The Design of the MasPar MP-1: A Cost Effective Massively Parallel Computer

John R. Nickolls

MasPar Computer Corporation
749 North Mary Avenue
Sunnyvale, CA 94086

Abstract

By using CMOS VLSI and replication of components effectively, massively parallel computers can achieve extraordinary performance at low cost. Key issues are how the processor and memory are partitioned and replicated, and how interprocessor communication and I/O are accomplished. This paper describes the design and implementation of the MasPar MP-1, a general purpose massively parallel computer system that achieves peak computation rates beyond a billion floating point operations per second, yet is priced like a minicomputer.

Massively Parallel System

Massively parallel computers use more than 1,000 processors to obtain computational performance unachievable by conventional processors [1,2,3]. The MasPar MP-1 system is scalable from 1,024 to 16,384 processors and its peak performance scales linearly with the number of processors. A 16K processor system delivers 30,000 MIPS peak performance where a representative instruction is a 32-bit integer add. In terms of peak floating point performance, the 16K processor system delivers 1,500 MFLOPS single precision (32-bit) and 650 MFLOPS double precision (64-bit), using the average of add and multiply times.

To effectively apply a high degree of parallelism to a single application, the problem data is spread across the processors. Each processor computes on behalf of one or a few data elements in the problem. This approach is called "data-level parallel" [4] and is effective for a broad range of compute-intensive applications.

Partitioning the computational effort is the key to high performance, and the simplest and most scalable method is data parallelism. The architecture of the MP-1 [5] is scalable in a way that permits its computational power to be increased along two axes: the performance of each processor, and the number of processors. This flexibility is well matched to VLSI technology where circuit densities continue to increase at a rapid rate. The scalable nature of massively parallel systems protects the customers' software investment while providing a path to increasing performance in successive products [6].

Because its architecture provides tremendous leverage, the MP-1 implementation is conservative in terms of circuit complexity, design rules, IC geometry, clock rates, margins, and power dissipation. A sufficiently high processor count reduces the need to have an overly aggressive (and thus expensive) implementation. Partitioning and replication make it possible to use low cost, low power workstation technology to build very high performance systems. Replication of key system elements happily enables both high performance and low cost.

Array Control Unit

Because massively parallel systems focus on data parallelism, all the processors can execute the same instruction stream. The MP-1 has a single instruction stream multiple data (SIMD) architecture that simplifies the highly replicated processors by eliminating their instruction logic and instruction memory, and thus saves millions of gates and hundreds of megabytes of memory in the overall system. The processors in a SIMD system are called processor elements (PEs) to indicate that they contain only the data path of a processor.

The MP-1 array control unit (ACU) is a 14 MIPS scalar processor with a RISC-style instruction set and a demand-paged instruction memory. The ACU fetches and decodes MP-1 instructions, computes addresses and scalar data values, issues control signals to the PE array, and monitors the status of the PE array. The ACU is implemented with a microcoded engine to accommodate the needs of the PE array, but most of the scalar ACU instructions execute in one 70 nsec clock. The ACU occupies one printed circuit board.

Processor Array

The MP-1 processor array (figure 1) is configurable from 1 to 16 identical processor boards. Each processor board has 1,024 processor elements (PEs) and associated memory arranged as 64 PE clusters (PECs) of 16 PEs per cluster. The processors are interconnected via the X-Net neighborhood mesh and the global multistage crossbar router network.

The processor boards are approximately 14" by 19" and use a high density connector to mate with a common backplane. A processor board dissipates less than 50 watts; a full 16K PE array and ACU dissipate less than 1,000 watts.

A PE cluster (figure 2) is composed of 16 PEs and 16 processor memories (PMEM). The PEs are logically arranged as a 4 by 4 array for the X-Net two-dimensional mesh interconnection. Each PE has a large internal register file shown in the figure as PREG. Load and store instructions move data between PREG and PMEM. The ACU broadcasts instructions and data to all PE clusters and the PEs all contribute to an inclusive-OR reduction

