N90-16453

BLITZEN:

A HIGHLY INTEGRATED MASSIVELY PARALLEL MACHINE*

D. W. Blevins, E. W. Davis[1], R. A. Heaton, and J. H. Reif[2]

The Microelectronics Center of North Carolina
Research Triangle Park, NC 27709-2889

ABSTRACT

The goal of the BLITZEN project is to construct a physically small, massively parallel machine. A highly integrated chip has been designed with 128 processing elements (PEs). A BLITZEN system consisting of 16,384 SIMD PEs will require only 128 PE array chips. This paper presents the PE architecture , the organization of PEs on the chip, and the feature set of the chip which has been custom designed and is being fabricated at the Microelectronics Center of North Carolina. Each PE has 1K bits of static RAM and performs bit-serial processing with functional elements for arithmetic, logic, and shifting. Unique local control features include modification of the global memory address by data local to each PE, and complementary operations based on a condition register. PEs on the chip are positioned in an 8 by 16 array. Data I/O is accomplished through a new method using a four-bit bus for each row of 16 PEs. The BLITZEN chip is one of the first to incorporate over 1.1 million transistors on a single die. It has been designed with MCNC's advanced 1.25 micron CMOS process to operate in excess of 20 MHz. A 16K PE system, operating at 20 MHz, can perform IEEE standard 32-bit floating point multiplication at a rate greater than 450 megaflops. Fixed point operations on 32 bit data can exceed the rate of one billion operations per second. Since the processors are bit-serial devices, performance rates improve with shorter word lengths. The bus oriented I/O scheme can transfer data at 10240 megabytes per second.

Keywords: massively parallel, custom VLSI, parallel processing, SIMD, MPP.

OVERVIEW AND MOTIVATION

Parallel machines make use of multiple processing elements executing simultaneously to speed up computation. For the purposes of this paper, we will consider a *massively* parallel machine to be a parallel machine with at least 10,000 processors. A number of massively parallel machines have been constructed, including the Massively Parallel Processor (MPP) built for NASA Goddard Space Flight Center by Goodyear Aerospace Corporation (now Loral Systems Group), the Distributed Array Processor (DAP) built by the British firm ICL, and the Connection Machine (CM) built by Thinking Machines, Inc. (Refs. 1, 7, 8, and 11). These projects demonstrated the feasibility of constructing machines with massive parallelism. Nevertheless, only a relatively small number (a few dozen) of the machines have been built so far and they have been utilized almost exclusively by *research* branches of government agencies, academic, and industrial organizations.

Miniaturization of Sequential Computing Machines

The situation now may be very similar to the development of the first mainframe computers in the late 40's: only a few general purpose computers existed. At that time, IBM made an early study which indicated that the worldwide use of computers would require only a few dozen mainframes (the rest of the computing equipment being calculators or special purpose machines). Nevertheless, a combination of advantageous engineering and economic factors resulted in the proliferation of computers. Central among these factors was the use of advanced electronic techniques to reduce the physical size, that is, to *miniaturize* computing machines. By miniaturization, we mean a high level of integration of the hardware onto VLSI components. Note that the process of miniaturizing sequential architectures has not necessarily at all degraded the computing power available to users. Miniaturization first allowed mainframe computing machines to be economically manufactured; and later, further improvements in integrated circuit technology allowed personal computing machines to be physically placed within the working environment of office workers, engineers, and scientists. In fact the development, for example, of miniaturized RISC architectures, has actually *improved* performance in many cases, by allowing higher execution rates.

BLITZEN: A Miniaturized Massively Parallel Machine

The central goal of the BLITZEN project is to develop a miniaturized massively parallel machine. The machine will be physically small while providing the performance associated with massively parallel processing. We are convinced that the development of such a miniaturized machine will have the same benefits as discussed above for conventional sequential machines:

(1) These miniaturized machines should be much more economical, allowing a much larger market for massively parallel machines.

* This work was supported in part by NASA Goddard Space Flight Center under Contract Number NAG-5-966 to the Microelectronics Center of North Carolina.

1. Dept. of Computer Science, North Carolina State Univ., Raleigh, NC 27695-8206.

2. Dept. of Computer Science, Duke Univ., Durham, NC 27706. He is also supported by contracts: ONR #N00014-80-C-0647, Air Force #AFOSR-87-0386, ONR #N00014-87-K-0310, NSF #CCR-8696134, DARPA/ARO #DAAL03-88-K-0195, and DARPA/ISTO #N00014-88-K-0458.

