SIMD AND MSIMD VARIANTS OF THE NON-VON SUPERCOMPUTER[1]

David Elliot Shaw

Department of Computer Science
Columbia University

ABSTRACT

Each member of the NON-VON family of supercomputers employs massive parallelism to obtain substantial performance improvements in a wide range of applications. The NON-VON 1 and NON-VON 3 machines, which are in various stages of construction at Columbia University, function for the most part as single instruction stream, multiple data stream (SIMD) [2] machines. We have recently begun to design an enhanced version of the machine, called NON-VON 4, which would be capable of functioning as an ensemble of one or more independent SIMD machines communicating through a high bandwidth interconnection network. This paper reviews the essential architectural features of the NON-VON family, and highlights the principal differences between NON-VON 4 and its SIMD predecessors.

1. OVERVIEW

NON-VON [3], [4] is a family of massively parallel supercomputers characterized by an extensive intermingling of processing and memory resources at various levels. The machines are intended to support the extremely rapid execution of many large scale data manipulation tasks, including relational database operations and other functions relevant to commercial data processing.

All members of the family incorporate a primary processing subsystem (PPS), which comprises a large number (as many as a million) of very simple, highly area-efficient small processing elements (SPE's), each associated with a small (64 bytes) local random access memory. The PPS is currently being implemented using custom nMOS VLSI circuits, each of which is to contain eight processing elements (at 1983 device dimensions). While this extremely fine granularity offers the potential for unprecedented computational concurrency, it also results in a local memory that is far too small to store meaningful programs. NON-VON's SPE's are thus forced to "import" their programs from one or more (depending on the version) external instruction sources.

The design of each member of the NON-VON family also includes a secondary processing system (SPS) based on a bank of "intelligent" disk drives, connected with the PPS through a high-bandwidth parallel interface. Because of funding limitations, we have not yet been able to undertake the implementation of an SPS for any version of the NON-VON machine. If implemented using contemporary technology, however, the machine might incorporate a reasonably large number of Winchester disk drives, each of moderat size. A small amount of processing hardware would be associated with each disk head. This hardware would allow records to be inspected "on the fly", to determine whether a given record is relevant t the operation at hand before transferring it to the PPS. The hardware would support certain othe operations, such as hashing, that play a key role in many disk-based NON-VON operations.

While the components we have just described are incorporated in all members of the NON-VON family there are certain essential differences between the three versions now being designed or constructed at Columbia. Specifically, NON-VON 1 and 3 include a single control processor, which i used to broadcast instructions for execution by the PE's in the PPS. With the exception of

[1]This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-80-G-0132.

Invited paper, COMPCON Spring '84, February, 1984

certain input/output operations discussed in Shaw [1982], these machines thus function in an instruction-synchronous (SIMD) mode, with all SPE's simulataneously executing a single instruction "in lock step".

NON-VON 4, on the other hand, would include a number of large processing elements (LPE's), each capable of serving as a control processor for some portion of the PPS. This should give NON-VON 4 the capacity for multiple instruction stream, multiple data stream (MIMD) and multiple SIMD (MSIMD) operations, multi-tasking and multi-user applications, and such problems as physical simulation for which the top of the NON-VON 3 tree would otherwise represent a significant communication bottleneck.

Additional enhancements anticipated for NON-VON 4 derive from the fact that a significant amount of storage (in the form of commercially available dynamic RAM chips) would be associated with each LPE. In addition to its use as storage for an individual LPE, this RAM should prove useful as swapping storage for the local RAM's incorporated in the SPE's with which it is associated. Finally, we hope to realize an additional multiplicative factor in total throughput in NON-VON 4 by reducing the effective instruction cycle time (which is equal to the time required for parallel inter-SPE communication) far below the estimated two microseconds projected for NON-VON 1 and 3 using a number of engineering techniques that are beyond the scope of this paper.

2. EVOLUTION OF THE NON-VON ARCHITECTURE

In order to minimize the risk involved in developing a highly unconventional supercomputer, we have adopted a three-stage, partially overlapped development strategy. We have begun by implementing and testing a relatively simple machine which nonetheless incorporates what we regard as the most essential elements of a full-scale NON-VON supercomputer. Architectural enhancements are to be added in stages, yielding incremental increases in power and generality without the introduction of an unmanageable increase in conceptual or engineering complexity at any single stage.

2.1. NON-VON 1

The first version we are actually implementing, which we now call NON-VON 1, is based on a custom nMOS VLSI chip that we have recently fabricated through the MOSIS "silicon brokerage" system. The chip is now undergoing extensive testing at Columbia, and has passed all tests successfully as of the time of this writing. Each chip contains a single SPE, including its own small local RAM. These single-SPE chips are to be interconnected to form the PPS, which is configured as a binary tree, with a single control processor attached to the root. A degenerate PPS consisting of three SPE chips has recently been constructed and, in preliminary tests, has performed according to specification. At least one additional fabrication run is anticipated prior to the fabrication of production quantities of parts in order to make improve certain electrical characteristics of the chip.