CH2843-1/90/0000/0025$01.00 © 1990 IEEE

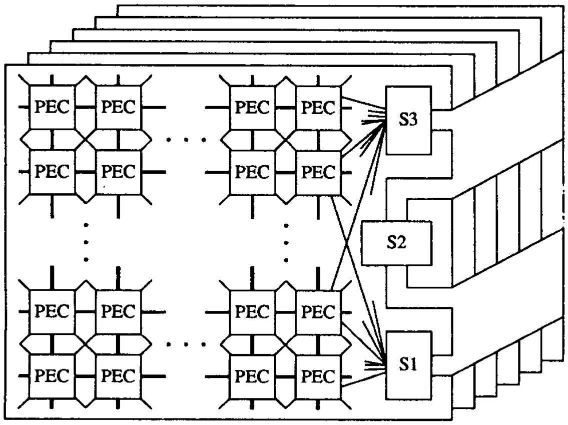

Figure 1. Array of PE Clusters

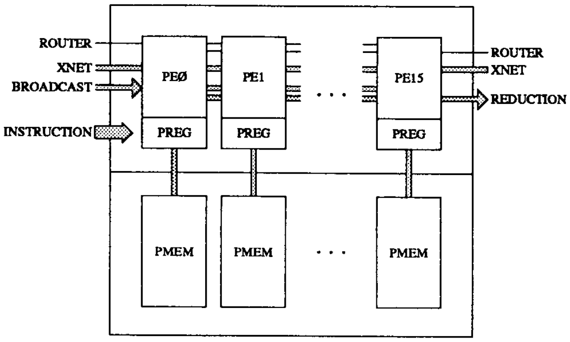

Figure 2. PE Cluster

tree received by the ACU. The 16 PEs in a cluster share an access port to the multistage crossbar router.

The MP-1 processor chip is a full-custom design that contains 32 identical PEs (2 PE clusters) implemented in two-level metal 1.6μ CMOS and packaged in a cost effective 164 pin plastic quad flat pack. The die is 11.6 mm by 9.5 mm, and has 450,000 transistors. A conservative 70 nsec clock yields low power and robust timing margins.

Processor memory, PMEM, is implemented with 1 Mbit DRAMs that are arranged in the cluster so that each PE has 16 Kbytes of ECC-protected data memory. A processor board has 16 Mbytes of memory, and a 16 board system has 256 Mbytes of memory. The MP-1 instruction set supports 32 bits of PE number and 32 bits of memory addressing per PE, so the memory system size is limited only by cost and market considerations.

As an MP-1 system is expanded, each increment adds PEs, memory, and communications resources, so the system always maintains a balance between processor performance, memory size and bandwidth, and communications and I/O bandwidth.

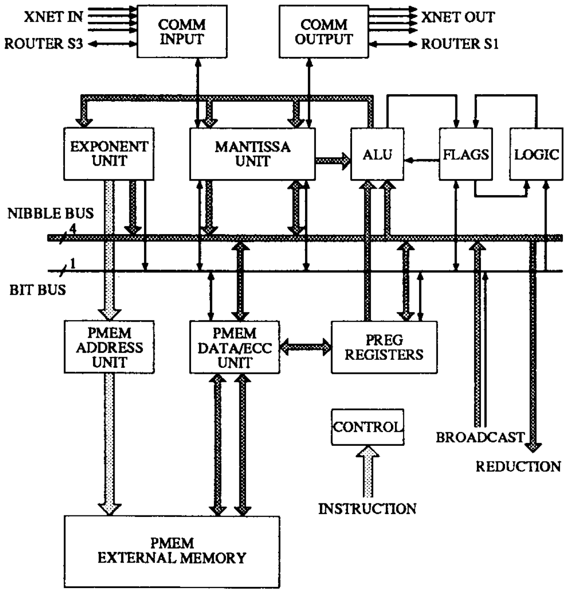

Figure 3. Processor Element and Processor Memory

Processor Elements

The MP-1 processor element (PE) design is different than that of a conventional processor because a PE is mostly data path logic and has no instruction fetch or decode logic. SIMD system performance is the product of the number of PEs and the speed of each PE, so the performance of a single PE is not as important as it is in conventional processors. Present VLSI densities and the relative tradeoffs between the number of processors and processor complexity encourage putting many PEs on one chip. The resulting design tradeoff between PE area and PE performance tends to reduce the PE architecture to the key essentials.

Each PE (figure 3) is designed to deliver high performance floating point and integer computation together with high memory bandwidth and communications bandwidth, yet have minimal complexity and silicon area to make it feasible to replicate many PEs on a single high-yield chip.

Like present RISC processors, each PE has a large on-chip register set (PREG) and all computations operate on the registers. Load and store instructions move data between the external memory (PMEM) and the register set. The register architecture substantially improves performance by reducing the need to reference external memory. The compilers optimize register usage to minimize load/store memory traffic.