(2) The miniaturized machines could be backplaned with conventional workstations, making the capabilities of massively parallel computation easily accessible to engineers and scientists.

(3) A miniaturized machine could potentially be used in environments that require very small size and power consumption, such as on space flights. For example, NASA plans to have such a machine as a component of the Space Station computing system.

This paper provides rationale for design decisions, many of which have the dual benefit of both insuring miniaturization and also improving performance.

The Project Team

The BLITZEN project involves a number of institutions in the Research Triangle area of North Carolina, including Duke University, North Carolina State University (NCSU), and the Microelectronics Center of North Carolina (MCNC). Project personnel included John Reif, Jonathan Rosenberg, and graduate students Jonathan Becher, Nigel Hooke and Lars Nyland of the Computer Science Dept. of Duke, Edward Davis of the Computer Science Dept.of NCSU, and Don Blevins and Fred Heaton of MCNC. The BLITZEN project has received partial support under a grant from NASA Goddard Space Flight Center.

Team effort to date has resulted in development of the processing element architecture (Refs. 4 and 5), custom design for the PE array chip, development of a full scale PE array simulator (Ref. 10), microcode for selected arithmetic operations, and the specification of an assembler language and architecture for the BLITZEN controller (Ref. 9). We are in the process of developing a prototype system and a high level parallel programming language which is an extension of C++ for the BLITZEN machine.

Organization of the Paper

In the next section, "Processing Element Architecture", we describe the bit serial processing element and provide some comparisons with the MPP and Connection Machine. Local control features and methods for memory access are emphasized. Following the discussion of individual PE architecture, we describe, in the section "PE Array Chip Architecture", the organization of PEs on the custom chip, with emphasis on our interconnection and I/O schemes. The section "Chip Feature Set", provides details of the custom chip design and instruction pipeline. An overview of system architecture concepts and software for BLITZEN is given in the final section, "BLITZEN Systems".

PROCESSING ELEMENT ARCHITECTURE

Each processing element in BLITZEN is a bit serial processor, with a variable length shift register and random access memory. The BLITZEN design used the MPP PE architecture, described in Ref. 2., as a starting point.

The existence of the MPP has provided experience with massively parallel processing such as that reported by the MPP Working Group (Ref. 6) and by K. E. Batcher, the chief architect of the MPP, (Ref. 3).

Our group has designed various improvements on the MPP PE architecture into BLITZEN:

(1) Incorporation of RAM *on-chip* for each PE.

Motivation: This allows the PE to access memory without off-chip delays.

(2) *Bus oriented I/O* with a four bit path for each set of 16 PEs.

Motivation This gives BLITZEN a total I/O capability of 4,096 bits per cycle. (In comparison, the MPP has a total I/O capability of 256 bits per cycle, and the Connection Machine has an I/O capability of 1,024 bits per cycle.)

(3) *Local* modification of RAM addressing.

Motivation: This allows on-chip memory accesses to be determined by the contents of each PE's shift register.

(4) *Local* conditional control of arithmetic and logic functions.

Motivation: This improves the performance of various arithmetic operations.

(5) *Bidirectional* shift register.

Motivation: This allows more flexible data movement.

(6) An *X-grid interconnect*, allowing eight neighbors per PE.

Motivation: This gives a factor of two improvement (over the NEWS grid) in diagonal data movement.

Note that (3) and (4) give the BLITZEN PE a degree of MIMD control, which can improve the flexibility and efficiency of the machine.

Figure 1 presents the functional elements of one BLITZEN PE and shows a similarity to the PE in the MPP. Blocks with double line boundaries are storage devices. There are six single-bit registers labelled A, B, C, G, K, and P. Two devices hold multiple bits. One is a variable length shift register which, in conjunction with registers A and B, has a capacity of 32 bits. The remaining storage device is a 1024 bit random access memory (RAM). Arithmetic and logical operations are performed by a full adder and a logic block. The above elements communicate primarily over a single bit data bus. A four bit I/O bus provides a path to pads of the chip for connection to external storage devices. An I/O bus is shared among 16 PEs on a chip. Following paragraphs discuss features that represent significant departures of BLITZEN from the MPP.

On-Chip Memory

An on-chip, static random access memory (RAM) is associated with each PE. From a processing point of view it is a 1024 by 1 bit RAM. A memory read operation reads the single bit specified by a ten bit address and places the value on the data bus. A memory write operation writes the value from the data bus into the location specified by a ten bit address.