Because only a single control processor will be incorporated in the NON-VON 1 prototype, the machine will be limited to SIMD applications, in which the control processor sends instructions to be executed concurrently by all processing elements. Although a complete system would also include an SPS based on a number of "intelligent" disk drives, we will not be developing a working SPS within the scope of our current research contract, as noted above. In short, NON-VON 1 will be limited to the execution of SIMD algorithms in which the argument and result data does not exceed the capacity of the PPS.

Unlike more recent versions of the architecture, NON-VON 1 performs all arithmetic and logical operations in a bit-serial fashion and is rather limited its choice of operands for most instructions. Because only one SPE is embedded o each chip, a relatively low priority was placed o the minimization of silicon area; detailed measurements of the NON-VON 1 layout have, however, formed the basis for the highly efficien floor plans now under development for use in late versions.

A large part of the evolution from NON-VON 1 to its successors was attributable to our experience in developing software for the machine. NON-VON was simulated at no less than five levels of detail (from the functional to circuit levels) in the course of its construction. The most useful from the viewpoint of architectural evolution was based on an instruction-level simulator implemented in LISP, which was used by several dozen researchers and students to design and test parallel implementations of a number of algorithms. This activity helped us to measure the frequency of execution of various NON-VON 1 instructions, to identify commonly instruction sequences, and to identify functional weaknesses in the instruction set. Our findings had a major influence on the design of later versions of the processing element.

2.2. NON-VON 2

For the sake of completeness, it is probably wort mentioning at this point that the name NON-VON 2 was assigned to an interesting architectural exercise that we do not currently plan to carry beyond the "paper-and-pencil" stage, although its central ideas may well influence future NON-VON

designs.

2.3. NON-VON 3

The machine we have come to call NON-VON 3, on the other hand, forms the basis for much of our current work. Like NON-VON 1, our NON-VON 3 prototype will include no disk drives and only a single control processor, and will thus capable of executing only SIMD algorithms in which the data do not exceed the capacity of the PPS. The machine will be similar in most respects to the original NON-VON 1 design, but will incorporate a number of improvements suggested by the results of our initial experiments in chip design and software development. In particular, the NON-VON 3 SPE will feature:

1. An area-efficient eight-bit ALU to replace the one-bit ALU incorporated in the prototype NON-VON 1 SPE chip.

2. Fewer local registers, based on NON-VON 1 area measurements and software simulation results.

3. A far better floor plan, formulated using precise measurements taken from the prototype chip.

4. A generalization of certain NON-VON 1 instructions to support the more efficient execution of many common instruction sequences.

The NON-VON 3 instruction set is nearly identical to, and with few exceptions, more general than the one employed in NON-VON 1. Some of the additions in fact correspond to commonly used macros in our existing NON-VON 1 software. Before adopting this instruction set, however, we were careful to insure that all existing NON-VON 1 software could be simply and mechanically translated into NON-VON 3 instructions, so that none of our work to date would be lost. Translated programs would take advantage of some, but not all of NON-VON 3's enhancements. In the future, of course, NON-VON 3 software will be written using NON-VON 3 instructions, allowing the exploitation of all of these features.

The development of NON-VON 3 has been greatly accelerated by the availability of a highly automated system for the specification, design, layout and testing of VLSI-based processing elements that we implemented for use in designing NON-VON 3. Within this semi-automatic development environment, changes in the instruction set may be realized in hardware in a fraction of the time that would otherwise be required, facilitating extensive experimentation and "fine tuning" of the floor plan.

2.4. NON-VON 4

The NON-VON 1 and 3 machines should serve to validate many of our most important architectural ideas, yielding major performance improvements on a number of problems amenable to SIMD execution. The more sophisticated NON-VON 4 architecture, though, is intended to provide for the highly efficient execution of a much wider range of computational tasks than NON-VON 1 and 3. At present, NON-VON 4 is in the earliest stages of preliminary design. It must thus be emphasized that much of the material we will be presenting must be regarded as tentative, and that many of our ideas have yet to be validated.

3. Organization of NON-VON 4

In the NON-VON 4 machine, each node above a certain fixed level in the PPS tree would be connected to its own LPE. (In a machine containing 256K SPE's, for example, somewhere between 511 and 1K-1 such LPE's might be provided.) Each LPE would include an off-the-shelf microprocessor (a Motorola 68000 or National Semiconductor 16032, for example), a reasonable amount (say, between 256 KBytes and 1 Megabyte), and a modest amount of custom hardware for interfacing with the rest of the machine.