Each PE has 40 32-bit registers available to the programmer and an additional 8 32-bit registers that are used internally to implement the MP-1 instruction set. With 32 PEs per die, the resulting 48 Kbits of register occupy about 30% of the die area, but represent 75% of the transistor count. Placing the registers on-chip yields an aggregate PE/PREG bandwidth of 117 gigabytes per second with 16K PEs. The registers are bit and byte addressable.

Each PE provides floating point operations on 32 and 64 bit IEEE or VAX format operands and integer operations on 1, 8, 16, 32, and 64 bit operands. The PE floating point/integer hardware has a 64-bit MANTISSA unit, a 16-bit EXPONENT unit, a 4-bit ALU, a 1-bit LOGIC unit, and a FLAGS unit; these units perform floating point, integer, and boolean computations. The floating point/integer unit uses more than half of the PE silicon area but provides substantially better performance than the bit-serial designs used in earlier massively parallel processors.

Most data movement within each PE occurs on the internal PE 4-bit NIBBLE BUS and the BIT BUS (figure 3). During a 32-bit or 64-bit floating point or integer instruction, the ACU microcode engine steps the PEs through a series of operations on successive 4-bit nibbles to generate the full precision result. For example, a 32-bit integer add requires 8 clocks: during each clock a nibble is fetched from a PREG register, a nibble is simultaneously obtained from the MANTISSA unit, the nibbles are added in the ALU, and the sum is delivered to the MANTISSA unit. At the same time, the ALU delivers a carry bit to the FLAGS unit to be returned to the ALU on the next step. The ALU also updates bits in the FLAGS unit that indicate overflow and zeroness.

The different functional units within the PE can be simultaneously active during each micro-step. For example, floating point normalization and de-normalization steps use the EXPONENT, MANTISSA, ALU, FLAGS, and LOGIC units together. The ACU issues the same micro-controls to all PEs, but the operation of each PE is locally enabled by the E-bit in its FLAGS unit. During a floating point operation, some micro-steps are data-dependent, so the PEs locally disable themselves as needed by the EXPONENT and MANTISSA units.

Because the MP-1 instruction set focuses on conventional operand sizes of 8, 16, 32, and 64 bits, MasPar can implement subsequent PEs with smaller or larger ALU widths without changing the programmer's instruction model. The internal 4-bit nature of the PE is not visible to the programmer, but does make the PE flexible enough to accommodate different front-end workstation data formats. The PE hardware supports both little-endian and big-endian format integers, VAX floating point F, D, and G format, and IEEE single and double precision floating point formats.

Along with the PE controls, the ACU broadcasts 4 bits of data per clock onto every PE nibble bus to support MP-1 instructions with scalar source operands. The PE nibble and bit bus also drive a 4-bit wide inclusive-OR reduction tree that returns to the ACU. Using the OR tree, the ACU can assemble a 32-bit scalar value from the OR of 16,384 32-bit PREG values in 8 clocks plus a few clocks of pipeline overhead.

Processor Memory

Because only load and store instructions access PMEM processor memory, the MP-1 overlaps memory operations with PE computation. When a load or store instruction is fetched, the ACU queues the operation to a separate state machine that operates independently of the normal instruction stream. Up to 32 load/store instructions can be queued and executed while PE computations proceed, as long as the PREG register being loaded or stored is not used by the PE in a conflicting way. A hardware interlock mechanism in the ACU prevents PE operations from using a PREG register before it is loaded and from changing a PREG register before it is stored. The optimizing compilers move loads earlier in the instruction stream and delay using registers that are being stored. The 40 registers in each PE assist the compilers in obtaining substantial memory/execution overlap.

The PMEM processor memory can be directly or indirectly addressed. Direct addressing uses an address broadcast from the ACU, so the address is the same in each PE. Using fast page mode DRAMS, a 16K PE system delivers memory bandwidth of over 12 gigabytes per second. Indirect addressing uses an address computed locally in each PE's PMEM ADDRESS UNIT and is a major improvement over earlier SIMD architectures[7] because it permits the use of pointers, linked lists, and data structures in a large processor memory. Indirect addressing is about one third as fast as direct addressing.

X-Net Mesh Interconnect

The X-Net interconnect directly connects each PE with its 8 nearest neighbors in a two-dimensional mesh. Each PE has 4 connections at its diagonal corners, forming an X pattern similar to the Blitzen[8] X grid network. A tri-state node at each X intersection permits communications with any of 8 neighbors using only 4 wires per PE.