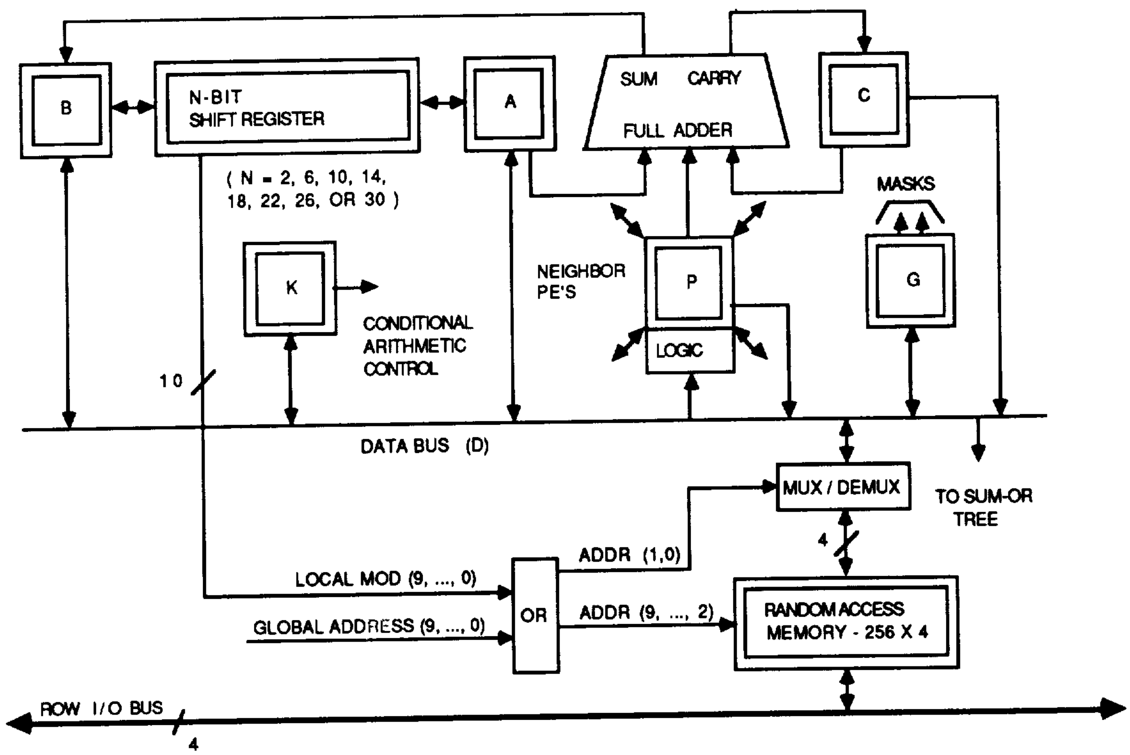

Figure 1. Functional elements of one BLITZEN PE.

Input/output operations view memory as a 256 by 4 bit RAM. I/O operations access memory using the eight most significant bits of the ten bit address, and transfer four bits between the I/O bus and memory.

Masking, the local control feature that can be used to enable or disable certain operations, is possible on all memory accesses.

Local Address Modification

In a SIMD machine, the control unit issues an instruction to all PEs. If a memory operation is involved, one address is delivered to all PEs. In BLITZEN, the global address can be modified at each PE. Conventional processors generally modify an address that appears in an instruction by adding index or base register values, or extracting an address from some location for indirect use. In a SIMD machine, logic that handles local modification of addresses must appear at each PE and be locally decoded. That is, the logic must appear at each of the 128 PEs on this chip. To conserve chip area the modification chosen is the logical OR of the global address with ten bits from the shift register. This can simulate indexing when data structures begin on appropriate power of two boundaries where the least significant bits are zeroes. When normal (unmodified) memory operations are issued, the global address is unchanged.

Figure 1 shows a ten bit bundle of signals from the shift register labeled "local mod". The ten most significant bits of the 16 bit section of the shift register are used to provide local address modification.

We believe BLITZEN is the first massively parallel machine with the ability to modify the global SIMD memory address in every PE. BLITZEN has addressing logic with every PE. Previously, a SIMD machine developed by DEC, and the Connection Machine 2, allowed a large group of processors to share indirect addressing logic.

