to avoid rereading a multiplicand and partial product for each bit of the multiplier from the relatively long and power-hungry bit-lines). Normalization support for floating-point adds would also be expensive. We have developed a variable-width processor [6] that contributes improved power and performance on multiply for roughly a factor of two increase in processor area.

Overall Architecture

Even though the majority of the computing power will come from the SIMD PEs in memory, we do not propose a pure SIMD machine but rather a MIMD-SIMD hybrid. Combining C•RAM, a high-performance microprocessor like the DEC Alpha [10] for every 256MB of memory, and a scalable interconnect [11], would make a multicomputer MIMD machine with each of its processors in turn having a massively parallel SIMD machine as its memory. With a terabyte of total system memory, each of 4096 microprocessors would have 128 SRAM (or 8 DRAM) based C•RAM chips. The MIMD-SIMD hybrid combines the flexibility of MIMD with the economics (especially the power economics) of SIMD.

Conclusions

A petaOPS system is obviously an extremely aggressive target, but a C•RAM design that focuses on power consumption and bandwidth makes it plausible. While the technologies we propose are far from “proven”, they are within the bounds of the imaginable with present fabrication processes and system engineering.

Acknowledgments

The authors are grateful for assistance from MOSAID Technologies, Northern Telecom Electronics, Canadian Microelectronics Corporation; and support from the Natural Sciences and Engineering Research Council of Canada and MICRONET

References

1. Dake Liu and Christer Svensson. Trading Speed for Low Power by Choice of Supply and Threshold Voltages. *IEEE Journal of Solid-State Circuits*, v8(1):10--17, January 1993.
2. Anantha P. Chandrakasan, Randy Allmon, Anthony Statakos, and Robert W. Brodersen. Design of Portable Systems. In *Custom Integrated Circuits Conference*, pages 12.1.1--12.1.8, May 1994.
3. Alex G. Dickinson and John S. Denker. Adiabatic Dynamic Logic. In *Custom Integrated Circuits Conference*, pages 12.6.1--12.6.4, May 1994.
4. Betty Prince, Semiconductor Memories: *A Handbook of Design, Manufacturing and Application* 2nd ed. Wiley, 1991.
5. Duncan G. Elliott, W. Martin Snelgrove, and Michael Stumm. Computational RAM: A Memory-SIMD Hybrid and its Application to DSP. In *Custom Integrated Circuits Conference*, pages 30.6.1--30.6.4, Boston, MA, May 1992.
6. Christian Cojocaru. Computational RAM: Implementation and Bit-Parallel Architecture. Master’s thesis, Carleton University, January 1995.
7. Nobuyuki Yamashita, Tohru Kimura, Yoshihiro Fujita, Yoshiharu Aimoto, Takashi Manaba, Shin’ichiro Okazaki, Kazuyuki Nakamura, and Masakazu Yamashina. A 3.84GIPS Integrated Memory Array Processor LSI with 64 Processing Elements and 2Mb SRAM. In *International Solid-State Circuits Conference*, pages 260--261, San Francisco, February 1994.
8. Maya Gokhale, Bill Holmes, Ken Iobst, Alan Murray, and Tom Turnbull. *A Massively Parallel Processor-in-Memory Array and its Programming Environment*. Technical Report SRC-TR-92-076, Supercomputer Research Center - Institute for Defense Analyses, 17100 Science Drive, Bowie, Maryland, November 1992.
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. Daniel Booberpuhl, Richard Witek, Randy Allmon, Robert Anglin, and Sharon Britton. A 200MHz 64b Dual-Issue CMOS Microprocessor. In *International Solid-State Circuits Conference*, pages 106--107, San Francisco, February 1992.
11. Zvonko G. Vranesic, Michael Stumm, David M. Lewis, and Ron White. Hector: A Hierarchically Structured Shared-Memory Multiprocessor. *Computer*, 24(1):72--79, January 1991.