The set of LPE's would be mutually interconnected through a high-bandwidth multistage interconnection network. While the details of this network have not yet been specified, we have recently begun to investigate the possibility of using a "folded" 2-log-n-stage banyan network operating on a circuit-switched basis. Our idea (which, it must be emphasized, is quite tentative) would be to incorporate logic within the individual switches that would support the rapid, parallel identification of some unblocked path through the network whenever any such path exists. This technique is closely related to the broadcasting and resolving functions used in the NON-VON PPS [Shaw, 1982].

Although we have no supporting data at present, we believe that the circuitry for such a network might be simple enough to allow the construction of a moderately high-order switch (that is, one with a number of independent input and output paths) using a single high-speed bipolar gate array chip. Since each switch chip would, in essence, function as a full crossbar switch, the use of higher-order switches would reduce the number of network paths subject to contention. Additional performance advantages should result from the the reduced number of chip-to-chip delays in a network built with higher-order chips, which would in general have fewer stages. Another potentially attractive feature of this "broadcast circuit switch" approach is the fact that later stages in the network are automatically bypassed

(and hence relieved of congestion) in cases where the source and destination LPE's are "close" (in a particular sense defined in terms of their addresses).

Some subset of the LPE's (perhaps one-eighth to one-quarter in a realistic system) would be equipped with Winchester disk drives of moderate capacity, but having the kind of specialized "intelligent head unit" hardware described above.

4. Operation of NON-VON 4

It will not be possible in the limited space remaining in this paper to describe all aspects of NON-VON 4 that distinguish its behavior from that of the SIMD NON-VON machines. A simple example chosen from one of the machine's original domains of intended application, however, will serve to illustrate the manner in which the SPS/LPE network complex would often be used. Our example is the relational join operation [1], which takes two relations (tables) as arguments and produces a new table as output. Each tuple (row) in the result relation is formed by concatenating two tuples, one from each argument relation, that have the same value for the corresponding attribute (column).

On a von Neumann machine, this operation is typically executed by sorting the two argument relations on their respective join attributes (the columns on which the comparison is to be made), then performing a "merge-like" operation to produce the result tuples; the running time of the algorithm is typically dominated by the O (n log n) time required for sorting. The SIMD NON-VON machines reduce this time to linear, and provide a significant savings in absolute running time, given certain assumptions about the input data. In NON-VON 4, however, these assumptions are weakened in practically important ways, and the time required to join large relations is reduced far below those of all database machines (actual and "paper") to which we have compared it.

In "internal" cases, where the data does not exceed the capacity of the PPS, the NON-VON algorithm [3] (for both SIMD and MIMD machines) involves the broadcasting of each join value through the PPS and the associative identification of all matching tuples in the other relation. In the "external" case, however, where the argument relations exceed the capacity of the PPS, the argument relations must be divided into key-disjoint partitions, which have the property that no join value appears in more than one partition.

An external algorithm executing in linear time, but under fairly restrictive assumptions, has been presented [3] for the SIMD NON-VON machines. In NON-VON 4, however, key-disjoint partitioning may be accomplished by storing the argument relation "spread across" the various disk heads in the SP hashing each join value passing under each head, and dynamically routing the tuples through the L network to a different disk head depending on it hashed join value. This process produces k different key-disjoint partition files, where k the number of disk heads involved, and requires time proportional to the capacity of the full fi divided by k.

We are presently investigating the use of an "internal analog" of the above hash partitioning algorithm to produce the join results after external hash partitioning is complete. The essential idea is to hash the join values of all tuples in the PPS (in constant time), and then t migrate each tuple through the LPE network to a PPS subtree determined by the resulting hash value. Each subtree would then apply the intern SIMD join algorithm in parallel, resulting in a speedup proportional to the number of subtrees employed. It should be noted, however, we have not yet completed the detailed statistical analysis necessary to balance the degree of speedup with the cost of overflow and recovery within the subtrees.

References

[1] E. F. Codd, "Relational Completeness of Data Base Sublanguages", in R. Rustin (ed.), Courant Computer Science Symposium 6: Data Base Systems, Prentice-Hall, Inc., 1972.

[2] M. Flynn, "Some Computer Organizations and their Effectiveness", in IEEE Transactions on Computers, vol. C-21, pp. 948-960, September, 1972.

[3] D. E. Shaw, "A Hierarchical Associative Architecture for the Parallel Evaluation of Relational Algebraic Database Primitives", Stanford Computer Science Department Report STAN-CS-79-778, October, 1979.

[4] D. E. Shaw, "The NON-VON Supercomputer", Columbia Computer Science Department Report, August, 1982.