Conditional Operations

BLITZEN provides additional new local control of PEs through the use of a programmable conditional operation test involving register K. When using the conditional feature, operations which are complements of each other can be performed at the same time in different PEs. The feature applies to operations involving logic at register P, or loading a value into register C. When a conditional operation is issued, processing is normal in all PEs where K = 0. In those PEs where K = 1 the results are complemented. Since both normal and complemented operations take place, based on testing a condition, this is like a restricted form of the high level IF-THEN-ELSE concept with both the THEN and ELSE clauses happening concurrently. When a conditional operation instruction is not used by the programmer, register K is available to hold a temporary value.

The conditional operation feature can be used to improve performance, by a factor near two, in non-restoring division algorithms where the next iterative step depends on the result of the current step. If the current step produces a negative partial remainder, the divisor is added at the next step. If the current step produces a positive partial remainder the divisor is subtracted at the next step. The approach to following both paths concurrently is to program the subtraction operation for conditional execution. By using the sign bit as the conditional flag in K, subtraction will take place in those PEs where K=0 and addition where K=1, as desired.

Bidirectional Shift Register and Data Paths

The MPP shift register is unidirectional. In BLITZEN it has been made bidirectional. In the MPP all bits shift during a shift operation, even if they are not selected under the current length setting. Since BLITZEN uses a section of the shift register to hold local address bits, the register design has been changed such that bits do not shift if they are not selected. This also lets the shift register be used to hold temporary variables.

Several smaller changes have been made, as compared to the original MPP PE. Bidirectional paths are provided between the data bus and all registers except C. Since a masked write operation is possible, the equivalence function between registers P and G has been eliminated. For a more detailed description of the BLITZEN PE architecture, see Refs. 4 and 5.

PE ARRAY CHIP ARCHITECTURE

Organization of PEs and Functional Components

The above PE architecture is used as the basis for the BLITZEN VLSI processor array chip. A single chip contains 128 PEs, each with 1K bits of locally addressable memory.

By placing 128 PEs and their local memory on a single chip, we make a major step toward miniaturization of the BLITZEN machine. Only 128 of these PE array chips are required for an entire 16,384 PE BLITZEN machine (In comparison the MPP processing element array chip contains eight PEs, and the system requires a total of 2048 such chips. The Connection Machine has 16 PEs per chip.).

A single PE is a building block for the chip architecture. PEs are organized into an 8 by 16 array on the chip. They are interconnected with a two dimensional grid for communication between PEs, as discussed in the next section.

Data is moved on and off the chip over a set of eight I/O buses, each with 16 PEs attached, as described in the section "BLITZEN I/O Scheme" Figure 2 shows the organization of PEs on the chip, including the X-grid interconnections, I/O buses, and some logic and control signals that are common to all PEs on the chip.

Message Routing Capability on the BLITZEN Machine

Why a Hypercube Interconnect Is Not Necessarily an Improvement Over a Grid - One major design decision was *not* to use a logarithmic diameter interconnection network, such as the hypercube used by the Connection Machine. Instead we used a variant of the two dimensional grid, namely the X-grid (due to C. Fiduccia), with diameter 128, which is the square of the number of processors. In spite of our background in theoretical computer science, we concluded that a logarithmic diameter network would be impractical for our needs. The key problems with logarithmic diameter networks, such as the hypercube, are:

(1) The number (namely 896) of I/O pads that would be required for hypercube edges exiting a processing element chip with 128 PEs is impossibly large.

(2) The inter-PE wiring requires large amounts of area, both on-chip and between chips.

A decision to use a hypercube interconnection network would make it very difficult to highly integrate our machine. Because of pin count and network area requirements, we would have been limited to only 16 PEs per chip, and even then only have 1/16 of the I/O pins required for a full hypercube interconnect. The result would be an interconnect with perhaps no greater communication capabilities than a two dimensional grid.