Figure 1 shows the X-Net connections between PE clusters. The PE chip has two clusters of 4 by 4 PEs and uses 24 pins for X-Net connections. The cluster, chip, and board boundaries are not visible and the connections at the PE array edges are wrapped around to form a torus. The torus facilitates several important matrix algorithms and can emulate a one-dimensional ring with two X-Net steps.

All PEs have the same direction controls so that, for example, every PE sends an operand to the North and simultaneously receives an operand from the South. The X-Net uses a bit-serial implementation to minimize pin and wire costs and is clocked synchronously with the PEs; all transmissions are parity checked. The PEs use the shift capability of the MANTISSA unit to generate and accumulate bit-serial messages. Inactive PEs can serve as pipeline stages to expedite long distance communication jumps through several PEs. The MP-1 instruction set [5] implements X-Net operations that move or distribute 1, 8, 16, 32, and 64 bit operands with time proportional to either the product or the sum of the operand length and the distance. The aggregate X-Net communication rate in a 16K PE system exceeds 20 gigabytes per second.

Multistage Crossbar Interconnect

The multistage crossbar interconnection network provides global communication between all the PEs and forms the basis for the MP-1 I/O system. The MP-1 network uses three router stages shown as S1, S2, and S3 in figure 1 to implement the function of a 1024 by 1024 crossbar switch. Each cluster of 16 PEs shares an originating port connected to router stage S1 and a target port connected to stage S3. Connections are established from an originating PE through stages S1, S2, S3, and then to the target PE. A 16K PE system has 1024 PE clusters, so each stage has 1024 router ports and the router supports up to 1024 simultaneous connections.

Originating PEs compute the number of a target PE and transmit it to the router S1 port. Each router stage selects a connection to the next stage based on the target PE number. Once established, the connection is bidirectional and can move data between the originating and target PEs. When the connection is closed, the target PE returns an acknowledgement. Because the router ports are multiplexed among 16 PEs, an arbitrary communication pattern takes 16 or more router cycles to complete.

The multistage crossbar is well matched to the SIMD architecture because all communication paths are equal length, and therefore all communications arrive at their targets simultaneously. The router connections are bit-serial and are clocked synchronously with the PE clock; all transmissions are parity checked. The PEs use the MANTISSA unit to simultaneously generate outgoing router data and assemble incoming router data.

The MP-1 router chip implements part of one router stage. The router chip connects 64 input ports to 64 output ports by partially decoding the target PE addresses [9]. The full-custom design is implemented in two-level metal 1.6μ CMOS and packaged in a 164 pin plastic quad flat pack. The die is 7.7 mm by 8.1 mm, and has 110,000 transistors. Three router chips are used on each processor board.

A 16K PE system has an aggregate router communication bandwidth in excess of 1.5 gigabytes per second. For random communication patterns the multistage router network is essentially equivalent to a 1024 by 1024 crossbar network with far fewer switches and wires.

Conclusion

Through a combination of massively parallel architecture, design simplicity, cell replication, CMOS VLSI, conservative clock rates, surface mount packaging, and volume component replication, the MasPar MP-1 family delivers very high performance with low power and low cost. The massively parallel design provides cost effective computing for today and a scalable growth path for tomorrow.

References

[1] S. F. Reddaway, "DAP, A Distributed Array Processor", *First Annual Symposium on Computer Architecture*, IEEE/ACM, Florida, 1973.

[2] Kenneth E. Batcher, "Design of a Massively Parallel Processor", *IEEE Transactions on Computers*, vol C-29, pp. 836-840, Sept 1980.

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

[4] David L. Waltz, "Applications of the Connection Machine", *Computer*, pp. 85-97, January 1987.

[5] Tom Blank, "The MasPar MP-1 Architecture", *Proceedings of IEEE Compcon Spring 1990*, IEEE, February 1990.

[6] Peter Christy, "Software to Support Massively Parallel Computing on the MasPar MP-1", *Proceedings of IEEE Compcon Spring 1990*, IEEE, February 1990.

[7] Ken Batcher, "The Architecture of Tomorrow's Massively Parallel Computer", *Frontiers of Massively Parallel Scientific Computing*, NASA CP 2478, September 1986.

[8] Edward W. Davis and John H. Reif, "Architecture and Operation of the BLITZEN Processing Element", *3rd Intl. Conf. on Supercomputing*, vol III, pp. 128-137, May 1988.

[9] Robert Grondalski, "A VLSI Chip Set for a Massively Parallel Architecture", *International Solid State Circuits Conference*, February 1987.