A Simple, Scalable Processor-in-Memory Microprocessor for HPC Systems

Michael P. Phillips

Artificial Intelligence Laboratory
Massachusetts Institue Of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: There is a need for computer systems which can provide large amounts of computing power to solve problems with enormous requirements in terms of memory, computation, and communication resources. Examples of such problems are protein folding, graphics rendering, physical simulations, and factoring. Traditional microprocessor designs suffer from inefficient use of silicon area, power consumption, scalability problems, and the legacy of how the designs have evolved over time.

Motivation: Advances in semiconductor fabrication technologies, such as modern transistor densities and new logic processes like processor-in-memory (PIM), might best be incorporated into new processor designs by starting from a completely clean slate. The goal is to develop a system scalable to at least one million processors with sufficiently high-bandwidth network interconnect and processing power to enable the system to set new standards for high-performance computing (HPC) systems.

Although there are many elements to this complex system, the design of the processor itself is certainly of interest. The need to replicate the processor one million times requires that the design be area- and power- efficient. Fast processing in a simple processor core will rely on being able to have fast access to memory (hence, PIM seems ideal) and on being able to support multiple simultaneous threads in order to hide latency when communicating with remote processors or accessing remote data.

Previous Work: Several promising investigations of the utility of PIM systems[1, 2, 3, 4] are part of the inspiration for this approach to processor design. There are several supercomputer implementations and designs based on networking a large number of processors, from cluster approaches like Beowulf, to more tightly integrated solutions from SGI and Sun. These solutions do not scale well to very large numbers of nodes. IBM's on-going Blue Gene project is an example of a project with motivations similar to ours. While not complete, the design appears to suffer from very low network bandwidth and may fall short of providing a sufficient amount of processing power to meet its goal of being a good platform for modeling protein-folding. It is difficult to evaluate at this stage, but it appears that PIM technology will not play a role.

Approach: We need a simple processor that is efficient in terms of area used for actual computation and power consumption. Traditional microprocessors spend an enormous amount of area on caches and control, with a comparatively small amount of area being spent on actual computation. This is consistent with being optimized for a single-thread of execution. Our processor can give up much of this complexity and focus on being able to execute many threads, switching among them to hide latency when data is needed but not yet available. By not including circuitry to perform complex operations such as out-of-order issue, speculative execution, and branch prediction, we believe we will save a significant amount of area.

With a small enough processor core, we can actually combine multiple cores on the same silicon die. This is advantageous for a couple of reasons. First, we observe that at current transistor sizes, clock rates, and die sizes, it is not possible to communicate across the die in a clock cycle or two. We prefer that the core be simple and compact, to keep communication delays between components to a minimum. Second, bigger problems can be solved on a single die if there are multiple cores each capable of executing multiple threads. It is preferable to keep network communication on-chip if possible for speed and bandwidth.

Integration of processor and memory is also key. Cache coherency protocols do not scale well to large numbers of processing nodes. Tight integration of processing elements and memory reduces the desire for caches, as the memory itself provides a wide high-bandwidth connection to the core. We believe that the view of a processor and off-chip memory connected by a high-latency low-bandwidth pipe is a thing of the past, thanks to innovations in memory fabrication technology[5].

Impact: The design of such a processor provides a key element to a new shared-memory HPC architecture. While it is not the whole story, it is vital that the processor be simple enough to be designed quickly and to make scaling to one million nodes or more feasible. Currently, we are working on processor core designs using FPGA synthesis and simulation tools. When the core is ready for further testing, it will be implemented on a Processor-In-Memory System Simulator[6] known as the "Moore board."

Future Work: The Moore board will provide a test environment suitable for evaluating several design variations. There are plenty of features that require some investigation before a specific design is settled upon. Handling multiple threads elegantly, providing hardware support for a VLIW-style instruction set architecture, providing capabilities support, and dealing with intra-node communication on and off chip are all examples of such features. A very large body of data will be obtained on the utility of such features and on their particular implementations. When a suitable processor design is selected, the task then shifts to prototyping a small system, a single board with a handful of such processors, for additional scalability testing and fine-tuning before a larger system is built.

[height=3in, width=3in]figures/phillips.eps

Potential configuration of several processing elements (PE) on a single die.

Research Support: Support for this research was provided by the Air Force Research Laboratory, agreement number F30602-98-1-0172, "Active Database Technology."

References:

[1] F. T. Chong, M. Oskin, T. Sherwood. Active Pages: A Computation Model for Intelligent Memory *ISCA 1998, Barcelona, Spain*

[2] C. E. Kozyrakis, S. Perissakis, D. Patterson, T. Anderson, K. Asanovic, N. Cardwell, R. Fromm, J. Golbus, B. Grbistad, K. Keeton, R. Thomas, N. Treuhaft, K. Yelick. Scalable Processors in the Billion-Transistor Era: IRAM *IEEE Computer*, September, 1997, pp. 75-78.

[3] Margolus, Norman. An Embedded DRAM Architecture for Large-Scale Spatial-Lattice Computations *ISCA 2000, Vancouver, Canada*. Proceedings, pp. 149-160.

[4] S. Perissakis, Y. Joo, J. Ahn, A. DeHon, J. Wawrzynek. Embedded DRAM for a Reconfigurable Array. *Proceedings of the 1999 Symposium on VLSI Circuits*, June 1999.

[5] Glaskowsky, Peter N. MoSys Explains 1T-SRAM Technology *Microprocessor Report*, Sept 1999.

[6] Huang, Andrew. Processor-In-Memory System Simulator. *MIT AI Lab Abstracts of Research Projects*, Fall 2000.