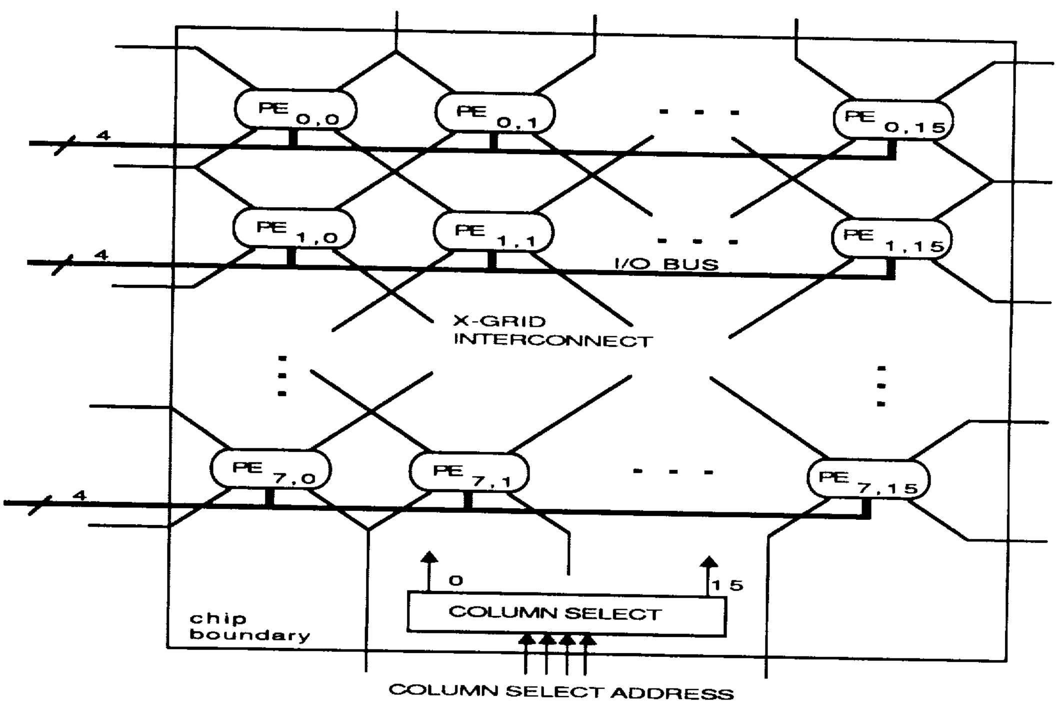

Figure 2. BLITZEN Chip Architecture

Another argument in favor of the grid interconnect is the empirical experience that a very large class of applications naturally require the grid interconnect.

The Connection Machine has some impressive built-in hardware for doing permutation message routing. Unfortunately, this routing circuitry uses a large fraction of their processor chip area and decreases the step rate of their machine. We decided that our need for a high performance, miniaturized architecture was more important than the need for message routing circuitry, (which can be replaced by software routing routines that are nearly as efficient.)

X-Grid Interconnection - Processing elements are interconnected in two ways on a chip: a grid interconnection for routing and a bus structure for I/O. Figure 2 shows the X-grid nearest neighbor routing network. PEs are arranged in a two dimensional grid with interconnection paths to neighbors in the eight compass directions N, NE, E, SE, S, SW, W, and NW. A routing operation transfers the state of P to the P register of a neighboring PE and accepts a new state from the PE in the opposite compass direction.

Four bidirectional routing connections are brought out of each PE from the four logical corners: NE, SE, SW, and NW. The connections intersect between PEs as shown in figure 2. A routing path is established by an operation which sends data out in one direction and accepts data in from one of the remaining directions. As an example, routing in the north direction can be achieved by sending P out to the NE and accepting P in from the SE. The data value on the SE input originated in the PE to the south. All PEs route the same direction in one processing cycle.

Eight paths can be established with four wires out of each PE by sending data on one wire, receiving data on one of the other three wires, and placing the remaining two wires in the high impedance state. This X-grid interconnects PEs on a chip and extends across chip boundaries so that an array of chips can be uniformly interconnected. Additional off-chip logic can provide various treatments of edges of the total array, as was done in the MPP system. The use of the X-grid allows a factor of two improvement in the frequently occurring case of diagonal data movement.

BLITZEN I/O Scheme - Data I/O is the critical path in any parallel machine. The MPP's I/O scheme is simple -- data is shifted in from the west edge of the array using the S–plane, and shifted out simultaneously along the east edge. In a BLITZEN system the array would be segmented along chip boundaries, so a natural extension to the MPP I/O scheme would be to have data flow in one side of a chip and out the other using the same S-plane idea. Thus BLITZEN would have data I/O occurring every 16 PEs, from west to east, using 32 pins.

At that time in the chip design activity, floorplanning predicted that the local static RAM should have a 256 by 4 aspect ratio. The RAM would have a four-bit interface, with further demultiplexing and multiplexing for the one-bit PE data bus. Since there were four data wires available per row of PEs on a chip, an alternative I/O approach was presented. The approach was to move, conceptually, the 16 output S-plane connections from the east edge to the west edge, and combine them with the 16 input S-plane connections to form eight bidirectional, four-bit I/O buses on each chip. Each four-bit bus is shared by the 16 PEs in a row. This scheme has several advantages, such as very high bandwidth, an easier interface for extending memory off-chip, the ability to broadcast data to all PEs simultaneously, fast data movement across the chip, and elimination of the S-plane.

Each chip has column select logic that is used in conjunction with the I/O buses. For normal I/O transfers, one PE in each row is active. The PE column index is the same for all rows and is given by a four bit address to the column select logic. In broadcast mode, data can be input to all PEs on a row, thus column selection is not used.

Video RAM (VRAM) chips are available with very high block data transfer rates, matching the rates of our PE I/O buses, and with four bit outputs, matching our four bit I/O buses. We plan to use one megabit VRAM chips, organized as 256K by 4, to augment the PE memory by 64K bits each. We will allow the 16 PEs along an I/O bus to share a vertically packaged VRAM chip.

CHIP FEATURE SET

The BLITZEN PE array chip was designed by the Microelectronics Center of North Carolina (MCNC) with two orthogonal constraints: maximize both integration and speed. The chip incorporates over 1.1 million transistors on a die 11.0 by 11.7 mm. It was designed with MCNC's 1.25 micron, two level metal, CMOS process. It is packaged in a 168 pin pin grid array and is designed for the JEDEC 3.3 volt power supply standard. The operating frequency is 20 MHz worst case, and power dissipation is 1.0 watt.

The chip contains 128 PEs positioned in an 8 by 16 array. Internally, a three stage pipeline enables BLITZEN to execute an instruction every cycle, as shown in figure 3. During the first cycle a 23 bit SIMD instruction from the control unit is latched and decoded into a fully horizontal 59 bit microinstruction. During the second stage of the pipeline the microinstruction is broadcast to all 128 PEs. In the final stage the instruction is executed. By issuing a fully horizontal microinstruction, no additional decoding logic was needed in the PEs. The encoding of the 23 bit instruction was optimized to minimize the amount of internal decoding.

Data transfers on the I/O bus take place in a single cycle as shown in the timing diagram in figure 4. If the I/O buses are used as an interface to high density video RAMs, blocks of data can be transferred quickly to and from the chip. Routing communication on the X-grid also takes place in a single cycle.

Figure 5 is the floorplan of a single PE. Each PE has access to its own 1K bits of memory, which are internally organized as 32 by 32 bits. Multiplexing is provided to select four out of 32 bits for interfacing to that PE's I/O bus. When a PE accesses memory for an operand, further selection of one out of four bits is needed. Address calculation logic (predecode) is also needed at each PE to support the indirect addressing mode provided by local modification of the global address. The execution unit of a PE, including the shifter and ALU, contains approximately 1130 transistors.

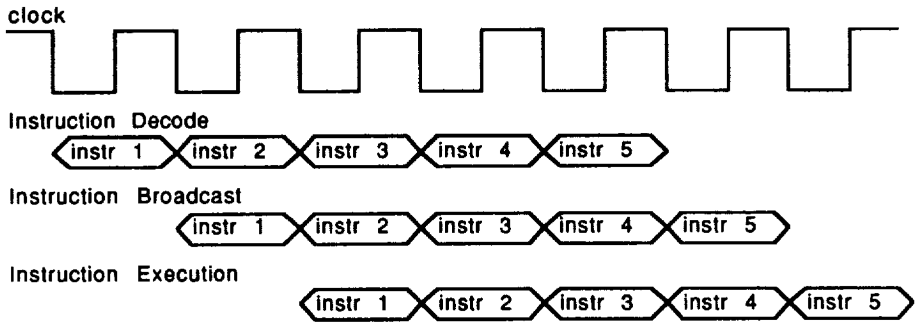

Figure 3. The instruction pipeline.

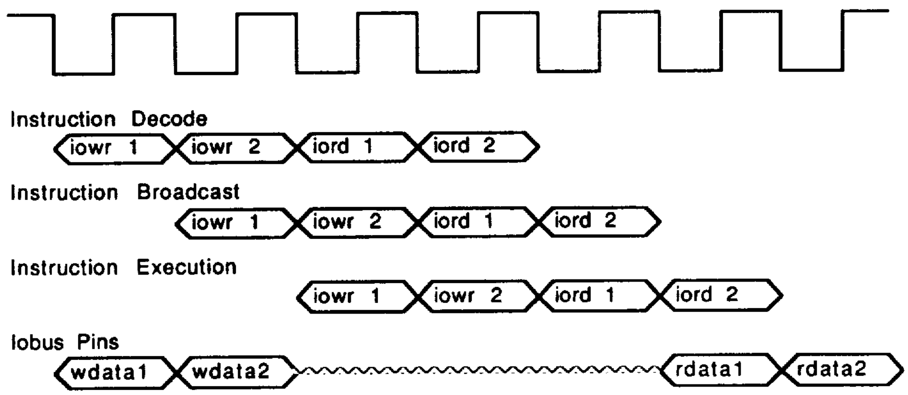

Figure 4. The instruction pipe for I/O bus transfers.

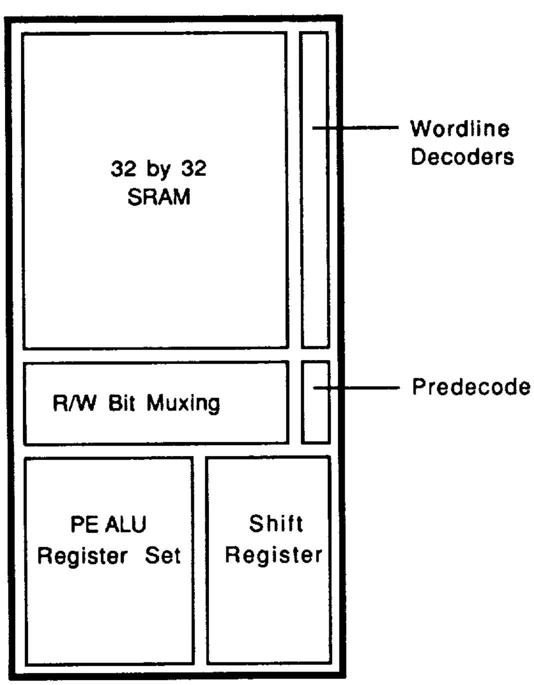

Figure 5. VLSI design floorplan for one PE.

BLITZEN SYSTEMS

In a top level view of the system architecture, major components are organized around two buses. An internal bus supports data transfers between register and memory components. The second bus is used for transfers between BLITZEN and a host computer. Massive SIMD processing takes place in the processing array. Data in the on-chip local memory is supplied from off-chip, video RAM data memory, with the transfers considered as I/O operations with respect to the array.

Instructions are broadcast from the control unit to all PEs in the array. More specifically, operation codes originate in microcoded routines stored in control memory, and local memory addresses are generated from the register set. Together they form an array instruction. Control logic manages the register set and sequences the microinstructions. A scalar microprocessor can be included for use as the processor running an application program. It executes scalar instructions and sends calls for array instructions to the sequencing logic in the control unit.

Two external interfaces are planned. The host interface is a narrow path that matches the host wordlength. It is used for downloading programs (both application and microcode) and transferring data at low bandwidth between BLITZEN and the host with it's peripherals. High speed peripherals communicate with BLITZEN through custom peripheral interface logic. This path accesses the data memory and is potentially very wide for very high bandwidth.

Data Memory

Each BLITZEN processing element has 1K bits of RAM on-chip for holding data. It is known that many applications can benefit from additional memory, but the 1K amount was governed by chip size and density limits. In BLITZEN, the memory limitation can be alleviated by off-chip data memory that is accessed across the I/O buses. The use of VRAM for this purpose was mentioned earlier. Data memory can be viewed as the primary data memory of the system with on-chip RAM treated as registers or data cache.

Using the high bandwidth I/O buses it is possible to change the content of all or part of the on-chip RAM very quickly. In one instruction cycle 32 bits (eight four-bit items) can be transferred between VRAM and each array chip. If the system is operating at 20 MHz, the total transfer rate is (4 bytes/chip)*(128 chips) per 50 nanoseconds, or 10.24 Gigabytes per second. In 128 instruction cycles, 32-bit data items can be transferred into (or out of) the on-chip RAM of each PE. In 4096 instruction cycles the entire 1K per PE RAM can be loaded. In 8192 cycles the content of RAM for the entire array can be swapped. Operating at 20 MHz, the time required to swap the total content is 409.6 microseconds.

Holographic Routing

J. Reif, at Duke, has invented a holographic message routing system, using electro-optical components yielding very high routing rates. He is developing this device under DARPA/ARO contract. K. Johnson from the Electro-optical Computing Center at University of Colorado, Boulder, is constructing a prototype of this system. We are developing microcode to allow BLITZEN to use this electro-optical routing device.

Programmer's Model

BLITZEN is a computing system whose primary computational resource is a single instruction stream, multiple data stream array processor with a massive number of processing elements. This massively parallel array operates in conjunction with several other major system components.

Programming BLITZEN takes place at several levels. At the lowest level is the machine language for the array. The hardware instruction set is specified in Ref. 4. Since the instruction set is concerned with single bit register transfers, it is not expected to be used by application programmers. Rather, it is the basis for a microcode development language, named BLITZ (Ref. 10), that couples array operations with control unit register transfers and sequencing operations. Commonly used routines corresponding to assembly language instructions such as load, store, add, floating point add, etc. are being written in BLITZ for inclusion in a microcode library whose routines can be called from a higher level language. An object oriented language based on C++ is being developed for application programming. High level language statements will be compiled into parallel assembly language statements that result in a calls to microcode routines which are executed on the array hardware.

Parallel PE Array Simulator

Prior to the existence of hardware, a software behavioral simulator known as "Zyglotron" was developed (Ref. 10).It is a "full scale" simulator in that it can simulate the entire 16,384 PE array with very high performance. Zyglotron is being used for microcode development, and can allow the development of algorithms and high level software to proceed concurrently with hardware system development. As noted in the abstract of Ref 10, " The simulator has achieved such high performance by taking advantage of a natural mapping that exists between massively parallel bit-serial machines and the vector architecture used in many high performance scientific super-computers." The simulator runs on the CONVEX C-1 vector processing machine and is written in C and in the CONVEX C-1 assembly language.

CONCLUSION

This paper has reported on the architecture and VLSI design of a new massively parallel processing array chip. The BLITZEN PE array chip, containing 1.1 million transistors, has been submitted to the Microelectronics Center of North Carolina for fabrication. The chips are the basis for a highly integrated, miniaturized, high performance, massively parallel machine that is currently under development.

The work reported in this paper resulted from the efforts of a group of researchers, mentioned in the overview section, participating in this project with the support of the Microelectronics Center of North Carolina. We also benefitted from discussions with Kenneth Batcher of Loral Systems Group concerning architecture of the MPP and local address modification schemes; with John Dorband of NASA Goddard SFC concerning conditional operations; and with Charles Fiduccia of General Electric who described their cross-omega machine with an eight neighbor grid interconnect. The interest and support of Milt Halem, NASA Goddard SFC, has been crucial to the success of this project.

REFERENCES

1. Batcher, K. E., "Design of a Massively Parallel Processor", *IEEE Trans. on Computers*, C-29(9), p 836-840.

2. Batcher, K. E., "Array Unit", *The Massively Parallel Processor*, J. L. Potter, Editor, The MIT Press, 1985.

3. Batcher, K. E., "The Architecture of Tomorrow's Massively Parallel Computer", *Proc. of the First Symposium on the Frontiers of Massively Parallel Scientific Computation*, 1986.

4. Blevins, D. W., E. W. Davis, and J. H. Reif, "Processing Element and Custom Chip Architecture for the BLITZEN Massively Parallel Processor", *MCNC Technical Report TR87-22*, 1987.

5. Davis, E. W., and J. H. Reif, "Architecture and Operation of the BLITZEN Processing Element", *Proc. of the Third International Conference on Supercomputing*, Boston, MA, May 1988.

6. Fischer, J. R., et al, "Report from the MPP Working Group to the NASA Associate Administrator for Space and Science Applications",*NASA Technical Memorandum 87819*, November,1987.

7. Hillis, W. D., *The Connection Machine*, The MIT Press, 1985.

8. Potter, J. L., Ed., *The Massively Parallel Processor*, MIT Press, 1985.

9. Rosenberg, J. B., and E. W. Davis, "BLITZ: Blitzen's Microcode Assembly Language Design Document, *MCNC Technical Report TR88-14*, 1988.

10. Rosenberg, J. B., J. Becher, and N. Hooke, "Vectorization Enables Full Scale Simulation of Massively Parallel (SIMD) Architectures", *Proceeding of the Third International Conference on Supercomputing*, Boston, MA, 1988.

11. Sharp, J. A., *An Introduction to Distributed and Parallel Processing*, Blackwell Scientific Publications, 1987.